



AgEcon SEARCH
RESEARCH IN AGRICULTURAL & APPLIED ECONOMICS

The World's Largest Open Access Agricultural & Applied Economics Digital Library

This document is discoverable and free to researchers across the globe due to the work of AgEcon Search.

Help ensure our sustainability.

Give to AgEcon Search

AgEcon Search

<http://ageconsearch.umn.edu>

aesearch@umn.edu

*Papers downloaded from **AgEcon Search** may be used for non-commercial purposes and personal study only. No other use, including posting to another Internet site, is permitted without permission from the copyright owner (not AgEcon Search), or as allowed under the provisions of Fair Use, U.S. Copyright Act, Title 17 U.S.C.*

May 1988

A.E.Res. 88-7

**THE ECONOMICS
OF HATCHERY PRODUCED
ALGAE AND BIVALVE SEED**

by

Julia A. Myers

and

Richard N. Boisvert

**Department of Agricultural Economics
Cornell University Agricultural Experiment Station
New York State College of Agriculture and Life Sciences
A Statutory College of the State University
Cornell University, Ithaca, New York 14853-7801**

It is the policy of Cornell University actively to support equality of educational and employment opportunity. No person shall be denied admission to any educational program or activity or be denied employment on the basis of any legally prohibited discrimination involving, but not limited to, such factors as race, color, creed, religion, national or ethnic origin, sex, age or handicap. The University is committed to the maintenance of affirmative action programs which will assure the continuation of such equality of opportunity.

THE ECONOMICS OF HATCHERY PRODUCED ALGAE AND BIVALVE SEED

by

Julia A. Myers and Richard N. Boisvert

ABSTRACT

The bivalve aquacultural industry in the northeastern United States has been strengthened by declining and fluctuating natural harvests in the surrounding coastal region. Hatcheries in which the bivalves are grown to a small seed size have gained acceptance as a link in the aquacultural process. These firms are potentially able to provide a consistent, reliable source of seed to the bivalve aquaculturist.

This research contributes to an understanding of the hatchery industry by examining factors that influence production costs. The research is based on observations of a working hatchery located in the northeast and on current literature. The data were used to develop the framework for a computer program that can estimate variable production costs. This program required the development of an algorithm that simulates the production process. The model sums costs at different stages of animal growth, which are further broken down into smaller units of time.

There was sufficient information available from the literature and hatchery observations to simulate the overall production system, but expenditure records did not separate costs between stages or between hatchery and open-water field operations. The empirical analysis focuses on estimating costs of algae production, because data on this process were more readily isolated from the rest of the hatchery and because feeding costs are recognized as a large proportion of total costs.

The FORTRAN computer program based on the model focuses on the cost of producing algae feed, but as more details become available about other hatchery production processes it would be a simple matter to incorporate them into the existing FORTRAN code. The computer program combines estimates of the cost of algae production with survival and growth rates and feeding efficiencies to determine the costs of feeding a batch of bivalves to various sizes.

The model is applied to raising Atlantic Oyster seed. Twelve initial simulations were made, assuming different values for important parameters. To explore the implications of these feed cost simulations for the marketing of bivalve seed, some rough estimates of total costs of production per bivalve were developed on the basis of others' estimates of the proportion that feed costs are of total costs. Under the most optimistic assumptions, bivalve seed can be produced and marketed profitably. These conclusions must be qualified by the fact that sufficient data were not available from the hatchery under study to estimate all components of total cost. Collection of these types of data would help in further testing of the algorithm as well as contributing to an overall understanding of hatchery costs at each stage of production. Equally important from a research perspective is the need for better data on hatchery survival rates and the size-age and feeding rates.

Table of Contents

	<u>Page</u>
<i>I - INTRODUCTION</i>	1
<i>II - FORMATION OF THE HATCHERY INDUSTRY</i>	2
Background.....	2
Population Dynamics	2
Demand Considerations	3
Alternatives	5
<i>III - HATCHERY PRODUCTION OF BIVALVE SEED</i>	7
The Hatchery.....	7
The Hatchery's Production System	7
The Hatchery's Physical Systems	13
Costs and Input Considerations.....	15
<i>IV - SIMULATING BIVALVE PRODUCTION COSTS</i>	17
Structure of the Model.....	17
Algorithms	19
Algae Price and Quantity	22
<i>V - EMPIRICAL TESTS OF THE SIMULATION MODEL</i>	25
The Data	25
Empirical Results.....	27
<i>VI - SUMMARY AND CONCLUSIONS</i>	35
References.....	37
APPENDIX A.....	39
APPENDIX B.....	83

List of Tables

1. Daily Consumption of Algae Cells by Oysters and Clams.....	10
2. Optimum Algae Concentrations for Oyster Larvae.....	11
3. Inputs to Bivalve Seed Production in the Hatchery	16
4. Production Costs in Various Production Stages in Bivalve Hatcheries of Two Sizes.....	18
5. Estimated Equations for Use in Simulation.....	26
6. Description of Parameters for Alternative Simulation Runs	28
7. Summary of Simulation Output for Initial Runs.....	29
8. Algae Consumption for Selected Simulations	30
9. Summary of Simulations to Test Sensitivity of Survival Parameter	31
10. Sensitivity of Simulation Results to the Age-Size and Feed-Size Relationships.....	32
11. Alternative Estimates of Total Costs of Production (\$ Per Oyster).....	34

List of Figures

	<u>Page</u>
1. Map of Long Island Coastal Region.....	3
2. Commercial Landings from New York Marine District Waters.....	4
3. Long Island Clam Harvest, 1966 - 1982.....	4
4. Production Locations in the Hatchery.....	8
5. Flow Between Hatchery Production Stages.....	9
6. Energy and Water Flow Within the Hatchery.....	14
7. Bivalve Code Flowchart.....	19
8. Algae Price Subroutine.....	23
9. Algae Quantity Subroutine.....	24

THE ECONOMICS OF HATCHERY PRODUCED ALGAE AND BIVALVE SEED

by

Julia A. Myers and Richard N. Boisvert*

I - INTRODUCTION*

Over the past 25 years, bivalve aquaculture in the northeastern United States has become a reliable supplier of marketable oysters and clams. However, to initiate production, the industry requires adequate supplies of bivalve seed. Seed can be supplied from either natural sets or artificial spawning in a hatchery.

A hatchery operator's ability to produce and market seed as an efficient alternative to naturally-spawned seed requires accurate information about the market, the production process and production costs. As with most managers in a new, rapidly developing industry, hatchery operators have difficulty assembling adequate information to make effective investment and production decisions.

The research reported here contributes to an understanding of the economics of artificially spawned bivalve seed by: a) describing the hatchery industry's development and its production processes and b) formulating a model, associated algorithms and computer code by which operators may calculate variable production costs.

Much of the information about production processes, input requirements and

prices was obtained in 1983 from one hatchery on the northeast coast. That being its first full year of operation, the firm's record keeping system was new and not well enough developed to facilitate accurate measurement of the quantities of inputs used at every stage of a bivalve seed's development. However, feed contributes importantly to the total variable cost of bivalve seed (Im *et al.* 1976; Gates *et al.* 1974 and Bolton 1982). Data on algae production were quite good. Thus, the algorithms that calculate algae production and feeding cost are developed in detail. The algorithms can also be used to examine the sensitivity of results with respect to different assumptions about survival rates, growth rates and feeding efficiency.

The remainder of the bulletin begins, in Section II, by placing the hatchery industry within the context of the historical decline in the natural bivalve harvest. Various aspects of the hatchery industry affecting the firm and its costs, such as the industry's information network, the issue of property rights and market structure are also discussed. Section III describes the production processes using one hatchery as an illustration. This section also outlines choices that affect quality, quantity and timing of inputs and outputs and contains a general discussion of production costs. Section IV describes algorithms that simulate the cost of production for each of the system's components. Section V presents the empirical application of the program to the American Atlantic Oyster, including tests using data obtained from the hatchery and sensitivity analyses for the program's important variables. Preliminary estimates of total production costs per bivalve are compared with bivalve seed prices to determine at what age (size) the bivalve seed should be sold. Section VI summarizes the major findings and their implications for a hatchery's production.

*J. A. Myers is a Research Associate at the University of California, San Diego, Laboratory for Mathematics and Statistics, and R. N. Boisvert is a professor in the Department of Agricultural Economics, Cornell University. The research draws extensively on Myers' thesis (1985) sponsored by the New York Sea Grant Institute under a grant from the Office of Sea Grant, NOAA. The cooperation from individuals at the hatchery where data were collected is gratefully acknowledged. The opinions expressed in the report are those of the authors and not necessarily those of the funding agency nor other cooperators in the research.

II - FORMATION OF THE HATCHERY INDUSTRY

In the northeastern United States, the long-standing clam and oyster fishing industry has given rise to the development of the bivalve hatchery industry. Over harvests and increased urbanization, with the consequent decline in the once-abundant natural bivalve population, have forced harvesters to look for alternatives. Among these alternatives are government regulation and incentives, but perhaps the most important alternative is the aquacultural industry, which is replacing traditional methods of harvest. Recently, hatcheries producing bivalves to seed size only have emerged as an alternative to natural bivalve spawnings and also as a link in the aquacultural production of marketable bivalves.

Background

Clams and oysters are harvested in all 14 states along the Atlantic Seaboard (U.S. Dept. of Commerce 1979). The most common oyster is the American Atlantic Oyster (*Crassostrea virginica*). Four species of clam--hard, soft, surf, and ocean quahog--constitute 99% of total U.S. landings by weight. The northern hard shell clam (*Mercenaria mercenaria*) accounts for 53% of the landings, although it constitutes only 17% of their value (Dressel and Fitzgibbon 1978).

Long Island, New England, and the Chesapeake Bay have historically been centers of clam and oyster fisheries (Ritchie 1977; Manzi *et al.* 1982b). Long Island Sound (the waters between Long Island and Connecticut), and Great South Bay, (the waters between Long Island and Fire Island), are the most productive of the New York areas (Figure 1). The seabeds there are hard, shallow and scattered with culch material to which the young animals attach (Korrington 1976). Additionally, the strong tidal currents carry abundant food into the bay (Bardach *et al.* 1972) and the bivalves are protected from many predators because the salinity is lower than in the surrounding ocean water.

Bivalves, especially oysters, have been an important food source for the New York

Bight coastal region. Oysters were used by Native Americans for food and trade and later by the colonialists (Terry 1977). Clams were fewer in number than oysters, but by the 1900's the rise of the cannery industry encouraged the growth of the clamming industry (Dressel and Fitzgibbon 1978).

The rise in clam production was interrupted after red meat shortages of the 1940's put increased pressure on the shellfish industry. Landings in the New York Marine District decreased from 6.8 million to 3.3 million pounds between 1943 and 1945 (Figure 2). Although landings later increased, the industry continued a pattern of boom and bust. In 1976 Long Island harvests reached a peak of 8.4 million pounds, worth about \$16.8 million (Freedman and Morris 1983). Great South Bay was considered the most important hard-shell clam industry in the world. It harvested 80% of Long Island's total landings, which represented 50% of the U.S. total (U.S. Dept. of Commerce 1974). Since then, the Long Island clam harvest has declined; by 1982, harvests were down by 60% from 1976 levels (Figure 3).

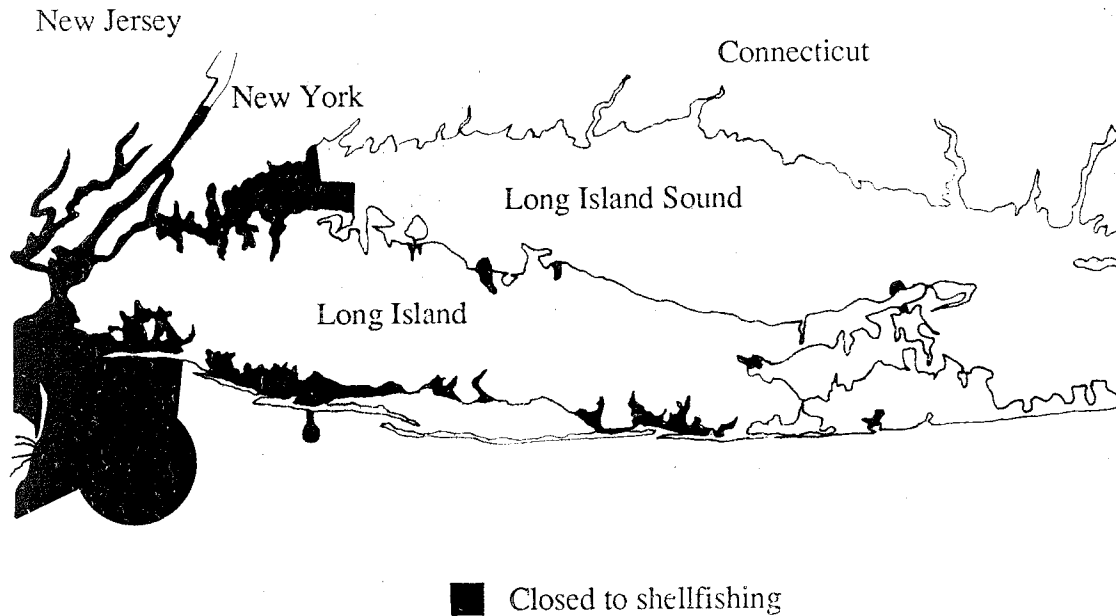
After a peak in 1904, oyster harvests in New York remained fairly stable between 1921 and 1952 but experienced a near-collapse in the 1960's (Figure 2). Revival of the industry through aquaculture has increased landings. Today, oyster production in the New York region is based entirely on aquaculture (Terry 1977).

Population Dynamics

To understand the dramatic changes in the Long Island clam and oyster industries, one must examine the interaction among competing factors that determine the cyclic pattern of bivalve populations. Although population fluctuations occur naturally, they have been exaggerated through over harvest and destruction of natural habitat.

An inherent problem of common property fisheries is overuse (Hardin 1968), which leaves too few animals unharvested for the species to repopulate. This problem is acute around Long Island because bivalves in

FIGURE 1. MAP OF LONG ISLAND COASTAL REGION



Source: Terry (1977)

the cool waters require a long maturation period and because a large proportion of the population must be left unharvested in order to sustain the population. Thus, one major reason for the periodic decline of the clam and oyster populations is the likelihood that harvests will occasionally exceed the maximum sustainable yield.

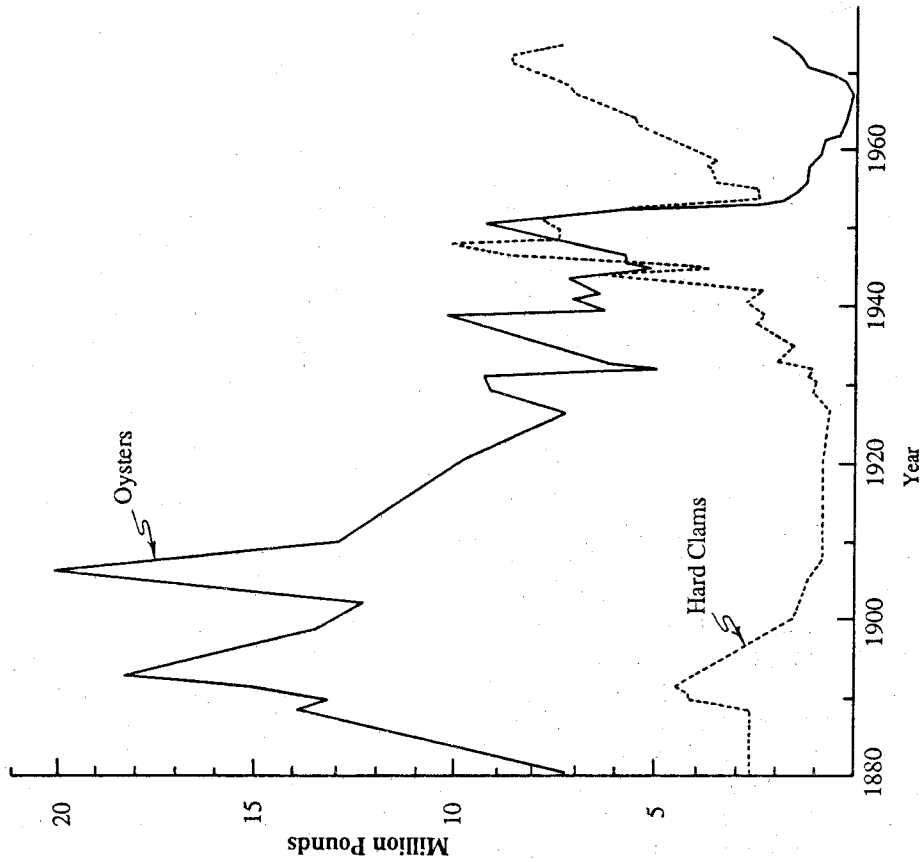
Competition with man has decreased the natural habitat of clams and oysters and adversely affected bivalve populations. As urbanization and shoreline use in the New York area intensified, natural spawning grounds were reduced. Siltation, which is particularly harmful to young spat, increased; and saltwater intrusion caused by the removal of sandbars decreased protection against natural predators (Freedman and Morris 1983). Because toxins accumulate in the flesh of oysters and clams, the increased sewage effluent and agricultural runoff in the area have

increased mortality rates as well as decreased the marketability of the bivalve. The actual numbers of oysters and clams available for harvest have been reduced further through the closure of contaminated natural spawning beds (Figure 1).

Demand Considerations

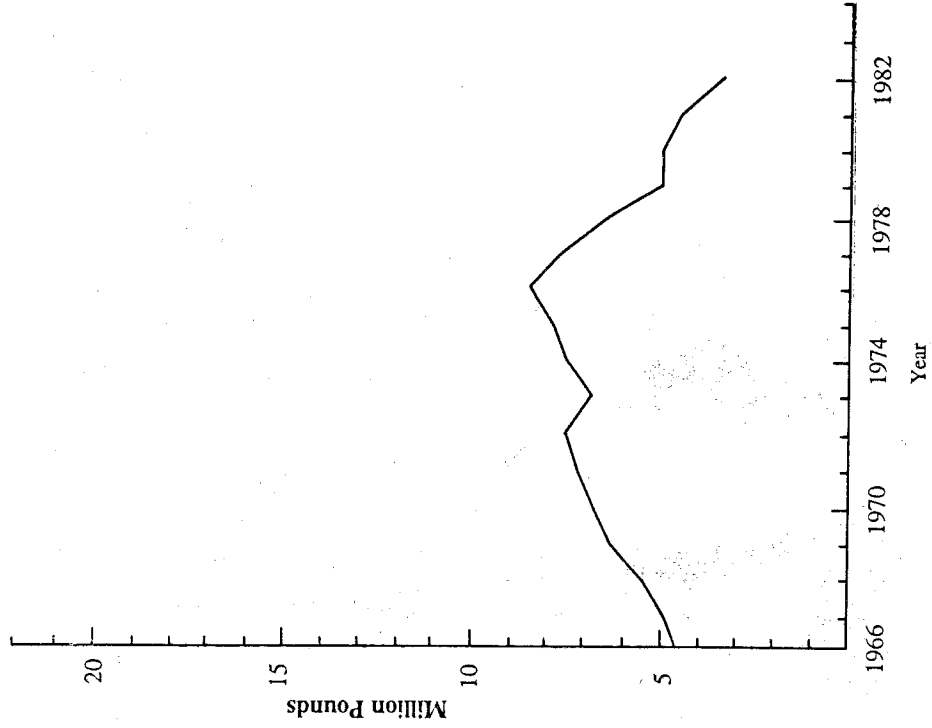
While many factors affecting the bivalve population dynamics have resulted in a reduction in their supply, the industry has also been affected by demand considerations. For example, although the income elasticity of demand for clams is positive, it is relatively low, giving rise to only a slow growth in demand for clams over time due to rising per capita income and increasing human population (Gates *et al.* 1974). The income elasticity of demand for oysters is positive as well, but per capita consumption has actually decreased

FIGURE 2. COMMERCIAL LANDINGS FROM NEW YORK MARINE DISTRICT WATERS



Sources: McHugh and Williams (1976); Terry (1977);
Regional Marine Resources Council (1974)

FIGURE 3. LONG ISLAND CLAM HARVEST



Source: Freedman and Morris (1983)

(Im and Langmo 1977). This decrease might have been caused by either the decrease in supply (Gates *et al.* 1974) and the associated rise in price, or by a change in consumers' tastes due to the fear of eating contaminated animals.

Alternatives

These problems have led to a disappearance of, or change in, the traditional shellfish industry. The scarcity of the hard clam has caused a shift to other species, such as surf clams (Dressel and Fitzgibbon 1978). Production has shifted southward to less-populated regions where competition for habitat is not so keen (Manzi *et al.* 1982a). Also, oyster imports have increased (Im and Langmo 1977).

Options to halt the gradual disappearance of the existing bivalve industry do exist. One option is a comprehensive government management plan to stop the decrease in, or even increase, the natural bivalve populations. Aquacultural production of marketable bivalves and hatchery production of bivalve seed are also viable alternatives, but they both require radical change in traditional harvesting techniques.

Regulation

Because most clam and oyster beds are common-property resources, they have suffered, as most common-property resources have, from over-use and abuse. Although size, time and method of catch are regulated, no consistent policy for predator control or return of culch material to the beds has been in force (Matthiessen 1970). Furthermore, these regulations have often been ineffective because of inadequate knowledge, failure to enforce existing laws, lack of cooperation and an absence of comprehensive planning.

The Regional Marine Resources Council's Shellfish Committee, created in the 1970's, partially filled the void by identifying 18 guidelines for improving the industry (Regional Marine Resources Council 1974). A Long Island clam union exists to aid members of the industry. On the consumer side, iden-

tification of potentially harmful pathogens, public health inspections and enforcement of closed beds will help maintain consumer confidence and increase demand.

To protect the industry, some Long Island governments have purchased young shellfish seed in order to seed public grounds (Smith 1982-83; Goldstein 1983). The population could also be increased by controlling predators, working the bottom for better spawning success, transferring animals to non-polluted grounds and controlling pollution, saltwater intrusion and siltation of habitat.

Unfortunately, implementation of such non-traditional resource policies is politically arduous and requires placation of special-interest groups and cooperation among affected parties. These measures are often expensive to establish and to enforce. The overall benefits to society from an increased bivalve population and the resulting strengthened harvesting industry must be measured against the costs of implementing any resource-regulating mechanisms.

Aquaculture

Aquaculture, the growing of organisms in water under controlled conditions, offers an alternative to traditional bivalve harvesting techniques. The primary benefits of controlling the environment are increased product supplies, although factors adversely affecting quality, such as predators, crowding and parasites, may be reduced.

The viability of an aquaculture industry may depend on the granting of exclusive rights to harvest or to develop a section of the ocean bottom aquaculturally. Marine resources have traditionally been common property (Terry 1977), but sedentary marine life on the bottom may be considered private property, although this right is difficult to substantiate (Hanson 1974; Terry 1977).

The states around the New York Bight area and towns on Long Island have a history of granting bottom leases to private individuals (Kochiss 1974; Terry 1977; Gates *et al.* 1974). In New York, the Department of En-

vironmental Conservation grants leases on the condition that the water quality is good and that no naturally-productive shellfish beds exist (Terry 1977). Great South Bay is held in trust by local townships; harvesters have full bottom rights to certain areas.

In general, economists support private ownership to increase efficiency and productivity (Agnello and Donnelley 1976; Terry 1977). Private beds experience five to ten times greater productivity (Dressel and Fitzgibbon 1978), probably because private harvesters can use less labor-intensive methods than can the public harvesters, who are required by the government to use inefficient harvesting techniques to control harvest quantity. Furthermore, without the assurance of exclusive rights to a harvest, there is no incentive for private harvesters to return culch, to control predators, to establish seed beds or to purchase seed to supplement the natural sets.

As stated above, the viability of the industry is demonstrated by the fact that all oysters on Long Island are currently produced by aquaculture (Terry 1977; Pillay 1976), and all oyster aquaculturists produce some hard clams (Korringa 1976). However, there has been increasing opposition led by local harvesters to the leasing of marine bottom, even of non-productive beds. Few leases have been granted recently (Terry 1977), and if these beds continue to remain in public control and open to all harvesters, the community must either exert better management controls or expect diminishing supply. On the other hand, private ownership might lead to an increase of marketable species but a decrease or disappearance of other naturally-occurring species.

Hatcheries

Potential advantages to hatchery-produced seed include greater reliability of production and better quality control. The development of the hatchery industry (which grows clam and other larvae into bottom-dwelling seed large enough to be planted in open waters for later cultivation or harvest-

ing) has come in response to natural reproduction failure in open water (Loosanoff and Davis 1963a; Henderson 1978). Hatcheries can potentially achieve better quality control, decreased mortality and faster growth (Bardach *et al.* 1972; Donohue *et al.* 1981), thus offering aquaculturists a reliable source of bivalve seed. Furthermore, local governments can supplement declining natural spawns with hatchery-produced seed, a practice local harvesters generally favor as an alternative to direct governmental support of open-water growout in areas where a few individuals can affect the local bivalve market (Miller 1977; Rhodes 1974).

Worldwide demand for oyster and clam seed is growing at an increasing rate because of decreasing natural stocks and improving hatchery techniques. Henderson (1978) has estimated the annual world market for hatchery seed to be \$2 billion, or 25% of the annual U.S. whole clam and oyster harvests. In the U.S., hatcheries exist in the Southeast, the Northwest and in parts of the Northeast. The U.S. leads in seed production, having pioneered many clam hatchery techniques, but its oyster techniques are still considered "primitive" (Bardach *et al.* 1972).

Opinions differ as to the sufficiency of seed availability. Clam seed is more readily available than oyster seed (Terry 1977), except for seed larger than 10 mm (Castagna and Kraeuter 1977; Manzi *et al.* 1982a). Supplies of oyster seed have proven unreliable and insufficient to meet demand at current price (Donohue *et al.* 1981; Im and Langmo 1977). The demand for hatchery-produced seed is cyclical and increases during natural set failures (Henderson 1978). If hatcheries could produce a reliable, inexpensive product that is more attractive to growers than natural seed, they could capture a larger share of the seed market.

Hatcheries currently operate with less than complete information for two reasons: a) lack of cooperation between affected parties in the industry (Henderson 1978); and b) inasmuch as no two hatcheries are operated exactly alike (Terry 1977), such information as is available from one is not entirely appli-

cable to another. Partly due to this lack of information, hatcheries are only marginally cost-effective (Terry 1977). Cooperation should be improved between hatchery operators and buyers, researchers and the government (Henderson 1978). This in turn should affect the flow of information and help industry development.

Despite the problems faced by hatcheries, many in the field believe that they will continue as a viable industry (Terry 1977). Genetic breeding and development of domestic bivalve strains are future potentials of the industry (Terry 1977). The development and success of the hatchery industry depends on the continued demand for seed, encouragement by government and the scientific community and the success of hatchery operators. The success of the individual operator depends in large part on an ability to compete in the market place. To be competitive, the hatchery operator must be able to make the best production decisions based on an accurate assessment of production costs. This is the subject of the following sections.

III - HATCHERY PRODUCTION OF BIVALVE SEED

Raising oyster and clam seed under controlled conditions requires technically-sophisticated, delicately-monitored production processes. The purpose of this section is to describe these processes, using the hatchery from which data were collected as a model. Although production systems vary among hatcheries, especially in algae and culch preparation and in facility and system designs, most operations are universal to all hatcheries. Techniques are varied primarily to take advantage of available natural resources at the site.

The Hatchery

The bivalve seed hatchery, from which data were collected for this study, was the outgrowth of an experimental plant begun in the late 1970's. This model facility was used

to help train staff and develop techniques utilizing the available natural resources in the surrounding area.

Construction of the permanent hatchery was completed in 1982. The building used for seed production has approximately 500 square meters of indoor floor space. There is an ample outdoor work area, as well as room available for expansion. The hatchery began partial production in 1982, and the first full production year was 1983. Both oysters and clams are grown, but production is now concentrated on oysters.

The hatchery is located a few hundred yards from a bay. The ownership of the bottom rights to part of the bay led to the development of a field operation ("outgrow") where some of the seed produced in the hatchery is grown to maturity. Most of these outgrow operations are conducted separately from those of the hatchery.

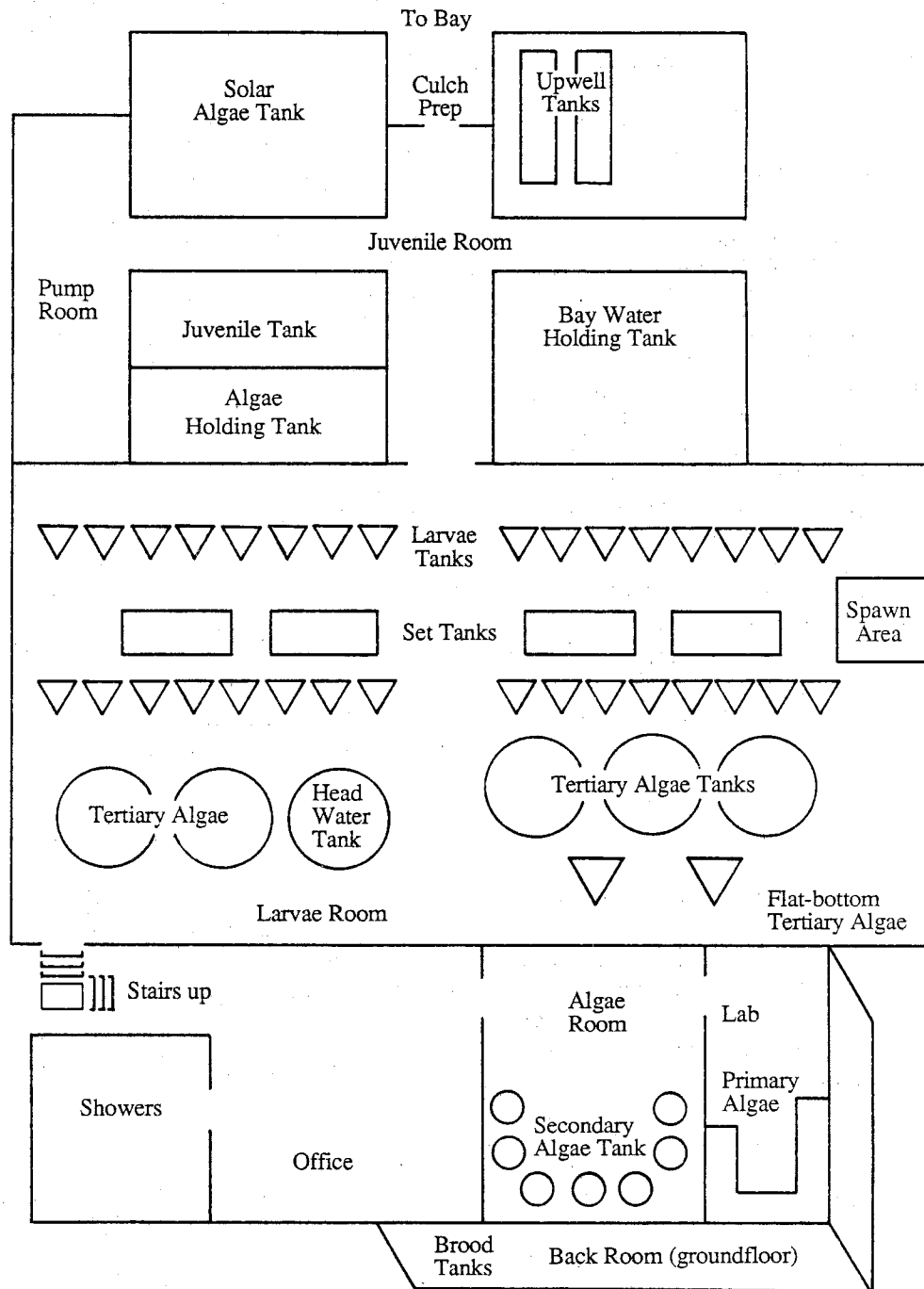
The Hatchery's Production System

The production process can be divided into six life-cycle stages for the bivalve:¹ 1) broodstock maintenance and conditioning; 2) spawning; 3) larvae development; 4) setting; 5) juvenile development and 6) field (or outgrow); and two support stages: 1) algae production and 2) culch production. Figure 4 shows the location of each production stage within the hatchery.

To maintain a low mortality rate, young bivalves must be treated with care during all phases of production, including inspection, cleaning and transfer. Exposure to severe environmental conditions or fouling by foreign organisms, such as fungus, bacteria and predators, can cause death or deformation of the animals or a poisoned product. Thus, particular attention must be given to cleanliness at each stage of production and to selec-

¹ Because the differences between clam and oyster production are minimal, the two processes will be treated as one throughout the remainder of this analysis.

FIGURE 4. PRODUCTION LOCATIONS IN THE HATCHERY



tion of non-toxic materials for all equipment that comes in direct contact with the animals. The quantity and quality of inputs used, such as energy, algae, culch and other materials, depend on the biological requirements of the

bivalve, the existing environmental conditions and on the hatchery's ability to purchase or produce those inputs. Inadequate or poor quality inputs will affect the survival of the bivalves.

As with most biological production processes, there is a trade-off between achieving maximum production and the cost of inputs. For instance, higher densities improve efficiency in the use of inputs such as water and labor; however, a high density decreases the survival and growth rates of the existing animals. The hatchery must weigh the marginal cost of providing ideal environmental conditions against the resulting increase in the value of the marginal output.

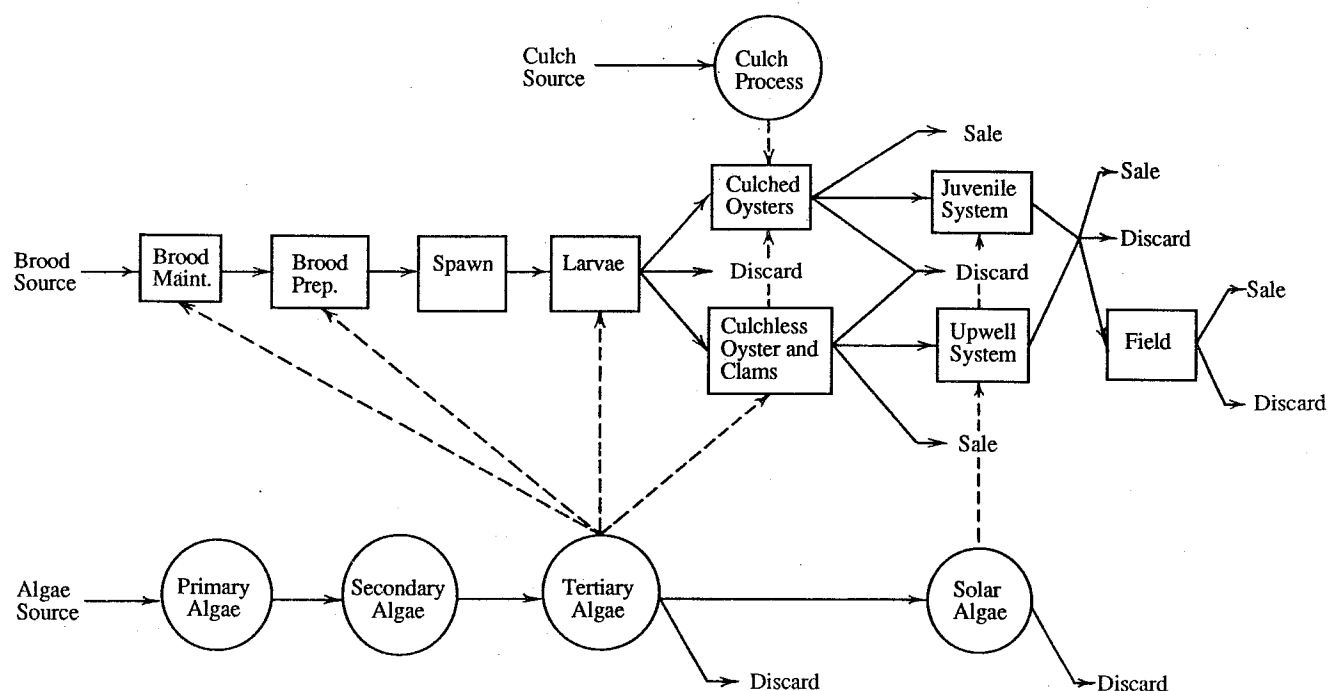
Proper timing of production is critical in two respects. First, if the hatchery begins production too early in the season, the animals will be ready for transfer to the juvenile tanks or outgrow facility before the water has reached a suitable temperature. Extending production too late may leave the hatchery with unsold inventory. Second, new spawnings and transfers between stages must be timed so as not to exceed the system's capacity in terms of space or ability to produce sufficient inputs. The optimal age of transfer

depends in part on the cost of maintaining the animal. The best way to understand the production system is to examine each of the components separately. (Figure 5 shows the flow between the production stages.)

Algae

Production of algae for feed in sufficient quantity and quality is important to the bivalve at all stages of growth and is one of the principal problems in profitably growing animals to market size entirely within a closed system (Gates *et al.* 1974; Epifanio 1975). Not much is known about the specific food requirements of the bivalve (Galtsoff 1964), especially about the adult bivalve (Epifanio 1975), but in the work that has been done, bivalve algae consumption is measured either by the number of cells consumed per animal or by the algae concentration made available to the animal. Existing estimates of the number of cells consumed by

FIGURE 5. FLOW BETWEEN HATCHERY PRODUCTION STAGES



oysters and clams at different stages of animal growth are in Table 1. Using the second and most common method, optimum algae concentrations for oyster larvae are given in Table 2.

The amount of algae assimilated, i.e., actually used for growth by the animal, is estimated at between 70% and 90% of the amount consumed (Epifanio 1975). Consumption is less than the filtering rate (the cells filtered per volume of water pumped), and the filtering rate in turn is less than the pumping rate. Filtering efficiency is affected by environmental conditions and was measured at between 34% and 58% by Baab and Associates (1973).

For bivalve feed, a mix of algae species is preferable (Ukeles 1975). The digestibility and nutrient value of the food depends on the chemical composition, size, texture, taste (Epifanio 1976), cell wall and toxicity (Ukeles 1975) of the algae species. Juveniles are less sensitive to algae quality and

size than are larvae, which require cells sized from 6-10 microns in diameter (Breese and Malouf 1975). Clams are generally less particular than are oysters about the quality of algae fed (Ukeles 1975).

The quantity of algae that the hatchery must produce to feed the bivalves depends on whether or not sea-water containing naturally occurring algae is made available to the animals. The proportion of its diet an animal is able to receive in this way depends on the concentration of algae in the seawater, on the pumping and filtering abilities of the animal, on the seasonal variation of that concentration and on the amount of water to which the animal is exposed. Also, some additional algae must be produced and held in reserve in case main supplies are contaminated. Producing excessive amounts can be costly, because algae can be stored only a few days.

The hatchery's algae stage is managed by a single employee, who spends more than one-half the total working hours exclusively

Table 1. Daily Consumption of Algae Cells by Oysters and Clams

Stage/Age of Oyster	Equivalent Stage at the Hatchery	Algae Cells Consumed Per Day	
		Oysters	Clams ^a
Larvae			
2 days		1.0×10^4	
12-15 days	Larval	4.0×10^4	
set size		5.0×10^4	
Spat			
3 weeks	Setting	2.0×10^5	1.0×10^5
8 weeks		3.0×10^6	
		2.4×10^7	
Seed	Juvenile	8.0×10^7	6.0×10^5
Adult	Juvenile/Outgrow		1.0×10^7
3 inches		1.1×10^8	

Sources: Claus and Adler (1970); Matthiessen and Toner (1966); Gates et al. (1974).

^aThe three consumption levels for clams refer to animals of 0.2, 2.0 and 8.0 mm in size, respectively. The corresponding stages for the oysters listed at the left are probably not exactly comparable to these sizes.

Table 2. Optimum Algae Concentrations for Oyster Larvae

Initial Larvae Shell Length (μ)	Food Concentration for Highest Growth Rate (μ l of packed cells/l)
77	2.5
78	2.5
104	10.0
104	10.0
139	10.0
146	20.0
201	40.0

Source: Rhodes and Landers (1973).

on algae production. The batch method, in which algae is cultured to a maximum density using successively larger containers, is used. Not all algae at a maximum age and density are used exclusively for food; instead, some are used to inoculate a larger tank.² This "modified batch" strategy requires additional labor (Matthiessen and Smith 1979).

The second-floor laboratory is where algae samples are analyzed for contamination and proper densities, nutrient solution is prepared to supplement the algae cultures, and initial algae stock is stored. Four different varieties are cultured: *Thalassiosira pseudoma* (diatom), *Pavlova lutherii*, *Pavlova lutherii* *Hoptaphyceae* and *Isochrysis galbana*. The latter represents 50% of the algae grown and 50% of the animal's diet. Because young animals require small algae cells, the diatom is used only during the juvenile stage.

² In contrast to techniques that centrifuge large quantities of seawater to concentrate naturally-occurring algae, this method selects only the best species of algae. The centrifuge process, which is commonly used on Long Island, is less expensive (Matthiessen and Smith 1979), but it requires an abundant and reliable source of naturally-occurring, desirable algae. The batch or modified batch process offers certain advantages for feeding efficiency because higher densities of algae can be obtained. With these processes, algae can be produced at densities of 10^4 cells/ml, as compared to densities in seawater of 10^4 cells/ml (Gates et al. 1974).

The primary algae production stage begins with tube cultures. Algae, in nutrient-rich well water, are allowed to mature two weeks to a maximum density of 3×10^5 cells/ml. The culture is then transferred to sterilized flask containers. After one week, this culture matures and is transferred to carboys. Water must remain cool and sufficiently aerated. Cool fluorescent lighting provides the algae with artificial sunlight 24 hours each day.

The secondary stage, located in the algae room next to the laboratory, consists of seven 757-litre tanks. Each tank is inoculated with 16 litres of primary algae culture; nutrients and well water are added. The number of tanks used depends on near-term food requirements. Optimum algae density of 4×10^5 cells/ml is reached in one week.

The secondary culture flows to the ground floor to inoculate the 3,785-litre tertiary algae tanks. The requirements at this stage are similar to those for previous cultures; and when a maximum density of 5 to 8×10^5 cells/ml is reached, the tertiary, or "direct-feed", stage is used to feed the bivalve broodstock and the animals at the larvae and set stages. It is also used to inoculate the next culture stage. After one week, any algae remaining are discarded.

The final algae tank is a sunken 49,205-litre tank in the juvenile room. This "solar" tank is inoculated from the tertiary tanks and allowed to bloom under natural sunlight provided by large, sloping windows over the tank. Maintenance is similar to that in other stages. When a suitable density is reached, the entire amount is transferred to the algae holding tank located in the same room; the solar tank is then reinnoculated. The juvenile bivalves are fed directly from the holding tank. After one week, the remaining algae are discarded.

Broodstock Maintenance and Conditioning

The quality of the bivalve produced depends to a degree on the genetic quality of the parental stock, which is selected on the

basis of size, shape and growth potential. The broodstock is kept in rectangular 825-litre tanks of aerated, filtered water. Cool water temperatures are maintained to prevent spontaneous reproduction.

Before spawning can occur, the mature adults must be "conditioned"; the water temperature is increased to 26-29°C, and the food supply is also increased. The conditioning time varies according to the temperature and the amount of food used, but oysters require approximately 35 days (Gates *et al.* 1974).

Spawning

Once the male and female bivalves have been conditioned, the spawning process takes two employees one to three hours to complete. Conditioned adults are induced to spawn by either introducing a sharp temperature increase or by adding previously-collected sperm (Breese and Malouf 1975). One female clam can produce as many as 24 million eggs at a time (Loosanoff and Davis 1963b). Within 24 to 48 hours, the gametes become free-swimming larvae.

Larvae Development

During the larvae stage, algae-rich seawater is circulated about the mobile animals ("veligers"). In the larvae room are 38 conical, 340-litre larvae tanks and two 700-litre, flat-bottom larvae tanks. These tanks are filled with warm, filtered seawater, which must be heated when seawater temperature is below 25°C. Approximately 14 litres of algae are added to each tank/day.

Oyster and clam larvae are maintained close to ideal densities for fastest growth and survival, viz., densities from 0.4 to 15 animals/ml and 5 animals/ml of water, respectively (Gates *et al.* 1974; Matthiessen and Toner 1966). Growth and survival rates at this and subsequent stages are affected not only by density and by food quantity and quality, but also by the temperature and available oxygen. Thus, water is maintained at the ideal temperature, and aeration is supplied continually to all tanks.

Maintenance of the larvae is labor-intensive. Besides daily feeding and monitoring, every 48 hours the tanks are cleaned, the larvae are sorted by size, and the dead animals are removed. This takes two employees nearly a full day to complete, depending on the number of tanks in use. Animals of similar size are collected together from different tanks and are then returned to a newly-filled tank, and algae are added. This culling process is an essential feature of the production process because it, in large part, determines the quality and the quantity of the final output.

The most critical period for the larvae occurs at metamorphosis, when the bivalves change from free-swimming larvae to bottom-dwelling spat. Setting occurs at 7 to 10 days for clams (Bardach *et al.* 1972; Gates *et al.* 1974) and 10 to 20 days for oysters (Gates *et al.* 1974). The clams at this stage measure 200 to 215 microns (Bardach *et al.* 1972; Gates *et al.* 1974) and oysters 300 to 325 microns. Prior to this metamorphosis, the animals must be transferred to the set tanks, which have been filled with culch.

Culch

Culch is the material onto which the oyster larvae set and attach permanently. Thus, before the larvae can be transferred to the setting tanks, culch is prepared. The choice of culch material depends on the type of grow-out anticipated rather than the relative cost of the materials (Matthiessen 1970; Gates *et al.* 1974).³ Before 1960, whole oyster or clam shell was used exclusively for bottom culch material (Matthiessen and Smith 1979); other techniques, such as the suspended-string method, could be used in lieu of bottom culch. Today, choices for culch include a variety of shell sizes, artificial materials and "culchless" oysters.

The culchless oyster is an oyster with no attached culch, that is, one that has been induced to set on a material, such as plastic

³Clams do not require culch since they do not attach firmly.

sheeting, from which it can be removed easily. Culchless oysters can be grown at greater densities than the culched variety inasmuch as clumping, which crowds and deforms the spat on the culch, is less of a problem. Culchless oysters may be less expensive to produce because culch production costs are avoided, and because shipping and handling of the culchless oyster is less costly. However, once transplanted, the culched oyster has a better chance of surviving, especially in waters with high siltation (Matthiessen 1970). Thus, culchless oysters (and clams) should be raised to larger sizes in the hatchery, potentially offsetting any other savings.

The preparation of culch is tedious, taking about 12 minutes to produce one litre of sorted, crushed shell. The prepared culch is then cleaned and spread on the bottom of the set tanks at densities of 200 to 350 ml/square foot.

Setting

Prior to setting, the larvae are transferred to eight 825-litre tanks filled with warm, filtered, aerated water at densities of 2 to 18 million animals per tank. Maintenance is similar to that for larvae; the same algae mix is used, although cleaning is less frequent (every 72 hours). The animals are fed 70 litres of algae/day/tank during the first week; this is increased to 140 litres by the second week.

Because the animals are delicate at this stage of their life, their transfer too early to the juvenile tanks could reduce the overall survival rate. The hatchery does not heat the juvenile tanks and must wait for the bay water to heat sufficiently by natural means before young spat can be transferred. However, if the animals remain too long in the set tanks, they cannot receive the large quantities of flowing water and algae necessary for rapid growth, nor can the tanks be used for subsequent batches. Animals as small as 300 microns and as large as 1.9 mm have been transferred out of the set tanks in the hatchery.

Juvenile Development

From the set tanks, the young spat are transferred into either the juvenile tanks or the upwell system, or they are sold. Most oyster and clam seed are sold at about 3 mm; smaller animals than this would experience too low of a survival rate.

The juvenile tank has a capacity of 26,500 litres. The culched oysters are placed in trays that measure 0.5 x 0.6 m; these trays are then placed on the bottom of the juvenile tank. The tank is filled with enough sieved bay water to cover the trays and is aerated. Approximately 75 litres of algae are added per day to achieve a cell count of 2×10^6 cells/ml. The tank is cleaned twice per week, at which time the animals are washed and culled for death, fouling, and deformation. By the time the animals are ready for sale or transfer to the field, there has been a significant reduction in their numbers.

The upwell system is used only for clams and culchless or micro-culched oysters. The spat are held in screened-bottomed buckets, which are suspended over 825-litre rectangular tanks. Water from the bay holding tank is siphoned into these tanks, creating a continuous flow of water that aerates the spat. This sieved water is at bay temperature: 17°C in May and 28°C in August. Algae from the holding tank are added at a rate of about 7,600 litres per day. Animals have been kept in this system until they have reached 2.3 to 8 mm in diameter, and then they have been sold or transferred to the field.

The Hatchery's Physical Systems

The facility was designed for efficiency and flexibility to allow for changes in production techniques in response to new scientific information, environmental conditions or market changes. Simple changes in internal arrangements, made possible by movable tanks that can be used for a variety of functions, can streamline production. The hatchery was designed to facilitate easy cleaning. Energy, water and air systems were coordinated to maximize efficiency.

Energy System

The amount and the type of energy used in the hatchery depend on the production method and on the prevailing environmental conditions. Natural lighting and heating are used wherever possible. Because the bay water temperature is cool, the hatchery must heat large quantities of water to fill the brood, larvae and set tanks.

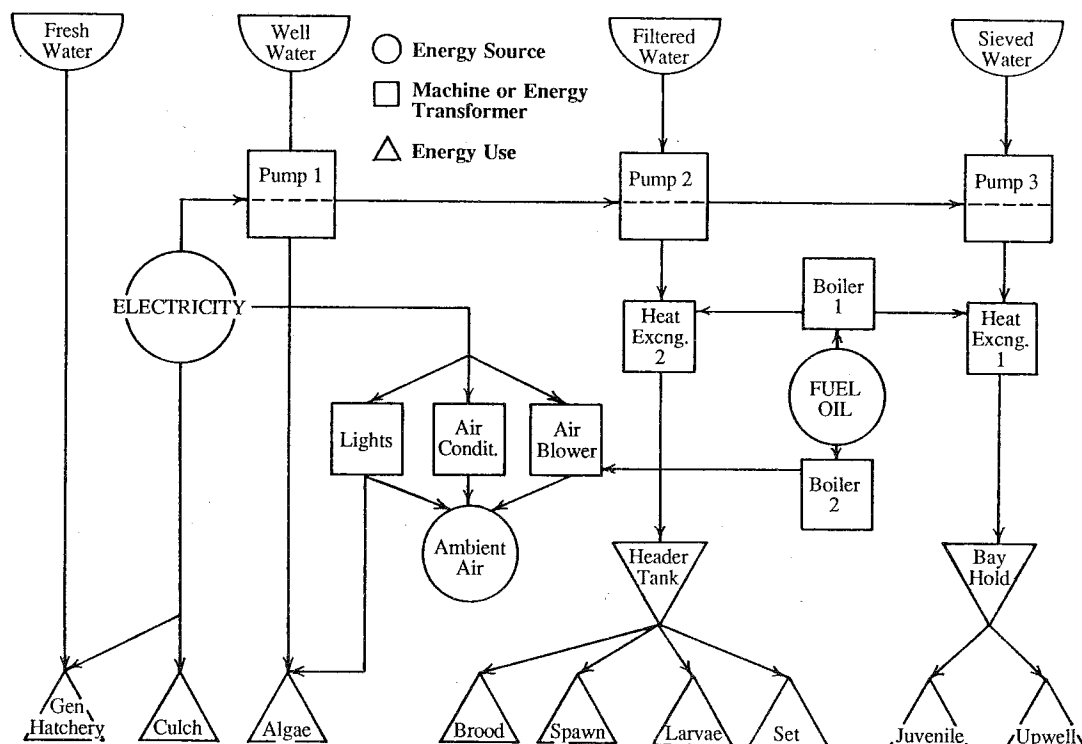
The energy flow is complex (Figure 6). Heat inputs include electricity for lighting and machinery and fuel oil for running the boilers to heat water and ambient air. Cooling inputs include air conditioners, bay water and ambient air. The relative contribution required from each of these sources to maintain ideal water and air temperatures is a function of the season. It is possible to determine the amount of electricity and fuel oil required if initial temperatures, final temperatures, and heating and cooling capacities are known.

Water System

Water is used in all phases of production: for all animal stages, for algae production and for cleaning. Water, the quality of which is critical to all hatchery operations (Bardach *et al.* 1972), can be sterilized with ultraviolet light or with chemicals. Fresh water for cleaning is obtained from a reservoir at no cost to the hatchery, and three pumps bring water from the bay (Figure 6). The first pump is for deep-well water, which is used exclusively for algae production. It is the purest of the three bay waters and is never heated. Pump 2 filters out particles greater than 1 to 5 microns; this water is used for larvae, brood and set tanks. The third pump delivers water to the juvenile room and sieves out particles greater than 700 microns.

The facility is designed to minimize the electrical costs of pumping. By using a series of different levels, water can be siphoned for various uses once it is within the

FIGURE 6. ENERGY AND WATER FLOW WITHIN THE HATCHERY



facility. Gravity is also used for the transfer of algae from the secondary to the tertiary stages and from the tertiary to the solar stage.

Air Flow System

If water is not kept in constant flow, the hatchery must provide aeration. Although the hatchery's air system is not elaborate, it is essential to algae and animal growth. Depending on size, all tanks are provided with a certain number of air tubes, except for the upwell system, which receives a constant flow of water. Air is pumped through these tubes from an air compressor that operates 24 hours per day. To reduce electricity consumption, the hatchery replaced its 5-HP air compressor with a 1-HP compressor, which has proven sufficient for the hatchery's needs.

Costs and Input Considerations

Before the decision to build a hatchery is made, a complete financial assessment should be conducted, including estimates of the initial investment cost, operating expenses and expected earnings. To date, existing hatcheries were estimated to have an average initial capital investment for building and equipment greater than \$100,000 (Matthiessen and Smith 1979), but technically-advanced hatcheries, such as this one, may cost even more. Using land with direct access to the ocean can add considerably to the total price, although in this case, the land was already owned.

Hatchery operating costs were estimated at between \$2,700 and \$10,000 per month for most facilities (Matthiessen and Smith 1979). The actual amount depends on the hatchery's output and design, on the existing environmental conditions, and on unforeseen production risk beyond the control of the operator.

To date, there are only a handful of published estimates on the cost of bivalve seed, and the average costs per animal range from \$0.0005 (Im and Langmo 1977) to \$0.01 (Matthiessen 1979) for smaller seed and from \$.029 to \$0.25 (Gates *et al.* 1974) for larger seed. The most comprehensive estimates have been made in recent years by Im and Langmo

(1977), Im *et al.* (1976) and Bolton (1982). They have attempted to estimate costs at each stage of production (e.g., brood, larvae etc.) using an economic engineering approach based on small, experimental hatcheries both on the Pacific and Atlantic coasts.

These studies are an important first step, but they fail to determine the cost of producing animals to different sizes in the same stage. This information is particularly critical at the final stage, that is, just prior to sale or transfer to outgrow facilities, because an aquaculturist purchases seed from a hatchery at a size in accordance with his production method, locality and production goals. It appears that 3 mm is a common size at which to sell the spat. Those who purchase larger seed face significantly less risk of mortality in open waters since they are more resilient to predation, siltation and climatic changes (Matthiessen and Smith 1979).

When larger seed are produced, it is usually because a) the hatchery cannot find buyers for seed not previously contracted for sale (usually the hatchery overproduces some animals to leave a margin of error), b) environmental conditions are such that the hatchery cannot transfer seed to their own "growout" facility or c) it is experimenting with growth techniques for larger animals. In other words, production to a larger size is not necessarily desirable but may be preferable to disposal of the animals if the hatchery can cover its costs at the margin.

Classifying Productive Inputs

An important first step in estimating the cost of raising bivalve seed to different sizes is to identify the input requirements for each stage of production. As outlined in Table 3, production of bivalve seed can be separated into five production stages and two support stages: 1) broodstock conditioning, 2) spawning, 3) larval development, 4) setting, and 5) juvenile development; and 1) culch and 2) algae preparation. Because input quantities change over time, they will also change over the duration of most stages. Therefore, in this analysis, the stages are broken down into smaller units of time, or periods. At the end of a period, there is an associated size and age

Table 3. Inputs to Bivalve Seed Production in the Hatchery

Inputs	Stages ^a								General Hatchery	Inputs	Stages ^a								General Hatchery
	Production					Support		Condition			Production					Support			
	1	2	3	4	5	1	2				1	2	3	4	5	1	2		
ELECTRICITY																			
Water pumping																			
Wellwater								x											
Filtered water	x	x	x	x															
Sieved water	x				x	x													
Air compressor	x	x	x	x	x	x		x											
Air cooling																			
Lab room									x										
Algae room									x										
Air heating									x										
Office																		x	
Algae room																			
Larvae room			x	x	x														
Back room	x			x															
Equipment																			
Office									x										
Laboratory									x										
Autoclave									x										
Crusher						x													
Sorter						x													
Lighting																			
Office																		x	
Laboratory									x									x	
Algae room									x										
Larvae room									x										
Back room									x										
Juvenile room									x										
FUEL OIL																			
Water heating																			
Heat exch.	x	x	x	x															
Filtered water	x	x	x	x															
Sieved water						x	x												
Space heating																			
Office																			
Algae room																		x	
Larvae room			x	x	x														
Back room	x			x															
LABOR																			
Tank maintenance									x	x	x	x	x	x				x	
Culling										x	x	x	x	x					
Feeding									x		x	x	x	x				x	
Transfers											x	x	x	x				x	
Cleaning																			
Management																			
Laboratory work									x	x	x	x	x	x				x	
Equipment maintenance																			
MATERIALS																			
Antibiotics																		x	
Cleaning																		x	
Glassware																		x	
Lightbulbs																		x	
Screening																		x	
Office supplies																		x	
Fresh water																		x	
Nutrient solutions																		x	
ALGAE																			
Direct-feed tanks									x		x	x							
Solar tanks												x	x	x					
From natural seawater														x	x				
CULCH																			
Micro													x						
Large													x						
MISCELLANEOUS																			
Travel and communications																			
Insurance																			
Permits																			
Shipping														x	x				
Loan payments										x	x								
Taxes																			

^aSee text and Figure 5 for identification of stages. General hatchery inputs are independent of stage. Intermediate inputs are produced in support stages.

of animal, as determined by the growth rate experienced in the hatchery. The cost per animal at any age can be estimated by measuring the inputs that are used in a period.

As with most production systems, inputs can be classified as either fixed or variable. Fixed inputs are those whose quantity cannot be changed during the length of time under consideration since the cost of quick

variation in their amount is large as to make such variations impractical (Mansfield 1979). Here, we are interested in smaller seed (produced in less than 4 months) as well as in larger animals that must be left in the hatchery longer. For convenience, the length of time under consideration is taken to be one year. Thus, fixed costs are assumed to include depreciation on building and equipment, taxes, and rental charges.

Some inputs are also fixed once a batch is started and so either do not vary with output or vary insignificantly. These inputs include general hatchery maintenance, such as for machinery and general hatchery cleaning. It is difficult to isolate these costs in terms of their applicability to any particular stage or period of the bivalve's life.

Variable costs differ in how much they are affected by the level of production output. First, because of the nature of hatchery production, which uses a modular-style (e.g., separate tanks), many inputs are only secondarily related to the number of animals within the system. For example, a larvae tank requires a certain amount of labor for cleaning and a certain amount of heated water no matter how many animals are in each tank even though the number of tanks used does depend on the total number of animals. Second, some inputs are related to the area where the animals are located and must be used no matter how many tanks are in that area, such as space heating. Third, some events take place only once and are not significantly affected by the number of animals, for example, sampling. Finally, there are those inputs, such as algae consumption, that vary directly with output.

It is the inputs that vary according to the number of animals in the system that are of greatest interest to this analysis, since it is these inputs that vary most with animal size. Inputs that do not vary directly with output are shared according to the relative length of time that the animal is retained in the hatchery, rather than the total output or the animal's size.

For this study, inputs are divided into six primary categories: electricity, fuel oil, labor, materials, algae and culch (Table 3). Although both algae and culch are support stages in the system, they are intermediate inputs that could be produced outside the hatchery and purchased.

In most hatcheries, it is not difficult to separate the total monthly operating costs into the total cost of each primary input. The real difficulty is in allocating a share of these primary input costs to different stages of pro-

duction. Im, Johnston and Langmo (1976) made some progress on such a breakdown in costs. A summary of their results is presented in Table 4. These data highlight the importance of algae production as a component in the variable costs of bivalve production and serve as a benchmark by which to calibrate the results from the bivalve cost simulation program described in the next section.

IV - SIMULATING BIVALVE PRODUCTION COSTS

This section describes a simulation model by which one may calculate input requirements, net output and the cost of growing bivalve seed. Within the model, primary input costs are allocated to different stages of production, thus facilitating the comparison of the marginal cost of growing bivalve seed to a larger size with the expected marginal revenue of doing so. The model was designed to facilitate its coding as an interactive computer program that could be used by hatchery operators.

Structure of the Model

The model can be used to calculate production costs, inputs and output for an entire batch of bivalves. Even though most hatcheries would have concurrent batches at different stages of development, focusing on a single batch makes it possible to allocate input costs to the different production stages and to generate the information necessary to determine the optimal timing of production. It is also assumed that only one production pathway is followed for each application of the model; that is, all animals in all stages are treated similarly regarding the choice of culch, timing of transfers and all other aspects of the production processes.

To compute inputs and costs for animals of different ages, each batch is initially divided into the production stages discussed above. These stages are subdivided into shorter periods so that costs can be determined at various points in time within a stage. This is particularly useful for long stages, such as the juvenile stage, where a more precise estimate of the change in cost with age is required. All periods within a

Table 4. Production Costs in Various Production Stages in Bivalve Hatcheries of Two Sizes^a

Stage	Percentage of Total Cost		Percentage of Variable Cost	
	Plant 1	Plant 2	Plant 1	Plant 2
Conditioning	2.7	0.2	3.9	0.2
Spawning	5.0	0.3	8.7	0.3
Larvae Rearing	15.0	27.8	21.4	25.1
Larvae Setting	7.2	18.8	7.6	13.5
Algae Production	38.9	33.3	29.4	39.0
Culch Preparation	3.7	9.3	7.2	14.7
Miscellaneous	27.5	10.3	21.8	7.1

^a The source of the data is Im, Johnston and Langmo (1976). Plant 1 is an experimental plant with an output of 15 bushels/week. Plant 2 is scaled up to a projected 800 bushels/week.

stage are of equal length, but the length and number of periods may differ amongst stages.

The number of bivalves surviving and their size are calculated at the end of each period, along with the per day averages over the period. The quantities and costs are calculated for each period and are averaged over each period as well. The costs incurred in the current period are added to those of previous periods to accumulate total costs up to the end of the period. This type of structure, in which there is a succession of repeated computations, lends itself easily to FORTRAN, the computer language in which the model is coded.⁴

As shown in Figure 7, the options to run the code are chosen in *step 1a*, and data, including price of inputs, are entered or computed in *steps 1b-1g*. After all data are entered, a batch is begun on the first day of the first month of production (*step 2*). Each stage *i*, beginning with the conditioning stage, is looped through *steps 3-14*, and within this stage loop, period *j* is incremented (*steps 5-13*). The passage of time during the course of a batch is accumulated within *j* (*step 6*), and thus the age of the bivalves is advanced.

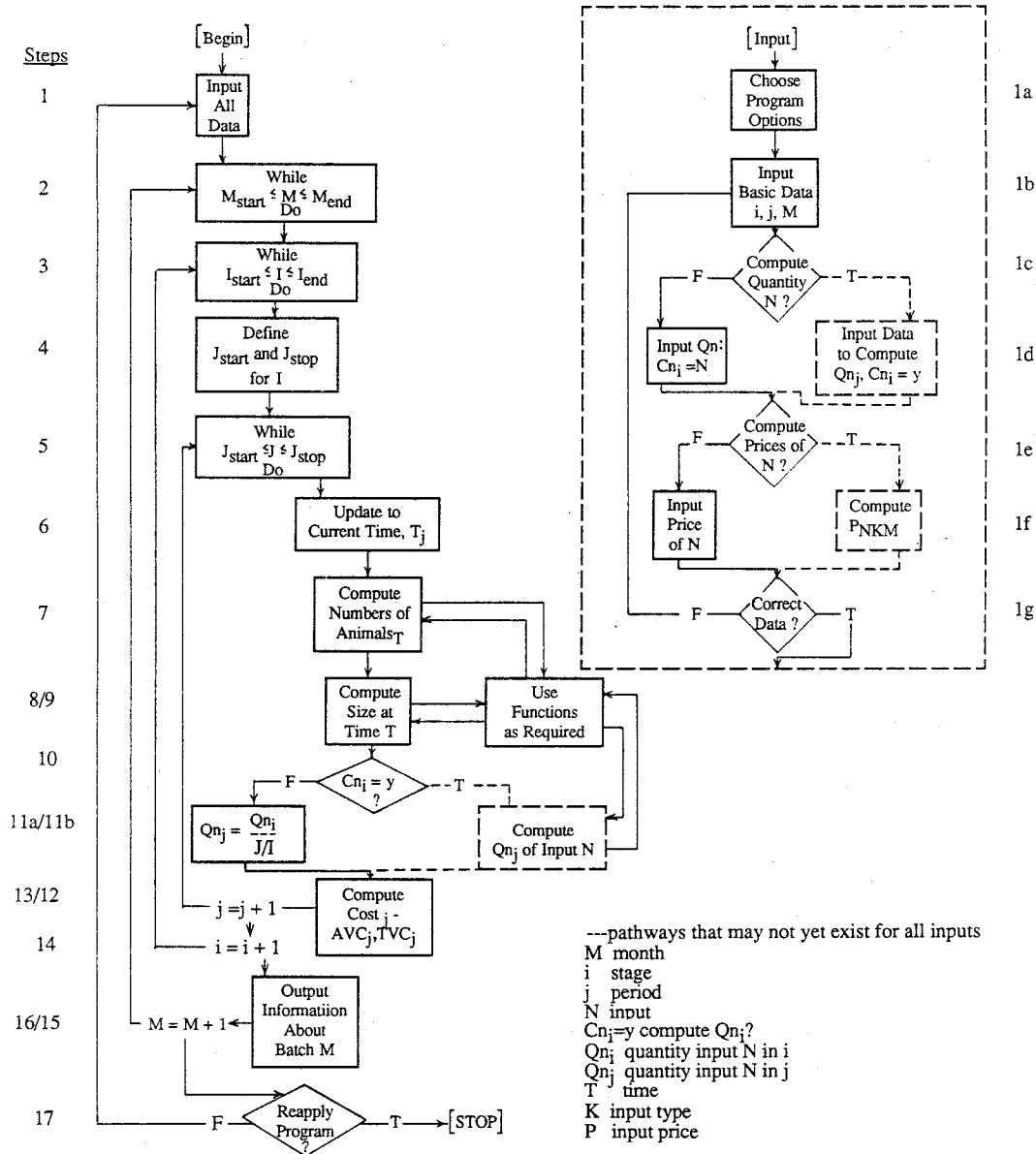
With this nested looping structure, any computation within the *j* loop will depend not only on *j*, but also on time, stage and the month the batch is begun.

Survival rates for bivalves (*step 7*), size of bivalves (*step 8*) and input quantities (*step 11*) are computed within the loop *j*, calling function subroutines, which are placed outside of the loop when used by more than one subroutine. Cumulative and average costs, based on information from previous steps, then are computed (*step 12*). The program output contains information generated about the batch beginning at month *M* (*step 15*); a new batch is begun on the first day of month *M+1* (*step 16*). By performing computations for each month, information about batches begun at different times during the production season can be generated.

The options included in the code allow the model to be adapted to fit a variety of different hatcheries. With little effort, the internal parameters of the code can be re-specified to change the number, units and name of inputs, and the number and name of stages, thus adapting the code to a hatchery's specific production methods. When running the program, the user may specify the length of each stage and the number of periods within that stage, thus determining in part the level of detail contained in the information generated. It is possible to skip entirely stages for which information is not required or for which data may be insufficient.

⁴ Preliminary development of the code, which concentrated on specific subroutines, is shown in Myers (1985). A listing of the completed code is given in Appendix A. It is being run on an IBM mainframe computer. Data can be entered entirely by file, entirely interactively, or by a combination of both.

FIGURE 7. BIVALVE CODE FLOWCHART



The user may also choose the way in which input prices and quantities are determined. If the code contains the necessary algorithms, the user may choose either to allow the program to compute input price and/or quantities (steps 1f, 11b) or enter these as fixed parameters (steps 1d, 1f). The modular design of the code facilitates the addition or

deletion of these price and quantity algorithms.

Algorithms

The algorithms for bivalve survival rates, bivalve size and production costs are

required in the model and are described here.⁵ In addition, the optional algorithms for algae price and quantity, which constitute the major focus of this analysis, are outlined. These algorithms, except those for price, are contained within the period loop and thus depend on time, period and stage. The prices for inputs used in a period are functions of the elapsed time up to period j and on the month during which that period occurs.

Survival Rate

The output from a batch of bivalves, as well as the quantity of inputs and production costs, depends on the initial number of bivalves in the batch and on the number of animals surviving from period to period. Death of a proportion of the population in a hatchery is unavoidable even though the mortality rate is lower than in the wild. Survival rates are affected by environmental conditions maintained in the tanks, but little is known about the trade-off between increasing the survival rates and cost of improving the conditions in the tanks. Survival rates will also vary by hatchery, although in all hatcheries survival should increase at an increasing rate because the animals become more resilient with age and culling rates go down with increasing age.

Hatcheries often overproduce at early stages to account for the declining numbers of animals over time, which is a result of natural mortality and the culling of inferior animals from the batch. Culling assures a faster overall growth rate and a superior final harvest. However, it is difficult to distinguish between the effects of mortality and culling on the number of remaining animals because both dead and inferior animals are removed in the same sorting process.

It is reasonable to assume that the number will decline over time. There are, however, few, if any, data on actual survival

rates under these controlled conditions. At the time data were collected for this study, the hatchery had not been in production long enough to have reliable data on survival rates. For this reason, a general exponential function reflecting a constant percentage mortality rate over time was written into the code. This allows the user to specify particular survival rates and facilitates testing the sensitivity of costs to changes in survival rates.

The number surviving at the end of each period is computed at *step 7* of the code by calling a function that relates survival at time t to the number of animals initially in the batch, i.e., the number successfully spawned.⁶ This function is:

$$(1) N_t = N_o(t)^\rho,$$

where N_t = the number of animals remaining at time t ; N_o = initial number of animals spawned; and ρ = a parameter to be set by the user, $0 > \rho > -1$.

While this general functional relationship is thought to be an accurate representation of an invertebrate population's survival rate (Emmel, 1976, p. 205), its tendency to go to infinity as t goes to zero caused a problem in writing the computer code. To be consistent with the use of t as an index of the passage of time for other calculations, it was necessary to have equation (1) at $t=0$ record the initial number of animals in the batch (N_o). Thus for convenience of programming, N_t was defined as the number of animals remaining at the end of any week t , where t is incremented in discrete time. Thus, to reconcile the initial estimate of the number of bivalves with the continuous time function above and the other functions in the model, the following function is used:

$$(2) N_t = N_o (t+1)^\rho.$$

⁵ The specific forms of the mathematical relationships used to describe the algorithms were chosen primarily because they were found to provide the "best" statistical results given the data used to estimate the parameters of the functions empirically. The empirical estimates are described in subsequent sections.

⁶ Although the cost to condition the adult population (stage 1), which spawns the initial egg population, affects the total cost of producing a batch, the number of adults is not important in determining the initial population in this model, because N_o is input by the user as a fixed parameter rather than as a function of the fecundity of the adults spawned.

An estimate of the average number of animals surviving during a period is necessary to calculate an average quantity of inputs used during j . The actual average number over the period from time a to b is calculated by:

$$(3) N_{(a,b)} = \frac{N_0}{(b-a)} \int_a^b (t+1)^\rho dt.$$

This integral was estimated in the code with a Reiman sum using the trapezoidal approximation to the area under a curve. The method gives a satisfactory estimate of the integral and increases the generalizability of the code since it does not depend on the actual form of the integrand.⁷

Size

Both the size of an individual bivalve at the end of each period and the average size during the period are estimated. This information is useful in assessing potential market price and is also necessary for computing the amount of inputs used within a period.

Data relating the size of bivalves to their age were difficult to find in the litera-

⁷ To estimate the definite integral $\int_a^b f(t)dt$ by the trapezoidal rule, one must calculate

$$A = \int_a^b f(t)dt \approx \sum_{k=1}^{n-1} f(t_k) \Delta t + \frac{f(a)+f(b)}{2} \Delta t.$$

For coding purposes, it was efficient to calculate

$$A = \sum_{k=1}^n f(t_k) \Delta t + \frac{f(a)-f(b)}{2} \Delta t$$

and $\Delta t = 1/7$, a daily increment. If f is continuous on $a \leq t \leq b$ and twice differentiable on $a < t < b$, then there is a number $a < \epsilon < b$ such that

$$\int_a^b f(t)dt = A - \frac{(b-a)}{12} f''(\epsilon) \Delta t^2$$

There is no set procedure for finding the value of ϵ such that this is true, but one can obtain some idea of the magnitude of the error by investigating the size of $f''(t)$ between a and b (Thomas, 1960, pp. 207-17 and pp. 385-88).

ture as well. However, Epifanio, Logan and Turk (1976) and Epifanio (1975) report a small amount of data relating the size H_t (in mm of shell height) to age in weeks (t) for both clams and oysters. These data suggest that the relationship is quadratic, at least for young bivalves:

$$(4) H_t = \alpha t + \beta t^2,$$

where one would expect $\alpha \geq 0$ and $\beta < 0$, (the parameters are chosen by the user). The only potential problem with an empirical specification of this function is that it may yield a maximum size at too early an age. This problem is not serious because the function behaves well over ages that are likely to exist in a hatchery setting.

Using this relationship, the average size during any period starting at time a and ending at b is given by:

$$(5) AH_{(a,b)} = \frac{1}{(b-a)} \int_a^b (\alpha t + \beta t^2) dt.$$

This integral also is approximated in the code using the trapezoidal rule, which computes the value of the function at each day of the period.

Costs

Estimating the variable cost to produce a single bivalve to a certain age or size involves calculating the total variable cost (TVC_t) for an entire batch at some time t and dividing by N_t , the number of animals surviving is given by:

$$(6) AVC_t = TVC_t / N_t.$$

This average variable cost per animal is derived at the end of each period.⁸

As stated above, each batch is initially divided into stages ($i=1, \dots, S$), which are in turn divided into periods. To facilitate cod-

⁸ The stage associated with the period must be greater than 1 since no animals are produced in the conditioning stage.

ing, and allow for a different number of periods in each stage, periods ($j=1, \dots, J$) are indexed consecutively over the entire batch. Thus total variable costs at time t^* (measured in weeks) is the sum of all variable costs ($PTVC_j$) across all periods included in the first t^* weeks:

$$(7) TVC_{t^*} = \sum_{j \in t^*} PTVC_j.$$

Because the code keeps track of periods and stages by incrementing the j th loop within the i th loop, costs up to the end of a period may also be thought of as the sum of the costs in all periods included in the completed stages ($i=1, \dots, S^*-1$) plus the sum of the costs for all completed periods up to time t^* included in the current stage, S^* ,

$$(8) TVC_{t^*} = \sum_{j \in (S=1, \dots, S^*-1)} PTVC_j + \sum_{j \in (S^* \cap t \leq t^*)} PTVC_j.$$

Variable costs during a period are of course determined by the quantity of the inputs used during that period and by the price of the input during the month in which the input is used. This is incorporated into the model as:

$$(9) PTVC_j = \sum_{n=1}^N \sum_{k=1}^K \sum_{r=1}^R P_{nkm_j} Q_{nkrj},$$

where P_{nkm_j} = price per unit of input n ,

type k , in the month during which j occurs; Q_{nkrj} = quantity of input n , type k , for use r used during j ; N = number of inputs n ; K = number of types k of input n ; R = number of uses of input n , type k ; and m_j = month during which j occurs.

Prices and Quantities

As discussed previously, prices and quantities are either computed by algorithms within the code or are input by the user, thus allowing the user to focus on specific parts in the production process. Although the model allows any of the input prices to be computed, this option is probably most useful in computing the cost, or internal price, of algae and culch, because these two inputs are often

intermediate inputs produced within the hatchery. This analysis focuses on the computation of this internal price for algae and on the use of this input as feed during the production of the bivalves.

As shown in equation (9), the model allows for a price for K different types of input n in each month of production, thus creating a 3-dimensional price array, P_{nkm} . The code requires a value for each element of the array regardless of whether the price is computed or input by the user so that a corresponding price can be found for inputs used during any month.

From the computations for the quantity of inputs used in each period, where a quantity is calculated for each type of each input for each period, Q_{nkj} , another 3-dimensional array is constructed. Determining the quantity of each type involves summing over the R different uses for the input. For expository purposes, this is included as an additional subscript on the quantity variable in equation (9), Q_{nkrj} . However, in the code, this sum of input quantities by use is calculated in the quantity subroutine, and this subscript does not appear in the quantity array used in the cost calculations.

If the quantity of input is entered by the user rather than computed by algorithms in the code (*step 1d*), a lump sum is entered as Q_{nkj} , which is then divided equally among each period of that stage (*step 11*).

Algae Price and Quantity

The decision to focus this report on algae production (price) and on the quantity of algae used for feed in bivalve production is justified for three reasons.

First, the production of algae in the hatchery was the most well defined and firmly established of the hatchery's production processes. It was easy to separate the inputs used in the production of algae from those used as direct inputs to bivalve production, because some different types were used and because the two processes were physically separated in the hatchery.

Second, the algae production model closely resembles the bivalve code structure because of the similarity in the production processes. By first writing the algae price code, potential problems with the full bivalve code could be anticipated.⁹

Third, and most important, is the fact that feed costs have been identified as the largest single component of hatchery costs, ranging from 29% and 33% of variable costs in early studies (Im *et al.* 1976) to as high as 85% in more recent studies (Bolton 1982). Thus, from a practical point of view,

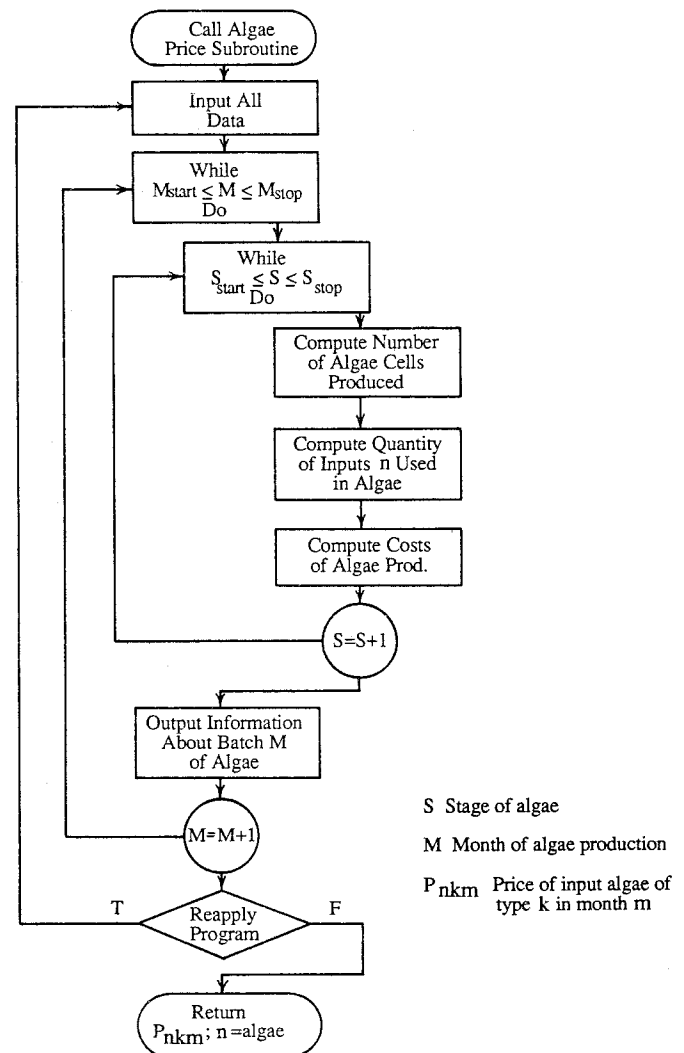
... the major bottleneck to commercialization of intensive bivalve culture systems [lies] in the ability to economically culture massive quantities of these suitable algal species. Intensive study of problems associated with such mass culture of algae is certainly warranted... (Epifanio 1975, p. 190).

These concerns become even more important as hatcheries attempt to raise animals to a larger size.

The algae system, described in Section III, can be broken into stages of relatively short, fixed duration, thus distinguishing different algae "types" by the age of the culture. Algae can be fed to the bivalves from different stages and as such provides one example of how different types of one input can be used within the code. These different types of algae are also distinguished on the basis of their different physical characteristics, such as density, and overall purity, and by the cost of production. In the subroutine where algae costs are computed, a price, which in this case equals production costs, is computed for each of the different algae types.

Figure 8 is a representation of the algorithms used to simulate algae production. As in the bivalve program, only one batch of algae is followed at a time. It is also assumed that only one algae species, or a single mixture of species, is produced in a batch and that all tanks within a stage are used. By assuming that a new batch is begun on the first

FIGURE 8. ALGAE PRICE SUBROUTINE



⁹ In fact, the two processes are very similar inasmuch as both operate on a batch-like principle. The transfer of algae to successively larger tanks as they mature is comparable to the transfer of animals between stages. Both systems are a function of the number of animals or cells; however, the number of organisms increases over time in the algae system due to cell reproduction, although cell size remains constant.

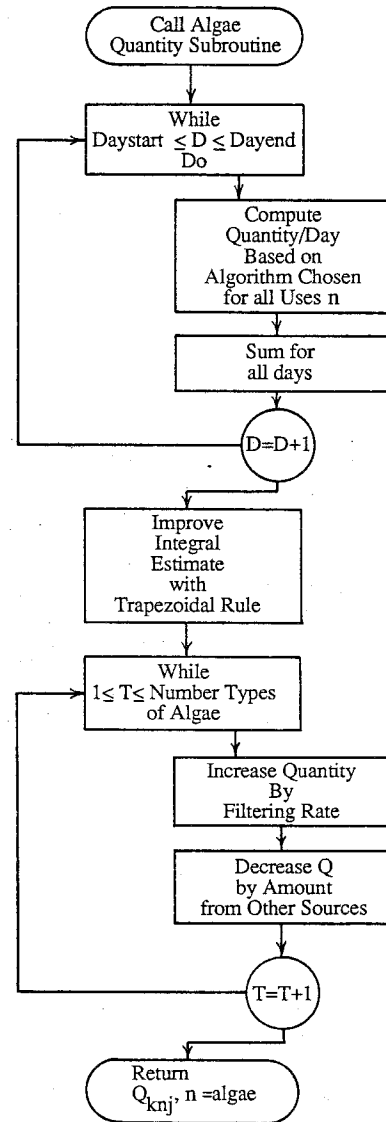
of every month, one can derive the 3-dimensional price array required by the bivalve program, P_{nkm_j} from equation (9) (where $n =$ algae), from the internal cost of producing each type of algae.

The cost of producing any stage of algae depends on the amount of inputs --electricity, labor and materials-- used during the stage and on the price of those inputs during the month in which the stage occurs. Because algae production in the hatchery is separate from bivalve production, there is a provision in the data input to specify different prices for those inputs used in producing algae from those used in other parts of bivalve production. The algorithms used to derive the internal costs of producing algae are discussed in detail in Myers (1985).

The algae quantity subprogram is used to compute the feed requirements for a batch of bivalves at any time t (see Figure 9). It allows for the feeding of algae from any stage of maturity to bivalves at any stage of growth, although it is likely that average production costs per unit of algae would be too high in the early stages. Exactly at what stage during the algae production process algae should start to be used for feed is an empirical question that can only be answered by the information generated from the program itself. More is said about this issue below.

To estimate the algae required for a batch of bivalves, one must know the relationship between feed consumption and some measure of a bivalve's age or size, in addition to the number of animals remaining in the batch at time t . In the literature, one finds feed requirements related directly to age (Epifanio 1975) and directly to size, as measured by shell height or dry weight (Epifanio *et al.* 1975 and Bolton 1982). To maintain the flexibility of the program, both options were included in the code. This was accommodated easily, because regardless of the relationship used --size and feed or age and feed-- the data are consistent with a function that is linear in logarithms. One might well interpret the relationship as a Cobb-Douglas production function.

FIGURE 9. ALGAE QUANTITY SUBROUTINE



In the first option contained in the code, the daily consumption of algae (10,000 cell units) at time t , C_t , is related to shell size in mm of shell height as determined from equation (4). Thus,

$$(10) \ln(H_t) = \delta + \gamma \ln C_t.$$

Solving for C_t and substituting,

$$(11) C_t = (e)^{-(\delta/\gamma)} (H_t)^{1/\gamma}.$$

Multiplying this expression by the number of animals surviving (N_t) and substituting from equations (7) and (9), the total daily feed requirements (in 10,000 cell units) for a batch at time t , C_t^* can be written entirely in terms of t :

$$(12) C_t^* = [(\alpha t + \beta t^2)^{1/\gamma}] [(t+1)^\rho N_o] [e^{-\delta/\gamma}].$$

The total number of algae cells consumed (in 10,000 cell units) during a period is then given by

$$(13) \bar{C}_t = e^{-\delta/\gamma} \int_a^b [(\alpha t + \beta t^2)^{1/\gamma} [(t+1)^\rho] dt.$$

Again, the values of the parameters are left to the user.

The second option relates the cells consumed, c_t , directly to age in weeks, thereby eliminating the need to evaluate equation (4). The new relationship is¹⁰

$$(14) \ln t = \xi + \eta \ln c_t.$$

Thus,

$$(15) c_t = (e)^{-(\xi/\eta)} t^{1/\eta};$$

$$(16) c_t^* = [t^{1/\eta}] [(t+1)^\rho N_o] [e^{-\xi/\eta}];$$

and

$$(17) \bar{c}_t = e^{-\xi/\eta} \int_a^b [t^{1/\eta}] [(t+1)^\rho N_o] dt.$$

Both of these options create the 2-dimensional array for algae quantity.¹¹ The price determined in the algae price routines can then be referenced to compute the cost of the algae inputs used in each period of bivalve production.

V - EMPIRICAL TESTS OF THE SIMULATION MODEL

The purpose of this section is to describe the results from initial experimentation with the bivalve cost simulation model. The decision to apply the model to oyster seed production (*Crassostrea virginica*) was based primarily on the availability of data. As emphasized above, the empirical results reported here focus on simulating the cost, or internal price, of algae and on the quantities of algae and total feed costs of raising a batch of bivalves to various sizes. The remainder of this section begins with a discussion of the data used for the analysis. Next, the empirical results are presented for a base case, after which a variety of sensitivity analyses are conducted to see how the empirical results are affected by changes in the key parameters of the model.¹² Finally, these results are compared with other cost and bivalve seed price information to provide initial indications of the optimum size at which to market the seed.

The Data

To experiment with the model, it was necessary to obtain two sets of information: a) the data required by the subprogram that calculates the cost or internal price of algae and b) data by which to estimate the parameters of the survival function, the age-size relationships and the algae feed functions [equations (1), (4), (10) and (14)]. The data for the algae price (internal cost) estimates are based in large measure on several weeks of observation at the hatchery during the summer of 1983. There were, however, a few items taken from other sources, but these were modified to be consistent with the other production processes and specific environmental conditions of the hatchery. These data are summarized in Appendix B and are described in detail by Myers (1985).

For any single type of bivalve, there were very few data in the literature on which to base the estimates of the parameters for

¹⁰ The variables C_t and c_t refer to feed consumption (in 10,000 cell units), calculated according to options 1 and 2, respectively.

¹¹ Both equations (13) and (17) are evaluated in the program using procedures similar to those for the survival and size equations above and reference equations (2) and (4) for each day of the period.

¹² Myers (1985) conducts considerable sensitivity analysis on the internal price of algae, so these results focus on changes in other parameters of the system.

equations (1), (4), (10) and (14). However, except for the survival equation, it was possible to combine data from several sources to derive consistent parameter estimates for the oyster.

Table 5 contains an empirical estimate of equation (4a), relating the size of oysters to age. The data are from Epifanio *et al.* (1976).

Within the simulation model, there are two alternatives for estimating the feed (algae) requirements for bivalves. One way is to relate feed directly to shell height as in equations (10a,b) of Table 5. To estimate equation (10a), equation (4a) was used first to predict shell height at some selected ages for which Claus and Adler (1970) report algae consumption data. These predicted shell heights are then regressed on the algae consumption figures (in log-linear form) to obtain equation (10a). Equation (10b) was estimated directly from data in Bolton (1982). A second way to estimate feed requirements accommodated in the model was to relate algae consumption directly to age. Data from Claus and Adler (1970) were used to estimate equation (14a) directly, while equation (14b) was estimated directly from information in Bolton (1982).¹³

The primary reason for developing these four ways of estimating feed requirements was to test the sensitivity of the model. It is somewhat surprising, but encouraging, to find that the algae production elasticities in these four equations are similar, ranging only from 0.35 to 0.42.

Data with which to estimate the survival rate parameters were among the most difficult to obtain. There were no population

¹³ Bolton (1982) estimates oyster feed requirements through a mathematical relationship between the whole weight of oysters and the daily consumption of algae (due to Pruder *et al.* (1977)):

$$Y = 8.2 x^{-0.21}$$

where x = whole weight in grams; and Y = algal cells cleared $\times 10^8$ /g whole weight/day. Because they report data on age and shell height for oysters of different whole weights, once this equation is used to estimate algal consumption based on weight, these predicted values can in turn be regressed on age and shell height to obtain equations (10b) and (14b).

Table 5. Estimated Equations for Use in Simulations

Equation Number ^a	Estimated Equation	
(1a) ^b	$\ln N_t = 8.55 - 0.255 \ln t$	
	(46.09)	(-6.26)
	$R^2 = 0.66$	[0.041]
(4a) ^{c,d}	$H_t = 0.737(t) - 0.00252(t)^2$	
	(22.14)	(-6.99)
	[0.033]	[0.0004]
(10)		
(a) ^f	$\ln H_t = -0.631 + 0.338 \ln C_t$	
	(-1.65)	(4.58)
	$R^2 = 0.81$	[0.074]
(b) ^e	$\ln H_t = -1.769 + 0.416 \ln C_t$	
	(-5.85)	(18.57)
	$R^2 = 0.98$	[0.022]
(14)		
(a) ^e	$\ln t = -0.331 + 0.345 \ln C_t$	
	(-0.85)	(4.62)
	$R^2 = 0.81$	[0.075]
(b) ^d	$\ln t = -2.244 + 0.407 \ln C_t$	
	(-5.644)	(13.83)
	$R^2 = 0.97$	[0.029]

^a See text for equation numbers and variable definitions. The numbers in parentheses and brackets are t-ratios and standard errors, respectively.

^b The source of the data is Eldridge, *et al.* (1979).

^c R^2 's are not applicable for restricted regressions.

^d The source of the data is Epifanio *et al.* (1976).

^e The source of the data is Bolton (1982).

^f The sources of data are Claus and Adler (1970) and Epifanio *et al.* (1976).

survival data for oysters raised in a hatchery, although Bolton (1982) makes allowance for a 20% mortality rate during the cultivation period. Because most assumptions in Bolton's experiment proved extremely optimistic, it seemed inadvisable to use this overall mortal-

ity rate as a base of comparison. Furthermore, there was no easy way to infer from this overall rate how mortality changes over time.

Since hatchery data on which to estimate survival rates of oysters or other bivalves were unavailable, the best alternative to obtain survival rates that are at least in the relevant range was to estimate equation (1) from data for hard clams planted in protective trays in a South Carolina estuary (Eldridge *et al.* 1979). The size of the clams planted was larger than bivalves raised in a hatchery, and the survival function for smaller bivalves had to be extrapolated beyond the limits of the data. Initially, one might expect this to lead to an overestimate of the survival of hatchery-sized bivalves. However, the data do reflect a less controlled environment than found in the hatchery, a factor that may partially offset the effect of size. The estimated equation, in log-linear form, is in Table 5; its parameters imply a constant mortality rate of 25.5%. When accumulated on a weekly basis, this function leads to over 60% mortality in a year's time, three times that for oysters in Bolton's (1982) prototype hatchery. This discrepancy is large, but it is unlikely that Bolton anticipated a year-long cultivation period. Equation (1a) from Table 5 implies about a 50% mortality after 18 weeks. This is closer to Bolton's initial assumptions. These two extremes are a useful range over which to test the sensitivity of the results.

Empirical Results

A computer program was designed that implements the simulation model described in detail above. This program was designed to be used by the hatchery operator to generate information about a batch of bivalves at different states of development. Input costs and requirements, and bivalve age, survival and size are reported at the end of each production period and as cumulative amounts. Most important, it gives an average cost per bivalve at the end of each period to give the user information about production timing and about the expected revenue from bivalve seed sale that might be expected.

Initial Simulations

A partial listing of the printout generated by the program is in Appendix B. It contains information for a batch of oysters under base run conditions (run 1). The batch is assumed to begin in April and last for 420 days, or 59 periods. Myers' (1985) input and price data were used to calculate the internal cost per unit of algae.¹⁴ The data reflect economic conditions at the hatchery in 1983. The batch contains one million oysters initially; the survival rate for bivalves is given by equation (1a), of Table 5; the age-size and feed-size relationships are given by equations (4) and (10a), respectively. The bivalve feeding filtering rate is assumed to be 50%, and 12% of the diet is assumed to come from algae in circulating sea water in the juvenile stage.

The assumptions underlying the base run and 11 other simulations are given in Table 6. These runs contain numerous combinations of the functional relationships described earlier in Table 5. The first three runs assume the same mortality rates but accommodate different ways of estimating the physical feed requirements. Runs 1 and 3 assume feed is directly related to size, although in run 2, feed is directly related to age. The results of these runs are summarized in Table 7.

For the base run, the number of bivalves surviving over the duration of the batch falls from a million at spawning to an estimated 938,000 at the beginning of the larval stage. After setting, the estimated number is 716,000. If the batch were to remain in the hatchery for 420 days (including 60 days for conditioning the broodstock), the

¹⁴ To facilitate comparisons, all other runs of the model are assumed to begin in April and run for 59 periods (420 days). The simulation was run over such a long duration primarily to test the behavior of the model since, due to bivalve size and the cost of feed, it is clear that bivalve seed would be sold or transferred to grow-out long before this last period was reached. Conditioning takes a month, and the spawning, larvae and setting stages were broken into periods of a few days each; but once the juvenile stage was reached, periods were a constant one week in length. For ease of exposition, much of the discussion in the text focuses on monthly intervals or on individual stages.

Table 6. Description of Parameters for Alternative Simulations Runs^a

Simulation	Survival Equation	Age-Size Equation	Feed-Size Equation	Feed-Age Equation
Run 1 (Base Run)	$N_0 = 1 \times 10^6$ $\rho = -0.0255$	$\alpha = 0.737$ $\beta = -0.00252$	$\delta = -0.631$ $\gamma = 0.0338$	N.A.
Run 2	same as run 1	same as run 1	N.A.	$\xi = -0.331$ $\eta = 0.345$
Run 3	same as run 1	same as run 1	$\delta = -1.769$ $\gamma = 0.0416$	N.A.
Run 4	$N_0 = 1 \times 10^6$ $\rho = -0.214$	same as run 1	same as run 1	N.A.
Run 5	$N_0 = 1 \times 10^6$ $\rho = -0.296$	same as run 1	same as run 1	N.A.
Run 6	$N_0 = 1 \times 10^6$ $\rho = -0.163$	same as run 1	same as run 1	N.A.
Run 7	$N_0 = 1 \times 10^6$ $\rho = -0.337$	same as run 1	same as run 1	N.A.
Run 8	same as run 1	$\alpha = 0.671$ $\beta = -0.00252$	same as run 1	N.A.
Run 9	same as run 1	$\alpha = 0.671$ $\beta = -0.00332$	same as run 1	N.A.
Run 10	same as run 1	same as run 1	$\delta = -0.631$ $\gamma = 0.412$	N.A.
Run 11	same as run 1	same as run 1	$\delta = -0.631$ $\gamma = 0.486$	N.A.
Run 12	same as run 1	$\alpha = 0.671$ $\beta = -0.00332$	$\delta = -0.631$ $\gamma = 0.486$	N.A.

^a See Table 5 and equations (1), (4), (10) and (15) in the text for parameter definitions and estimated relationships.

number of animals surviving would be approximately 357,000.

The estimated average size of the bivalves ranges from just under 2 mm after the setting stage to just over 33 mm at the end of the simulation. As the batch progresses through time, the additional cost of feeding larger bivalves is partially offset by declining

numbers. Because the feed cost per surviving bivalve includes the accumulated costs of feeding those that did not survive, the costs rise rather rapidly throughout the batch. At the end of period 18 (about 100 days), the feed cost per animal is still less than one cent, but if left in the hatchery through the 59th period, feed costs would average more than a dollar per surviving bivalve.

Table 7. Summary of Simulation Output for Initial Runs

Period ^a	Stage	Cumulative Days to End of Period	Animals Surviving (thousands)	Average Size of Animal (mm)	Cumulative Cost per Bivalve		
					Run 1 ^b (Base) (\$)	Run 2 (Proportion of Run 1)	Run 3
6	Setting	49	716	1.98	^c	0.98	8.77
10	Juvenile	77	594	4.84	^c	0.97	5.72
14	Juvenile	105	534	7.61	0.003	0.98	4.29
18	Juvenile	133	495	10.30	0.009	0.99	3.59
22	Juvenile	161	467	12.91	0.022	1.01	3.16
26	Juvenile	189	446	15.44	0.045	1.04	2.86
30	Juvenile	217	429	17.89	0.082	1.06	2.63
34	Juvenile	245	414	20.26	0.137	1.09	2.46
38	Juvenile	273	402	22.55	0.214	1.12	2.31
42	Juvenile	301	391	24.76	0.316	1.16	2.20
46	Juvenile	329	382	26.88	0.449	1.19	2.10
50	Juvenile	357	373	28.93	0.616	1.23	2.01
54	Juvenile	385	366	30.90	0.822	1.27	1.94
59	Juvenile	420	357	33.24	1.144	1.32	1.86

^aPeriod 6 is through setting (about 50 days including adult conditioning, spawning and larval stages); the remaining periods are a week in length.

^bSee Table 6 for assumptions about runs.

^cLess than 0.001.

Data for the other runs in Table 7 suggest that these results can be quite sensitive to the assumptions made regarding the feed equations. In the base run, feed requirements are related directly to the size of the animal, whereas in run 2 it is assumed that feed requirements are determined by equation (14a) relating feed directly to age. Assuming the same survival rates, this alternative way of calculating feed leads to a slightly lower estimate of feed cost per surviving animal through period 18, but then cost rises to 1.3 times that of the base run by period 59. The feed relationships in these two runs were estimated from the same basic data; despite these differences, the results are quite consistent.

The cumulative costs from run 3 are in sharp contrast to those for runs 1 and 2. In this latter run, costs are consistently higher than in the base case, and within the same

run the differential is most extreme in the early periods. This result is somewhat surprising, given that the production elasticity for algae is higher in runs 3, 4 and 6 than in the base run. However, the intercepts on the size or age axes of the production function are lower than for the base run. Thus, even though the percentage gain in size is larger for each one-percent change in algae consumption, absolute feed requirements to raise these oysters to any size over the duration of the batch remains higher than in the base run.

The implications of this result lie in the importance of measuring feed requirements accurately in early stages of the growth process. That is, if one were to look only at a single animal, the tendency would be to conclude that feed costs would be relatively unimportant compared to costs for older animals. However, because the number of animals falls throughout the process, costs (due

to what appear to be minor cost differences in early periods) accumulate rapidly as more animals die and as total costs are averaged over fewer survivors. Thus, reliable estimates for both feed costs at different points in the growth process and the number of animals surviving are more critical in estimating costs for raising bivalves than for larger animals (e.g., cattle or other livestock) with much lower mortality rates.

A Digression on Algae Production

The primary focus of the analysis so far has been on the costs of feeding a batch of bivalves and the effect of survival and feeding ratios on costs. However, it is evident (Table 8) that cumulative algae consumption in runs 2 and 3 is a smaller multiple of the base run consumption than it is for cost in early periods and is a larger multiple

in later periods. The primary reason for this is that feed costs for bivalves are also a function of the implicit price or cost of algae that is raised for feed. The program uses algorithms from Myers (1985) to calculate the price of algae, which are assumed to be constant in any month but which can vary by month because of seasonal differences.

The "algae price" algorithm assumes that algae are produced in a batch process, which is divided into 6 stages covering about 6 to 7 weeks. During the first four stages, most of the algae are needed to inoculate the subsequent stage. Thus, the net production of algae available is negligible until the tertiary stage (about 6 weeks). Up to this point the algae are grown in well water and the batch is quite pure, thus making it ideal for feeding to bivalves at the larval and set stages. These tertiary algae tanks are located (in the study's hatchery) adjacent to the larval tanks, which

Table 8. Algae Consumption for Selected Simulations

Period ^a	Cumulative Algae Consumption			Ratio of Cost to Algae Consumption ^c	
	Run 1 ^b (Base Case) (10 bil. cells)	Run 2 (Proportion of Run 1)	Run 3	Run 2	Run 3
6	48	0.98	8.05	1.00	1.09
10	580	0.97	4.79	1.00	1.19
14	2,153	0.98	3.68	1.00	1.17
18	5,067	1.01	3.10	1.00	1.16
22	9,513	1.04	2.73	0.97	1.16
26	15,603	1.07	2.47	0.97	1.16
30	23,382	1.11	2.27	0.95	1.16
34	32,846	1.15	2.12	0.95	1.16
38	43,947	1.20	2.00	0.93	1.16
42	56,605	1.25	1.89	0.93	1.16
46	70,711	1.30	1.81	0.92	1.16
50	86,133	1.36	1.74	0.90	1.16
54	102,720	1.42	1.67	0.89	1.16
59	124,850	1.50	1.61	0.88	1.16

^aSee Table 7.

^bSee Table 7.

^cRatio of column 7 (Table 7) to column 3 (Table 8) and column 8 (Table 7) to column 4 (Table 8), respectively.

helps keep the costs of moving algae around the hatchery to a minimum. There are an estimated 9,650 litres of algae available at the tertiary stage, most of which is fed to the larvae and animals at set. Some of the tertiary algae is also used to inoculate a solar stage to which seawater, rather than well water, has been added, thus making it unusable for the sensitive larvae. After another week or two, there are an estimated 67,000 litres of algae available to feed juvenile bivalves. Because the cost per cell of algae in the solar stage is about 25% of that in the tertiary stage, it is not surprising that in the early periods feed cost is higher relative to algae consumption than it is in later periods (Table 8).

A further explanation for the behavior of costs relative to algae consumption lies in a slightly higher algae cost estimate in the summer months because of the air conditioning required to maintain cool temperatures suitable for algae production. For example, assuming the batch is started in April, about

82% of the cost of algae in the solar stage is due to labor, 5% to materials and 13% to electricity, much of which is used for cooling. Thus, as the bivalve production moves into autumn, cooling requirements are reduced and production costs fall by as much as 3% relative to a batch available in July.

Implications of Different Survival Rates

The accuracy of hatchery cost estimates derived by this analysis depends on the number of animals surviving at any stage. To test the sensitivity of these results, the survival parameter was changed by plus and minus one and two standard derivatives from the least squares estimate of equation (1). A decrease in the absolute value of the parameter, as in runs 4 and 6, implies a lower mortality rate, whereas an increase in its absolute value leads to higher mortality (runs 5 and 7).

The results of these sensitivity tests are summarized in Table 9. The change in

Table 9. Summary of Simulations to Test Sensitivity of Survival Parameter

Period ^a	Number of Bivalves Surviving				Total Feed Cost				Cumulative Feed Cost Per Bivalve			
	Run				Run				Run			
	4 ^b	5	6	7	4	5	6	7	4	5	6	7
	(Proportion of Run 1)				(Proportion of Run 1)				(Proportion of Run 1)			
6	1.06	0.95	1.13	0.89	1.05	0.96	1.11	0.91	0.99	1.01	0.98	1.02
10	1.09	0.92	1.21	0.85	1.08	0.93	1.18	0.87	0.99	1.01	0.97	1.02
14	1.11	0.90	1.25	0.82	1.10	0.91	1.23	0.84	0.99	1.01	0.98	1.02
18	1.12	0.89	1.29	0.80	1.11	0.90	1.26	0.82	0.99	1.01	0.98	1.02
22	1.13	0.89	1.32	0.78	1.12	0.90	1.29	0.80	0.99	1.01	0.98	1.02
26	1.14	0.88	1.34	0.77	1.13	0.89	1.31	0.79	0.99	1.01	0.98	1.02
30	1.15	0.87	1.36	0.76	1.13	0.88	1.33	0.78	0.99	1.01	0.98	1.02
34	1.15	0.87	1.38	0.75	1.13	0.88	1.34	0.77	0.99	1.01	0.98	1.02
38	1.16	0.86	1.39	0.75	1.14	0.87	1.36	0.76	0.99	1.01	0.98	1.02
42	1.16	0.86	1.40	0.74	1.15	0.87	1.37	0.76	0.99	1.01	0.98	1.02
46	1.17	0.86	1.42	0.73	1.15	0.87	1.38	0.75	0.99	1.01	0.97	1.02
50	1.17	0.85	1.43	0.73	1.15	0.86	1.39	0.75	0.99	1.01	0.97	1.02
54	1.18	0.85	1.44	0.72	1.16	0.86	1.40	0.74	0.99	1.01	0.97	1.02
59	1.18	0.85	1.45	0.72	1.16	0.86	1.41	0.74	0.99	1.01	0.97	1.03

^aSee Table 7 for explanation of periods.

^bSee Table 6 for assumptions about runs.

the number of surviving bivalves is predictable directly from the parameter changes. Because of the form of the survival function, mortality is assumed to occur at a compound rate. Thus, for increases in mortality, the proportion of bivalves surviving, relative to changes in the base run, falls as the batch moves to later stages. Total feed costs (relative to the base run) fall as well, but not quite so fast. Exactly the opposite is true when mortality rates are reduced.

Perhaps the most important result from this analysis is that despite significant differences in the number of surviving animals (from 0.72 to 1.5 times the base run), cumulative costs per bivalve are never any more than plus or minus 3% of the base run costs. The important implication of this result for hatchery owners is that relatively accurate estimates of per-unit variable costs can be made somewhat independently of good estimates of survival. This is critical in compar-

ing per-unit costs to the potential price of seed to determine when to sell.

Sensitivity of the Relationship of Age and Feed to Size

Although the average cost per bivalve is relatively insensitive to the assumptions made about survival rates, the results of the initial simulations suggest that this is not the case for the relationships between age and size and between feed and size. The importance of these relationships is perhaps best understood by systematically changing the parameters of equations (4) and (10a), assuming the other assumptions in run 1 are constant. The results of this experimentation are summarized in Table 10.

To begin the analysis, the coefficient on the linear term was changed by plus and minus one and two standard deviations. It was readily apparent that although the change

Table 10. Sensitivity of Simulation Results to the Age-Size and Feed-Size Relationships

Period ^a	Average Size of Bivalve					Cumulative Feed Cost per Bivalve ^c				
	Run ^b	Run	Run	Run	Run	Run	Run	Run	Run	Run
	8	9	10	11	12	8	9	10	11	12
	(Proportion of Run 1)					(Proportion of Run 1)				
6	0.91	0.91	1.00	1.00	0.91	0.76	0.75	0.58	0.40	0.33
10	0.91	0.90	1.00	1.00	0.90	0.75	0.74	0.38	0.20	0.17
14	0.91	0.90	1.00	1.00	0.90	0.75	0.73	0.29	0.13	0.10
18	0.91	0.89	1.00	1.00	0.89	0.75	0.72	0.25	0.09	0.07
22	0.90	0.88	1.00	1.00	0.88	0.75	0.71	0.22	0.08	0.06
26	0.90	0.88	1.00	1.00	0.88	0.74	0.70	0.20	0.06	0.05
30	0.90	0.87	1.00	1.00	0.87	0.74	0.69	0.18	0.06	0.04
34	0.90	0.86	1.00	1.00	0.86	0.74	0.67	0.17	0.05	0.04
38	0.90	0.86	1.00	1.00	0.86	0.74	0.66	0.16	0.05	0.03
42	0.90	0.85	1.00	1.00	0.85	0.73	0.65	0.15	0.04	0.03
46	0.90	0.84	1.00	1.00	0.84	0.73	0.64	0.15	0.04	0.03
50	0.89	0.83	1.00	1.00	0.83	0.73	0.63	0.14	0.04	0.03
54	0.89	0.83	1.00	1.00	0.83	0.72	0.61	0.14	0.03	0.02
59	0.89	0.82	1.00	1.00	0.82	0.72	0.60	0.13	0.03	0.02

^aSee Table 7 for explanation of periods.

^bSee Table 6 for assumptions about runs.

^cTotal feed costs as a proportion of those in run 1 are the same as the feed costs per bivalve.

in feed costs was more dramatic than the change in bivalve size, both remained about proportional to the change in the parameter, α . Thus, only one of these runs, where α is reduced by two standard deviations, is reported (run 8). This change reduced the estimated size by about 90% of the base run size while reducing feed costs to between 72 to 76% of base run costs.

The next step in the analysis was to change the coefficient on the t^2 term in equation (4). In run 9, the coefficient on the linear term was the same as in run 11, but the absolute value of the coefficient on the squared term was increased by two standard deviations. The effect of this change on both size and feed costs was minimal in the early stages that would be most relevant to a hatchery producing seed to be sold or transferred to growout facilities. At later stages, the effect of this parameter change becomes increasingly important.

The most significant results from Table 10, however, seem to be the large reductions in feed costs due to an increase in the production elasticity of algae in equation (10a). By raising the production elasticity to one standard deviation above its original level, costs fall to just under 60% of the base run (run 10) in period 6 and to 13% by the end of the simulation. Increasing the production elasticity by an additional one standard deviation leads to further, but not quite so dramatic, cost reductions (run 11).

Preliminary Observations about Total Costs of Production

Because data were unavailable from the hatchery on other aspects of the production of oyster seed, it was impossible to implement all parts of the simulation model and estimate costs other than feed costs. However, at least two previous studies contain estimates of the proportion of total costs accounted for by algae feed. On the basis of these estimates, it is possible to make some preliminary observations about total production costs and thus about when during the batch costs will exceed the price that a hatchery operator might receive for the seed.

As reported in Table 4, Im *et al.* (1976) estimate that algae feed costs are about one-third of total production costs; Bolton (1982) reports estimates ranging from 15 to 85% of total costs. Feed costs, as a percentage of total costs, probably increase with age. However, the nature of this relationship is not known; so to obtain some idea of what this wide range in estimates implies for the simulation results above, two kinds of sensitivity analysis were conducted. The first was to assume that feed costs are a constant fraction of total costs (first three panels of Table 11). By assuming feed costs are 33, 59 and 85% of total costs, one is able to reflect all but the low range of Bolton's estimates. It was believed that this low end of the range would apply only to very young bivalves. To reflect increasing relative feed costs, the last two panels in the table assume that costs rise linearly from 15 to 85% of feed costs from period 1 through 59 and period 1 through 30, respectively.

These sensitivity results have important implications for marketing bivalve seed in that Matthiessen and Smith (1979) argued that oyster seed (2 to 3 mm in size) could be purchased at from \$2.00 to \$5.00 per thousand. Assuming these prices, base run conditions and feed costs at 33% of total, seed would have to be sold prior to the 11th period (5.5 mm shell height) to cover costs if the price were \$2.00 per thousand, and before the 13th period (6.9 mm shell height) if the price were to be \$5.00 per thousand. This "window of opportunity" would occur much earlier (periods 5 and 6) under the most costly conditions (run 3), but only slightly later (periods 14 and 17) for the least cost scenario (run 10).¹⁵

These results are remarkably consistent, given the wide range of parameters in the age-size and size-feed relationship reflected in these runs. The "marketing win-

¹⁵ Since this is a "batch process" and per bivalve feed costs are relatively insensitive to survival rates, it's unlikely that these marketing opportunities would change much if different survival rates were assumed. This does not mean, however, that survival rates are unimportant in terms of the overall profitability of the batch.

Table 11. Alternative Estimates of Total Cumulative Costs of Production (\$ Per Oyster)

Period ^a	Run 1 ^b	Run 2	Run 3	Run 8	Run 9	Run 10
33.3%^c						
6	d	d	0.009	d	d	d
10	0.001	0.001	0.061	0.001	0.001	0.001
14	0.008	0.008	0.275	0.006	0.006	0.002
18	0.026	0.026	0.808	0.020	0.019	0.006
22	0.065	0.066	1.845	0.049	0.046	0.014
26	0.135	0.140	3.584	0.100	0.094	0.027
30	0.247	0.262	6.247	0.183	0.169	0.045
(0.002) ^e	11	11	5	12	12	14
(0.005)	13	13	6	14	14	17
(0.010)	15	15	7	16	15	21
59%^c						
6	d	d	0.005	d	d	d
10	0.001	0.001	0.034	0.001	0.001	c
14	0.004	0.004	0.155	0.003	0.003	0.001
18	0.015	0.015	0.456	0.011	0.011	0.004
22	0.037	0.037	1.041	0.027	0.026	0.008
26	0.076	0.079	2.022	0.057	0.053	0.015
30	0.139	0.148	3.526	0.103	0.095	0.025
(0.002) ^e	12	13	6	13	13	16
(0.005)	15	15	6	16	16	20
(0.010)	17	17	8	18	18	24
85%^c						
6	d	d	0.004	d	d	d
10	0.001	0.001	0.024	d	d	d
14	0.003	0.003	0.108	0.002	0.002	0.001
18	0.010	0.010	0.317	0.008	0.007	0.003
22	0.026	0.026	0.722	0.019	0.018	0.006
26	0.045	0.055	1.404	0.039	0.037	0.010
30	0.097	0.103	2.448	0.072	0.067	0.018
(0.002) ^e	13	13	6	14	14	18
(0.005)	16	16	7	17	17	22
(0.010)	18	18	9	20	20	26
15-85%^c						
6	d	d	0.015	d	d	d
10	0.002	0.002	0.078	0.001	0.001	0.001
14	0.008	0.008	0.298	0.006	0.006	0.002
18	0.024	0.024	0.758	0.018	0.018	0.006
22	0.054	0.054	1.522	0.040	0.038	0.012
26	0.100	0.103	2.642	0.074	0.070	0.020
30	0.164	0.174	4.161	0.122	0.112	0.030
(0.002) ^e	11	11	5	11	11	14
(0.005)	13	13	5	14	14	18
(0.010)	15	15	6	16	16	21
15-85%^c						
6	d	d	0.011	d	d	d
10	0.001	0.001	0.055	0.001	0.001	d
14	0.006	0.005	0.197	0.004	0.004	0.002
18	0.016	0.015	0.480	0.012	0.011	0.004
22	0.033	0.033	0.935	0.025	0.023	0.007
26	0.060	0.062	1.584	0.044	0.041	0.012
30	0.100	0.103	2.448	0.072	0.066	0.018
(0.002) ^e	12	12	5	12	12	15
(0.005)	14	14	6	15	15	20
(0.010)	17	17	6	18	18	25

^aSee Table 7 for explanation of periods.

^bSee Table 6 for assumptions about runs.

^cThese are the percentages that feed costs are assumed to be of total cost. In the last two panels, costs are assumed to increase linearly from 15 to 85% of feed cost from periods 1 through 30 and periods 1 through 59, respectively.

^dLess than 0.0005.

^eActual period during which costs reach \$0.002, \$0.005 and \$0.010, respectively, per bivalve.

dow" is extended by only 2 or 3 weeks if feed costs actually turn out to be 85% rather than 33% of total production costs or if feed costs are assumed to rise to 85% of total costs over 30 rather than 59 periods. A doubling of the price (from \$0.005 to \$0.01 would extend the marketing period by only an additional 2 to 3 weeks as well.¹⁶

It is also important to recognize that during these "break-even" marketing periods, it is estimated that the size of the seed is much larger than the 2 to 3 mm on which the prices are based. This implies that the hatchery owners may be in an advantageous bargaining position to sell seed that meets the size requirements earlier, thus reducing costs, or by asking for a higher price for larger seed, with its higher survival rates during growout.

VI - SUMMARY AND CONCLUSIONS

The bivalve aquacultural industry in the northeastern United States has been strengthened in recent years by declining and fluctuating natural harvests in the surrounding coastal region. The hatchery industry, in which the bivalves are grown only to a small seed size, has gained acceptance as an important link in the aquacultural process; the firms in the industry are potentially able to provide a consistent, reliable source of seed with which the bivalve aquaculturist begins the production process. They have also gained acceptance as a seed supplier to local governments, who artificially seed depleted natural bivalve beds with the seed purchased from hatcheries. Hatcheries as an industry are likely to play a larger role in the overall production of bivalves if they can overcome some of the technical and economic problems with which they must contend.

The techniques involved in the production of bivalve seed in hatcheries have evolved only recently, yet are technically sophisticated compared to other open-water aquacultural techniques. The production pro-

cesses differ, but each requires a precise sequence of steps under carefully-controlled and -monitored environmental conditions. Mature adult clams or oysters taken from hatchery-maintained broodstock are induced to spawn. The gametes are transferred into large, conical larvae tanks where they freely circulate in warm, algae-rich seawater. The concentrated algae used for feed are grown in the hatchery or taken from seawater. After several weeks, the bivalves undergo a metamorphosis where they change from free-swimming larvae to bottom-dwelling spat. Prior to this time, the animals are transferred to set tanks where they are induced to attach to culch -- a material purchased or produced in the hatchery generally from crushed bivalve shell. Once the animals have passed through the delicate setting period, they are transferred to the juvenile tanks where they can be exposed to large quantities of algae and seawater. The animals remain in these tanks until they reach a suitable size for transfer to open-waters.

There are a number of production options available to the hatcherist, the most apparent being the design of the production system and the source of inputs. The algae production method is the greatest cause of variations affecting cost among hatcheries. These methods include the centrifuging of algae cells from seawater, continuous culture methods, or a batch process in which production is separated into distinct stages. The type of culch used and the number of animals produced as culched or culchless (attached or non-attached to the culch) also differ among hatcheries. The hatcherist has a great deal of discretion regarding the optimum survival rate of the animals in relation to the quantity and quality of inputs used. Finally, and most central to this research, is the choice of when to sell seed.

Despite the growth of the industry and advancements in the technical processes of hatcheries, little is known about the economics of the hatchery. There have been several important studies on this aspect of hatcheries (Im, *et al.*, 1976 and by Im and Langmo, 1977). These and other studies fail to pay sufficient attention to the wide variation that exists among hatcheries and to break

¹⁶ This price was included because there has undoubtedly been an increase in the price of seed between that reported by Matthiessen and Smith (1979) and 1982, the year for which the algae cost data apply.

cost down by period of the animal's life. Estimates of cost by period are essential to determine the optimal time (*viz.* size) at which to sell the seed, inasmuch as the optimal selling age is at the point where marginal cost is equal to the price that could potentially be received.

The purpose of this research is to contribute to the existing body of knowledge about the hatchery industry by examining the factors that influence production costs. The research is partly based on observations of a working hatchery located in the northeast and on current literature, with particular attention paid to the variations that exist among hatcheries. The data collected were used to develop the framework for a computer program that could estimate variable production costs. The program routine was intended to enable the individual hatcherist to determine the optimal size at which to sell the bivalve seed.

Creation of this program required the development of an algorithm that simulates the production process. The model sums costs at different stages of animal growth, which are further broken down into smaller units of time. In this way a cost can be computed at any given point in the animals' lives. The cost incurred during any given period of time is a function of the quantity of inputs used and their prices. Thus, the major component of the model was designed to determine the quantity of inputs used and the cost incurred during any given period.

There was sufficient information available from the literature and hatchery observations to simulate the overall production system. Unfortunately, records concerning the expenditures did not distinguish between hatchery and open-water field operations nor did they break costs down by stage. Consequently, data were not available to allow for all aspects of the model to be completed in detail. Data on the batch algae production process used by the hatchery were, however, more readily isolated from the rest of the hatchery because of the physical separation of the two systems in the hatchery. Thus, the focus of the empirical analysis was on the estimation of costs of algae production and is

justified since feeding costs are recognized as a large proportion of total costs.

The FORTRAN computer program that is based on the model focuses on the cost of producing algae feed, but as more details become available about other hatchery production processes it would be a simple matter to incorporate them into the existing FORTRAN code. The computer program calculates the cost, or internal price of algae to the hatchery, and then combines this information with survival and growth rates and feeding efficiencies to determine total (and per unit) feeding costs to raise a batch of bivalves to various sizes. The algorithms are used to examine the sensitivity of the algae production parameters and to the parameters of the bivalve growth, survival and feeding relationships.

The model is applied to raising Atlantic Oyster seed, using data on algae production from the hatchery and data on survival rates, feeding efficiency and growth rates available from the literature. Twelve initial simulations were made. The output from each contained a summary of input prices and requirements and costs for bivalves at different periods throughout the production process, along with the age, size and number of bivalves and the total and average costs per bivalve.

The 12 simulations reflect a range of assumptions about survival rates, growth rates and algae feeding rates. In the base run, feed requirements are related directly to animal size. The feed cost per surviving bivalve remains less than one cent up to 100 days but rises rapidly thereafter. In a second run, based on the same data to relate feed directly to age, feed cost remains lower for the first 100 days, but then rises to 1.3 times the base run.

As might be expected, the feed costs per bivalve are extremely sensitive to the parameters of the feeding relationships and to the age-size relationship. This is not particularly encouraging because there seem to be few sources of data in the literature from which to estimate these relationships. The one encouraging thing is that feed costs are

less sensitive to these parameters for very young bivalves than for older ones, and it is the young bivalve that is of most interest to a hatchery. Another important result is that for wide differences in the numbers of animals surviving, the feed cost per surviving animal (including the accumulated cost of feeding those that did not survive) varied by no more than plus or minus 3%. Having good estimates of feed costs per animal in the face of considerable uncertainty about survival rates may well facilitate planning when to sell the bivalve seed or transfer them for growout.

To explore the implications of these feed cost simulations for the marketing of bivalve seed, some rough estimates of total costs of production per bivalve were developed on the basis of others' estimates of the proportion that feed costs are of total costs. Total costs were estimated for several simulations assuming feed costs were 1) one-third, 2) nearly 60%, 3) 85% of total costs, and 4) a linear function of total costs starting at 15% and rising to 85% by the end of the production period. While the time period at which price just covered costs differed significantly over the scenarios, within a given scenario the "break-even" marketing periods varied only by a week or two in spite of diverse assumptions about the fraction feed costs are of total costs.

In conclusion, this research demonstrates that it is possible and practical to estimate algae and feeding costs for a seed producing bivalve hatchery using a computer program to simulate the production system. The most optimistic assumptions suggest that bivalve seed can be produced and marketed profitably. These conclusions must be qualified by the fact that sufficient data were not available from the hatchery under study to estimate all components of total cost. Collection of these types of data would help in further testing of the algorithm as well as contributing to an overall understanding of hatchery costs at each stage of production. Equally important from a research perspective is the need for better data on hatchery survival rates and the size-age and feeding rates. What data do exist in the literature on which to estimate these rates lead to quite different implications for feed costs, and none of the

existing data sets contains more than a few observations. Thus, in the absence of more extensive research data on which to estimate these important physical relationships, anyone attempting to use this kind of software for management decisions is well advised to use information that is specific to a particular hatchery.

References

- Agnello, R. J. and L. P. Donnelley. "Externalities and Property Rights in the Fisheries". Land Economics, 52 (1976):518-529.
- Baab, J. S., G. L. Hamm, K. C. Hanes, A. Chu and O. A. Roels. "Shellfish Mariculture in an Artificial Upwelling System." Proceedings of the National Shellfisheries Association, 63 (1973):63-68.
- Bardach, J. E., J. Ryther and W. McLarney. Aquaculture - The Farming and Husbandry of Freshwater and Marine Organisms. New York: John Wiley and Sons Inc., 1972.
- Bolton, E. (ed.) Intensive Marine Bivalve Cultivation in a Controlled Recirculating Seawater Prototype System. DEL-SG-07-82, University of Delaware, 1982.
- Breese, W. P. and R. E. Malouf. "Hatchery Manual for the Pacific Oyster". Oregon State University Sea Grant College Program, Agriculture Exper. Station, Special Report Number 443, December, 1975.
- Castagna, M. and J. N. Kraeuter. "Mercenaria Culture Using Stone Aggregate for Predator Protection". Proceedings of the National Shellfisheries Association, 67 (1977):1-6.
- Claus, G. and C. Adler. "Optimal Conditions for Oysters Grown in Closed Environments". Proceedings of the Marine Technical Society, 1 (1970):671-679.
- Donohue, M. J., D. A. Storey and P. G. Allen. "Technical and Economic Feasibility of Alternative Aquacultural Production Systems for American Oysters and Other Bivalve Mollusks in New England". Massachusetts Agricultural Experiment Station Research Bulletin, Number 668, June 1981.
- Dressel, D. M. and D. S. Fitzgibbon. "The U.S. Molluscan Shellfish Industry". United States Dept. of Commerce, National Oceanic and Atmospheric Administration, National Marine Fisheries Service, Washington, D.C., pp. 255-261, in "Drugs and Food from the Sea; Myth or Reality?", P.N. Kaul, and C.J. Sindermann (eds.), University of Oklahoma, 1978.
- Eldridge, P., A. Eversole and J. Whetstone. "Comparative Survival and Growth of Hard Clams *Mercenaria mercenaria*, Planted in Trays Subtidally and Intertidally at Varying Densities in a South Carolina Estuary". Proceedings of the National Shellfisheries Association, 69 (1979):30-39.
- Emmel, T. G. Population Biology. New York: Harper and Row, Publishers, 1976.
- Epifanio, C. E. "Culture of Bivalve Mollusks in Recirculating Systems: Nutritional Requirements". Proceedings of the First International Conference on Aquaculture Nutrition, pp. 173-194, October, 1975.
- Epifanio, C. E., C. M. Logan and C. Turk. "Culture of Six Species of Bivalves in a Recirculating Seawater System". Proceedings of the 10th European Symposium of Marine Biology, 1976.

- Freedman, M. and T. Morris. "A Dwindling Harvest Industry". Sunday Newsday, Long Island, New York, Suffolk Edition, p.3, March 6, 1983.
- Galtsoff, P. S. "The American Oyster *Crassostrea virginica* Gmelin". Fishery Bulletin of the Fish and Wildlife Service, 64(1964):397-301.
- Gates, J. M., G. C. Matthiessen and C. A. Griscon. "Aquaculture in New England". Sea Grant University of Rhode Island, Marine Technical Report Series Number 18, Kinston, 1974.
- Goldstein, M. "Plan for Clams - Without Rocking the Boat". Sunday Newsday, Long Island Newspaper, Suffolk Edition, p.21, March 7, 1983
- Hanson, J.A. (ed.) "Open Sea Mariculture". Pennsylvania: Hutchinson and Ross, 1974.
- Hardin, G. "The Tragedy of the Commons". Science, 162 (1968):1243-1247.
- Henderson, S. P. "Shellfish Hatcheries, an Industry View," in Drug and Food from the Sea: Myth or Reality? P.N. Kaul and C.J. Sindermann, eds., University of Oklahoma, 1978.
- Im, K. H., R. S. Johnston and R. D. Langmo. "The Economics of Hatchery Production of Pacific Oyster Seed: A Research Progress Report". Proceedings of the National Shellfisheries Association, 66 (1976):81-94.
- Im, K. H. and R. D. Langmo. "Economic Analysis of Producing Pacific Oyster Seed in Hatcheries". Proceedings of the National Shellfisheries Association, 67 (1977):17-28.
- Kochiss, J. M. "Oystering from New York to Boston". Connecticut: Wesleyan University Press, 1974.
- Korringa, P. "Farming the Cupped Oyster of the Genus *Crassostrea*". New York: Elsevier Scientific Publishing Co., 1976.
- Loosanoff, V. L. and H. C. Davis. "Rearing of Bivalve Mollusks". Advances in Marine Biology 1(1963a):1-136.
- Loosanoff, V. L. and H. C. Davis. "Shellfish Hatcheries and Their Future". United States Department of the Interior, Fish and Wildlife Service, Washington, D.C., Commercial Fisheries Review, 25, 1963b.
- Mansfield, E. "Microeconomics - Theory and Application". New York: WW Norton and Co., 1979.
- Manzi, J. J., V. G. Burrell Jr. and H. Q. Clawson. "Commercialization of the Hard Clam (*Mercenaria mercenaria*), Mariculture in South Carolina". Preliminary Report, in J. W. Avault Jr. (ed.), Proceedings of the Twelfth Annual Meeting World Mariculture Society. Louisiana State University, 1982a.
- Manzi, J. J., V. G. Burrell Jr., H. Q. Clawson, F. S. Stevens and M. B. Maddox. "Hard Clam, *Mercenaria mercenaria* (Linne), Mariculture in South Carolina", draft for review only, Oct., 1982b.
- Matthiessen, G. C. "Review of Oyster Culture and the Oyster Industry in North America". Number 2528, Woods Hole Oceanographic Institute, 1970.
- Matthiessen, G. C. "The Oyster Industry of Massachusetts and the Introduction of Exotic Species" in R. Mann (ed.), Exotic Species and Mariculture, pp. 213-224, MIT Press, Cambridge, 1979.
- Matthiessen, G. C. and L. J. Smith. "Analysis of Methods for the Culture of *Crassostrea virginica* in New England". Proceedings of the World Mariculture Society, 10 (1979):609-623.
- Matthiessen, G. C. and R. C. Toner. "Possible Methods of Improving the Shellfish Industry of Martha's Vineyard, Duke's County, Massachusetts". Marine Research Foundation, Inc., Edgartown, Massachusetts, 1966.
- McHugh, J. and A. Williams. "Historical Statistics of the Fisheries of the New York Bight", New York Sea Grant Institute, Albany, New York, 1976.
- Miller, R. "A View Through the Porthole by a Commercial Fisherman" in O. Terry (ed.) Proceedings of the Symposium on Aquaculture in New York State, 1977.
- Myers, J. M. Production and Feeding Costs of a Bivalve Seed Hatchery. Unpublished M.S. thesis, June 1985.
- Pillay, T. V. R. "The State of Aquaculture 1975". FAO Technical Conference on Aquaculture, May 26 - June 2, 1976, Kyoto, Japan, pp. 1-13.
- Regional Marine Resources Council, Committee of the Nassau-Suffolk Regional Planning Board. "Guidelines for Management of Long Island Hard Clam Resources", pp. 2-13, 1974.
- Rhodes, E. W. and W. S. Landers. "Growth of Oyster Larvae *Crassostrea virginica* of Various Sizes in Different Concentrations of the chrysophyte, *Isochrysis galbana*". Proceedings of the National Shellfisheries Association, 63(1973):53-59.
- Rhodes, T. "Comments of a Bayman" in O. Terry and D. Chase (eds.) Proceedings of the Symposium on Mariculture in New York State, Southampton College, New York, 1974.
- Ritchie, T. P. "A Comprehensive Review of the Commercial Clam Industries in the United States". Delaware Sea Grant Program Publication, #DEL-SG-26-76, United States Government Printing Office, Washington, D.C., 1977.
- Smith, B. STOP Project Director, interviews, 1982-83.
- Terry, O. W. "Aquaculture," MESA New York Bight Atlas Monograph Number 17, New York Sea Grant Institute, Albany, New York, June 1977.
- Thomas, G. B., Jr. Calculus and Analytic Geometry. 3rd ed. Reading, Mass.: Addison-Wesley Publishing Co., Inc. 1960.
- Ukeles, R. "View on Bivalve Larvae Nutrition". Proceedings of the First International Conference on Aquaculture Nutrition, pp. 127-162, 1975.
- United States Department of Commerce, NOAA, NMFS. "New York Landings, Annual Summary 1973". Current Fisheries Statistics Number 6412, p.2, Washington, D.C., 1974.
- United States Department of Commerce, NOAA, NMFS. "Fishery Statistics of the United States". Washington, D.C., 1979.

APPENDIX A

```

0001          PROGRAM ORGNSM
0002          -----
0003          ORGNSM:
0004          -----
0005          C VARIABLES DO NOT CHANGE WHEN ORGANISM CHANGES
0006
0007          C STRUCTURE:  GENERAL PROGRAM INFORMATION
0008          C OPTION TO CALL SUBR. INFORM TO DISPLAY PROGRAM INFOR.
0009          C CALLS SUBR. READIN TO READ IN ALL DATA
0010          C CALLS SUBR. BODY TO DO COMPUTATIONS
0011          C OFFERS OPTION TO RERUN PROGRAM WITH MINOR CHANGES
0012
0013          C????????????????????????????????????????????????????????????????????
0014          C GENERAL PROGRAM INFORMATION:
0015          -----
0016
0017          C TITLE :  ORGANISM COST SIMULATION MODEL
0018          C AUTHOR:  JULIA A. MYERS
0019          C PLACE :  CORNELL UNIVERSITY
0020          C DATE  :  DECEMBER 6, 1986
0021
0022          C LANGUAGE:  FORTRAN FF - IBM COMPATIBLE
0023          C NONSTANDARD - MIXED CHARACTER COMMON BLOCKS, INCLUDE,
0024          C 'S' IN VARIABLE NAMES
0025          C DOCUMENTATION:  USER SPREAD-SHEET (FOR INPUTTING DATA)
0026
0027
0028          C THIS INTERACTIVE FORTRAN PROGRAM CONTAINS THE BASIC STRUCTURE
0029          C OF A MODEL NECESSARY TO SIMULATE THE COST OF PRODUCING
0030          C ANY ORGANISM THAT GROWS BY STAGES. THE COST AND QUANTITY OF INPUTS
0031          C IS DETERMINED AT EACH STAGE OF THE ORGANISMS' LIVES AND AT
0032          C DIFFERENT PERIODS IN THOSE STAGES.
0033          C THE PROGRAM'S MODULAR DESIGN ALLOWS SUBROUTINES TO BE ADDED
0034          C FOR SPECIFIC INPUTS AND STAGES OF THE PRODUCTION PROCESS
0035          C THEREFORE, IT IS POSSIBLE TO DETERMINE ALL OR ONLY PART OF THE TOTAL
0036          C COST OF PRODUCTION. EACH SUBROUTINE CONTAINS A DESCRIPTION OF
0037          C THE PROGRAM VARIABLES THAT MUST BE RETURNED TO THE PROGRAM.
0038          C THE CURRENT CAPABILITIES OF THE PROGRAM ARE INITIALIZED IN BLOCK
0039          C DATA, BLK1. THESE MUST BE RESET WHEN SUBROUTINES ARE ADDED OR
0040          C DELETED. SEE THE ACCOMPANYING PROGRAM DOCUMENTATION.
0041          C THE PROGRAM WAS TESTED FOR THE PRODUCTION OF BIVALVE SEED IN AN
0042          C ENCLOSED HATCHERY. THUS, SOME SUBROUTINES ARE SPECIFIC TO
0043          C THIS ORGANISM AND THE APPROPRIATE CHANGES MUST BE MADE FOR OTHERS.
0044
0045          C NAME      DESCRIPTION OF REQUIRED ASSOCIATED FILES INCLUDED IN PROGRAM
0046          C IPAR1    PARAMETERS NECESSARY FOR GENERAL COMMON BLOCK VARIABLES
0047          C ICMATN   VARIABLES IN COMMON NECESSARY TO RUN GENERAL PROGRAM
0048          C ICMATN   VAR. IN COMMON FOR UNIT NUMBERS OF READ AND WRITE
0049          C ICBLK1   VAR. IN COMMON USED IN SUBR. BLOCK DATA BLK1
0050          C ICBLKO   VAR. IN COMMON, USED IN SUBR. BLOCK DATA BLK1 FOR ORG.
0051          C ICORG   VAR. IN COMMON SPECIFIC TO ORGANISM
0052          C (NOTE:  OTHER FILES MAY BE REQUIRED FOR ORGANISM SPECIFIC ROUTINES)
0053          C-----
0054          C 0056
0055          C 0057
0056          C 0058

```

```

0059          C FLOW OF CONTROL:
0060          C ORGNSM
0061          C I
0062          C
0063          C I
0064          C I
0065          C I
0066          C I
0067          C I
0068          C I
0069          C I
0070          C I
0071          C I
0072          C I
0073          C I
0074          C I
0075          C I
0076          C I
0077          C I
0078          C I
0079          C I
0080          C I
0081          C I
0082          C I
0083          C I
0084          C I
0085          C I
0086          C I
0087          C I
0088          C I
0089          C I
0090          C I
0091          C I
0092          C I
0093          C I
0094          C I
0095          C I
0096          C I
0097          C I
0098          C I
0099          C I
0100          C I
0101          C I
0102          C I
0103          C I
0104          C I
0105          C I
0106          C I
0107          C I
0108          C I
0109          C I
0110          C I
0111          C I
0112          C I
0113          C I
0114          C I
0115          C I
0116          C I
0117          C I
0118          C I
0119          C I
0120          C I
0121          C I
0122          C I

```

```

0091          C NAME      DESCRIPTION OF REQUIRED SUBROUTINES AND FUNCTIONS BY APPEARANCE
0092          C ORGNSM   MAIN PROGRAM
0093          C INFORM   READS OUT INFORMATION CONCERNING THE PROGRAM
0094          C READIN   READS IN OPTION INFO AND CALLS ALL SUBROUTINES TO READ IN
0095          C ROUT    READS OUT DATA FROM READIN
0096          C READS   READS IN DATA TO RUN GENERAL PROGRAM
0097          C ROUTS   READS OUT DATA FROM READS
0098          C READO   READS IN DATA SPECIFIC TO ORGANISM
0099          C ROUTO   READS OUT DATA FROM READO
0100          C READN1  READS IN DATA TO COMPUTE QUANT OF INPUT 1
0101          C READN2  READS IN DATA TO COMPUTE QUANT OF INPUT 2
0102          C READN3  READS IN DATA TO COMPUTE QUANT OF INPUT 3
0103          C READN4  READS IN DATA TO COMPUTE QUANT OF INPUT 4
0104          C READN5  READS IN DATA TO COMPUTE QUANT OF INPUT 5
0105          C READN6  READS IN DATA TO COMPUTE QUANT OF INPUT 6
0106          C ROUTN1  READS OUT DATA ENTERED IN READN1
0107          C ROUTN2  READS OUT DATA ENTERED IN READN2
0108          C ROUTN3  READS OUT DATA ENTERED IN READN3
0109          C ROUTN4  READS OUT DATA ENTERED IN READN4
0110          C ROUTN5  READS OUT DATA ENTERED IN READN5
0111          C ROUTN6  READS OUT DATA ENTERED IN READN6
0112          C PRICE1  COMPUTES PRICE OF INPUT 1
0113          C PRICE2  COMPUTES PRICE OF INPUT 2
0114          C PRICE3  COMPUTES PRICE OF INPUT 3
0115          C PRICE4  COMPUTES PRICE OF INPUT 4
0116          C PRICE5  COMPUTES PRICE OF INPUT 5
0117          C PRICE6  COMPUTES PRICE OF INPUT 6
0118          C BODY   LOOPS THRU EACH MONTH TO CALL $ AND QUANT SUBR.
0119          C NUMBRB COMPUTES NUMBER OF ORGANISM AT PERIOD,J
0120          C SIZE   COMPUTES SIZE OF ORGANISM AT PERIOD J
0121          C QUANT1 COMPUTES QUANT OF INPUT 1 USED IN J
0122          C QUANT2 COMPUTES QUANT OF INPUT 2 USED IN J

```

C	QUANT3	COMPUTES	QUANT	OF	INPUT	3	USED	IN	J	0123	C	ONOBEG	R	CMAIN	MAXJI	NO. OF ORGANISMS AT BEGIN	0187
C	QUANT4	COMPUTES	QUANT	OF	INPUT	4	USED	IN	J	0124	C	ONOFJ	R	CMAIN	MAXJI	AVG. NO. OF ORGANISMS DURING EACH PERIOD, J	0188
C	QUANT5	COMPUTES	QUANT	OF	INPUT	5	USED	IN	J	0125	C	ONOFJ	R	CMAIN	MAXJI	NO. OF ORGANISMS AT END OF EACH PERIOD, J	0189
C	QUANT6	COMPUTES	QUANT	OF	INPUT	6	USED	IN	J	0126	C	OTYPE	I	CBLK1		TYPE OF ORGANISM TO DO CALCULATIONS FOR	0190
C	COSTJ	SUMS	AND	COMPUTES	COST	OF	INPUTS	IN	PERIOD, J	0127	C	OPNSOK	I	CBLK1	NOOFN	CAN PROGRAM COMPUTE PRICE OF INPUT, N?	0191
C	WRTOU	PRINTS	COSTS	AND	QUANTITIES					0128	C					0=YES, 1=NO	0192
C	WOUTN1	PRINTS	ADDITIONAL	INFO	GENERATED	ABOUT	INPUT	1		0129	C	OPNNEW	C*1	CMAIN	NOOFI	DOES THE USER WANT TO ENTER N IN I?	0193
C	WOUTN2	PRINTS	ADDITIONAL	INFO	GENERATED	ABOUT	INPUT	2		0130	C					CAN PROGRAM COMPUTE INPUT, N, IN STAGE, I?	0194
C	WOUTN3	PRINTS	ADDITIONAL	INFO	GENERATED	ABOUT	INPUT	3		0131	C	OPNOK	I	CBLK1	NOOFI	0=YES, 1=NO	0195
C	WOUTN4	PRINTS	ADDITIONAL	INFO	GENERATED	ABOUT	INPUT	4		0132	C	OPTION	I	CMAIN	NOOFN	OPTIONS 1-3 THAT IS CHOSEN TO COMPUTE Q	0197
C	WOUTN5	PRINTS	ADDITIONAL	INFO	GENERATED	ABOUT	INPUT	5		0133	C	OPQOK	I	CBLK1	3	CAN PROGRAM DO EACH OPTION FOR COMPUTING?	0198
C	WOUTN6	PRINTS	ADDITIONAL	INFO	GENERATED	ABOUT	INPUT	6		0134	C					0=YES, 1=NO	0199
C	AGENSZ	FUNCTION	RELATING	AGE	TO	SURVIVAL				0135	C	OPTOK	I	CBLK1	NOOFN	CAN USER CHANGE NO. OF TYPES, T OF INPUT, N?	0200
C	AGESZE	FUNCTION	RELATING	AGE	TO	SIZE				0136	C					0=YES, 1=NO	0201
C	BLK1	BLOCK	DATA	FOR	ORGANISM	DEPENDENT	VARIABLES			0137	C	OUTON	I	CBLK1	NOOFN	IS IT POSSIBLE TO WRITE ADDITIONAL INFO N?	0202
C	BLK0	BLOCK	DATA	FOR	ORGANISM	DEPENDENT	VARIABLES			0138	C	PASEDM	I	CMAIN	MAXJI	IF PAST MONTHS OF PROD., SET TO 1	0203
C	(NOTE: OPTIONAL SUBROUTINES LISTED UNDER APPROPRIATE SUBROUTINES)									0139	C	PRICET	R	CMAIN	NOOFN	PRICE OF TYPE, T, OF INPUT, N, IN MONTH, M.	0204
C										0140	C				MAXT		0205
C										0141	C				MAXM		0206
C										0142	C				NOOFN	DOES USER WANT TO COMPUTE PRICE OF INPUT, N?	0207
C										0143	C	QOPNS	C*1	CMAIN	NOOFN	PRINT OUT ADDITIONAL INFO ABOUT INPUT N?	0208
C										0144	C	QOUTQN	C*1	CBLK1	NOOFN	UNIT NUMBER FOR READ FILE	0209
C										0145	C	RF	I	CUNIT		UNIT NUMBER FOR READ INTERACTIVE	0210
C										0146	C	RI	I	CUNIT		AVG SIZE OF ORGANISM DURING PERIOD	0211
C										0147	C	SZEOFJ	R	CMAIN	MAXJI	SIZE OF ORGANISM AT END OF PERIOD	0212
C										0148	C	SZEPUR	R	CMAIN	MAXJI	NO. OF STAGES TO DO COMPUTATIONS FOR (OP=1), 0213	0213
C										0149	C	TOTALI	I	CMAIN	NOOFI	QUANTITY OF TYPE, T, OF INPUT, N, USED IN	0214
C										0150	C	TQUANI	R	CMAIN	NOOFN	MONTH, M FOR STAGE, I.	0215
C										0151	C				MAXT		0216
C										0152	C				MAXM		0217
C										0153	C				NOOFN	QUANTITY OF TYPE, T, OF INPUT, N, USED IN	0218
C										0154	C	TQUANJ	R	CMAIN	NOOFN	MONTH, M, IN PERIOD, J.	0219
C										0155	C				MAXT		0220
C										0156	C	WKSUPJ	R	CMAIN		ELAPSED WEEKS UP TO END OF PERIOD, J	0221
C										0157	C	WF	I	CUNIT		UNIT NUMBER FOR WRITE FILE	0222
C										0158	C	WI	I	CUNIT		UNIT NUMBER FOR WRITE INTERACTIVE	0223
C										0159	C	WO	I	CUNIT		UNIT NUMBER TO WRITE OUTPUT FILE	0224
C										0160	C	WON	I	CUNIT	NOOFN	UNIT TO WRITE TO N DIFFERENT OUTPUT FILES	0225
C										0161	C					MAXT	0226
C										0162	C						0227
C										0163	C						0228
C										0164	C						0229
C										0165	C						0230
C										0166	C						0231
C										0167	C						0232
C										0168	C	DAYS11	R	CORG		DAYS IN I=1, CONDITIONING	0233
C										0169	C	EQNO	C*50	CBLKO	1	EQUATION RELATING AGE TO SURVIVAL	0234
C										0170	C	EQNOA	R	CBLKO		PARAMETER A FOR AGE NUMBER EQUATION	0235
C										0171	C	EQNOAF	R	CBLKO		PARAMETER A RECOMMENDATION FOR AGE/NO EQ.	0236
C										0172	C	EQQ5	C*50	CBLKO	2	EQUATION RELATING AGE TO QUANT N=5	0237
C										0173	C	EQQ5A	R	CBLKO	2	PARAMETER A FOR QUANT N=5 EQUATION	0238
C										0174	C	EQQ5AF	R	CBLKO	2	RECOMMENDATION FOR PARAM. A FOR QUANT5 EQ	0239
C										0175	C	EQQ5B	R	CBLKO	2	PARAMETER B FOR QUANT N=5 EQUATION	0240
C										0176	C	EQQ5BF	R	CBLKO	2	RECOMMENDED VALUE FOR PARAM B FOR QUANT5 EQ	0241
C										0177	C	EQSZ	C*50	CBLKO	1	EQUATION RELATING AGE TO SIZE	0242
C										0178	C	EQSZA	R	CBLKO		PARAMETER A FOR AGE SIZE EQUATION	0243
C										0179	C	EQSZAF	R	CBLKO		RECOMMENDED VALUE FOR PARAM A FOR SIZE EQ	0244
C										0180	C	EQSZB	R	CBLKO		PARAMETER B FOR AGE SIZE EQUATION	0245
C										0181	C	EQSZBF	R	CBLKO		RECOMMENDED VALUE FOR PARAM B FOR SIZE EQ	0246
C										0182	C	FILTER	R	CORG	MAXI	FEDDING FILTERING RATE, AS A FRACTION	0247
C										0183	C	NAME	C*1	CBLKO	NOOFO	NAME OF ORGANISM PROGRAM IS SPECIFIC TO	0248
C										0184	C	NAMEO	C*10	CBLKO	NOOFO	NAME OF ORGANISM PROGRAM	0249
C										0185	C	NAMEI	C*10	CBLKO	NOOFI	NAME OF EACH STAGE, I	0250
C										0186	C	NAMEN	C*10	CBLKO	NOOFN	NAME OF EACH INPUT, N	

```
C 251  WRITE(WI,50000)  WO
C 252  READ(RI,*) UNIT
C 253  WO = UNIT
C 254  DO 200 N=1,NOOFN
C 255  IF (COUTQ(N).EQ. 0) THEN
C 256  WRITE(WI,51000)  NAMEN(N),  WON(N)
C 257  READ(RI,*)  UNITN(N)
C 258  WON(N) = UNITN(N)
C 259  ENDDIF
C 260  CONTINUE
C 261  WF = WI
C 262  RF = RI
C 263  RERUN = .TRUE.
C 264  CALL REREAD(WI,RI,RERUN)
C 265  GOTO 100
C 266  ENDDIF
C 267
C 268
C 269
C 270  FORMAT (A1)
C 271
C 272  10000 FORMAT (' ' ,15X,A,'COST SIMULATION MODEL',' ',15X,32('-'))/0335
C 273  2  '//'' DO YOU WANT TO SEE THE PROGRAM INFORMATION? Y OR N'
C 274
C 275  30000 FORMAT(' ' ,15X,' DO YOU WANT TO RERUN THE PROGRAM WITH ONLY A ',
C 276  2  'FEW CHANGES?',' ','Y OR N')
C 277
C 278  49000 FORMAT(' ' ,15X,' GIVE A NEW DATE OR PROGRAM IDENTIFIER - UP TO 50 ',
C 279  2  'CHARACTER IN QUOTES.//' (PREVIOUS DATE= ,A,')')
C 280
C 281  50000 FORMAT(' ' ,15X,'WHAT UNIT DO YOU WANT THE NEW OUTPUT FILES ',
C 282  2  'TO GO TO? - INTEGER 10 TO 99//' (PREVIOUS UNIT = ,I2,')')
C 283  51000 FORMAT(' ' ,15X,'WHAT UNIT DO YOU WANT THE ADDITIONAL INFORMATION ',
C 284  2  'ABOUT ,A//', TO GO TO IF THE OPTION TO PRINT IS ',
C 285  3  'CHOSEN?//' (PREVIOUS UNIT = ,I2,')')
C 286
C 287  STOP 'PROGRAM COMPLETED'
C 288  END
C 289
C 290  SUBROUTINE INFORM
C 291  C ORGNM:
C 292  C-----
C 293  C SUBROUTINE CAN CHANGE WITH ORGANISM
C 294  C NO VARIABLES RETURNED
C 295  C STRUCTURE: WRITES OUT INFORMATION CONCERNING PROGRAM USE
C 296  C 1. PARAMETERS
C 297  C 3. COMMON BLOCKS
C 298  C INCLUDE (ICBLK1)
C 299  C INCLUDE (ICUNIT)
C 300  C INCLUDE (ICMAIN)
C 301  C INCLUDE (ICBLKO)
C 302  C LOCAL VARIABLES
C 303  C-----
C 304
C 305  C OPTION TO RERUN WITH A FEW CHANGES TO DATA
C 306  WRITE(WI,30000)
C 307  READ(RI,1000) QUEST2
C 308  RESPECIFY FILE READ AND WRITE UNIT NUMBERS TO INTERACTIVE
C 309  AND OUTPUT FILES TO NEW FILES
C 310  IF (QUEST2.EQ. 'Y') THEN
C 311  WRITE(WI,49000)  DATE
C 312  READ(RI,*) DATE
C 313
C 314
C 315
C 316
C 317
C 318
C 319
C 320
C 321
C 322
C 323
C 324
C 325
C 326
C 327
C 328
C 329
C 330
C 331
C 332
C 333
C 334
C 335
C 336
C 337
C 338
C 339
C 340
C 341
C 342
C 343
C 344
C 345
C 346
C 347
C 348
C 349
C 350
C 351
C 352
C 353
C 354
C 355
C 356
C 357
C 358
C 359
C 360
C 361
C 362
C 363
C 364
C 365
C 366
C 367
C 368
C 369
C 370
C 371
C 372
C 373
C 374
C 375
C 376
C 377
C 378
```



```
INTEGER I, N, O           0379
CHARACTER*1  QUEST       0380
2  I2, ' TYPES'/' THE USER CAN ',             0443
3  'CHOOSE THE NUMBER OF TYPES OF THE FOLLOWING INPUTS: ' 0444
                                0445
32000 FORMAT (2X,A10)     0446
                                0447
40000 FORMAT( ' ' / ' THE PROGRAM IS CAPABLE OF COMPUTING THE FOLLOWING ', 0448
2  'INPUT QUANTITIES: ' ) 0449
                                0450
41000 FORMAT( ' ',IX,A, ' IN THE ',A10, ' STAGE' ) 0451
                                0452
50000 FORMAT( ' ' / ' THE PROGRAM IS CAPABLE OF COMPUTING THE FOLLOWING ', 0453
2  'INPUT PRICES: ' )   0454
                                0455
51000 FORMAT( ' ',IX,A10) 0456
                                0457
60000 FORMAT( ' ' / ' NOTE: ALTHOUGH THE PROGRAM CAN COMPUTE FOR ',I2,   0458
2  'MONTHS, FROM ',A9, 'TO ',A10/ ' ', 'THE PRODUCTION SEASON ', 0459
3  'MUST RUN SEQUENTIALLY FROM THE 1ST TO THE LAST MONTH. ' ) 0460
                                0461
70000 FORMAT( ' ' / ' FOLLOW THESE GUIDELINES TO ENTER REQUIRED DATA: ' / 0462
2  ' USE A DECIMAL POINT UNLESS INTEGER IS SPECIFIED. ' /          0463
3  ' USE NO COMMAS, NO DOLLAR SIGNS. ' /                            0464
4  ' USE NO POSITIVE SIGNS; USE NEGATIVE WHEN APPLICABLE. ' /0465
5  ' ALWAYS LEAVE A SPACE BETWEEN ENTRIES ON THE SAME LINE. ' /0466
6  ' ENTER 0. IF NO VALUE EXISTS FOR THE REQUEST. ' /             0467
7  ' USE METRIC UNITS UNLESS OTHERWISE SPECIFIED. ' /           0468
8  ' ERRORS CAN BE CORRECTED PERIODICALLY. ' )                  0469
                                0470
                                0471
RETURN                   0472
END                       0473
                                0474
                                0475
                                0476
                                0477
                                0478
                                0479
                                0480
                                0481
                                0482
                                0483
                                0484
                                0485
                                0486
                                0487
                                0488
                                0489
                                0490
                                0491
                                0492
                                0493
                                0494
                                0495
                                0496
                                0497
                                0498
                                0499
                                0500
                                0501
                                0502
                                0503
                                0504
                                0505
                                0506
2  I2, ' TYPES'/' THE USER CAN ',             0443
3  'CHOOSE THE NUMBER OF TYPES OF THE FOLLOWING INPUTS: ' 0444
                                0445
32000 FORMAT (2X,A10)     0446
                                0447
40000 FORMAT( ' ' / ' THE PROGRAM IS CAPABLE OF COMPUTING THE FOLLOWING ', 0448
2  'INPUT QUANTITIES: ' ) 0449
                                0450
41000 FORMAT( ' ',IX,A, ' IN THE ',A10, ' STAGE' ) 0451
                                0452
50000 FORMAT( ' ' / ' THE PROGRAM IS CAPABLE OF COMPUTING THE FOLLOWING ', 0453
2  'INPUT PRICES: ' )   0454
                                0455
51000 FORMAT( ' ',IX,A10) 0456
                                0457
60000 FORMAT( ' ' / ' NOTE: ALTHOUGH THE PROGRAM CAN COMPUTE FOR ',I2,   0458
2  'MONTHS, FROM ',A9, 'TO ',A10/ ' ', 'THE PRODUCTION SEASON ', 0459
3  'MUST RUN SEQUENTIALLY FROM THE 1ST TO THE LAST MONTH. ' ) 0460
                                0461
70000 FORMAT( ' ' / ' FOLLOW THESE GUIDELINES TO ENTER REQUIRED DATA: ' / 0462
2  ' USE A DECIMAL POINT UNLESS INTEGER IS SPECIFIED. ' /          0463
3  ' USE NO COMMAS, NO DOLLAR SIGNS. ' /                            0464
4  ' USE NO POSITIVE SIGNS; USE NEGATIVE WHEN APPLICABLE. ' /0465
5  ' ALWAYS LEAVE A SPACE BETWEEN ENTRIES ON THE SAME LINE. ' /0466
6  ' ENTER 0. IF NO VALUE EXISTS FOR THE REQUEST. ' /             0467
7  ' USE METRIC UNITS UNLESS OTHERWISE SPECIFIED. ' /           0468
8  ' ERRORS CAN BE CORRECTED PERIODICALLY. ' )                  0469
                                0470
                                0471
RETURN                   0472
END                       0473
                                0474
                                0475
                                0476
                                0477
                                0478
                                0479
                                0480
                                0481
                                0482
                                0483
                                0484
                                0485
                                0486
                                0487
                                0488
                                0489
                                0490
                                0491
                                0492
                                0493
                                0494
                                0495
                                0496
                                0497
                                0498
                                0499
                                0500
                                0501
                                0502
                                0503
                                0504
                                0505
                                0506
C 1. PARAMETERS
C 2. ARGUMENTS
C 3. COMMON BLOCKS
C 4. INCLUDE (IPAR1)
C 5. ARGUMENTS
C 6. LOGICAL RED, RERED, WRT, REWRT, HREAD
C 7. COMMON BLOCKS
C 8. INCLUDE (ICMAIN)
C 9. INCLUDE (ICBLK1)
```

```

0507 INCLUDE (ICBLKO)
0508 INCLUDE (ICUNIT)
0509 C 4. LOCAL VARIABLES
0510 INTEGER I, INFO, K, K2, KGOTO, KLAST, N, NOOPK, OPN,
0511 2
0512 LOGICAL CHECK, RIPROB, RUN1
0513 CHARACTER * 1 QFILE, QUEST, QUEST2, QUEST3
0514 PARAMETER (NOOPK = 8)
0515 DIMENSION INFO(NOOPK)
0516
0517 C NAME DESCRIPTION OF LOCAL VARIABLES AND ARGUMENTS
0518 -----
0519 C CHECK USED TO DETERMINE IF LOOP HAS BEEN ENTERED
0520 C HREAD ARGUE: IOSTAT TO SHOW PROBLEM IN READING FILE (2-WAY COMM.)
0521 C I LOOP COUNTER FOR STAGE, I
0522 C INFO INFORMATION LINE NUMBER
0523 C K COUNTER FOR LINE NUMBER
0524 C K2 COUNTER FOR SUBROUTINE LINE NUMBERS
0525 C KGOTO NO. OF THE INFO LINE TO REINPUT DATA
0526 C KLAST THE LAST INFO LINE, K INPUT
0527 C N LOOP COUNTER FOR INPUTS
0528 C NOOPK NUMBER OF INPUT LINES
0529 C OPN TO COUNT UP THE NO. OF N TO ENTER FOR EACH I
0530 C QFILE WANT TO USE A DATA FILE?
0531 C QUEST GENERAL QUESTION FOR USER (IS WRITTEN OVER)
0532 C QUEST2 OPTION TO STOP, REDO, OR CONTINUE WITH DATA ENTRY
0533 C QUEST3 OPTION TO SEE OPTION INFORMATION
0534 C R UNIT NUMBER FOR READ
0535 C RED ARGUE: UNIT NUMBERS FOR READ
0536 C RERED ARGU: UNIT NO. FOR 2ND ENTRY POINT - ALWAYS INTERACTIVE
0537 C RERUN ARGU: TO ALLOW INFO PRINTING IF NOT RUN1, FROM 2ND ENTRY PNT0537
0538 C REMWT ARGU: UNIT NO FOR 2ND ENTRY POINT - ALWAYS INTERACTIVE
0539 C RIPROB TO DETERMINE IF FILE PROBLEM HAS BEEN ENCOUNTERED
0540 C RUN1 USED TO PRINT INFO THE FIRST RUN THROUGH
0541 C RUN2 ARGUE: IF FIRST RUN THROUGH DATA INPUT ALREADY DONE
0542 C UNIT UNIT NO FOR OUTPUT FILE TO GO TO
0543 C W UNIT NUMBER FOR WRITE
0544 C WRG ARGUE: UNIT NUMBERS FOR WRITE
0545 C Z 2ND COUNTER FOR LINE NUMBERS
0546 -----
0547
0548 C SET ALL LINES TO 0 IN ORDER TO READ IN THE FIRST TIME
0549 DO 100 K=1,NOOPK
0550 INFO(K) = 0
0551 CONTINUE
0552 RUN1 = .TRUE.
0553 RUN2 = .FALSE.
0554
0555 C INITIALIZE FILE VARIABLES
0556 RIPROB = .FALSE.
0557 HREAD = 0
0558
0559 C OPTION TO TERMINATE
0560 WRITE(WI,20000)
0561 READ(RI,1000) QUEST
0562 IF (QUEST.EQ. 'Y') STOP 'COMPLETE SPREAD SHEET'
0563
0564 C READ IN PROGRAM DATA/IDENTIFIER AND OUTPUT FILE UNIT NUMBER
0565 WRITE(WI,25000)
0566 READ(RI,*) DATE
0567 WRITE(WI,26000)
0568 READ(RI,*) UNIT
0569 W = UNIT
0570
0571 C OPTION TO INPUT BY SEQUENTIAL FILE
0572 WRITE(WI,30000)
0573 READ(RI,1000) QFILE
0574 IF (QFILE.EQ. 'Y') THEN
0575 WRITE(WI,31000)
0576 READ(RI,*) NAMEF1
0577 WRITE(WI,32000)
0578 W = WF
0579 R = RF
0580 WRT = WF
0581 RED = RF
0582
0583 ELSE
0584 W = WI
0585 R = RI
0586 WRT = WI
0587 RED = RI
0588 NAMEF1 = 'NOFILE'
0589 ENDIF
0590
0591 C OPTION INFORMATION IDENTIFIER
0592 WRITE(WI,39000)
0593
0594 C LIST OPTIONS AVAILABLE TO READ DATA IF OPTION TO DISPLAY CHOSEN
0595 250 WRITE(W,40000)
0596 WRITE(W,40010)
0597 READ(R,1000) QUEST3
0598 IF (QUEST3.EQ. 'Y') THEN
0599 IF (OPOOK(1).EQ. 0) THEN
0600 WRITE(W,40100)
0601 WRITE(W,40150)
0602 ENDIF
0603 IF (OPOOK(2).EQ. 0) THEN
0604 WRITE(W,40200)
0605 WRITE(W,40150)
0606 ENDIF
0607 IF (OPOOK(3).EQ. 0) WRITE(W,40300)
0608 WRITE(W,40500)
0609 READ(R,1000,ERR=600,IOSTAT=HREAD) QUEST
0610
0611 C LIST INPUTS CALCULATABLE IN EACH STAGE
0612 DO 300 N=1,NOOPN
0613 WRITE(W,41100) N, NAMEN(N)
0614 CHECK = .TRUE.
0615 DO 310 I=1,NOOFI
0616 IF (OPNOK(I,N).EQ. 0) THEN
0617 WRITE(W,41200) NAMEI(I)
0618 CHECK = .FALSE.
0619
0620 ENDIF
0621 CONTINUE
0622 IF (CHECK) WRITE(W,41300)
0623
0624 C USER CHOOSES AN OPTION
0625 WRITE(W,42000)
0626 READ(R,*,ERR=600,IOSTAT=HREAD) OPTION
0627 ENDIF
0628
0629
0630
0631
0632
0633
0634

```

```

0635 C OPTION 1 REQUIRES DEFINITION OF THE FIRST STAGE OF INTEREST
0636 IF (OPTION .EQ. 1) THEN
0637   WRITE(W,43000)
0638 READ(R,*,ERR=600,IOSTAT=HREAD) BEGINI
0639 IF (BEGINI .LE. 1 .OR. BEGINI .GT. NOOFI) THEN
0640   WRITE(W,43100)
0641   GOTO 320
0642   ENDF
0643 ELSE
0644   BEGINI = 1
0645   ENDF
0646
0647 C USER CHOOSES WHICH INPUTS TO READ IN AS QUANTITY ONLY, INITIALIZE TO
0648 C 1 ONLY THOSE THAT ARE COMPUTABLE ARE OFFERED
0649 IF (OPTION .NE. 3) THEN
0650   WRITE(W,44000)
0651   DO 350 N=1,NOOFN
0652     WRITE(W,41100) N, NAMEN(N)
0653     CHECK = .TRUE
0654     DO 360 I=1,NOOFI
0655       IF (I .LT. BEGINI) THEN
0656         OPNNEW(I,N) = 'E'
0657       ELSEIF (OPNOK(I,N) .EQ. 0 .AND. I .GE. BEGINI) THEN
0658         WRITE(W,41200) NAMEI(I)
0659         READ(R,1000,ERR=600,IOSTAT=HREAD) OPNNEW(I,N)
0660         CHECK = .FALSE
0661       ELSEIF (OPNOK(I,N) .NE. 0 .AND. I .GE. BEGINI) THEN
0662         OPNNEW(I,N) = 'E'
0663       ENDF
0664     CONTINUE
0665   IF (CHECK) WRITE(W,41300)
0666   CONTINUE
0667 ENDF
0668
0669 C IF NO STAGES TO BE COMPUTED FOR AN INPUT,
0670 C THEN ANYNOK(N) = TOTAL STAGES TO COMPUTE
0671 DO 370 N=1,NOOFN
0672   ANYNOK(N) = 0
0673   DO 380 I=BEGINI,NOOFI
0674     IF (OPNNEW(I,N) .EQ. 'C') THEN
0675       OPN = 0
0676     ELSEIF (OPNNEW(I,N) .EQ. 'E') THEN
0677       OPN = 1
0678     ENDF
0679     ANYNOK(N) = ANYNOK(N) + OPN
0680   CONTINUE
0681   TOTALI = NOOFI - BEGINI + 1
0682
0683 C OPTION TO CHECK AND REINPUT OPTION INFORMATION
0684 WRITE (WI,45000)
0685 READ(RI,1000) QUEST
0686 IF (QUEST .EQ. 'Y') CALL ROUT(WI)
0687 IF (NAMEFI .EQ. 'NOFILE') THEN
0688   WRITE(WI,45100)
0689   READ(RI,1000) QUEST
0690   IF (QUEST .EQ. 'Y') GOTO 250
0691   ENDF
0692
0693 C OPTION TO PRINT OPTION INFORMATION
0694 WRITE(WI,45300)
0695 READ(RI,1000) QUEST
0696
0697
0698
0699
0700
0701
0702
0703
0704
0705
0706
0707
0708
0709
0710
0711
0712
0713
0714
0715
0716
0717
0718
0719
0720
0721
0722
0723
0724
0725
0726
0727
0728
0729
0730
0731
0732
0733
0734
0735
0736
0737
0738
0739
0740
0741
0742
0743
0744
0745
0746
0747
0748
0749
0750
0751
0752
0753
0754
0755
0756
0757
0758
0759
0760
0761
0762

```

```

0763 IF (HREAD .LT. 0) WRITE(WI,60000)
0764 IF (HREAD .GT. 0) WRITE(WI,61000)
0765 IF (HREAD .NE. 0) RIPROB = .TRUE.
0766 QFILE = 'N'
0767 HREAD = 0
0768 W = WI
0769 R = RI
0770 WRT = WI
0771 RED = RI
0772 W = WI
0773 R = RI
0774 W = WI
0775 R = RI
0776 INFO(K) = 0
0777 CONTINUE
0778 WRT = WI
0779 RED = RI
0780 W = WI
0781 R = RI
0782 W = WI
0783 R = RI
0784 W = WI
0785 R = RI
0786 WRT = WI
0787 RED = RI
0788 W = WI
0789 R = RI
0790 W = WI
0791 R = RI
0792 W = WI
0793 R = RI
0794 W = WI
0795 R = RI
0796 W = WI
0797 R = RI
0798 W = WI
0799 R = RI
0800 W = WI
0801 R = RI
0802 W = WI
0803 R = RI
0804 W = WI
0805 R = RI
0806 W = WI
0807 R = RI
0808 W = WI
0809 R = RI
0810 W = WI
0811 R = RI
0812 W = WI
0813 R = RI
0814 W = WI
0815 R = RI
0816 W = WI
0817 R = RI
0818 W = WI
0819 R = RI
0820 W = WI
0821 R = RI
0822 W = WI
0823 R = RI
0824 W = WI
0825 R = RI
0826 W = WI
0827 R = RI

C RETURN POINT FROM MAIN PROGRAM TO RERUN THE PROGRAM WITH FEW CHANGES
ENTRY RERED(REWRT, RERED, RERUN)
WRT = WI
RED = RI
W = WI
R = RI

C OPTION TO CHECK DATA THAT HAS BEEN INPUT
WRITE(WI,62000)
READ(RI,1000) QUEST
IF (QUEST .EQ. 'Y') THEN
  WRITE(WI,62100)
  READ(RI,1000) QUEST
  IF (QUEST .EQ. 'Y') CALL ROUT(WI)
  Z = 1
  WRITE(WI,62200) Z
  READ(RI,1000) QUEST
  IF (QUEST .EQ. 'Y') CALL ROUTS(WI)
  Z = Z + 1
  WRITE(WI,62210) NAME, Z
  READ(RI,1000) QUEST
  IF (QUEST .EQ. 'Y') CALL ROUTO(WI)
  DO 605 N=1,NOOFN
    Z = Z + 1
    IF (ANYNOK(N) .LT. TOTALI) THEN
      WRITE(WI,62300) Z, NAMEN(N)
      READ(RI,1000) QUEST
      IF (QUEST .EQ. 'Y') THEN
        IF (N .EQ. 1) CALL ROUTN1(WI)
        IF (N .EQ. 2) CALL ROUTN2(WI)
        IF (N .EQ. 3) CALL ROUTN3(WI)
        IF (N .EQ. 4) CALL ROUTN4(WI)
        IF (N .EQ. 5) CALL ROUTN5(WI)
        IF (N .EQ. 6) CALL ROUTN6(WI)
      ENDIF
    ENDIF
  CONTINUE
ENDIF
605 CONTINUE
ENDIF

C OPTION TO TERMINATE, REDO INTERACTIVELY, OR CONTINUE
C PROBLEM WITH FILE, OFFER A DIFFERENT SET OF CHOICES
610 IF (RIPROB) THEN
  WRITE(WI,63000)
  WRITE(WI,63100)
  ELSEIF (.NOT. RIPROB) THEN
    WRITE(WI,63000)
    WRITE(WI,63200)
  ENDIF

C CONTROL TRANSFERRED ACCORDING TO REQUEST
C AND INPUT LINES ARE SET TO 0 OR 1 ACCORDINGLY

```

```

0827 READ(RI,1000) QUEST2
0828 IF ((QUEST2.NE.'S') .AND. (QUEST2.NE.'R') .AND. (RIPROB)) THEN
0829 GOTO 610
0830 ELSEIF (QUEST2 .EQ. 'S') THEN
0831 GOTO 800
0832 ELSEIF (QUEST2 .EQ. 'R') THEN
0833 RIPROB = .FALSE.
0834 NAMEF1 = 'NOFILE'
0835 RUN1 = .TRUE.
0836 RUN2 = .FALSE.
0837 W = WI
0838 R = RI
0839 DO 620 K=1,NOOFFK
0840 INFO(K) = 0
0841 CONTINUE
0842 WRT = WI
0843 RED = RI
0844 GOTO 250
0845 ELSEIF (QUEST2 .EQ. 'C') THEN
0846 DO 700 K=1,NOOFFK
0847 INFO(K) = 1
0848 CONTINUE
0849 ENDIF
0850 RUN1 = .FALSE.
0851 RUN2 = .TRUE.
0852 RERUN = .FALSE.
0853 W = WI
0854 R = RI
0855 W = WI
0856 R = RI
0857 W = WI
0858 R = RI
0859 W = WI
0860 R = RI
0861 W = WI
0862 R = RI
0863 W = WI
0864 R = RI
0865 W = WI
0866 R = RI
0867 W = WI
0868 R = RI
0869 W = WI
0870 R = RI
0871 W = WI
0872 R = RI
0873 W = WI
0874 R = RI
0875 W = WI
0876 R = RI
0877 W = WI
0878 R = RI
0879 W = WI
0880 R = RI
0881 W = WI
0882 R = RI
0883 W = WI
0884 R = RI
0885 W = WI
0886 R = RI
0887 W = WI
0888 R = RI
0889 W = WI
0890 R = RI

C OPTION TO CORRECT LINE BY CHOOSING A LINE TO CORRECT, EXCEPT OPTIONS
710 WRITE(WI,70000) NOOFFK
READ(RI,*) KGOTO
IF (KGOTO .LT. 0 .OR. KGOTO .GT. NOOFFK) THEN
  GOTO 710
ELSEIF (KGOTO .GT. 0) THEN
  INFO(KGOTO) = 0
  GOTO 400
ENDIF

C OPTION TO PRINT DATA ENTERED
800 WRITE(WI,80000)
READ(RI,1000) QUEST
IF (QUEST .EQ. 'Y') THEN
  CALL ROUTS(WO)
  CALL ROUTO(WO)
  IF (ANYNOK(1) .LT. TOTALI) CALL ROUTN1(WO)
  IF (ANYNOK(2) .LT. TOTALI) CALL ROUTN2(WO)
  IF (ANYNOK(3) .LT. TOTALI) CALL ROUTN3(WO)
  IF (ANYNOK(4) .LT. TOTALI) CALL ROUTN4(WO)
  IF (ANYNOK(5) .LT. TOTALI) CALL ROUTN5(WO)
  IF (ANYNOK(6) .LT. TOTALI) CALL ROUTN6(WO)
ENDIF

IF (QUEST2.EQ.'S') STOP 'DATA ENTRY HALTED'

1000 FORMAT (A1)

20000 FORMAT ('///', 'COMPLETE THE PROGRAM DOCUMENTATION SPREAD',
2 ' SHEET BEFORE CONTINUING.',
3 ' DO YOU WANT TO TERMINATE TO COMPLETE THIS FORM?',
4 ' Y OR N')

```

```

25000 FORMAT(' ',' ','ENTER THE DATE OR A PROGRAM IDENTIFIER - ',' ,
2          'UP TO 50 CHARACTERS IN SINGLE QUOTES')
26000 FORMAT(' ',' ','TO WHAT UNIT SHOULD THE OUTPUT FILES BE SENT?',' ,
2          '- INTEGER 10 TO 99')
30000 FORMAT(' ',' ','DATA CAN BE ENTERED INTERACTIVELY OR BY FILE.'/'
2          ' ','DO YOU WANT TO USE AN EXISTING DATA FILE? ',' ,
3          'Y OR N')
31000 FORMAT(' ',' ','ENTER THE NAME OF THE DATA FILE - ',' ,
2          'UP TO 10 CHARACTERS IN SINGLE QUOTES')
32000 FORMAT(' ',' ','THE NAME OF EACH SECTION CURRENTLY BEING READ ',' ,
2          'FROM THE FILE WILL APPEAR.'/' ','YOU WILL ',' ,
3          'THEN HAVE THE OPPORTUNITY TO LOOK AT AND TO CHANGE ',' ,
4          'THE DATA.')
39000 FORMAT(' ',' ','OPTION INFORMATION (MAY NOT BE CHANGED IF FILE ',' ,
2          'USED) ',' ,51(-)')
40000 FORMAT(' ',' ','YOU HAVE SEVERAL OPTIONS FOR ENTERING DATA: '/'
2          ' ','CHOOSE AN OPTION THAT IS COMPATIBLE WITH THE PRO',' ,
3          'GRAMS CAPABILITIES AND',' ','WITH THE DATA THAT IS ',' ,
4          'AVAILABLE TO YOU, THE USER.'/'
5          ' ','REFER TO THE PROGRAM DOCUMENTATION FOR THE ',' ,
6          'DATA THAT WILL BE REQUIRED.')
40010 FORMAT(' ',' ','DO YOU WANT TO SEE THE INFORMATION CONCERNING ',' ,
2          'THE OPTION INFORMATION? Y OR N')
40100 FORMAT(' ',' ','OPTION 1: ',' ,9(-) ',' ,YOU MUST CHOOSE AT WH ',' ,
2          'ICH STAGE TO BEGIN COMPUTATIONS.'/' ','FOR THOSE ',' ,
3          'STAGES BEFORE THE FIRST STAGE OF INTEREST, YOU ',' ,
4          'MUST ENTER',' ','THE TOTAL COST AND ',' ,
5          'ELAPSED TIME FOR THOSE STAGES NOT COMPUTED INDIVID ',' ,
6          'UALLY.'/' ','FOR THOSE STAGES AFTER THE FIRST STA ',' ,
7          'GE OF INTEREST, YOU HAVE A CHOICE;')
40150 FORMAT(' ',' ','FOR EACH INPUT IN EACH STAGE, YOU MAY CHOOSE TO ENT ',' ,
2          'ER THE QUANTITY',' ','OF THAT INPUT USED IN A STAGE ',' ,
3          'OR YOU MAY CHOOSE TO ALLOW',' ','THE PROGRAM TO COM ',' ,
4          'PUTE THIS QUANTITY')
40200 FORMAT(' ',' ','OPTION 2: ',' ,9(-)')
40300 FORMAT(' ',' ','OPTION 3: ',' ,9(-) ',' ,3X, 'PROGRAM COMPUTES ',' ,
2          'THE QUANTITY OF ALL INPUTS USED IN EVERY STAGE.')
40500 FORMAT(' ',' ','CONTINUE?')
41000 FORMAT(' ',' ','THE QUANTITY OF INPUT USED CAN BE COMPUTED BY ',' ,
2          'THE PROGRAM'/' FOR THE FOLLOWING STAGES:')
41100 FORMAT(' ',' ','INPUT ' ,12, ',' ,1X, A10, ':')
41200 FORMAT(' ',' ,2X, A10)
41300 FORMAT(' ',' ,2X, 'NO STAGES ARE COMPUTABLE FOR THIS INPUT')
42000 FORMAT(' ',' ','NOW, CHOOSE ONE OF THE ABOVE OPTIONS BY ',' ,
2          'NUMBER.')
43000 FORMAT(' ',' ','YOU HAVE CHOSEN OPTION 1. AT WHICH STAGE DO YOU ',' ,
2          '9954
0891
0892
0893
0894
0895
0896
0897
0898
0899
0900
0901
0902
0903
0904
0905
0906
0907
0908
0909
0910
0911
0912
0913
0914
0915
0916
0917
0918
0919
0920
0921
0922
0923
0924
0925
0926
0927
0928
0929
0930
0931
0932
0933
0934
0935
0936
0937
0938
0939
0940
0941
0942
0943
0944
0945
0946
0947
0948
0949
0950
0951
0952
0953
0954
0895
0896
0897
0898
0899
0900
0901
0902
0903
0904
0905
0906
0907
0908
0909
0910
0911
0912
0913
0914
0915
0916
0917
0918
0919
0920
0921
0922
0923
0924
0925
0926
0927
0928
0929
0930
0931
0932
0933
0934
0935
0936
0937
0938
0939
0940
0941
0942
0943
0944
0945
0946
0947
0948
0949
0950
0951
0952
0953
0954
0955
0956
0957
0958
0959
0960
0961
0962
0963
0964
0965
0966
0967
0968
0969
0970
0971
0972
0973
0974
0975
0976
0977
0978
0979
0980
0981
0982
0983
0984
0985
0986
0987
0988
0989
0990
0991
0992
0993
0994
0995
0996
0997
0998
0999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
'WANT TO BEGIN COMPUTATION?')
43100 FORMAT(' ',' ','FOR OPTION 1, FIRST STAGE MUST BE GREATER THAN 1, '
2          'AND NO GREATER THAN THE NUMBER OF STAGES. PLEASE ',' ,
3          'REINPUT')
44000 FORMAT(' ',' ','FOR EACH INPUT, YOU CAN CHOOSE TO ENTER THE ',' ,
2          'QUANTITY OF AN INPUT USED, ',' ,OR YOU CAN CHOOSE ',' ,
3          'TO LET THE PROGRAM COMPUTE THE QUANTITIES.'/'
4          ' ','(STAGES NOT LISTED MUST BE ENTERED AS QUANTITIES.)'/'
5          ' ,2X, 'ENTER C TO LET THE PROGRAM COMPUTE THE QUANTITY, '
6          'IN A STAGE OR',' ,2X, 'ENTER E TO ENTER YOUR OWN ',' ,
7          'QUANTITY.')
45000 FORMAT(' ',' ','DO YOU WANT TO SEE THE OPTION INFORMATION? ',' ,
2          'Y OR N')
45100 FORMAT(' ',' ','DO YOU WANT TO CHANGE ANY OF THIS INFORMATION? '/'
2          ' ','IF YES, THEN ALL INFORMATION MUST BE CHANGED')
45200 FORMAT(' ',' ','DO YOU WANT TO HALT THE PROGRAM AT THIS POINT?')
45300 FORMAT(' ',' ','DO YOU WANT TO PRINT THE OPTION INFORMATION?')
50000 FORMAT(' ',' ','YOU WILL NOW BE ASKED TO ENTER ALL DATA ',' ,
2          'NECESSARY TO RUN THIS PROGRAM'/'
3          ' ','EACH SECTION BEGINS A NEW SET OF INPUT LINE NUMBERS.'/'
4          ' ','FOLLOW THE PD. SPREAD SHEET FOR THE LINE NUMBERS ',' ,
5          'OF EACH SECTION')
50010 FORMAT(' ',' ','DATA MUST BE ENTERED FOR THE FOLLOWING SECTIONS:')
51000 FORMAT(' ',' ','SECTION ' ,11, ': ' ,2X, 'GENERAL DATA'/' ,25(-)')
50020 FORMAT(' SECTION ' ,11, ': ' ,2X, 'GENERAL DATA')
50030 FORMAT(' SECTION ' ,11, ': ' ,2X, A10, ' DATA')
50040 FORMAT(' SECTION ' ,11, ': ' ,2X, A10, ' DATA')
51100 FORMAT(' ',' ','SECTION ' ,11, ': ' ,2X, A, ' DATA'/' ,27(-)')
53000 FORMAT(' ',' ','SECTION ' ,12, ': ' ,2X, A10, 1X, 'DATA'/' ,27(-)')
59000 FORMAT(' ',' ****DATA ENTRY COMPLETED *****')
60000 FORMAT(' ',' ','END OF FILE HAS BEEN REACHED BEFORE DATA ENTRY ',' ,
2          'COMPLETED.')
3          ' ','PROGRAM CAN NOT CONTINUE WITHOUT CORRECTIONS'/(/)
61000 FORMAT(' ',' ','AN ERROR WAS DETECTED IN THE DATA FILE.'/'
2          ' ','PROGRAM CAN NOT CONTINUE WITHOUT CORRECTIONS')
62000 FORMAT(' ',' ','DATA ENTERED IN ANY SECTION CAN NOW BE CHECKED ',' ,
2          'FOR ERRORS. ONE AT A TIME.'/'
3          ' ','DO YOU WANT TO SEE ANY OF THE ENTERED DATA?')
62100 FORMAT(' ',' ','DATA CONCERNING OPTIONS? (CAN NOT BE CHANGED)')
62200 FORMAT(' ',' ','DATA FOR GENERAL SECTION ' ,12, '?')
62210 FORMAT(' ',' ','DATA FOR ' ,A, ' SECTION ' ,12, '?')
62300 FORMAT(' ',' ','DATA FOR SECTION ' ,12, ' CONCERNING ' ,A, '?')

```

```

1019 63000 FORMAT(' ',' ','YOU NOW HAVE SEVERAL OPTIONS CONCERNING ALL ' ,
1020 2 ' ','2X','THE DATA ENTERED. YOU CAN: ' /
1021 3 ' ','2X','ENTER S TO STOP AFTER AN OPTION TO PRINT THE DATA' /
1022 3 ' ','2X','ENTER R TO REDO DATA ENTIRELY INTERACTIVELY' )
1023
1024
1025 63100 FORMAT(' ','ENTER S OR R' )
1026
1027 63200 FORMAT(' ','2X','ENTER C TO CONTINUE EXECUTION WITH CORRECTIONS' /
1028 2 ' ','ENTER S, R, OR C' )
1029
1030 70000 FORMAT(' ',' YOU MAY NOW CORRECT ENTERED DATA ONE SECTION ' ,
1031 2 ' ','AT A TIME' /
1032 3 ' ','2X','ENTER 0 TO MAKE NO CORRECTIONS, OR' /
1033 4 ' ','2X','ENTER THE NUMBER OF THE SECTION IN WHICH YOU ' ,
1034 5 ' ','WISH TO MAKE CORRECTIONS' /
1035 6 ' ','ENTER AN INTEGER 0 TO',1X,I2)
1036
1037 80000 FORMAT(' ',' ','DO YOU WANT A PRINTED COPY OF ALL OF THE DATA ' ,
1038 2 ' ','ENTERED? Y OR N.' )
1039
1040 RETURN
1041 END
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
C SUBROUTINE ROUT(WRT)
C -----
C ORGNSM:
C -----
C SUBROUTINE DOES NOT VARY
C VARIABLES DO NOT CHANGE
C STRUCTURE : DISPLAYS OR PRINTS OUT DATA FROM SUBROUTINE READIN
C 1. PARAMETERS
C INCLUDE (IPAR1)
C INTEGER WRT
C 2. ARGUMENTS
C 3. COMMON BLOCKS
C INCLUDE (ICMAIN)
C INCLUDE (ICBLK1)
C INCLUDE (ICBLK0)
C INCLUDE (ICUNIT)
C 1. LOCAL VARIABLES AND ARGUMENTS
C INTEGER I, K, N, W
C NAME DESCRIPTION OF LOCAL VARIABLES AND ARGUMENTS
C I LOOP COUNTER FOR STAGES
C K INPUT LINE NUMBER
C N LOOP COUNTER FOR INPUTS
C W UNIT NUMBERS FOR WRITE
C WRT ARGUE: UNIT NUMBER FOR WRITE
C -----
C TRANSLATE ARGUMENTS TO LOCAL VARIABLES
W = WRT

```

```

1083 C DISPLAY OR PRINT ALL DATA ENTERED IN SUBROUTINE READIN
1084 C HEADING FOR OUTPUT
1085 WRITE(W,10000) NAME, DATE
1086 IF (NAMEF1.NE. 'NOFILE') WRITE(W,20000) NAMEF1
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
C OPTION INFORMATION
WRITE(W,30000) OPTION
IF (OPTION.EQ. 1) WRITE(W,31000)
IF (OPTION.EQ. 2) WRITE(W,32000)
IF (OPTION.EQ. 3) WRITE(W,33000)
IF (OPTION.EQ. 1) WRITE(W,34000) NAMEI(BEGINI)
IF (OPTION.EQ. 1 .OR. OPTION.EQ. 2) THEN
WRITE(W,40000) (NAMEI(I), I=1,NOOFI)
DO 100 N=1,NOOFN
WRITE(W,41000) NAMED(N), (OPNNEW(I,N), I=1,NOOFI)
100 CONTINUE
ENDIF
C THE NUMBER AND NAME OF EACH INPUT SUBROUTINE
K = 1
WRITE(W,45000)
WRITE(W,50000) K
K=K+1
WRITE(W,51000) K, NAME
DO 200 N=1,NOOFN
K = K+1
IF(ANYNOK(N) .LT. TOTALI) WRITE(W,52000) K, NAMED(N)
200 CONTINUE
10000 FORMAT('1',A10,' COST SIMULATION MODEL: INPUTS ENTERED' /
2 ' ','49(',-')' /
3 ' ','DATE: ',A50/' ','OPTION DATA ENTERED' /)
20000 FORMAT(' ','NAME OF FILE USED:',2X,A10)
30000 FORMAT(' ','OPTION CHOSEN:',1X,I1)
31000 FORMAT(' ','15X','COST UP TO FIRST STAGE OF INTEREST IS ENTERED' /
2 ' ','15X','USER CHOOSES INPUTS TO COMPUTE IN REMAINING STAGES' )
32000 FORMAT(' ','15X','USER CHOOSES WHICH INPUTS TO COMPUTE IN EACH ' ,
2 ' ','STAGE' )
33000 FORMAT(' ','15X','ALL INPUTS IN ALL STAGES ARE COMPUTED ' ,
2 ' ','BY THE PROGRAM' )
34000 FORMAT(' ','15X','THE FIRST STAGE OF INTEREST IS THE',1X,A10,
2 ' ','STAGE' )
40000 FORMAT(' ',' ','IF C, THE PROGRAM COMPUTES THE QUANTITY' /
2 ' ','IF E, THE USERS ENTERS THE QUANTITY' /
3 ' ','IF -, THE QUANTITY IS NEITHER ENTERED ' ,
4 ' ','NOR COMPUTED' / /
5 ' ','INPUT',T15,7(A10,1X))
41000 FORMAT(' ','A10,T15,7(A1,10X))
45000 FORMAT(' ',' ','THE NUMBER OF EACH SECTION FOR WHICH DATA ' ,

```

```

2          'MUST BE ENTERED.')
50000 FORMAT(' ', 2X, 'SECTION', 1X, I2, ':', GENERAL DATA')
51000 FORMAT(' ', 2X, 'SECTION', 1X, I2, ':', 'A, ' DATA')
52000 FORMAT(' ', 2X, 'SECTION', 1X, I2, ':', 1X, A10, 1X, 'DATA')

RETURN
END

SUBROUTINE READS(WRT, RED, HREAD, RUN2)
-----
C ORGNSM:
C-----
C SUBROUTINE DOES NOT VARY
C VARIABLES DO NOT CHANGE
C **NOTE: SOME INFO(K) MAY HAVE TO BE FORCED TO 0 DEPENDING ON HOW
C ORGANISM DEPENDENT SUBROUTINES ARE MANAGED
C
C STRUCTURE: READS IN GENERAL DATA TO RUN PROGRAM
C READS IN PRICE OF EACH INPUT OR ELSE CALLS PRICE, N, TO
C COMPUTE PRICE
C OPTION TO CHECK, CORRECT, OR PRINT DATA
C
C 1. PARAMETERS
C 2. ARGUMENTS
C 3. COMMON BLOCKS
C 4. LOCAL VARIABLES
C 5. INTEGER I, INFO, K, K2, KGOTO, KLAST, M, N, NOOFK, R, T, W
C LOGICAL CHECK, RUN1
C CHARACTER * 1 QUEST, QUEST2
C PARAMETER (NOOFK = 54)
C DIMENSION INFO(NOOFK)
C
C NAME DESCRIPTION OF LOCAL VARIABLES AND ARGUMENTS
C-----
C CHECK TO DETERMINE IF A LOOP HAS BEEN ENTERED
C HREAD ARGUE: ISOSTAT TO INDICATE PROBLEMS IN READING FILE
C I LOOP COUNTER FOR STAGES
C INFO INFORMATION LINE NUMBER
C K NUMBER OF THE INFO LINE
C K2 LOOP COUNTER TO FORCE REINPUT
C KGOTO LINE NUMBER TO CORRECT
C KLAST LAST INFO LINE NUMBER TO BE INPUT
C M LOOP COUNTER FOR MONTH
C N LOOP COUNTER FOR INPUT
C NOOFK NUMBER OF INPUT LINES
C QUEST GENERAL VARIABLE FOR USER QUESTION (IS WRITTEN OVER)

```

```

1147 C QUEST2 OPTION TO STOP, REDO, OR CONTINUE DATA ENTRY
1148 C R UNIT NUMBER FOR READ
1149 C RED ARGUE: FOR UNIT NUMBERS TO READ
1150 C RUN1 ARGUE: UNIT NO. TO READ FOR NEXT LEVEL SUBR.
1151 C RUN2 FIRST RUN THROUGH
1152 C RUN3 ARGUE: INDICATES IF ISNT FIRST RUN THROUGH
1153 C T ARGUE: INDICATES TO NEXT LEVEL SUBROUTINE NOT 1ST RUN THRU
1154 C T LOOP COUNTER FOR TYPE OF INPUT, I
1155 C W UNIT NUMBER FOR WRITE
1156 C WRT ARGUE: FOR UNIT NUMBERS TO WRITE
1157 C WRT2 ARGUE: UNIT NO. TO WRITE FOR NEXT LEVEL SUBR.
1158 C WW ARGUE: UNIT NO. TO WRITE AS OUTPUT FILE
1159 C
1160 C-----
1161 C
1162 C NOT FIRST RUN, IMMEDIATELY OFFER OPTION TO CHECK AND CHANGE DATA
1163 IF (RUN2) THEN
1164 DO 120 K=1,NOOFK
1165 INFO(K) = 1
1166 CONTINUE
1167
1168 WRT2 = WI
1169 RED2 = RI
1170 W = WI
1171 R = RI
1172 RUN1 = .FALSE.
1173 RUN3 = .TRUE.
1174
1175 WRITE(WI, 60000)
1176 READ(RI, 1000) QUEST
1177 IF (QUEST .EQ. 'Y') CALL ROUTS(WI)
1178
1179
1180
1181 WRITE(WI, 70000) NOOFK
1182 READ(RI, *) KGOTO
1183 IF (KGOTO .LT. 0 .OR. KGOTO .GT. NOOFK) THEN
1184 GOTO 100
1185 ELSEIF (KGOTO .GT. 0) THEN
1186 INFO (KGOTO) = 0
1187 ELSEIF (KGOTO .EQ. 0) THEN
1188 GOTO 850
1189 ENDIF
1190
1191 ELSE
1192 INITIALIZE INPUT LINES TO READ IN ALL DATA THE FIRST RUN
1193 C ASSIGN ARGUE. TO LOCAL UNIT NUMBERS TO BE CONSISTENT W/ OPTION
1194 W = WRT
1195 R = RED
1196 WRT2 = WRT
1197 RED2 = RED
1198 DO 110 K=1,NOOFK
1199 INFO(K) = 0
1200 CONTINUE
1201 RUN1 = .TRUE.
1202 RUN3 = .FALSE.
1203 ENDIF
1204
1205 C READ IN DATA NECESSARY TO RUN THE GENERAL PROGRAM
1206 400 HREAD = 0
1207
1208 C BEGINNING AND ENDING MONTHS OF PRODUCTION
1209 K = 1
1210 IF (INFO(K) .EQ. 0) THEN
1211 WRITE(W, 21000) K

```

```

1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274

```

```

1275 READ(R,*,ERR=900,IOSTAT=HREAD) BEGINM
1276 FORCE ALL INFO(K) TO 0 IN CASE OF REINPUT
1277 DO 405 K2 = 2,NOOFK
1278 INFO(K2) = 0
1279 CONTINUE
1280 RUN2 = .FALSE.
1281 ENDIF
1282 K = K+1
1283 IF (INFO(K).EQ. 0) THEN
1284 WRITE(W,22000) K
1285 READ(R,*,ERR=900,IOSTAT=HREAD) ENDM
1286 FORCE ALL REMAINING INFO(K) TO 0 IN CASE OF REINPUT
1287 DO 406 K2 = 3,NOOFK
1288 INFO(K2) = 0
1289 CONTINUE
1290 RUN2 = .FALSE.
1291 ENDIF
1292
1293 C DETERMINE THE LENGTH OF STAGES, I, AND THE NO. OF PERIODS, J, IN EACH
1294 K = K+1
1295 IF (INFO(K).EQ. 0) THEN
1296 WRITE(W,23000) K
1297 DO 410 I=BEGINI,NOOFI
1298 WRITE(W,23100) NAMEI(I)
1299 READ(R,*,ERR=900,IOSTAT=HREAD) DAYS(I)
1300 CONTINUE
1301 ENDIF
1302 K = K+1
1303 IF (INFO(K).EQ. 0) THEN
1304 WRITE(W,24000) K
1305 DO 420 I=BEGINI,NOOFI
1306 WRITE(W,25100) NAMEI(I)
1307 READ(R,*,ERR=900,IOSTAT=HREAD) JPERI(I)
1308 CONTINUE
1309 ENDIF
1310
1311 C DAYS AND COST UP TO THE END OF OPTION 1
1312 K = K+1
1313 IF (INFO(K).EQ. 0) THEN
1314 IF (OPTION.EQ. 1) THEN
1315 WRITE(W,25000) K, NAMEI(BEGINI)
1316 READ(R,*,ERR=900,IOSTAT=HREAD) DAYOP1
1317 ENDIF
1318 ENDIF
1319 K = K+1
1320 IF (INFO(K).EQ. 0) THEN
1321 IF (OPTION.EQ. 1) THEN
1322 WRITE(W,26000) K, NAMEI(BEGINI)
1323 READ(R,*,ERR=900,IOSTAT=HREAD) CSOP1
1324 ENDIF
1325 ENDIF
1326
1327 C LOOP THROUGH EACH INPUT, N, TO OFFER PRICE OPTION
1328 IF (RUN1) WRITE(W,50000)
1329 DO 500 N=1,NOOFN
1330 IF (RUN1) WRITE(W,50100) N,NAME(N)
1331 IF (THE OPTION EXISTS TO COMPUTE INPUT PRICE, OFFER IT
1332 K = K+1
1333 IF (INFO(K).EQ. 0) THEN
1334 IF (OPNSOK(N).EQ. 0) THEN
1335 WRITE(W,51000) K, NAME(N)
1336 READ(R,1000,ERR=900,IOSTAT=HREAD) QOPNS(N)
1337 ELSE
1338 QOPNS(N) = 'N'
1339
1475 WRITE(W,51100) K, NAME(N)
1476 ENDIF
1477 SET RELEVANT INFO LINES TO 0 TO FORCE REINPUTTING
1478 IF (QOPNS(N).EQ. 'Y') THEN
1479 INFO(K+5*NOOFN) = 0
1480 ELSEIF (QOPNS(N).EQ. 'N') THEN
1481 CONTINUE
1482 ENDIF
1483 ENDIF
1484 CONTINUE
1485
1486 C LOOP THRU EACH INPUT, N, TO ENTER INFO IF OPTION TO COMPUTE $ = NO
1487 IF (RUN1) WRITE(W,30000) NOOFN
1488 DO 600 N=1,NOOFN
1489 C NUMBER OF TYPES, T, OF EACH INPUT, N
1490 K = K+1
1491 IF (INFO(K).EQ. 0 .AND. OPTOK(N).EQ. 0 .AND. QOPNS(N).EQ. 'N') THEN1357
1492 WRITE(W,31000) K, NAME(N)
1493 READ(R,*,ERR=900,IOSTAT=HREAD) NOOFT(N)
1494 SET RELEVANT INFO LINES TO 0 TO FORCE REINPUT
1495 INFO(K+1) = 0
1496 INFO(K+3) = 0
1497 NEXT FORCING DOESNT WORK. MAKE COMMENT FOR NOW
1498 INFO(K+5*NOOFN) = 0
1499 ENDIF
1500
1501 C NAME OF EACH TYPE, T
1502 K = K+1
1503 IF (INFO(K).EQ. 0 .AND. OPTOK(N).EQ. 0 .AND. QOPNS(N).EQ. 'N') THEN1369
1504 IF (NOOFT(N).GT. 1) THEN
1505 WRITE(W,32000) K, T, NAME(N)
1506 READ(R,*,ERR=900,IOSTAT=HREAD) NAMEI(N,T)
1507 CONTINUE
1508 ELSE
1509 NAMEI(N,1) = 'ONE'
1510 ENDIF
1511 ENDIF
1512
1513 C DOES THE PRICE VARY BY MONTH?
1514 K = K+1
1515 IF (INFO(K).EQ. 0 .AND. QOPNS(N).EQ. 'N') THEN
1516 IF ((ENDM-BEGINM).GT. 0) THEN
1517 WRITE(W,53000) K, NAME(N)
1518 READ(R,1000,ERR=900,IOSTAT=HREAD) NSBYM(N)
1519 SET RELEVANT INFO LINES TO 0 TO FORCE REINPUT
1520 INFO(K+1) = 0
1521 ELSE
1522 NSBYM(N) = '-'
1523 ENDIF
1524 ENDIF
1525
1526 C PRICE OF EACH TYPE, T, IN ALL PRODUCTION MONTHS
1527 K = K+1
1528 IF (INFO(K).EQ. 0 .AND. QOPNS(N).EQ. 'N') THEN
1529 WRITE(W,54000) K, NAME(N), NAMEU(N)
1530 DO 620 T=1,NOOFT(N)
1531 IF (NOOFT(N).GT. 1) WRITE(W,55000) NAMEI(N,T)
1532 IF (NSBYM(N).EQ. 'Y') THEN
1533 DO 630 M=BEGINM,ENDM
1534 WRITE(W,56000) NAMEI(N,M)
1535 READ(R,*,ERR=900,IOSTAT=HREAD) PRICET(N,T,M)
1536
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402

```



```

1403 CONTINUE
1404 ELSE
1405 READ(R,*,ERR=900,IOSTAT=HREAD) PRICET(N,T,1)
1406 DO 640 M=BEGINM,ENDM
1407 PRICET(N,T,M) = PRICET(N,T,1)
1408 CONTINUE
1409 ENDIF
1410 CONTINUE
1411 CONTINUE
1412 CONTINUE
1413 CONTINUE
1414 CONTINUE
1415 C IF OPTION TO COMPUTE PRICE IS CHOSEN, CALL CORRESPONDING PRICE SUBR.
1416 DO 550 N=1,NOOFTN
1417 K = K+1
1418 IF (INFO(K).EQ.0 .AND. QOPNS(N).EQ.'Y') THEN
1419 WRITE(WI,52000) K, NAMEN(N)
1420 IF (N.EQ.1) CALL PRICE1(WRT2,RED2,HREAD,RUN2,RUN3)
1421 IF (N.EQ.2) CALL PRICE2(WRT2,RED2,HREAD,RUN2,RUN3)
1422 IF (N.EQ.3) CALL PRICE3(WRT2,RED2,HREAD,RUN2,RUN3)
1423 IF (N.EQ.4) CALL PRICE4(WRT2,RED2,HREAD,RUN2,RUN3)
1424 IF (N.EQ.5) CALL PRICE5(WRT2,RED2,HREAD,RUN2,RUN3)
1425 IF (N.EQ.6) CALL PRICE6(WRT2,RED2,HREAD,RUN2,RUN3)
1426 IF (HREAD.NE.0) GOTO 900
1427 ENDIF
1428 CONTINUE
1429
1430
1431 C INPUT QUANT. IF OPTION TO COMPUTE QUANT. OF N IN STAGE, I, ISNT CHOSEN
1432 IF(RUN1.AND.(OPTION.EQ.1.OR.OPTION.EQ.2)) WRITE(W,41000)
1433 DO 700 N=1,NOOFTN
1434 K = K+1
1435 DOES THE QUANTITY VARY BY MONTH?
1436 IF (INFO(K).EQ.0 .AND. ANYNOK(N).NE.0) THEN
1437 IF (OPTION.EQ.1.OR.OPTION.EQ.2) THEN
1438 IF ((ENDM-BEGINM).GT.0) THEN
1439 WRITE(W,40000) K, NAMEN(N)
1440 READ(R,1000) NOBYM(N)
1441 SET RELEVANT INFO LINES TO 0
1442 INFO(K+1) = 0
1443 ELSE
1444 NOBYM(N) = '-'
1445 ENDIF
1446 ENDIF
1447 CONTINUE
1448
1449 C QUANTITY OF EACH TYPE, I, OF EACH INPUT, N, USED IN EACH MONTH, M
1450 K = K+1
1451 IF (INFO(K).EQ.0) THEN
1452 IF (OPTION.EQ.1.OR.OPTION.EQ.2) THEN
1453 WRITE(W,41100) K, NAMEN(N), NAMENU(N)
1454 CHECK = .TRUE.
1455 DO 710 I=1,NOOFT(N)
1456 IF (NOOFT(N).GT.1 .AND. ANYNOK(N).NE.0) THEN
1457 WRITE(W,55000) NAMET(N,I)
1458 ENDIF
1459 DO 720 I=BEGINI,NOOFI
1460 IF (OPNEW(I,N).EQ.'E') THEN
1461 WRITE(W,41300) I, NAMEI(I)
1462 IF (NOBYM(N).EQ.'Y') THEN
1463 DO 730 M=BEGINM,ENDM
1464 WRITE(W,56000) NAMEM(M)
1465 READ(R,*,ERR=900,IOSTAT=HREAD)TQUANI(I,N,T,M)1466
630 CONTINUE
640 PRICET(N,T,M) = PRICET(N,T,1)
650 CONTINUE
660 CONTINUE
670 CONTINUE
680 CONTINUE
690 CONTINUE
700 CONTINUE
710 CONTINUE
720 CONTINUE
730 CHECK = .FALSE.
CONTINUE
ELSE
READ(R,*,ERR=900,IOSTAT=HREAD)TQUANI(I,N,T,1)
DO 740 M=BEGINM,ENDM
TQUANI(I,N,T,M) = TQUANI(I,N,T,1)
CONTINUE
CHECK = .FALSE.
ENDIF
ELSE
DO 750 M=BEGINM,ENDM
TQUANI(I,N,T,M) = -1.
CONTINUE
ENDIF
710 CONTINUE
IF (CHECK) WRITE(W,41500)
ENDIF
700 CONTINUE
KLAST = K
W = WI
R = RI
WRT2 = WI
RED2 = RI
C OPTION TO CHECK DATA
WRITE(WI,60000)
READ(RI,1000) QUEST
IF (QUEST.EQ.'Y') THEN
CALL ROUTS(WI)
ENDIF
C OPTION TO STOP OR REDO
IF (RUN1) THEN
WRITE(WI,64000)
READ(RI,1000) QUEST2
IF (QUEST2.EQ.'S') THEN
GOTO 850
ELSEIF (QUEST2.EQ.'R') THEN
GOTO 400
ELSE
RUN1 = .FALSE.
RUN3 = .TRUE.
ENDIF
ENDIF
C REASSIGN I TO INPUT LINES TO STOP FROM REENTERING DATA
DO 800 K=1,NOOFTK
INFO(K) = 1
CONTINUE
800 CONTINUE
C OPTION TO CORRECT DATA, LINE BY LINE, BY CHOOSING A LINE
810 WRITE(WI,70000) NOOFFK
INFO(K) = 1
CONTINUE
810 READ(RI,*) KGOTO
IF (KGOTO.LI.0 .OR. KGOTO.GT.NOOFFK) THEN
GOTO 810
ELSE IF (KGOTO.GT.0) THEN
INFO(KGOTO) = 0
GOTO 400
ENDIF
1467 1468
1469 1470
1471 1472
1473 1474
1475 1476
1477 1478
1479 1480
1481 1482
1483 1484
1485 1486
1487 1488
1489 1490
1491 1492
1493 1494
1495 1496
1497 1498
1499 1500
1501 1502
1503 1504
1505 1506
1507 1508
1509 1510
1511 1512
1513 1514
1515 1516
1517 1518
1519 1520
1521 1522
1523 1524
1525 1526
1527 1528
1529 1530

```

```

C OPTION TO PRINT THE DATA ENTERED
850 WRITE(WI,80000)
READ(RI,1000) QUEST
IF (QUEST.EQ.'Y') THEN
  CALL ROUTS(WO)
ENDIF

IF(QUEST2.EQ.'S') STOP 'PROGRAM STOP IN DATA ENTRY SECTION 1'

1000 FORMAT (A1)

21000 FORMAT(' ','I2',' THE NUMBER OF THE MONTH THAT THE PRODUCTION',
2 SEASON BEGINS',' ,T6,'ENTER AN INTEGER:')
1541
1542
22000 FORMAT(' ','I2',' THE NUMBER OF THE MONTH THAT THE PRODUCTION',
2 SEASON ENDS' /' ,T6,'ENTER AN INTEGER:')
1543
1544
23000 FORMAT(' ','I2',' THE NUMBER OF DAYS IN EACH STAGE.')
1545
1546
23100 FORMAT(' ','T6','IN THE',1X,A10,1X,'STAGE:')
1547
1548
24000 FORMAT(' ','I2',' THE NUMBER OF PERIODS IN EACH STAGE. ',
2 'ENTER AN INTEGER:')
1549
1550
25000 FORMAT(' ','I2',' FOR OPTION 1, THE NUMBER OF DAYS UP TO THE',
2 'FIRST STAGE OF INTEREST, ',A10)
1551
1552
26000 FORMAT(' ','I2',' FOR OPTION 1, WHAT IS THE TOTAL COST UP TO THE',
2 'FIRST STAGE OF INTEREST, ',A)
1553
1554
30000 FORMAT(' ','I2',' YOU WILL NOW BE ASKED TO ENTER DATA FOR EACH OF',
2 'THE',1X,' INPUTS')
1555
1556
31000 FORMAT(' ','I2',' THE NUMBER OF DISTINCT TYPES OF ',A/
2 'AX','ENTER AN INTEGER GREATER OR EQUAL TO 1')
1557
1558
32000 FORMAT(' ','I2',' THE NAME OF TYPE ',I2,' OF ',A/ ' ENTER UP TO ',
2 '10 CHARACTERS, IN SINGLE QUOTES')
1559
1560
40000 FORMAT(' ','I2',' DOES THE QUANTITY OF ',A,' VARY BY MONTH?')
1561
1562
41000 FORMAT(' ','I2',' QUANTITY INFORMATION: (GENERAL SECTION) /',
2 'FOR OPTIONS 1 AND 2, ENTER THE QUANTITY OF ',
3 'EACH INPUT USED EACH STAGE')
1563
1564
41100 FORMAT(' ','I2',' ,1X,' QUANTITY OF ',A,1X,A6,':')
1565
1566
41300 FORMAT(' ','5X,'IN STAGE',I2,' ',1X,A)
1567
1568
41500 FORMAT(' ','4X,'NO STAGES REQUIRE QUANTITIES OF THIS INPUT')
1569
1570
50000 FORMAT(' ','I2',' ,FOR SOME INPUTS YOU CAN CHOOSE TO ',
2 'LET THE PROGRAM COMPUTE INPUT PRICE OR /',
3 ' ', ' YOU CAN CHOOSE TO INPUT THE PRICE/UNIT YOURSELF.')
1571
1572
50100 FORMAT(' ','I2',' ,PRICE OF',1X,A,':')
1573
1574
51000 FORMAT(' ','I2',' DO YOU WANT THE PROGRAM TO COMPUTE THE PRICE',
2 ' ', ' OF ',A,'?')
1575
1576
51100 FORMAT(' ','I2',' PRICE CAN NOT BE COMPUTED BY THE PROGRAM FOR ',A)
1577
1578
52000 FORMAT(' ','I2',' DATA FOR COMPUTING PRICE OF ',A,
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658

```

```

1659 C K INPUT LINE NUMBER
1660 C M LOOP COUNTER FOR MONTHS
1661 C N LOOP COUNTER FOR INPUTS
1662 C T LOOP COUNTER FOR TYPES OF INPUT
1663 C W UNIT NUMBER FOR WRITE
1664 C WRT ARGUMENT FOR UNIT NUMBER TO WRITE
1665 -----
1666 C
1667 C
1668 C TRANSLATE ARGUMENTS TO LOCAL VARIABLES
1669 W = WRT
1670 C
1671 C GENERAL INFORMATION
1672 WRITE(W,10000) NAME, DATE
1673 K = 1
1674 WRITE(W,11000) K, NAMEM(BEGINM), BEGINM
1675 K = K+1
1676 WRITE(W,12000) K, NAMEM(ENDM), ENDM
1677 WRITE(W,12500) (I, I=BEGINM,NOOFT)
1678 K = K+1
1679 WRITE(W,13000) K, (DAYSI(I), I=BEGINM,NOOFT)
1680 K = K+1
1681 WRITE(W,14000) K, (JPERI(I), I=BEGINM,NOOFT)
1682 K = K+1
1683 IF (OPTION.EQ.1) WRITE(W,15000) K, BEGINI, DAYOPI
1684 K = K+1
1685 IF (OPTION.EQ.1) WRITE(W,16000) K, BEGINI, CSOPI
1686 K = K+1
1687 C
1688 C CALCULATE OR INPUT PRICE?
1689 WRITE(W,40000)
1690 DO 500 N=1,NOOFT
1691 WRITE(W,41000) NAMEM(N)
1692 K = K+1
1693 WRITE(W,42000) K, QOPNS(N)
1694 CONTINUE
1695 500
1696 C
1697 C LOOP THRU EACH N TO WRITE INFO WHEN PRICE OPTION WAS NOT CHOSEN
1698 DO 600 N=1,NOOFT
1699 WRITE(W,41000) NAMEM(N)
1700 C
1701 C NUMBER AND NAME OF TYPES, I
1702 K = K+1
1703 IF (QOPNS(N).EQ.'N') THEN
1704 IF (OPTOK(N).EQ.0) WRITE(W,31000) K, NOOFT(N)
1705 ENDF
1706 K = K+1
1707 IF (QOPNS(N).EQ.'N') THEN
1708 IF (OPTOK(N).EQ.0 .AND. NOOFT(N).GT.1)
1709 WRITE(W,32000) K, (T, NAMEM(N,T), T=1,NOOFT(N))
1710 ENDF
1711 2
1712 C
1713 C DOES PRICE VARY BY MONTH?
1714 K = K+1
1715 IF (QOPNS(N).EQ.'N' .AND. (ENDM-BEGINM).GT.0) THEN
1716 WRITE(W,44000) K, NSBYM(N)
1717 ENDF
1718 C
1719 C PRICE OF EACH TYPE, T, IN EACH INPUT MONTH
1720 K = K+1
1721 IF (QOPNS(N).EQ.'N') THEN
1722 WRITE(W,45000)K,NAMEM(N),NAMEM(N),NAMEM(M),M=BEGINM,ENDM)1721
1723 ELSE
1724
1725 WRITE(W,45100)K,NAMEM(N),NAMEM(N),NAMEM(M),M=BEGINM,ENDM)1723
1726 ENDF
1727 DO 610 T=1,NOOFT(N)
1728 IF (NOOFT(N).GT.1) WRITE(W,46000) NAMEM(N,T)
1729 WRITE(W,46500) (PRICET(N,T,M), M=BEGINM,ENDM)
1730 CONTINUE
1731 610
1732 C
1733 C 600 CONTINUE
1734
1735 C IF OPTION TO COMPUTE PRICE WAS CHOSEN, WRITE THE NO. OF EACH $ SUBR.
1736 DO 550 N=1,NOOFT
1737 K = K+1
1738 IF (QOPNS(N).EQ.'Y') WRITE(W,43000) K, NAMEM(N)
1739 CONTINUE
1740 550
1741 C
1742 IF (OPTION.EQ.1 .OR. OPTION.EQ.2) WRITE(W,30000)
1743 DO 700 N=1,NOOFT
1744 WRITE(W,34010) NAMEM(N), NAMEM(N)
1745 CONTINUE
1746 700
1747 C
1748 C DOES THE QUANTITY VARY BY MONTH?
1749 K = K+1
1750 IF ((OPTION.EQ.1 .OR. OPTION.EQ.2) .AND. (ENDM-BEGINM).GT.0)THEN1745
1751 WRITE (W,34100) K, NQBYM(N)
1752 ENDF
1753 1747
1754 C WHAT IS THE QUANTITY OF EACH INPUT USED?
1755 K = K+1
1756 IF (OPTION.EQ.1 .OR. OPTION.EQ.2) THEN
1757 WRITE(W,34000) K, (I, I=BEGINM,NOOFT)
1758 DO 710 T=1,NOOFT(N)
1759 IF (NOOFT(N).GT.1) WRITE(W,35000) NAMEM(N,T)
1760 DO 720 M=BEGINM,ENDM
1761 WRITE(W,36000)NAMEM(M),(TQUANI(I,N,T,M),I=BEGINM,NOOFT)1756
1762 CONTINUE
1763 720
1764 CONTINUE
1765 710
1766 ENDF
1767 700
1768 C
1769 C 10000 FORMAT('1',A,' COST SIMULATION MODEL: INPUT ENTERED (CONT.)')/
1770 2 ' ',A/','GENERAL DATA '/')
1771
1772 C 11000 FORMAT(' ','I2','I2',' MONTH THAT PRODUCTION SEASON BEGINS:',T43,
1773 2 A10,(' ','I2,')')
1774
1775 C 12000 FORMAT(' ','I2','I2',' MONTH THAT PRODUCTION SEASON ENDS:',T43,
1776 2 A10,(' ','I2,')')
1777
1778 C 12500 FORMAT(' '//','T6,7(I2,' STAGE',2X)'/','T6,7(8(' '-'),2X))
1779
1780 C 13000 FORMAT(' '//','I2','I2',' NUMBER OF DAYS IN EACH STAGE: '/
1781 2 ' ','T5,7(F9.2,IX))
1782
1783 C 14000 FORMAT(' '//','I2','I2',' NUMBER OF PERIODS IN EACH STAGE: '/
1784 2 ' ','3X,7(I8,2X))
1785
1786 C 15000 FORMAT(' '//','I2','I2',' FOR OPTION 1, THE NUMBER OF DAYS ',
1787 2 'UP TO STAGE ',I1,','T54,F10.2)
1788
1789 C 16000 FORMAT(' ','I2','I2',' FOR OPTION 1, THE COST UP TO STAGE ',
1790 2 I1,','T54,F10.2//)
1791
1792 C 31000 FORMAT(' ','I2','I2',' NUMBER OF TYPES : ',I2)

```

```

32000 FORMAT(' ,I2,' , TYPE NAMES:' / ' , 3X,4(5(I2,' =',A10,1X) / ' ')
33000 FORMAT(' / / ' , , QUANTITY USED OF EACH INPUT IN EACH STAGE : /
2      , , , , (WHERE = -1, PROGRAM , , ,
3      , , , , WILL COMPUTE THE QUANTITY LATER IN EXECUTION')
34010 FORMAT(' / ' , A,1X,A, , ':')
34100 FORMAT(' ,I2,' , QUANTITY VARIES BY MONTH?:', 2X,A)
34000 FORMAT(' ,I2,' , QUANTITY USED:' / ' , T21,7(2X,I2,' STAGE',6X))
35000 FORMAT(' ,4X,' OF',1X,A)
36000 FORMAT(' ,5X,' IN',1X,A10,' :', T20,7(G14.7,1X))
40000 FORMAT(' , INFORMATION PERTAINING TO PRICE:' / ' , 31(' ')
41000 FORMAT(' / ' , INPUT:', 2X,A)
42000 FORMAT(' ,I2,' , PRICE COMPUTED BY PROGRAM?:', 1X,A)
43000 FORMAT(' / ' , I2,' , SECTION TO COMPUTE', 1X,A,1X,' PRICE ',
2      , (SEE SEPARATE INPUT LIST)')
44000 FORMAT(' ,I2,' , PRICE VARIES BY MONTH?:', 2X,A)
45000 FORMAT(' ,I2,' , PRICE OF ' , A , ' PER ' , A , ': ' / ' , T18,6(A,5X) /
2      , , , T18,6(A,5X))
45100 FORMAT(' ,I2,' , PRICE OF ' , A , ' PER ' , A , ' (COMPUTED BY THE ' ,
2      , ' PROGRAM):' / ' , T18,6(A10,5X) / ' , , T18,6(A10,5X))
46000 FORMAT(' , T6,A10,' :')
46500 FORMAT(' , T16,6(G14.7,1X) / ' , T16,6(G14.7,1X))

RETURN
END

C
SUBROUTINE READO(WRT,RED,HREAD,RUN2)
-----
C ORGNSM:
C -----
C SUBROUTINE VARIES WITH ORGANISM
C VARIABLES CHANGE WITH ORGANISM
C VARIABLES THAT MUST BE RETURNED:
C THE TYPE OF ORGANISM : NAME(O)
C THE NO. OF ORGANISMS AT BEGINNING : ONOBEG
C THE TYPE OF SYSTEM USED : NAMES
C CURRENT ORGANISM: BIVALVES

C STRUCTURE: READS IN THE ABOVE MENTIONED DATA
C READS IN OTHER DATA SPECIFIC TO ORGANISM
C OPTION TO CHECK, CORRECT, OR PRINT DATA

```

```

1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850

```

```

1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915

```

```

C 1. PARAMETERS
C 2. ARGUMENTS
C 3. COMMON BLOCKS
C 4. LOCAL VARIABLES
C O LOOP COUNTER FOR ORGANISM
C HREAD ARGUE: ISOSTAT TO INDICATE PROBLEMS IN READING FILE
C INFO INFORMATION LINE NUMBER
C K NUMBER OF THE INFO LINE
C KGOTO LINE NUMBER TO CORRECT
C LAST INFO LINE NUMBER TO BE INPUT
C NOOFK NUMBER OF INPUT LINES
C QUEST GENERAL VARIABLE FOR USER QUESTION (IS WRITTEN OVER)
C QUEST2 OPTION TO STOP, REDO, OR CONTINUE DATA ENTRY
C R UNIT NUMBER FOR READ
C RED ARGUE: FOR UNIT NUMBERS TO READ
C RED2 ARGUE: UNIT NO. TO READ FOR NEXT LEVEL SUBR.
C RUN1 FIRST RUN THROUGH
C RUN2 ARGUE: INDICATES IF ISNT FIRST RUN THROUGH
C W UNIT NUMBER FOR WRITE
C WRT ARGUE: FOR UNIT NUMBERS TO WRITE
C WRT2 ARGUE: FOR UNIT NUMBERS TO WRITE
C WW ARGUE: UNIT NO. TO WRITE TO OUTPUT FILE
C NOT FIRST RUN, IMMEDIATELY OFFER OPTION TO CHECK AND CHANGE DATA
IF (RUN2) THEN
DO 120 K=1,NOOFK
INFO(K) = 1
CONTINUE
WRT2 = WI
RED2 = RI
W = WI
R = RI
WRITE(WI,60000) NAME
RUN1 = .FALSE.
READ(RI,1000) QUEST
IF (QUEST.EQ.'Y') CALL ROUTO(WI)
WRITE(WI,70000) NAME, NOOFK
READ(RI,*) KGOTO
IF (KGOTO.LT.0 .OR. KGOTO.GT. NOOFK) THEN
GOTO 100
ELSEIF (KGOTO.GT. 0) THEN
INFO (KGOTO) = 0
ELSEIF (KGOTO.EQ. 0) THEN
GOTO 850

```

```

1915      ENDIF
1916
1917      ELSE
1918      C INITIALIZE INPUT LINES TO READ IN ALL DATA THE FIRST RUN
1919      C ASSIGN ARGUE. TO LOCAL UNIT NUMBERS TO BE CONSISTENT W/ OPTION
1920      W = WRT
1921      R = RED
1922      WRT2 = WRT
1923      RED2 = RED
1924      DO 110 K=1,NOOFK
1925          INFO(K) = 0
1926      CONTINUE
1927      RUN1 = .TRUE.
1928      ENDIF
1929
1930      C READ IN DATA SPECIFIC TO ORGANISM
1931      400 HREAD = 0
1932
1933      C THE TYPE OF ORGANISM
1934      K = 1
1935      IF (INFO(K) .EQ. 0) THEN
1936          WRITE(W,20000) K, NAME
1937          DO 405 O=1,NOOFO
1938              WRITE(W,20100) O, NAMEO(O)
1939          CONTINUE
1940          READ(R,*,ERR=900,IOSTAT=HREAD) OTYPE
1941          ENDIF
1942
1943      C THE TYPE OF SYSTEM USED
1944      K = K+1
1945      IF (INFO(K) .EQ. 0) THEN
1946          WRITE(W,20015) K
1947          READ (R,*,ERR=900,IOSTAT=HREAD) NAMES
1948          ENDIF
1949
1950      C INFORMATION CONCERNING CONDITIONING STAGE
1951      K = K+1
1952      IF (INFO(K) .EQ. 0) THEN
1953          WRITE(W,20020) K, NAMEI(1)
1954          READ (R,*,ERR=900,IOSTAT=HREAD) DAYSII
1955          ENDIF
1956      K = K+1
1957      IF (INFO(K) .EQ. 0) THEN
1958          WRITE(W,20025) K, NAMEO(OTYPE), NAMEI(1)
1959          READ (R,*,ERR=900,IOSTAT=HREAD) NOOII
1960          ENDIF
1961
1962      C NUMBER OF ORGANISMS THAT ARE BEGUN
1963      K = K+1
1964      IF (INFO(K) .EQ. 0) THEN
1965          WRITE(W,30000) K, NAMEO(OTYPE), NAMEI(2)
1966          READ(R,*,ERR=900,IOSTAT=HREAD) ONOBEG
1967          ENDIF
1968
1969      C CHOOSE PARAMETERS FOR AGE NUMBER ALGORITHM
1970      IF (RUN1) WRITE(W,40000) EQNO(1)
1971      K = K+1
1972      IF (INFO(K) .EQ. 0) THEN
1973          WRITE (W,41000) K, EQNOAF
1974          READ (R,*,ERR=900,IOSTAT=HREAD) EQNOA
1975          ENDIF
1976
1977      C CHOOSE PARAMETERS FOR SIZE/AGE ALGORITHM
1978
1979      IF (RUN1) WRITE(W,50000) EQSZ(1)
1980      K = K+1
1981      IF (INFO(K) .EQ. 0) THEN
1982          WRITE(W,51000) K, EQSZAF
1983          READ(R,*,ERR=900,IOSTAT=HREAD) EQSZA
1984          ENDIF
1985      K = K+1
1986      IF (INFO(K) .EQ. 0) THEN
1987          WRITE(W,52000) K, EQSZBF
1988          READ(R,*,ERR=900,IOSTAT=HREAD) EQSZB
1989          ENDIF
1990
1991      KLAST = K
1992      WRT2 = WI
1993      RED2 = RI
1994      W = WI
1995      R = RI
1996
1997      C OPTION TO CHECK DATA
1998      WRITE(WI,60000) NAME
1999      READ(RI,1000) QUEST
2000      IF (QUEST .EQ. 'Y') THEN
2001          CALL ROUTO(WI)
2002          ENDIF
2003
2004      C OPTION TO STOP OR REDO
2005      IF (RUN1) THEN
2006          WRITE(WI,64000) NAME
2007          READ(RI,1000) QUEST2
2008          IF (QUEST2 .EQ. 'S') THEN
2009              GOTO 850
2010          ELSEIF (QUEST2 .EQ. 'R') THEN
2011              GOTO 400
2012          ENDIF
2013      RUN1 = .FALSE.
2014      ENDIF
2015
2016      C REASSIGN 1 TO INPUT LINES TO STOP FROM REENTERING DATA
2017      DO 800 K=1,NOOFK
2018          INFO(K) = 1
2019      CONTINUE
2020
2021      C OPTION TO CORRECT DATA, LINE BY LINE, BY CHOOSING A LINE
2022      810 WRITE(WI,70000) NAME, NOOFK
2023      READ(RI,*) KGOTO
2024      IF (KGOTO .LT. 0 .OR. KGOTO .GT. NOOFK) THEN
2025          GOTO 810
2026      ELSE IF (KGOTO .GT. 0) THEN
2027          INFO(KGOTO) = 0
2028          GOTO 400
2029      ENDIF
2030
2031      C OPTION TO PRINT THE DATA ENTERED
2032      850 WRITE(WI,80000) NAME
2033      READ(RI,1000) QUEST
2034      IF (QUEST .EQ. 'Y') THEN
2035          CALL ROUTO(WO)
2036          ENDIF
2037
2038      IF (QUEST2 .EQ. 'S') STOP 'PROGRAM STOP IN DATA ENTRY SECTION 1'
2039
2040
2041
2042

```

```

1000 FORMAT (A1)
2000 FORMAT(' ',I2,'. CHOOSE ONE OF THE FOLLOWING ',A,' TYPES:')
20015 FORMAT(' ',I2,'. NAME OF THE SYSTEM USED.'/',' ',4X,'UP TO 50 ',
2 'CHARACTERS IN QUOTES')
20100 FORMAT(' ',T6,I1,' = ',A)
20020 FORMAT(' ',I2,'. NUMBER OF DAYS IN THE ',A,' STAGE.')
20025 FORMAT(' ',I2,'. NUMBER OF ',A,' IN THE ',A,' STAGE')
30000 FORMAT(' ',I2,'. NUMBER OF ',A,' THAT A BATCH IS BEGUN WITH IN ',
2 'THE ',A,' STAGE')
40000 FORMAT(' '// THE FOLLOWING EQUATION IS USED TO EQUATE AGE IN ',
2 'WEEKS TO NUMBER SURVIVING:.'/',' ',4X,A)
41000 FORMAT(' ',I2,'. CHOOSE SURVIVAL PARAMETER A (SUGGESTED: ',
2 G14.7,')')
50000 FORMAT(' '// THE FOLLOWING EQUATION IS USED TO EQUATE AGE IN ',
2 'WEEKS TO SIZE IN MM:.'/',' ',4X,A)
51000 FORMAT(' ',I2,'. CHOOSE SIZE PARAMETER A (SUGGESTED: ',
2 G14.7,')')
52000 FORMAT(' ',I2,'. CHOOSE SIZE PARAMETER B (SUGGESTED: ',
2 G14.7,')')
60000 FORMAT(' '// 'DO YOU WANT TO CHECK DATA FOR ',A/)
64000 FORMAT(' ',ENTER S TO STOP EXECUTION, AFTER AN OPTION TO',
2 ' PRINT DATA FOR ',A/
3 ' ',ENTER R TO REDO ALL DATA IN THIS SECTION ',
3 'INTERACTIVELY'/
4 ' ',ENTER C TO CONTINUE EXECUTION WITH CORRECTIONS ',
5 'LINE BY LINE'/
6 ' ',ENTER S, R, OR C'/)
70000 FORMAT(' ',YOU MAY CORRECT DATA LINE AT A TIME FOR ',A/
2 ' ',2X,'ENTER 0 TO MAKE NO CORRECTIONS OR'/
3 ' ',2X,'ENTER THE NUMBER OF THE LINE YOU WANT TO CORRECT.'/2086
4 ' ',ENTER AN INTEGER FROM 0 TO ',I1,I2)
80000 FORMAT(' '// DO YOU WANT A PRINTED COPY OF DATA ENTERED FOR ',A,
2 ' SECTION?')
900 RETURN
END

C SUBROUTINE ROUTO(WRT)
C -----
C ORGNISM:
C -----

```

```

2107 C SUBROUTINE VARIES WITH ORGANISM
2108 C VARIABLES CHANGE WITH ORGANISM, BUT SOME MUST BE RETURNED
2109 C VARIABLES REQUIRED ARE THOSE REQUIRED IN READO
2110 C CURRENT ORGANISM: BIVALVES
2111
2112 C STRUCTURE: DISPLAYS OR PRINTS DATA FROM SUBROUTINE READO
2113
2114 C 1. PARAMETERS
2115 C INCLUDE (IPAR1)
2116 C 2. ARGUMENTS
2117 C INTEGER WRT
2118 C 3. COMMON BLOCKS
2119 C INCLUDE (ICMAIN)
2120 C INCLUDE (ICBLK1)
2121 C INCLUDE (ICBLKO)
2122 C INCLUDE (ICUNIT)
2123 C INCLUDE (ICORG)
2124 C 4. LOCAL VARIABLES
2125 C INTEGER K, W
2126
2127 C NAME DESCRIPTION OF LOCAL VARIABLES AND ARGUMENTS
2128 C -----
2129 C K INPUT LINE NUMBER
2130 C W UNIT NUMBER FOR WRITE
2131 C WRT ARGUE: UNIT NUMBER FOR WRITE
2132 C -----
2133
2134 C TRANSLATE ARGUMENTS TO LOCAL VARIABLES
2135 W = WRT
2136
2137 C GENERAL INFORMATION
2138 WRITE(W,10000) NAME, DATE, NAME
2139 K = 1
2140 WRITE(W,10500) K, NAME, OTYPE, NAMEO(OTYPE)
2141 K = K+1
2142 WRITE(W,10515) K, NAMES
2143 K = K+1
2144 WRITE(W,10520) K, NAMEI(1), DAYSII
2145 K = K+1
2146 WRITE(W,10525) K, NAMEI(1), NOOI1
2147 K = K+1
2148 WRITE(W,30000) K, NAMEO(OTYPE), ONOBEG
2149
2150 C PARAMETERS FOR SIZE/NUMBER EQUATION
2151 WRITE(W,40000) EQNO(1)
2152 K = K+1
2153 WRITE(W,41000) K, EQNOA
2154
2155 C PARAMETERS FOR AGE/SIZE EQUATION
2156 WRITE(W,50000) EQSZ(1)
2157 K = K+1
2158 WRITE(W,51000) K, EQSZA
2159 K = K+1
2160 WRITE(W,52000) K, EQSZB
2161
2162 10000 FORMAT('1',A,' COST SIMULATION MODEL: INPUT ENTERED (CONT.)')
2163 2 ' ',A/,' ',A,' DATA '/')
2164
2165 10500 FORMAT(' ',I2,'. TYPE OF ',A,T46,I2,' = ',A10)
2166
2167 10515 FORMAT(' '// ',I2,'. NAME OF SYSTEM',T46,A)
2168
2169
2170

```

```

10520 FORMAT(' ','I2',' NUMBER DAYS IN STAGE ',A,T46,G10.5)
2171
2172
10525 FORMAT(' ','I2',' NUMBER BIVALVES IN STAGE ',A,T46,G10.5)
2173
2174
30000 FORMAT(' ','I2',' NUMBER OF ',A,' BATCH BEGUN WITH:',T46,G12.7)2175
2176
40000 FORMAT(' ','I2',' EQUATION TO DETERMINE SURVIVAL:',' ',4X,A)
2177
41000 FORMAT(' ','I2',' PARAMETER A = ',G14.7)
2178
50000 FORMAT(' ','I2',' EQUATION TO EQUATE AGE TO SIZE:',' ',4X,A)
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234

SUBROUTINE READN1(WRT,RED,HREAD,RUN2)
RETURN
END

SUBROUTINE READN2(WRT,RED,HREAD,RUN2)
RETURN
END

SUBROUTINE READN3(WRT,RED,HREAD,RUN2)
RETURN
END

SUBROUTINE READN4(WRT,RED,HREAD,RUN2)
RETURN
END

SUBROUTINE READN5(WRT,RED,HREAD,RUN2)
-----
C
C ORGNSM:
C -----
C SUBROUTINE VARIES WITH ORGANISM
C VARIABLES CHANGE WITH ORGANISM
C VARIABLES NOT RETURNED TO ORGNSM
C VARIABLES NECESSARY ONLY FOR QUANT5, IF OPTION TO COMPUTE QUANTITY=Y
C CURRENT ORGANISM: BIVALVES

```

```

C STRUCTURE: READS IN DATA NECESSARY TO COMPUTE QUANTITY OF INPUT 5
C OPTIONS TO CHECK, PRINT, CORRECT DATA
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298

C 1. PARAMETERS
C INCLUDE (IPAR1)
2239
C 2. ARGUMENTS
C INTEGER HREAD, RED, RED2, WRT, WRT2
2240
C LOGICAL RUN2
2241
C 3. COMMON BLOCKS
C INCLUDE (ICMAIN)
2242
C INCLUDE (ICBLK1)
2243
C INCLUDE (ICBLKO)
2244
C INCLUDE (ICUNIT)
2245
C INCLUDE (ICORG)
2246
C 4. LOCAL VARIABLES
C INTEGER EQ, I, INFO, K, KGOTO, KLAST, M, NOOFK, R, T, W
2247
C LOGICAL RUN1
2248
C CHARACTER * 1 QUEST, QUEST2
2249
C PARAMETER (NOOFK=9)
2250
C DIMENSION INFO(NOOFK)
2251
C NAME DESCRIPTION OF LOCAL VARIABLES AND ARGUMENTS
2252
C EQ LOOP COUNTER FOR EQUATIONS
2253
C HREAD ARGUE: IOSTAT TO INDICATE PROBLEMS READING FILE
2254
C I LOOP COUNTER FOR INPUTS
2255
C INFO INDICATOR OF INPUT LINE
2256
C K INPUT LINE NUMBER
2257
C KGOTO INPUT LINE NO. TO GOTO TO CORRECT
2258
C KLAST LAST INPUT LINE INPUT
2259
C M LOOP COUNTER FOR MONTHS
2260
C NOOFK NUMBER OF INPUT LINES
2261
C QUEST GENERAL INTERACTIVE USER QUESTION (IS OVERWRITTEN)
2262
C QUEST2 OPTION TO STOP, READ, OR CONTINUE
2263
C R UNIT NUMBER FOR READ
2264
C RED ARGUE: UNIT NUMBER FOR READ
2265
C RED2 ARGUE: UNIT NUMBER FOR READ FOR NEXT LEVEL SUBR.
2266
C RUN1 IS IT THE FIRST RUN?
2267
C RUN2 ARGUE: IS IT GREATER THAN THE FIRST RUN
2268
C T LOOP COUNTER FOR TYPES
2269
C W UNIT NUMBER FOR WRITE
2270
C WRT ARGUE: UNIT NUMBER TO WRITE TO NEXT LEVEL SUBROUTINE
2271
C WRT2 ARGUE: UNIT NUMBER TO WRITE TO NEXT LEVEL SUBROUTINE
2272
C -----
C IF NOT 1ST RUN, IMMEDIATELY OFFER TO DISPLAY AND CHANGE DATA
2273
C IF (RUN2) THEN
2274
C DO 120 K=1,NOOFK
2275
C INFO(K) = 1
2276
C CONTINUE
2277
C WRT2 = WI
2278
C RED2 = RI
2279
C W = WI
2280
C R = RI
2281
C RUN1 = .FALSE.
2282
C WRITE(WI,60000) NAMEN(5)
2283
C READ(RI,1000) QUEST
2284
C IF (QUEST .EQ. 'Y') CALL ROUTN5(WI)
2285
C 100 WRITE(WI,70000) NAMEN(5), NOOFK
2286
C READ(RI,*) KGOTO
2287
C IF (KGOTO .LT. 0 .OR. KGOTO .GT. NOOFK) THEN
2288
C GOTO 100
2289

```

```

2299 ELSEIF (KGOTO .GT. 0) THEN
2300 INFO(KGOTO) = 0
2301 ELSEIF (KGOTO .EQ. 0) THEN
2302 GOTO 850
2303 ENDFIF
2304
2305 ELSE
2306 INITIALIZE VARIABLES TO 0 TO READ IN DATA FIRST TIME
2307 C ASSIGN ARGUE UNIT NO. TO LOCAL UNIT NUMBERS
2308 DO 110 K=1,NOOFK
2309 INFO(K) = 0
2310 CONTINUE
2311 RUN1 = .TRUE.
2312 W = WRT
2313 R = RED
2314 WRT2 = WRT
2315 RED2 = RED
2316 ENDFIF
2317
2318 C READ IN ALL DATA
2319 400 HREAD = 0
2320
2321 C CHOOSE ALGORITHM AND PARAMETERS TO COMPUTE QUANTITY OF ALGAE
2322 K = 1
2323 IF (INFO(K) .EQ. 0) THEN
2324 WRITE(W,10000) K
2325 WRITE(W,10100) (EQ, EQQ5(EQ), EQ=1,2)
2326 READ(R,*,ERR=900,IOSTAT=HREAD) EQQ5
2327 INFO(K+1) = 0
2328 INFO(K+2) = 0
2329 ENDFIF
2330 K = K+1
2331 IF (INFO(K) .EQ. 0) THEN
2332 WRITE(W,11000) K, EQQ5AF(EQQQ5)
2333 READ(R,*,ERR=900,IOSTAT=HREAD) EQQ5A(EQQQ5)
2334 ENDFIF
2335 K = K+1
2336 IF (INFO(K) .EQ. 0) THEN
2337 WRITE(W,11100) K, EQQ5BF(EQQQ5)
2338 READ(R,*,ERR=900,IOSTAT=HREAD) EQQ5B(EQQQ5)
2339 ENDFIF
2340
2341 C FACTORS AFFECTING QUANTITY REQUIRED
2342 C FEEDING FILTERING RATE
2343 K = K+1
2344 IF (INFO(K) .EQ. 0) THEN
2345 WRITE(W,20000) K
2346 DO 200 I=BEGIN1,NOOF1
2347 IF (OPNNEW(I,5) .EQ. 'C') THEN
2348 WRITE(W,20100) I, NAMEI(I)
2349 READ(R,*,ERR=900,IOSTAT=HREAD) FILTER(I)
2350 ENDFIF
2351 CONTINUE
2352
2353 C FRACTION OF EACH TYPE OF ALGAE (IF  $\phi$  1) USED IN EACH STAGE
2354 K = K+1
2355 IF (INFO(K) .EQ. 0) THEN
2356 IF (NOOFT(5) .GT. 1) THEN
2357 WRITE(W,21000) K
2358 DO 210 I=BEGIN1,NOOF1
2359 IF (OPNNEW(I,5) .EQ. 'C') THEN
2360
2361
2362

```

```

2363 WRITE(W,20100) I, NAMEI(I)
2364 DO 211 T=1,NOOFT(5)
2365 IF (NOOFT(5) .GT. 1) WRITE(W,20200) NAMEI(5,T)
2366 READ(R,*,ERR=900,IOSTAT=HREAD) NSREQ(I,T)
2367 CONTINUE
2368 ENDFIF
2369
2370 210 CONTINUE
2371 ELSE
2372 DO 220 I=BEGIN1,NOOF1
2373 IF (OPNNEW(I,5) .EQ. 'C') THEN
2374 NSREQ(I,1) = 1.
2375 ENDFIF
2376 CONTINUE
2377 ENDFIF
2378
2379 C AVAILABILITY OF ALGAE BY MONTH FROM SEAWATER, PRODUCED AND PURCHASED
2380 K = K+1
2381 IF (INFO(K) .EQ. 0) THEN
2382 IF ( ( (ENDM-BEGINM) .GT. 0) THEN
2383 WRITE(W,30000) K
2384 READ(R,1000,ERR=900,IOSTAT=HREAD) SEABYM
2385 ELSE
2386 SEABYM = '-'
2387 ENDFIF
2388 INFO(K+1) = 0
2389 ENDFIF
2390 K = K+1
2391 IF (INFO(K) .EQ. 0) THEN
2392 WRITE(W,30100) K
2393 DO 300 I=BEGIN1,NOOF1
2394 IF (OPNNEW(I,5) .EQ. 'C') THEN
2395 WRITE(W,20100) I, NAMEI(I)
2396 IF (SEABYM .EQ. 'Y') THEN
2397 DO 301 M=BEGINM,ENDM
2398 WRITE(W,30200) NAMEM(M)
2399 READ(R,*,ERR=900,IOSTAT=HREAD) SEACEL(I,M)
2400 CONTINUE
2401 ELSE
2402 READ(R,*,ERR=900,IOSTAT=HREAD) SEACEL(I,1)
2403 DO 302 M=BEGINM,ENDM
2404 SEACEL(I,M) = SEACEL(I,1)
2405 CONTINUE
2406 ENDFIF
2407 ENDFIF
2408 CONTINUE
2409 ENDFIF
2410
2411 C QUANTITY OF ALGAE PRODUCED IF PRICE OPTION NOT CHOSEN
2412 C IF PRICE OPTION CHOSEN, THEN PRICE SUBR. SHOULD DETERMINE QUANT.
2413 K = K+1
2414 IF (INFO(K) .EQ. 0) THEN
2415 IF (QOFPNS(5) .EQ. 'N') THEN
2416 IF ( ( (ENDM-BEGINM) .GT. 0) THEN
2417 WRITE(W,31000) K
2418 READ(R,1000,ERR=900,IOSTAT=HREAD) N5ABYM
2419 ELSE
2420 N5ABYM = '-'
2421 ENDFIF
2422 INFO(K+1) = 0
2423 ENDFIF
2424 K = K+1
2425 IF (INFO(K) .EQ. 0) THEN
2426

```



```

2427 IF (QOPNS(5) .EQ. 'N') THEN
2428   WRITE(W,31100) K
2429   DO 311 T=1,NOOFT(5)
2430     WRITE(W,20200) NAMED(5,T)
2431     IF (NSABYM .EQ. 'Y') THEN
2432       DO 312 M=BEGINM,ENDM
2433         WRITE(W,30200) NAMED(M)
2434         READ(R,*,ERR=900,IOSTAT=HREAD) N5AVAL(T,M)
2435         CONTINUE
2436       ELSE
2437         READ(R,*,ERR=900,IOSTAT=HREAD) N5AVAL(T,1)
2438         DO 313 M=BEGINM,ENDM
2439           N5AVAL(T,M) = N5AVAL(T,1)
2440           CONTINUE
2441         ENDF
2442       ENDF
2443     ENDF
2444   ENDF
2445 ENDF
2446 ENDF
2447 ENDF
2448 ENDF
2449 ENDF
2450 ENDF
2451 ENDF
2452 ENDF
2453 ENDF
2454 ENDF
2455 ENDF
2456 ENDF
2457 ENDF
2458 ENDF
2459 ENDF
2460 ENDF
2461 ENDF
2462 ENDF
2463 ENDF
2464 ENDF
2465 ENDF
2466 ENDF
2467 ENDF
2468 ENDF
2469 ENDF
2470 ENDF
2471 ENDF
2472 ENDF
2473 ENDF
2474 ENDF
2475 ENDF
2476 ENDF
2477 ENDF
2478 ENDF
2479 ENDF
2480 ENDF
2481 ENDF
2482 ENDF
2483 ENDF
2484 ENDF
2485 ENDF
2486 ENDF
2487 ENDF
2488 ENDF
2489 ENDF
2490 ENDF

312 CONTINUE
313 CONTINUE
311 ENDF

KLAST = K
WRT2 = WI
RED2 = RI
W = WI
R = RI

C OPTION TO CHECK DATA
WRITE(WI,60000) NAMED(5)
READ(RI,1000) QUEST
IF (QUEST .EQ. 'Y') THEN
  CALL ROUTINS(WI)
ENDF

C OPTION TO STOP OR REDO
IF (RUNI) THEN
  WRITE(WI,64000) NAMED(5)
  READ(RI,1000) QUEST2
  IF (QUEST2 .EQ. 'S') THEN
    GOTO 850
  ELSEIF (QUEST2 .EQ. 'R') THEN
    GOTO 400
  ELSEIF (QUEST .EQ. 'C') THEN
    CONTINUE
  ENDF
  RUN1 = .FALSE.
ENDF

C REASSIGN 1 TO INPUT LINES TO STOP FROM REENTERING DATA
DO 800 K=1,NOOFK
  INFO(K) = 1
CONTINUE

800 CONTINUE

C OPTION TO CORRECT DATA, LINE BY LINE, BY CHOOSING A LINE
810 WRITE(WI,70000) NAMED(5), NOOFK
READ(RI,*) KGOTO
IF (KGOTO .LT. 0 .OR. KGOTO .GT. NOOFK) THEN
  GOTO 810
ELSEIF (KGOTO .GT. 0) THEN
  INFO(KGOTO) = 0
  GOTO 400
ENDF

C OPTION TO PRINT THE DATA ENTERED

```

```

850 WRITE(WI,80000) NAMED(5)
READ(RI,1000) QUEST
IF (QUEST .EQ. 'Y') THEN
  CALL ROUTINS(WO)
ENDF

IF (QUEST2 .EQ. 'S') STOP 'PROGRAM STOP IN DATA ENTRY INPUT 5'

C OPTION TO PRINT OUT ADDITIONAL INFORMATION ABOUT INPUT 5
IF (OOUTN(5) .EQ. 0) THEN
  WRITE(WI,90000)
  READ(RI,1000) QOUTN(5)
  IF (QOUTN(5) .EQ. 'Y') THEN
    WRITE(WI,91000) WO
    READ(RI,*) WON(5)
  ENDF
ENDF

1000 FORMAT (A1)

10000 FORMAT (' ', I2, ' TO COMPUTE ALGAE QUANTITY, CHOOSE ONE OF ',
2 ' THE FOLLOWING EQUATIONS', ' ', 4X, ' THAT BEST ',
2 ' RELATES AGE IN WEEKS TO 10,000 CELLS CONSUMED.', ' ',
3 ' ', 4X, ' (THE EQUATION WILL BE SOLVED FOR THE NUMBER ',
4 ' OF CELLS CONSUMED PER DAY.)', ' ')

10100 FORMAT (' ', 3X, I2, ': ', A)

11000 FORMAT (' ', I2, ' CHOOSE ALGAE PARAMETER A (SUGGESTED: ',
2 ' 614.7, ')')

11100 FORMAT (' ', I2, ' CHOOSE ALGAE PARAMETER B (SUGGESTED: ',
2 ' 614.7, ')')

20000 FORMAT (' ', I2, ' WHAT IS THE FEEDING FILTERING RATE OF THE ',
2 ' ORGANISM AT EACH STAGE', ' ',
3 ' ', 4X, ' FOR WHICH ALGAE QUANTITY IS TO BE COMPUTED')

20100 FORMAT (' ', 3X, ' IN STAGE ', I2, ', ', A)

21000 FORMAT (' ', I2, ' WHAT IS THE FRACTION OF EACH TYPE OF ALGAE ',
2 ' REQUIRED IN EACH STAGE (TOTAL 100%): ')

20200 FORMAT (' ', 6X, ' OF TYPE ', A)

30000 FORMAT (' ', I2, ' DOES THE AVAILABILITY OF ALGAE FROM ',
2 ' SEAWATER CHANGE BY MONTH? ')

30100 FORMAT (' ', I2, ' FRACTION OF DIET BIVALVES RECEIVE FROM ',
2 ' SEAWATER: ')

30200 FORMAT (' ', 7X, A)

31000 FORMAT (' ', I2, ' DOES THE AVAILABILITY OF ALGAE PRODUCED ',
2 ' CHANGE BY MONTH? ')

31100 FORMAT (' ', I2, ' TOTAL AMOUNT OF EACH TYPE OF ALGAE AVAILABLE')

60000 FORMAT (' ', I2, ' DO YOU WANT TO CHECK THE DATA ENTERED FOR ', A)

64000 FORMAT (' ', I2, ' ENTER S TO STOP EXECUTION AFTER AN OPTION TO ',
2 ' PRINT DATA FOR ', A,
3 ' ', ' ENTER R TO REDO ALL ALGAE DATA INTERACTIVELY /'

```

```

2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554

```

```

4 ' ' , 'ENTER C TO CONTINUE WITH OPTION TO CORRECT ' ,
5 'LINE BY LINE' /
6 ' ' , 'ENTER S , R , OR C' )

70000 FORMAT ( ' ' / ' YOU MAY NOW CORRECT DATA A LINE AT A TIME FOR ' , A /
' ' , ' 2X , ENTER 0 TO MAKE NO CORRECTIONS, OR ' /
' ' , ' 2X , ENTER THE NO. OF THE LINE YOU WANT TO CORRECT ' /
' ' , ' ENTER AN INTEGER FORM 0 TO ' , I X , I 2 )

80000 FORMAT ( ' ' / ' DO YOU WANT A PRINTED COPY OF DATA ENTERED FOR ' , A ,
' ? ' Y OR N' )

90000 FORMAT ( ' ' / ' , ' DO YOU WANT TO PRINT ADDITIONAL ' ,
' INFORMATION ABOUT ALGAE? Y OR N' )

91000 FORMAT ( ' ' , ' TO WHAT OUTPUT FILE UNIT DO YOU WANT THIS TO GO TO? ' ,
' - INTEGER 10 TO 99 ' / ' (MAIN OUTPUT FILE HAS ' ,
' A UNIT = ' , I 2 , ' ) ' )

900 RETURN
END

SUBROUTINE READN6(WRT,RED,HREAD,RUN2)
RETURN
END

SUBROUTINE ROUTN1(WRT)
RETURN
END

SUBROUTINE ROUTN2(WRT)
RETURN
END

SUBROUTINE ROUTN3(WRT)
RETURN
END

SUBROUTINE ROUTN4(WRT)
RETURN
END

SUBROUTINE ROUTN5(WRT)
RETURN
END

```

```

C -----
C ORGNSM:
C -----
C SUBROUTINE VARIES WITH ORGANISM
C VARIABLES CHANGE, BUT NOT RETURNED TO ORGNSM
C VARIABLES REQUIRED ARE THOSE TO PRINT OUT VARIABLES ENTERED
C IN READN5, WHICH ARE SPECIFIC TO THE ORGANISM
C CURRENT ORGANISM: BIVALVES
C STRUCTURE: DISPLAYS OR PRINTS OUT DATA FOR SUBROUTINE READN5
C 1. PARAMETERS
C INCLUDE(IPAR1)
C 2. ARGUMENTS
C INTEGER WRT
C 3. COMMON BLOCKS
C INCLUDE(ICBLK1)
C INCLUDE(ICBLKO)
C INCLUDE(ICMAIN)
C INCLUDE(ICORG)
C 4. LOCAL VARIABLES
C INTEGER I, K, M, T, W
C NAME DESCRIPTION OF LOCAL VARIABLES AND ARGUMENTS
C I LOOP COUNTER FOR STAGE
C K INPUT LINE NUMBER
C M LOOP COUNTER FOR MONTH
C T LOOP COUNTER FOR TYPES
C W UNIT NUMBER FOR WRITE
C WRT ARGUMENT FOR UNIT NUMBER TO WRITE
C -----
C TRANSLATE ARGUMENTS TO LOCAL VARIABLES
W = WRT
C WRITE OUT HEADING
WRITE(W,5000) NAME, DATE, NAMEN(5)
C ALGORITHM TO COMPUTE QUANTITY
K = 1
WRITE(W,10000) K, QEQQ5, EQQ5(QEQQ5)
K = K+1
WRITE(W,11000) K, EQQ5A(QEQQ5)
K = K+1
WRITE(W,11100) K, EQQ5B(QEQQ5)
WRITE(W,20000) (I, I=BEGIN1,NOOF1)
C QUANTITY REQUIRED
C FEEDING FILTERING RATE
K = K + 1
WRITE(W,21000) K, (FILTER(I), I=BEGIN1,NOOF1)
C FRACTION OF EACH TYPE USED IN DIET
K = K+1
IF (NOOFT(5) .GT. 1) THEN
WRITE(W,21100) K
DO 211 T=1,NOOFT(5)
IF (NOOFT(5) .GT. 1) WRITE(W,21200) NAMET(5,T)
WRITE(W,21300) (NSREQ(I,T), I=BEGIN1,NOOF1)
211 CONTINUE
ENDIF

```

```

2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2680
2681
2682

```

```

C AVAILABILITY
K = K+1
IF ((ENDM-BEGINM) .GT. 0) WRITE(W,30000) K, SEABYM
K = K+1
WRITE(W,30100) K
DO 300 M=BEGINM, ENDM
WRITE(W,30200) NAMEM(M), (SEACEL(I,M), I=BEGINI, NOOFTI)
300 CONTINUE
K = K+1
IF (QOPNS(5) .EQ. 'N' .AND. (ENDM-BEGINM) .GT. 0) THEN
  ENDIF
  WRITE(W,31000) K, NSABYM
K = K+1
IF (QOPNS(5) .EQ. 'N') THEN
  WRITE(W,31100) K, (NAMEM(M), M=BEGINM, ENDM)
  DO 310 T=1, NOOFT(5)
  IF (NOOFT(5) .GT. 1) WRITE(W,21200) NAMEM(5,T)
  WRITE(W,31300) (NSAVAL(T,M), M=BEGINM, ENDM)
310 CONTINUE
ENDIF
5000 FORMAT('1',A,' COST SIMULATION MODEL: INPUT ENTERED (CONT.)') /
2 'A50','A,1X,'DATA' /
10000 FORMAT(' ',I2,'. AGE/ALGAE EQUATION CHOSEN: ',I1/' ',4X,A)
11000 FORMAT(' ',I2,'. PARAMETER A = ',G14.7)
11100 FORMAT(' ',I2,'. PARAMETER B = ',G14.7/)
20000 FORMAT(' ',I2,'. IN THE FOLLOWING, WHERE QUANTITY=0, QUANTITY IS ',
2 'ENTERED'/' ',6X,6(I2,' STAGE',2X)/)
21000 FORMAT(' ',I2,'. FILTERING RATE'/' ',6X,6(G9.2,1X))
21100 FORMAT(' ',I2,'. FRACTION OF EACH ALGAE TYPE REQUIRED ',
2 'PER STAGE:')
21200 FORMAT(' ',4X,' TYPE ',A)
21300 FORMAT(' ',6X,6(G9.2,1X))
30000 FORMAT(' ',I2,'. FRACTION FROM SEA VARIES BY MONTH?: ',A1)
30100 FORMAT(' ',I2,'. FRACTION OF DIET FROM SEA ALGAE IN EACH ',
2 'STAGE, EACH MONTH:')
30200 FORMAT(' ',6X,A/' ',6X,6(G9.2,1X)/)
31000 FORMAT(' ',I2,'. AVAILABILITY OF ALGAE VARIES BY MONTH?: ',A)
31100 FORMAT(' ',I2,'. AVAILABILITY EACH TYPE BY MONTH: '/
2 ' ',6X,6(A10,1X)/' ',6X,6(A10,1X))
31300 FORMAT(' ',6X,6(G9.2,1X)/' ',6X,6(G9.2,1X))

RETURN
END

```

```

2683
2684
2685
2686
2687
2688
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2740
2741
2742
2743
2744
2745
2746

```

```

SUBROUTINE ROUTN6 (WRT)
RETURN
END

SUBROUTINE PRICE1 (WRT, RED, HREAD, RUN2, RUN3)
RETURN
END

SUBROUTINE PRICE2 (WRT, RED, HREAD, RUN2, RUN3)
RETURN
END

SUBROUTINE PRICE3 (WRT, RED, HREAD, RUN2, RUN3)
RETURN
END

SUBROUTINE PRICE4 (WRT, RED, HREAD, RUN2, RUN3)
RETURN
END

SUBROUTINE PRICE5 (WRT, RED, HREAD, RUN2, RUN3)
-----
C ORGNISM
C -----
C SUBROUTINE VARIES WITH ORGANISM
C VARIABLES CHANGE WITH ORGANISM
C VARIABLES NEED BE RETURNED (ONLY IF PRICE OPTION CHOSEN) :
C NUMBER OF TYPES - NOOFT(5)
C NAME OF EACH TYPE - NAMEI(5,I)
C PRICE PER LITRE - PRICET(5,T,M)
C AMOUNT AVAILABLE- N5AVAL(T,M)
C CURRENT ORGANISM: BIVALVE

C STRUCTURE: MAIN SUBROUTINE, PRICES, AND 8 LESSER SUBROUTINES,
C MAIN SUBROUTINE DOES THE FOLLOWING:
C -CALLS SUBROUTINE RINS$, ROUT$5
C -LOOPS THROUGH PRODUCTION MONTHS
C -LOOPS THROUGH PRODUCTION STAGES
C CALCULATES AVAILABLE ALGAE QUANTITY
C UPDATE$S CURRENT PRODUCTION MONTH MJ
C CALLS SUBROUTINE ELEC TO FIND KWHR
C CALLS SUBROUTINE FUEL TO FIND LITRES OF OIL

```

```

2747
2748
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2810

```

C	CALLS SUBROUTINE LABOR TO FIND HOURS OF LABOR	2811	C	NAME	2875
C	CALLS SUBROUTINE GOODS TO FIND MATERIAL UNITS	2812	C	NO. OF AIRCOND. IN ROOM OR EACH STAGE	2876
C	CALLS SUBROUTINE COST TO FIND COST OF STAGE	2813	C	NO. OF DIFFERENT GOOD TYPES, G	2877
C	UPDATE2S STAGE TO S+1	2814	C	NO. OF DIFFERENT LABOR TYPES, I	2878
C	CALLS SUBROUTINE WRIT55 -PRINT BY MONTH BATCH BEGUN	2815	C	NO. OF STAGES TO PRODUCE A SINGLE BATCH	2879
C	UPDATE2S MONTH TO M+1	2816	C	NUMBER OF TANKS PER STAGE	2880
C	-ALLOWS PROGRAM TO BE RERUN WITH A FEW DATA CHANGES	2817	C	NUMBER OF TANKS FED BY COMPRESSOR	2881
C	??	2818	C	NUTRIENT SOLUTION ADDED PER L OF H2O/STG	2882
C	PRICE OF ALGAE SUBROUTINES - GENERAL INFORMATION	2819	C	PRICE OF GOOD, TYPE G	2883
C	-----	2820	C	PRICE PER KWHR OF ELECTRICITY	2884
C	-----	2821	C	PRICE PER HOUR OF TYPE L LABOR	2885
C	-----	2822	C	FRACTION EFFICIENCY RATING OF WATER PUMP	2886
C	-----	2823	C	OUTPUT CAPACITY RATING OF WATER PUMP	2887
C	-----	2824	C	PRICE PER LITRE OF FUEL OIL	2888
C	-----	2825	C	WRITE OUT OUTPUT FOR PRICES CALCULATIONS?	2889
C	-----	2826	C	NUMBER OF STAGES IN THE ROOM	2890
C	-----	2827	C	TOTAL TYPE L LABOR PER STAGE, PLUS INNOC.	2891
C	-----	2828	C	TOTAL TYPE L LABOR PER STAGE/LITR, + INNOC	2892
C	-----	2829	C	COST OF ALL INPU, I, PER STAGE, + INNOC.	2893
C	-----	2830	C	TOTAL CELLS OF ALGAE PRODUCED	2894
C	-----	2831	C	COST OF INPUT, I, PER STAGE, PLUS INNOC.	2895
C	-----	2832	C	TOTAL COST PER CELL OF ALGAE PRODUCED	2896
C	-----	2833	C	COST OF INPUT, I, PER STAGE, + INNOCULATION	2898
C	-----	2834	C	COST OF TYPE L LABOR PER STAGE, + INNOC.	2900
C	-----	2835	C	COST OF ALL INPUT, I, PER LITRE PER STAGE	2901
C	-----	2836	C	UNITS OF TYPE G GOOD USED PER STAGE	2902
C	-----	2837	C	ELECTRICITY USED PER STAGE, LESS INNOCULA.	2904
C	-----	2838	C	TYPE L LABOR USED PER STAGE, LESS INNOCULA.	2905
C	-----	2839	C	ALGAE PRODUCED IN EACH STAGE	2906
C	-----	2840	C	OIL CONSUMPTION PER STAGE, LESS INNOCULATION	2908
C	-----	2841	C	TYPE G GOOD USED PER STAGE, PLUS INNOC.	2909
C	-----	2842	C	ELC. PER LITRE PER STAGE, PLUS INNOC.	2910
C	-----	2843	C	ELEC USED PER STAGE, PLUS INNOC.	2912
C	-----	2844	C	LABOR, TYPE LE, PER STAGE, PLUS INNOC.	2913
C	-----	2845	C	OIL CONSUMED PER STAGE, PLUS INNOCULATION	2914
C	-----	2846	C	OIL CONSUMED PER LITRE OF ALGAE	2915
C	-----	2847	C	COMPRESSOR TUBES PER TANK PER STAGE	2917
C	-----	2848	C	AIR COMPRESSOR INPUT RATING	2918
C	-----	2849	C	AUTOCCLAVE INPUT WATT RATING	2919
C	-----	2850	C	LIGHTBULB INPUT RATING	2920
C	-----	2851	C	-----	2921
C	-----	2852	C	-----	2922
C	-----	2853	C	-----	2923
C	-----	2854	C	-----	2924
C	-----	2855	C	-----	2925
C	-----	2856	C	-----	2926
C	-----	2857	C	-----	2927
C	-----	2858	C	-----	2928
C	-----	2859	C	-----	2929
C	-----	2860	C	-----	2930
C	-----	2861	C	-----	2931
C	-----	2862	C	-----	2932
C	-----	2863	C	-----	2933
C	-----	2864	C	-----	2934
C	-----	2865	C	-----	2935
C	-----	2866	C	-----	2936
C	-----	2867	C	-----	2937
C	-----	2868	C	-----	2938
C	-----	2869	C	-----	
C	-----	2870	C	-----	
C	-----	2871	C	-----	
C	-----	2872	C	-----	
C	-----	2873	C	-----	
C	-----	2874	C	-----	

C 1. COMMON BLOCK VARIABLES
 INCLUDE (IPAR1)
 INCLUDE (ICN5)
 INCLUDE (ICUNIT)
 INCLUDE (ICBLK1)
 INCLUDE (ICORGE)
 INCLUDE (ICELKO)
 INCLUDE (ICMAIN)
 C 2. ARGUMENT VARIABLES
 I INTEGER STAGE, TIME,
 I WRT, WRT2, RED, RED2, HREAD,
 I LASSTG
 LOGICAL RUN2, RUN3
 CHARACTER*9 MONBEG

```

3003 TDAY = 0.
3004 DO 30 S = 1, NUMSTG
3005
3006 C ACCUMULATE DAYS TO DETERMINE WHAT MONTH CURRENT STAGE IS
3007 BEING PRODUCED IN (FIRST CONVERT TO AN INTEGER)
3008 STAGE = S
3009 TDAY = DAYSTG(S) + TDAY
3010 RTDAY = TDAY + 0.9999999
3011 ITDAY = RTDAY
3012 ID = (ITDAY - 1) / 30
3013 TIME = M + ID
3014 IF ((ENDPRO-BEGPRO).EQ.11 .AND. TIME.GT.12) TIME = TIME - 12
3015
3016 C STOP CALCULATIONS IF STAGE CONTINUES INTO NON-PRODUCTION MONTHS
3017 IF (TIME.GT.ENDPRO) THEN
3018 MONAVA(S) = 'NOT AVAIL.'
3019 TLITRE(S) = 0.
3020 LFOOD(S) = 0.
3021 TCELL(S) = 0.
3022 IF (S.GT.1) THEN
3023 LFOOD(S-1) = TLITRE(S-1)
3024 TCELL(S-1) = CPERML(S-1) * 1000. * TLITRE(S-1)
3025 ENDF
3026
3027 C OTHERWISE CALCULATE ALL STAGE QUANTITIES AND COSTS
3028 ELSE
3029 MONTH THAT ALGAE IS AVAILABLE AND LAST STAGE OF PRODUCTION
3030 MONAVA(S) = NAMEM(TIME)
3031 LASSTG = S
3032
3033 C QUANTITIES PRODUCED AND AVAILABLE - CHECK FOR LAST STAGE
3034 TLITRE(S) = (H2OTNK(S)+INQTNK(S)) * NUMTNK(S) - DISSTG(S)
3035 TCELL(S) = CPERML(S) * 1000. * TLITRE(S)
3036 IF (S.EQ.NUMSTG) THEN
3037 LFOOD(S) = TLITRE(S)
3038 ELSE
3039 LFOOD(S) = TLITRE(S) - INQTNK(S+1) * NUMTNK(S+1)
3040 ENDF
3041
3042 C CALCULATE INPUT QUANTITIES - BY CALLING SUBROUTINES
3043 CALL ELEC(STAGE, TIME)
3044 CALL FUEL (STAGE, TIME)
3045 CALL LABOR(STAGE, TIME)
3046 CALL GOODS(STAGE, TIME)
3047
3048 C CALCULATE INPUT COSTS
3049 CALL COST(STAGE, TIME)
3050
3051 C TRANSLATE PRICES VARIABLES INTO ORGNM VARIABLES
3052 PRICET(5,S,TIME) = TCELL(S)
3053 N5AVAL(S,TIME) = TCELL(S)
3054 ENDF
3055
3056 30 CONTINUE
3057 IF (QWRT$5.EQ.'Y') CALL WRT$5(MONBEG,LASSTG,WO)
3058
3059 20 CONTINUE
3060
3061 C OFFER AN OPTION TO SEE LAST OUTPUT
3062 WRITE (WI,290)
3063 FORMAT (' ',' ',' ','DO YOU WANT TO SEE THE COMPUTATIONS FOR THE ',
3064 2 'LAST BATCH OF ALGAE? ')
3065 READ(RI,1000) QUEST
3066
2939
2940
2941
2942
2943
2944
2945
2946
2947
2948
2949
2950
2951
2952
2953
2954
2955
2956
2957
2958
2959
2960
2961
2962
2963
2964
2965
2966
2967
2968
2969
2970
2971
2972
2973
2974
2975
2976
2977
2978
2979
2980
2981
2982
2983
2984
2985
2986
2987
2988
2989
2990
2991
2992
2993
2994
2995
2996
2997
2998
2999
3000
3001
3002

```

3. LOCAL VARIABLES
INTEGER ID, ITDAY, M, MSTART, MSTOP, S
REAL TDAY, RTDAY
CHARACTER*1 ANSWR3, QUEST

C NAME DESCRIPTION OF LOCAL VARIABLES AND ARGUMENTS
C ANSWR3 OPTION TO REDO WITH A FEW CHANGES
C HREAD ARGUE: IOSTAT TO HELP INDICATE PROBLEMS IN READING FILE
C ID ADDITION TO MONTH TO UPDATE FOR CURRENT STAGE
C ITDAY INTEGER TO MONTH TO UPDATE FOR CURRENT STAGE
C LASSTG ARGUE: LAST STAGE OF ALGAE CALCULATED IN THE BATCH
C M LOOP COUNTER FOR MONTHS
C MONBEG ARGUE: MONTH IN WHICH STAGE=1 OF ALGAE IS BEGUN
C MSTART FIRST MONTH OF ALGAE PRODUCTION TO LOOP THROUGH
C MSTOP LAST MONTH OF ALGAE PRODUCTION TO LOOP THROUGH
C RED ARGUE: TRANSLATE UNIT READ FROM ORGNM TO NEXT LEVEL SUBROUTINE
C RED2 ARGUE: TRANSLATE UNIT READ TO NEXT LEVEL SUBROUTINE
C RTDAY CONVERT TOTAL DAYS ELAPSED TO REAL
C RUN2 ARGUE: INDICATES THAT ONE RUN THRU ALGAE PRICE ALREADY DONE
C RUN3 ARGUE: INDICATES THAT 1 RUN THRU FROM ABOVE LEVEL SUBR. DONE
C S LOOP COUNTER FOR STAGE
C STAGE ARGUE: CURRENT STAGE OF ALGAE THAT CALCULATIONS ARE FOR
C TDAY TOTAL ELAPSED DAYS
C TIME ARGUE: CURRENT MONTH THAT CURRENT STAGE IS IN
C QUEST OPTION TO SEE LAST OUTPUT ON SCREEN
C WRT ARGUE: TRANSLATE UNIT WRITE FROM ORGANISM TO NEXT LEVEL SUBR.
C WRT2 ARGUE: UNIT TO WRITE TO NEXT LEVEL SUBROUTINE
C -----

C TRANSLATE ARGUMENTS
WRT2 = WRT
RED2 = RED

C CALL SUBROUTINE TO INPUT ALL DATA
CALL RIN\$5 (WRT2,RED2,RUN2,RUN3)

C OFFER A SECOND INPUT ENTRY POINT TO ENABLE MINOR INPUT CHANGES
ANSWR3 = 'X'
10 IF (ANSWR3.EQ.'Y') CALL INPUT2(WRT2,RED2,RUN2,RUN3)

C OFFER AN OPTION TO PRINT OUTPUT FOR ALGAE PRICE
WRITE(WI,400)
400 READ (RI,1000) QWRT\$5
FORMAT(' ',' ','DO YOU WANT A PRINTED COPY OF THE ALGAE PRICE ',
2 'COMPUTATIONS? Y OR N')

C ASSIGN THE MONTHS TO BE CALCULATED
MSTART = BEGPRO
MSTOP = ENDPRO
DO 20 M = MSTART,MSTOP

C LOOP THROUGH MONTHS OF PRODUCTION - M IS THE MONTH BATCH IS BEGUN
DO 20 M = MSTART,MSTOP

C NAME OF THE MONTH IN WHICH PRODUCTION BEGINS
C ALLOW PRINTING OF AT LEAST ONE STAGE
MONBEG = NAMEM(M)
LASSTG = NUMSTG
MONTH = M

C LOOP THROUGH EACH STAGE TO CALCULATE COSTS


```

2000 FORMAT(' ',' ','ENTER DATA CONCERNING ALGAE PRODUCTION')
201 FORMAT(' ','1. DATE OF COMPUTATIONS (TO 25 CHARACTERS IN QUOTES)',I3)
202 FORMAT(' ','2. PRICE PER KWHR OF ELECTRICITY',F10.2)
203 FORMAT(' ','3. PRICE PER LITRE OF FUEL OIL',F10.2)
204 FORMAT(' ','4. PRICE PER LITRE OF FUEL OIL',F10.2)
205 FORMAT(' ','5. PRICE PER LITRE NUTRIENT SOLUTION',F10.2)
206 FORMAT(' ','6. PRICE PER HOUR OF EACH LABOR TYPE',F10.2)
207 FORMAT(' ','7. PRICE OF TYPE',I1)
C LINE NUMBER 7-19
08 IF (INFO(8).EQ.0) THEN
  WRITE(W,208)
  READ(R,*,ERR=8) WATAIR
  ENDIF
09 IF (INFO(9).EQ.0) THEN
  WRITE(W,209)
  READ(R,*,ERR=9) WATBLB
  ENDIF
10 IF (INFO(10).EQ.0) THEN
  WRITE(W,210)
  READ(R,*,ERR=10) BBLIF
  ENDIF
11 IF (INFO(11).EQ.0) THEN
  WRITE(W,211)
  READ(R,*,ERR=11) WATAUT
  ENDIF
12 IF (INFO(12).EQ.0) THEN
  WRITE(W,212)
  READ(R,*,ERR=12) HPCOM
  ENDIF
13 IF (INFO(13).EQ.0) THEN
  WRITE(W,213)
  READ(R,*,ERR=13) HRSCOM
  ENDIF
14 IF (INFO(14).EQ.0) THEN
  WRITE(W,214)
  READ(R,*,ERR=14) NUMTUB
  ENDIF
15 IF (INFO(15).EQ.0) THEN
  WRITE(W,215)
  READ(R,*,ERR=15) COMEFF
  ENDIF
16 IF (INFO(16).EQ.0) THEN
  WRITE(W,216)
  READ(R,*,ERR=16) PMPEFF
  ENDIF
17 IF (INFO(17).EQ.0) THEN
  WRITE(W,217)
  READ(R,*,ERR=17) PMPGPM
  ENDIF
18 IF (INFO(18).EQ.0) THEN
  WRITE(W,218)
  READ(R,*,ERR=18) HPPMP
  ENDIF
19 IF (INFO(19).EQ.0) THEN
  WRITE(W,219)
  DO 81 I=1,NUMLAB
  WRITE(W,219) I
  READ(R,*,ERR=19) HRLABW(L)
  CONTINUE
81 ENDIF
208 FORMAT(' ','8. WAIT RATING OF THE AIR CONDITIONER')
3195
3196
3197
3198
3199
3200
3201
3202
3203
3204
3205
3206
3207
3208
3209
3210
3211
3212
3213
3214
3215
3216
3217
3218
3219
3220
3221
3222
3223
3224
3225
3226
3227
3228
3229
3230
3231
3232
3233
3234
3235
3236
3237
3238
3239
3240
3241
3242
3243
3244
3245
3246
3247
3248
3249
3250
3251
3252
3253
3254
3255
3256
3257
3258
209 FORMAT(' ','9. AVERAGE WAIT RATING OF LIGHTBULBS')
210 FORMAT(' ','10. AVERAGE LIFETIME OF LIGHTBULB-IN HOURS')
211 FORMAT(' ','11. WAIT RATING OF AUTOCLAVE OR STERILIZER')
212 FORMAT(' ','12. HORSEPOWER RATING OF AIR COMPRESSOR')
213 FORMAT(' ','13. HOURS PER DAY THAT COMPRESSOR OPERATES')
214 FORMAT(' ','14. TOTAL NUMBER OF AIR COMPRESSOR TUBES')
215 FORMAT(' ','15. EFFICIENCY RATING OF AIR COMPRESSOR')
216 FORMAT(' ','16. EFFICIENCY RATING OF WATER PUMP')
217 FORMAT(' ','17. GALLONS/MINUTE RATING OF WATER PUMP')
218 FORMAT(' ','18. HORSEPOWER RATING OF WATER PUMP')
219 FORMAT(' ','19. HOURS PER WEEK OF EACH LABOR TYPE')
220 FORMAT(' ','20. HOURS OF TYPE',I1)
C LINES 20-35
20 DO 4000 K=20,35
  IF (INFO(K).EQ.0) RETYP1=0
  CONTINUE
4000 IF (INFO(20).EQ.0) THEN
  WRITE(W,220) MAXI
  READ(R,*,ERR=20) NUMSTG
  ENDIF
  IF (RETY1.EQ.0) THEN
  WRITE(W,222)
  WRITE(W,222) (S, S=1,NUMSTG)
  ENDIF
  IF (INFO(21).EQ.0) THEN
  WRITE(W,221)
  READ(R,*,ERR=21) (NUTLTR(S), S=1,NUMSTG)
  ENDIF
  IF (INFO(22).EQ.0) THEN
  WRITE(W,222)
  READ(R,*,ERR=22) (NUMTNK(S), S=1,NUMSTG)
  ENDIF
  IF (INFO(23).EQ.0) THEN
  WRITE(W,223)
  READ(R,*,ERR=23) (STGRM(S), S=1,NUMSTG)
  ENDIF
  IF (INFO(24).EQ.0) THEN
  WRITE(W,224)
  READ(R,*,ERR=24) (DAYSTG(S), S=1,NUMSTG)
  ENDIF
  IF (INFO(25).EQ.0) THEN
  WRITE(W,225)
  READ(R,*,ERR=25) (CPERML(S), S=1,NUMSTG)
  ENDIF
  IF (INFO(26).EQ.0) THEN
  WRITE(W,226)
  READ(R,*,ERR=26) (H2OTNK(S), S=1,NUMSTG)
  ENDIF
  IF (INFO(27).EQ.0) THEN
  WRITE(W,227)
  READ(R,*,ERR=27) (INQTNK(S), S=1,NUMSTG)
  ENDIF
  IF (INFO(28).EQ.0) THEN
  WRITE(W,228)
  READ(R,*,ERR=28) (DISSTG(S), S=1,NUMSTG)
  ENDIF
  IF (INFO(29).EQ.0) THEN
  WRITE(W,229)
  READ(R,*,ERR=29) (TUBTNK(S), S=1,NUMSTG)
  ENDIF
  IF (INFO(30).EQ.0) THEN
  WRITE(W,230)
  READ(R,*,ERR=30) (BLBRM(S), S=1,NUMSTG)
  ENDIF

```

```

3259
3260
3261
3262
3263
3264
3265
3266
3267
3268
3269
3270
3271
3272
3273
3274
3275
3276
3277
3278
3279
3280
3281
3282
3283
3284
3285
3286
3287
3288
3289
3290
3291
3292
3293
3294
3295
3296
3297
3298
3299
3300
3301
3302
3303
3304
3305
3306
3307
3308
3309
3310
3311
3312
3313
3314
3315
3316
3317
3318
3319
3320
3321
3322

```



```

3451          PRINT2 = 2
3452          ENDIF
3453          IF (COPYA .EQ. 'Y' .AND. COPYB .EQ. 'Y') THEN
3454              PRINT1 = 1
3455              PRINT2 = 2
3456          ENDIF
3457          IF (COPYA .NE. 'Y' .AND. COPYB .NE. 'Y') GOTO 98
3458
3459
3460
3461
3462
3463
3464
3465
3466
3467
3468
3469
3470
3471
3472
3473
3474
3475
3476
3477
3478
3479
3480
3481
3482
3483
3484
3485
3486
3487
3488
3489
3490
3491
3492
3493
3494
3495
3496
3497
3498
3499
3500
3501
3502
3503
3504
3505
3506
3507
3508
3509
3510
3511
3512
3513
3514
-----
C SUBROUTINE ROUT5$
C
C ORGNSM:
C-----
C SUBROUTINE VARIES WITH ORGANISM
C VARIABLES CHANGE WITH ORGANISM
C VARIABLES NOT RETURNED TO ORGNSM, ONLY TO PRICES
C CURRENT ORGANISM: BIVALVES
C
C STRUCTURE:  READS OUT DATA RIN$S IN SUBROUTINE INPUT,
C              WHICH IS USED TO CALCULATE PRICES, ALGAE
C
C 1. COMMON BLOCKS
C   INCLUDE (IPAR1)
C   INCLUDE (ICN5$)
C   INCLUDE (ICUNIT)
C   INCLUDE (ICBLK1)
C   INCLUDE (ICBLKO)
C   INCLUDE (ICMAIN)
C 2. LOCAL VARIABLES
C   CHARACTER*1 COPYA, COPYB
C   INTEGER PRINT1, PRINT2, G, L, P, M, S, W
C-----
C NAME  DESCRIPTION OF LOCAL VARIABLES AND ARGUMENTS
C-----
C COPYA OPTION TO PRINT INPUT DATA
C COPYB OPTION TO WRITE INPUT DATA
C G     LOOP COUNTER FOR GOODS
C L     LOOP COUNTER FOR MONTHS
C M     LOOP COUNTER FOR DIFFERENT PRINTING OPTIONS
C P     PRINTING INITIAL VALUE (1 OR 2)
C PRINT1 PRINTING FINAL VALUE (1 OR 2)
C S     LOOP COUNTER STAGE
C W     UNIT NUMBER TO WRITE
C-----
C TRANSLATE UNIT NUMBERS TO UNIT NUMBERS USED IN ORGNSM
W = WO
C
C OPTION TO PRINT INPUT
WRITE(WI,3000)
3000 FORMAT(' ', 'DO YOU WANT TO PRINT THE ALGAE PRICE DATA ',
2 ' ENTERED?')
1000 FORMAT(A1)
3001 WRITE(WI,3001)
3001 FORMAT(' ', 'DO YOU WANT TO SEE THE ALGAE DATA ON THE SCREEN? ',
2 ' Y OR N')
READ(RI,1000) COPYB
IF (COPYA .EQ. 'Y' .AND. COPYB .NE. 'Y') THEN
PRINT1 = 1
PRINT2 = 1
ENDIF
IF (COPYB .EQ. 'Y' .AND. COPYA .NE. 'Y') THEN
3515          PRINT2 = 2
3516          ENDIF
3517          IF (COPYA .EQ. 'Y' .AND. COPYB .EQ. 'Y') THEN
3518              PRINT1 = 1
3519              PRINT2 = 2
3520          ENDIF
3521          IF (COPYA .NE. 'Y' .AND. COPYB .NE. 'Y') GOTO 98
3522
3523
3524
3525
3526
3527
3528
3529
3530
3531
3532
3533
3534
3535
3536
3537
3538
3539
3540
3541
3542
3543
3544
3545
3546
3547
3548
3549
3550
3551
3552
3553
3554
3555
3556
3557
3558
3559
3560
3561
3562
3563
3564
3565
3566
3567
3568
3569
3570
3571
3572
3573
3574
3575
3576
3577
3578
-----
DO 99 P = PRINT1,PRINT2
IF (P .EQ. 2) V = WI
WRITE(W,3300) NAME, DATE
WRITE(W,301) DATE2
WRITE(W,3302)
WRITE(W,302) PKUHR
WRITE(W,303) POLL
WRITE(W,304) (PGOOD(G), G=1,NUMGOD)
WRITE(W,305) NUMLAB
DO 83 L=1,NUMLAB
WRITE(W,306) L, PLABOR(L)
CONTINUE
83
WRITE(W,3307)
WRITE(W,308) WATAIR
WRITE(W,309) WATBLB
WRITE(W,310) BLBLIF
WRITE(W,311) WATAUT
WRITE(W,312) HPCOM
WRITE(W,313) HRSOCM
WRITE(W,314) NUMTUB
WRITE(W,315) COMEFF
WRITE(W,316) PMPFP
WRITE(W,317) PMPGPM
WRITE(W,318) HPPMP
WRITE(W,319)
DO 84 L=1,NUMLAB
WRITE(W,319) L, HRLABW(L)
CONTINUE
84
WRITE(W,3320)
WRITE(W,320) NUMSTG
WRITE(W,3321) (S, S=1,NUMSTG)
WRITE(W,321) (NUTLR(S), S=1,NUMSTG)
WRITE(W,322) (NOMNK(S), S=1,NUMSTG)
WRITE(W,323) (STGRM(S), S=1,NUMSTG)
WRITE(W,324) (DAYSTG(S), S=1,NUMSTG)
WRITE(W,325) (CPERM(S), S=1,NUMSTG)
WRITE(W,326) (HZOTNK(S), S=1,NUMSTG)
WRITE(W,327) (INQTNK(S), S=1,NUMSTG)
WRITE(W,328) (DISSTG(S), S=1,NUMSTG)
WRITE(W,329) (TUBTNK(S), S=1,NUMSTG)
WRITE(W,330) (BLBRM(S), S=1,NUMSTG)
WRITE(W,331) (HRSBLE(S), S=1,NUMSTG)
WRITE(W,332) (HRSAUT(S), S=1,NUMSTG)
WRITE(W,333) (NUMAIR(S), S=1,NUMSTG)
WRITE(W,3334)
DO 89 L=1,NUMLAB
WRITE(W,334) L, (HRLABS(L), S=1,NUMSTG)
CONTINUE
89
WRITE(W,335) (NAMALG(S), S=1,NUMSTG)
WRITE(W,3336)
WRITE(W,336) BEGPRO, NAMEM(BEGPRO)
WRITE(W,337) ENDPRO, NAMEM(ENDPRO)

```

```

3579 WRITE(W,3338) (S, S=1, NUMSTG)
3580 WRITE(W,3339)
3581 DO 85 M=BEGPRO,ENDPRO
3582 WRITE(W,338) M, NAMEM(M), (HRSAIR(M,S), S=1, NUMSTG)
3583 CONTINUE
85
C
3584 RESET UNIT NUMBER
3585 W = WI
3586
3587 CONTINUE
3588
3589
3590
3591
3592
3593
3594
3595
3596
3597
3598
3599
3600
3601
3602
3603
3604
3605
3606
3607
3608
3609
3610
3611
3612
3613
3614
3615
3616
3617
3618
3619
3620
3621
3622
3623
3624
3625
3626
3627
3628
3629
3630
3631
3632
3633
3634
3635
3636
3637
3638
3639
3640
3641
3642
3643
3644
3645
3646
3647
3648
3649
3650
3651
3652
3653
3654
3655
3656
3657
3658
3659
3660
3661
3662
3663
3664
3665
3666
3667
3668
3669
3670
3671
3672
3673
3674
3675
3676
3677
3678
3679
3680
3681
3682
3683
3684
3685
3686
3687
3688
3689
3690
3691
3692
3693
3694
3695
3696
3697
3698
3699
3700
3701
3702
3703
3704
3705
3706

```

B T41, 6(F13.2,2X)/ ' , T41, 6(F13.2,2X))
333 FORMAT('0', 33, NUMBER OF AIR CONDIT. PER ROOM,
T41, 6(F13.1,2X)/ ' , T41, 6(F13.1,2X))
3334 FORMAT('0', 34, HRS OF LABOR PER STAGE')
334 FORMAT(' , 4X, 'OF TYPE', I1, T41, 6(F13.2,2X)) / ' , T41, 6(F13.2,2X))
335 FORMAT('0', 35, NAME OF EACH ALGAE STAGE',
T41, 6(A10,5X) / ' , T41, 6(A10,5X))
3356
3336 FORMAT('0', *MONTH INFORMATION')
336 FORMAT('0', 36, MONTH BEGIN PRODUCTION', 2X, I2, 1X, A)
337 FORMAT('0', 37, MONTH END PRODUCTION', 4X, I2, 1X, A)
3338 FORMAT('0', T43, 6(3X, I2, 'STAGE', 5X) / ' , T43, 6(3X, I2, 'STAGE', 5X))
3339 FORMAT('0', 38, HOURS AIR CONDIT. ON PER DAY/ROOM')
338 FORMAT('0', 3X, I2, 1X, A, T41, 6(F13.2,2X) / ' , T41, 6(F13.2,2X))
98 CONTINUE
3658
3659
3660
3661
3662
3663
3664
3665
3666
3667
3668
3669
3670
3671
3672
3673
3674
3675
3676
3677
3678
3679
3680
3681
3682
3683
3684
3685
3686
3687
3688
3689
3690
3691
3692
3693
3694
3695
3696
3697
3698
3699
3700
3701
3702
3703
3704
3705
3706

C
C 1. COMMON BLOCK VARIABLES
INCLUDE (IPAR1)
INCLUDE (ICN5\$)
INCLUDE (ICUNIT)
C 2. ARGUMENT VARIABLES
INTEGER STAGE, TIME
C 3. LOCAL VARIABLES
INTEGER MJ, S
REAL AIRWHR, AUTWHR, COMIN, COMWH, INQKWH, PERCOM,
2. PMPCON, PMPIN, PMPINL, PMPWHR, WATPH
C NAME DESCRIPTION OF LOCAL VARIABLES AND ARGUMENTS
C AIRWHR ELEC. ENERGY FOR AIR CONDITIONING
C AUTWHR ELEC. ENERGY FOR STERILIZING
C COMIN ELEC INPUT FOR AIR COMPRESSOR
C COMWH ELEC ENERGY FOR AIR COMPRESSOR
C INQKWH ELEC ENERGY USED FOR INOCULATION
C MJ MONTH IN WHICH CURRENT STAGE IS PRODUCED
C PERCOM FRACTION OF TOTAL AIR COMPRESSOR USED/DAY
C PMPCON CONVERSION FACTOR FOR GAL/MIN TO LITRE/MIN
C PMPIN PUMP ELECTRICAL INPUT
C PMPINL ELEC POWER INPUT TO PUMP/LITRE
C PMPWHR ELEC. ENERGY TO PUMP ALL WATER/STAGE
C S CURRENT STAGE
C STAGE ARGUE: TO TRANSLATE CURRENT STAGE OF ALGAE, S

C ORGNSM:

C SUBROUTINE VARIES WITH ORGANISM
C VARIABLES CHANGE, RETURNED ONLY TO PRICES
C CURRENT ORGANISM: BIVALVES
C STRUCTURE: CALCULATES WHR FOR MAJOR ELECTRICAL USE, AND SUMS FOR
TOTAL KWHR PER STAGE, PER LITRE OF ALGAE PRICE PRODUCTION
3679
3680
3681
3682
3683
3684
3685
3686
3687
3688
3689
3690
3691
3692
3693
3694
3695
3696
3697
3698
3699
3700
3701
3702
3703
3704
3705
3706

```

C TIME ARGUE: TO TRANSLATE CURRENT MONTH STAGE IS IN TO MJ      3707
C WATHP CONVERSION FACTOR FOR WATTS/HP                          3708
C -----                                                    3709
C INITIALIZE LOCAL                                             3710
C DATA WATHP/ 745.7 /                                          3711
C CONVERT PUMP FROM GPM TO L/HR 3.78537L/G * 60M/H           3712
C DATA PMPCON/ 227.122/                                        3713
C -----                                                    3714
C TRANSLATE ARGUMENTS INTO LOCAL VARIABLES                    3715
C S = STAGE                                                    3716
C MJ = TIME                                                    3717
C -----                                                    3718
C LIGHTING                                                    3719
C LIWHR(S) = (WATBLB * BLBRM(S) * HRSBLB(S) / STGRM(S)) * DAYSTG(S) 3720
C -----                                                    3721
C AUTOCLAVE                                                    3722
C AUTWHR = WATAUT * HRSAUT(S)                                  3723
C -----                                                    3724
C AIRCOMPRESSOR: TEST FOR 0, FIND MACHINE INPUT, FIND % TOTAL MACHINE 3725
C IF (COMEFF.EQ.0. .OR. NUMTUB.EQ.0.) THEN                     3726
C   COMWHR = 0.                                                3727
C ELSE                                                          3728
C   COMIN = (HPCOM * WATHP * HRSOCM) / COMEFP                3729
C   PERCOM = (TUBTNK(S) * NUMTNK(S)) / NUMTUB                3730
C   COMWHR = COMIN * PERCOM * DAYSTG(S)                        3731
C ENDIF                                                        3732
C -----                                                    3733
C PUMPING: TEST FOR 0, FIND MACHINE INPUT, FIND MACHINE INPUT PER    3734
C LITRE, AND FIND INPUT FOR ALL REQUIRED WATER                 3735
C IF (PMPGPM.EQ.0. .OR. PMPPEFF.EQ.0.) THEN                    3736
C   PMPWHR = 0.                                                3737
C ELSE                                                          3738
C   PMPIN = (HPMP * WATHP) / PMPPEFF                          3739
C   PMPINL = PMPIN / (PMPGPM * PMPCON)                         3740
C   PMPWHR = PMPINL * H2OTNK(S) * NUMTNK(S)                   3741
C ENDIF                                                        3742
C -----                                                    3743
C AIR CONDITIONING                                            3744
C AIRWHR = (WATAIR * NUMAIR(S) * HRSAIR(MJ,S) / STGRM(S)) * DAYSTG(S) 3745
C -----                                                    3746
C SUM ALL WHR , CHANGE TO KWHR                                3747
C TKWHR(S) = (LITWHR(S) + AUTWHR + COMWHR + PMPWHR + AIRWHR) * .001 3748
C -----                                                    3749
C FIND KWHR PLUS INNOCULATION - CHECK FOR FIRST STAGE        3750
C IF (S.EQ.1) THEN                                           3751
C   INQKWH = 0.                                                3752
C   TTKWHR(S) = TKWHR(S)                                       3753
C ELSE                                                          3754
C   INQKWH = INQTNK(S) * NUMTNK(S) * TTKWHL(S-1)             3755
C   TTKWHR(S) = INQKWH + TKWHR(S)                               3756
C ENDIF                                                        3757
C -----                                                    3758
C FIND KWHR PER LITRE                                         3759
C IF (TLITRE(S) .LE. 0.) STOP 'ERROR IN ELEC; TOTAL LITRES = 0' 3760
C TTKWHL(S) = TTKWHR(S) / TLITRE(S)                           3761
C -----                                                    3762
C RETURN                                                      3763
C END                                                          3764
C -----                                                    3765
C -----                                                    3766
C -----                                                    3767
C -----                                                    3768
C -----                                                    3769
C -----                                                    3770

C SUBROUTINE FUEL(STAGE, TIME)
C -----
C ORGNSM:
C -----
C SUBROUTINE VARIES WITH ORGANISM
C VARIABLES CHANGE, BUT RETURNED ONLY TO PRICES
C CURRENT ORGANISM: BIVALVES
C -----
C STRUCTURE: COMPUTES LITRES OF OIL USED PER STAGE OF ALGAE PRODUCTION
3781 3782
3783 3784
3785 3786
3787 3788
3789 3790
3791 3792
3793 3794
3795 3796
3797 3798
3799 3799
3800 3800
3801 3801
3802 3802
3803 3803
3804 3804
3805 3805
3806 3806
3807 3807
3808 3808
3809 3809
3810 3810
3811 3811
3812 3812
3813 3813
3814 3814
3815 3815
3816 3816
3817 3817
3818 3818
3819 3819
3820 3820
3821 3821
3822 3822
3823 3823
3824 3824
3825 3825
3826 3826
3827 3827
3828 3828
3829 3829
3830 3830
3831 3831
3832 3832
3833 3833
3834 3834

C 1. COMMON BLOCK VARIABLES
C   INCLUDE (IPAR1)
C   INCLUDE (ICNS$)
C 2. ARGUMENT VARIABLES
C   INTEGER STAGE
C 3. LOCAL VARIABLES
C   INTEGER S
C NAME DESCRIPTION OF LOCAL VARIABLES AND ARGUMENTS
C S STAGE
C S TRANSLATES ARGUMENT FOR CURRENT STAGE
C STAGE CURRENT STAGE OF ALGAE BEING PRODUCED
C -----
C TRANSLATE ARGUMENTS INTO LOCAL VARIABLES
C S = STAGE
C *** NOTE: BECAUSE THIS IS ONLY A DUMMY ROUTINE, ACCURATE ALGAE
C COSTS CANNOT BE COMPUTED FOR MONTHS IN WHICH HEATING
C WOULD PLAY A SIGNIFICANT ROLE
C ***
C FUEL CANNOT BE COMPUTED AT THIS TIME. SET VARIABLE = 0.
C TOLL(S) = 0.
C TTOIL(S) = 0.
C RETURN
C END
C -----
C SUBROUTINE LABOR(STAGE, TIME)
C -----
C ORGNSM:
C -----
C SUBROUTINE VARIES WITH ORGANISM
C VARIABLES CHANGE, RETURNED ONLY TO PRICES
C CURRENT ORGANISM: BIVALVES
C STRUCTURE: CALCULATES THE QUANTITY OF LABOR TYPE L USED
C PER STAGE, PER LITRE OF ALGAE PRODUCTION
C 1. COMMON BLOCK VARIABLES
C   INCLUDE (IPAR1)
C   INCLUDE (ICNS$)
C 2. ARGUMENT VARIABLES

```

```

3835 C 3. LOCAL VARIABLES, TIME
3836 C 3. LOCAL VARIABLES
3837 C 1. COMMON BLOCK VARIABLES
3838 C INCLUDE (IPARI)
3839 C INCLUDE (ICN5$)
3840 C 2. ARGUMENT VARIABLES
3841 C 3. INTEGER STAGE, TIME
3842 C 3. LOCAL VARIABLES
3843 C INTEGER G, S, MJ
3844 C REAL IQQOTY(S)
3845 C NAME DESCRIPTION OF LOCAL VARIABLES AND ARGUMENTS
3846 C -----
3847 C IQQOTY GOOD TYPE G REQUIRED FOR INNOCULATION
3848 C G LOOP COUNTER FOR GOODS
3849 C MJ TRANSLATES ARGUMENT TIME FOR USE IN SUBROUTINE
3850 C S TRANSLATES ARGUMENT STAGE FOR USE IN SUBROUTINE
3851 C STAGE ARGUE: CURRENT ALGAE STAGE
3852 C TIME ARGUE: MONTH THAT CURRENT STAGE IS IN
3853 C -----
3854 C TRANSLATE ARGUMENTS INTO LOCAL VARIABLES
3855 C S = STAGE
3856 C MJ = TIME
3857 C TALLAB(S) = 0.
3858 C DO 10 L = 1, NUMLAB
3859 C TLABY(L,S) = HRLABW(L)/7. *DAYSTG(S)/FLOAT(NUMSTG) +HRLABS(L,S)
3860 C 3861
3862 C ADD INNOCULATION - TEST FOR FIRST STAGE
3863 C IF (S.EQ.1) THEN
3864 C IQLATY(L) = 0.
3865 C TTLATY(L,S) = TLABY(L,S)
3866 C ELSE
3867 C IF (TLITRE(S-1).LE.0.) STOP 'ERROR LABOR: TLITRE_0'
3868 C IQLATY(L) = INQNK(S)*NUMNK(S) * (TTLATY(L,S-1)/TLITRE(S-1))
3869 C TTLATY(L,S) = TLABY(L,S) + IQLATY(L)
3870 C ENDIF
3871 C 3872
3873 C SUM ALL TYPES L
3874 C ALLAB(S) = TLABY(L,S) + ALLAB(S)
3875 C TALLAB(S) = TTLATY(L,S) + TALLAB(S)
3876 C 10 CONTINUE
3877 C 3878
3879 C TOTAL ALL LABOR USED PER LITRE
3880 C IF (TLITRE(S).LE.0.) STOP 'ERROR LABOR2: TLITRE_0'
3881 C TALLAL(S) = TALLAB(S) / TLITRE(S)
3882 C RETURN
3883 C END
3884 C
3885 C SUBROUTINE GOODS(STAGE,TIME)
3886 C -----
3887 C ORGNSM:
3888 C -----
3889 C SUBROUTINE VARIES WITH ORGANISM
3890 C VARIABLE CHANGE, RETURNED ONLY TO PRICES
3891 C CURRENT ORGANISM: BIVALVES
3892 C 3893
3894 C 3895
3896 C 3897
3898 C 3899
3899 C 3900
3900 C 3901
3901 C 3902
3902 C 3903
3903 C 3904
3904 C 3905
3905 C 3906
3906 C 3907
3907 C 3908
3908 C 3909
3909 C 3910
3910 C 3911
3911 C 3912
3912 C 3913
3913 C 3914
3914 C 3915
3915 C 3916
3916 C 3917
3917 C 3918
3918 C 3919
3919 C 3920
3920 C 3921
3921 C 3922
3922 C 3923
3923 C 3924
3924 C 3925
3925 C 3926
3926 C 3927
3927 C 3928
3928 C 3929
3929 C 3930
3930 C 3931
3931 C 3932
3932 C 3933
3933 C 3934
3934 C 3935
3935 C 3936
3936 C 3937
3937 C 3938
3938 C 3939
3939 C 3940
3940 C 3941
3941 C 3942
3942 C 3943
3943 C 3944
3944 C 3945
3945 C 3946
3946 C 3947
3947 C 3948
3948 C 3949
3949 C 3950
3950 C 3951
3951 C 3952
3952 C 3953
3953 C 3954
3954 C 3955
3955 C 3956
3956 C 3957
3957 C 3958
3958 C 3959
3959 C 3960
3960 C 3961
3961 C 3962

```

```

C SUBROUTINE: DETERMINES COST OF GIVEN ALGAE INPUT QUANT. FOR PRICE N=3963
3964
C 1. COMMON BLOCK VARIABLES
3965 INCLUDE (IPAR1)
3966 INCLUDE (ICN5$)
C 2. ARGUMENT VARIABLES
3967 INTEGER TIME, STAGE
C 3. LOCAL VARIABLES
3968 INTEGER G, I, L, S
3969 REAL CLABTY, TCGOTY, CGODTY
C NAME DESCRIPTION OF LOCAL VARIABLES AND ARGUMENTS
3970 -----
C CGODTY COST OF GOOD, TYPE G, LESS INNOCULATION
C CLABTY COST OF LABOR, TYPE L, LESS INNOCULATION
C G LOOP COUNTER FOR GOODS
C I LOOP COUNTER FOR INPUTS
C L LOOP COUNTER FOR LABOR TYPES
C S LOOP COUNTER FOR STAGES
C STAGE ARGUE: CURRENT STAGE OF ALGAE
C TCGOTY TOTAL COST OF GOODS, TYPE G, LESS INNOCULATION
C TIME ARGUE: MONTH THAT CURRENT STAGE IS IN
3971 -----
C TRANSLATE ARGUMENT INTO LOCAL VARIABLES
3972 S = STAGE
3973 M = TIME
C FIND COST OF EACH INPUT TYPE LESS INNOCULATION, FILE INTO ARRAY
3974 C1NP(1,S) = TKWHR(S) * PKWHR
3975 C2NP(2,S) = TOIL(S) * POIL
3976 C3NP(3,S) = 0.
3977 DO 10 L = 1, NUMLAB
3978 CLABTY = TLABTY(L,S) * PLABOR(L)
3979 C1NP(3,S) = CLABTY + C1NP(3,S)
3980 CONTINUE
3981 C1NP(4,S) = 0.
3982 DO 20 G = 1, NUMGOD
3983 CGODTY = TCGOTY(G,S) * PGOD(G)
3984 C1NP(4,S) = CGODTY + C1NP(4,S)
3985 CONTINUE
C FIND THE COST OF EACH INPUT TYPE PLUS INNOCULATION
3986 TC1NP(1,S) = TTKWHR(S) * PKWHR
3987 TC1NP(2,S) = TTOIL(S) * POIL
3988 DO 30 L = 1, NUMLAB
3989 TLABTY(L) = TLABTY(L,S) * PLABOR(L)
3990 TC1NP(3,S) = TLABTY(L) + TC1NP(3,S)
3991 CONTINUE
3992 TC1NP(4,S) = 0.
3993 DO 40 G = 1, NUMGOD
3994 TCGOTY = TCGOTY(G,S) * PGOD(G)
3995 TC1NP(4,S) = TCGOTY + TC1NP(4,S)
3996 CONTINUE
C TOTAL COSTS
4000 TCALIN(S) = 0.
4001 TCALIN(S) = 0.
4002 DO 50 I = 1,4
4003 COST PER LITRE OF EACH INPUT
4004 TCINPL(I,S) = TC1NP(I,S) / TLITRE(S)
4005
C SUBROUTINE: DETERMINES COST OF GIVEN ALGAE INPUT QUANT. FOR PRICE N=3963
4027
C SUM OF ALL INPUT COSTS
4028 CALIN(S) = C1NP(I,S) + CALIN(S)
4029
C SUM OF ALL INPUT COSTS PLUS INNOCULATION
4030 TCALIN(S) = TC1NP(I,S) + TCALIN(S)
4031
C CONTINUE
4032
C TOTAL COST PLUS INNOCULATION - PER LITRE, CELL FOOD
4033 TCLITR(S) = TCALIN(S) / TLITRE(S)
4034
C CHECK FOR 0
4035 IF (TCELL(S) .EQ. 0.) THEN
4036 TCELL(S) = 0.
4037
C TCELL(S) = TCALIN(S) / TCELL(S)
4038
C ENDIF
4039
C RETURN
4040
C END
4041
C SUBROUTINE WRT$5(MONBEG, LASSTG, WRT)
4042 -----
C ORGNM:
4043
C SUBROUTINE VARIES WITH ORGANISM
4044
C VARIABLES CHANGE, RETURNED ONLY TO PRICES
4045
C CURRENT ORGANISM: BIVALVES
4046
C STRUCTURE: OUTPUTS DATA BY THE MONTH IN WHICH ALGAE STAGE 1 IS BEGUN
4047
C 1. COMMON BLOCK VARIABLES
4048 INCLUDE (IPAR1)
4049 INCLUDE (ICBLK1)
4050 INCLUDE (ICN5$)
4051 INCLUDE (ICUNIT)
4052 INCLUDE (ICBLKO)
4053 INCLUDE (ICMAIN)
C 2. ARGUMENT VARIABLES
4054 INTEGER LASSTG, WRT
4055 CHARACTER*9 MONBEG
4056 INTEGER S, W
C NAME DESCRIPTION OF LOCAL VARIABLES AND ARGUMENTS
4057 -----
C LASSTG ARGUE: LAST STAGE OF ALGAE THAT CALCULATIONS ARE DONE FOR
4058 MONBEG ARGUE: MONTH IN WHICH CURRENT BATCH OF ALGAE WAS BEGUN
C S LOOP COUNTER FOR STAGES
4059 WRT ARGUE: UNIT NUMBER TO WRITE
C W UNIT NUMBER TO WRITE
4060 -----
C TRANSLATE UNIT NUMBERS TO NUMBERS USED IN ORGNM
4061 W = WRT
4062

```

```

1491 WRITE(W,1101) NAME, DATE
1492 WRITE(W,101) DATE2
1493 WRITE(W,102) MONBEG
1494 WRITE(W,1103) (S,
1495 S = 1, NUMSTG)
1496 WRITE(W,103) (MONAVA( S), S = 1, NUMSTG)
1497 WRITE(W,101) (TLTIRE( S), S = 1, LASSTG)
1498 WRITE(W,104) (LFOOD( S), S = 1, LASSTG)
1499 WRITE(W,105) (TCELL( S), S = 1, LASSTG)
1500 WRITE(W,108) (TCALIN( S), S = 1, LASSTG)
1501 WRITE(W,109) (TCLTR(S), S = 1, LASSTG)
1502 WRITE(W,1092) (TCELL( S), S = 1, LASSTG)
1503 WRITE(W,110) (CALIN( S), S = 1, LASSTG)
1504 WRITE(W,113) (TCINP(1,S), S = 1, LASSTG)
1505 WRITE(W,114) (TCINPL(1,S), S = 1, LASSTG)
1506 WRITE(W,115) (CINP(1,S), S = 1, LASSTG)
1507 WRITE(W,116) (TTKWR(1,S), S = 1, LASSTG)
1508 WRITE(W,117) (TKWHL( S), S = 1, LASSTG)
1509 WRITE(W,118) (TKWHR( S), S = 1, LASSTG)
1510 WRITE(W,119) (TCINP(2,S), S = 1, LASSTG)
1511 WRITE(W,120) (TCINPL(2,S), S = 1, LASSTG)
1512 WRITE(W,121) (CINP(2,S), S = 1, LASSTG)
1513 WRITE(W,122) (TTOIL( S), S = 1, LASSTG)
1514 WRITE(W,123) (TTOILL( S), S = 1, LASSTG)
1515 WRITE(W,124) (TOIL( S), S = 1, LASSTG)
1516 WRITE(W,125) (TCINP(3,S), S = 1, LASSTG)
1517 WRITE(W,126) (TCINPL(3,S), S = 1, LASSTG)
1518 WRITE(W,127) (CINP(3,S), S = 1, LASSTG)
1519 WRITE(W,128) (TALLA( S), S = 1, LASSTG)
1520 WRITE(W,129) (TALLAL( S), S = 1, LASSTG)
1521 WRITE(W,130) (ALLAR( S), S = 1, LASSTG)
1522 WRITE(W,131) (TCINP(4,S), S = 1, LASSTG)
1523 WRITE(W,132) (TCINPL(4,S), S = 1, LASSTG)
1524 WRITE(W,133) (CINP(4,S), S = 1, LASSTG)
1525
1526 1101 FORMAT ('1///', A10, ' COST SIMULATION MODEL: ALGAE PRICE ',
1527 2 ' COMPUTATIONS', /, '59('-'/' DATE: ', A25/
1528 3 ' OUTPUT GENERATED')
1529 101 FORMAT ('0', DATE OF PRICE COMPUTATIONS: ', A///)
1530 102 FORMAT ('0', 'BATCH BEGUN: ', 2X, A/)
1531 1103 FORMAT ('0', T43, 6(3X, I2, 'STAGE', 5X) /, /, T43, 6(3X, I2, 'STAGE', 5X) //)
1532
1533 103 FORMAT ('0', 'ALGAE AVAILABLE FOR USE IN MONTH',
1534 B T41, 6(4X, A9, 2X) /, /, T41, 6(4X, A9, 2X))
1535 1031 FORMAT ('0', '*TOTAL LITRES PRODUCED',
1536 B T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1537 104 FORMAT ('0', 'LITRES OF ALGAE AVAILABLE',
1538 B T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1539 105 FORMAT ('0', 'CELLS OF ALGAE PRODUCED',
1540 B T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1541 108 FORMAT ('0', '*TOTAL COST OF STAGE',
1542 B T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1543 109 FORMAT ('0', 'IX, TOTAL COST PER LITRE OF ALGAE',
1544 B T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1545 1092 FORMAT ('0', 'IX, TOTAL COST PER CELL OF ALGAE',
1546 B T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1547 110 FORMAT ('0', 'IX, TOTAL COST OF STAGE, LESS INNOCULATION',
1548 B T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1549 113 FORMAT ('0', '*TOTAL COST OF ELECTRICITY',
1550 B T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1551
1552 4091 T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1553 4092 T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1554 4093 T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1555 4094 T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1556 4095 T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1557 4096 T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1558 4097 T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1559 4098 T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1560 4099 T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1561 4100 T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1562 4101 T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1563 4102 T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1564 4103 T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1565 4104 T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1566 4105 T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1567 4106 T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1568 4107 T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1569 4108 T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1570 4109 T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1571 4110 T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1572 4111 T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1573 4112 T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1574 4113 T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1575 4114 T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1576 4115 T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1577 4116 T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1578 4117 T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1579 4118 T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1580 4119 T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1581 4120 T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1582 4121 T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1583 4122 T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1584 4123 T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1585 4124 T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1586 4125 T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1587 4126 T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1588 4127 T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1589 4128 T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1590 4129 T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1591 4130 T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1592 4131 T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1593 4132 T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1594 4133 T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1595 4134 T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1596 4135 T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1597 4136 T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1598 4137 T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1599 4138 T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1600 4139 T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1601 4140 T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1602 4141 T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1603 4142 T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1604 4143 T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1605 4144 T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1606 4145 T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1607 4146 T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1608 4147 T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1609 4148 T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1610 4149 T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1611 4150 T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1612 4151 T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1613 4152 T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1614 4153 T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1615 4154 T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1616
1617 114 FORMAT ('0', T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1618 115 FORMAT ('0', T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1619 116 FORMAT ('0', T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1620 117 FORMAT ('0', T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1621 118 FORMAT ('0', T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1622 119 FORMAT ('0', T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1623 120 FORMAT ('0', T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1624 121 FORMAT ('0', T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1625 122 FORMAT ('0', T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1626 123 FORMAT ('0', T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1627 124 FORMAT ('0', T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1628 125 FORMAT ('0', T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1629 126 FORMAT ('0', T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1630 127 FORMAT ('0', T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1631 128 FORMAT ('0', T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1632 129 FORMAT ('0', T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1633 130 FORMAT ('0', T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1634 131 FORMAT ('0', T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1635 132 FORMAT ('0', T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1636 133 FORMAT ('0', T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1637 134 FORMAT ('0', T41, 6(G13.4, 2X) /, /, T41, 6(G13.4, 2X))
1638 135 RETURN
1639 136 END
1640
1641 SUBROUTINE PRICE6 (WRT, RED, HREAD, RUN2, RUN3)
1642 RETURN
1643 END
1644
1645 SUBROUTINE BODY
1646 -----
1647 C

```

```

C ORGNSM: 4219
C ----- 4220
C SUBROUTINE DOES NOT VARY WITH ORGANISM 4221
C VARIABLES DO NOT CHANGE 4222
C 4223
C 4224
C 4225
C 4226
C 4227
C 4228
C 4229
C 4230
C 4231
C 4232
C 4233
C 4234
C 4235
C 4236
C 4237
C 4238
C 4239
C 4240
C 4241
C 4242
C 4243
C 4244
C 4245
C 4246
C 4247
C 4248
C 4249
C 4250
C 4251
C 4252
C 4253
C 4254
C 4255
C 4256
C 4257
C 4258
C 4259
C 4260
C 4261
C 4262
C 4263
C 4264
C 4265
C 4266
C 4267
C 4268
C 4269
C 4270
C 4271
C 4272
C 4273
C 4274
C 4275
C 4276
C 4277
C 4278
C 4279
C 4280
C 4281
C 4282

C STRUCTURE: OPTION TO RUN FOR LESS THAN FULL PRODUCTION SEASON
C LOOPS THROUGH EACH MONTH OF PRODUCTION
C LOOPS THROUGH EACH STAGE
C DETERMINES PERIODS IN EACH STAGE
C LOOPS THROUGH EACH PERIOD OF EACH STAGE
C UPDATES CURRENT DAYS AND MONTH
C CALLS NUMBER
C CALLS INPUT QUANTITY SUBROUTINES (QUANTN)
C CALLS COST (COSTJ)
C CALLS SUBROUTINE TO OUTPUT (WRTOU, WOUTO)

C 1. PARAMETERS
C INCLUDE (IPARL)
C 2. ARGUMENTS
C INTEGER INOW, JNOW, LASTJ, MNOW
C REAL DAYNOW, WKNOW
C CHARACTER * 10 MBEGUN
C 3. COMMON BLOCKS
C INCLUDE (ICMAIN)
C INCLUDE (ICBLK1)
C INCLUDE (ICBLK2)
C INCLUDE (ICUNIT)
C 4. LOCAL VARIABLES
C INTEGER ADDTOM, I, J, M, T
C CHARACTER * 1 QUEST
C -----
C NAME DESCRIPTION OF LOCAL VARIABLES ARGUMENTS
C ADDTOM UPDATE MONTH THAT PRODUCTION IS OCCURRING IN
C DAYNOW ARGUMENT: ELAPSED DAYS TO END OF PERIOD
C I LOOP COUNTER FOR STAGES
C INOW ARGUMENT: CURRENT STAGE THAT J IS IN
C J LOOP COUNTER FOR PERIODS
C JNOW ARGUMENT: CURRENT PERIOD COMPUTING FOR
C LASTJ ARGUMENT: LAST J THAT CALCULATIONS ACCOMPLISHED
C M LOOP COUNTER FOR MONTHS
C MBEGUN MONTH THAT BATCH IS BEGUN FOR CURRENT CALCULATIONS
C MNOW ARGUMENT: CURRENT MONTH THAT J IS OCCURRING IN
C QUEST RUN FOR LESS THAN THE FULL PRODUCTION SEASON?
C T LOOP COUNTER FOR TYPES
C WKNOW ARGUMENT: ELAPSED WEEKS AT END OF J
C -----
C OPTION TO RUN PROGRAM FOR LESS MONTHS THAN FULL PRODUCTION SEASON
WRITE(WI,20000)
READ(RI,1000) QUEST
IF (QUEST.EQ.'Y') THEN
WRITE(WI,21000)
READ(RI,*) M1,M2
ELSE
M1 = BEGINM
M2 = ENDM
ENDIF
1000 FORMAT (A1)
20000 FORMAT('////' DO YOU WANT TO RUN THE PROGRAM FOR LESS THAN THE

```

```

2 21000 FORMAT(' ' FULL PRODUCTION SEASON? ' ' , 'Y OR N ' ) 4283
2 ' ' TYPE THE NUMBER OF THE MONTH TO BEGIN COMPUTATION ' , 4284
2 'AND THE MONTH TO END ' ' , 'TYPE TWO INTEGERS ' ) 4285
4286
4287
4288
4289
4290
4291
4292
4293
4294
4295
4296
4297
4298
4299
4300
4301
4302
4303
4304
4305
4306
4307
4308
4309
4310
4311
4312
4313
4314
4315
4316
4317
4318
4319
4320
4321
4322
4323
4324
4325
4326
4327
4328
4329
4330
4331
4332
4333
4334
4335
4336
4337
4338
4339
4340
4341
4342
4343
4344
4345
4346

C COMPUTE FOR A NEW BATCH ON THE BEGINNING OF EACH MONTH
DO 100 N=MI,M2

C NAME THE MONTH THAT PRODUCTION BEGINS
MBEGUN = NAMEM(M)

C LOOP THROUGH STAGES
DAYNOW = 0.
DO 200 I=1,NOOFI
INOW = I

C DEFINE ALL PERIODS SEQUENTIALLY THROUGH ALL STAGES

C OPTION 1: SKIP ALL I BEFORE 1ST STAGE OF INTEREST - 1
IF (OPTION.EQ.1 .AND. I.LT.(BEGINJ-1)) THEN
BEGINJ(I) = 0
ENDJ(I) = 0
JPERI(I) = 0
DAYSJ(I) = 0.
GOTO 200

C OPTION 1: DEFINE 1ST J TO BE STAGE BEFORE 1ST STG. OF INTER4311
ELSEIF (OPTION.EQ.1 .AND. I.EQ.(BEGINJ-1)) THEN
BEGINJ(I) = 1
ENDJ(I) = 1
JPERI(I) = 1
DAYSJ(I) = DAYOPI

C OPTION 2,3: DEFINE 1ST PERIOD TO BE 1ST PERIOD OF I=1
ELSEIF (OPTION.NE.1 .AND. I.EQ.(BEGINJ)) THEN
BEGINJ(I) = 1
ENDJ(I) = 1
JPERI(I) = 1
DAYSJ(I) = 1

C OPTIONS ALL: DEFINE ALL SUBSEQUENT PERIODS AFTER THE 1ST
PERIOD AS ACCUMULATING THROUGH THE STAGES
ELSE
BEGINJ(I) = ENDJ(I-1) + 1
ENDJ(I) = BEGINJ(I) + JPERI(I) - 1
DAYSJ(I) = DAYSJ(I) / JPERI(I)
ENDIF

C LOOP THROUGH EACH PERIOD, BY STAGE
DO 300 J=BEGINJ(I),ENDJ(I)
JNOW = J
LASTJ = J

C ACCUMULATE DAYS AND INCREMENT TIME VARIABLES
DAYNOW = DAYNOW + DAYSJ(I)
WKNOW = DAYNOW / 7.
ADDTOM = IFIX(DAYNOW-1) / 30
MNOW = M + ADDTOM

C ALLOW FOR FULL YEAR CYCLE
(USE THE FACT THAT M IS AN INTEGER TO SUBTRACT PROPERLY) 4345
IF (ENDM-BEGINM.EQ.11) MNOW=MNOW-(12*(MNOW-1)/12)4346

```

```

4347 MOFJ (J) = MNOW
4348 IOFJ (J) = I
4349 DAYUPJ(J) = DAYNOW
4350 WKSUPJ = WKSNOW
4351
4352 HALT IF BEYOND MONTH OF PRODUCTION
4353 IF (MNOW .GT. ENDM) THEN
4354 PASEDM(J) = 1
4355 GOTO 250
4356
4357
4358 C COMPUTE QUANTITY OF N, COSTS, AND NUMBER OF ANIMALS
4359 ELSE
4360 PASEDM(J) = 0
4361
4362 C COMPUTE NUMBER AND SIZE OF ORGANISMS IN J
4363 CALL NUMBER(INOW, JNOW, WKSNOW, MNOW, DAYNOW)
4364 CALL SIZE (INOW, JNOW, WKSNOW, MNOW, DAYNOW)
4365
4366 C COMPUTE QUANTITY OF EACH INPUT, N, USED IN J
4367 IF (I .GE. BEGINI) THEN
4368 DO 400 N=1, NOOFN
4369 CALL SUBR. TO COMPUTE QUANTITY IF CHOSEN
4370 IF (OPNEW(I, N) .EQ. 'C') THEN
4371 IF (N.EQ.1) CALL QUANT1(INOW, JNOW, WKSNOW, MNOW, DAYNOW)
4372 IF (N.EQ.2) CALL QUANT2(INOW, JNOW, WKSNOW, MNOW, DAYNOW)
4373 IF (N.EQ.3) CALL QUANT3(INOW, JNOW, WKSNOW, MNOW, DAYNOW)
4374 IF (N.EQ.4) CALL QUANT4(INOW, JNOW, WKSNOW, MNOW, DAYNOW)
4375 IF (N.EQ.5) CALL QUANT5(INOW, JNOW, WKSNOW, MNOW, DAYNOW)
4376 IF (N.EQ.6) CALL QUANT6(INOW, JNOW, WKSNOW, MNOW, DAYNOW)
4377
4378 QUANTITY ENTERED, THEN QUANT. IS SAME EACH PERIOD
4379 ELSEIF (OPNEW(I, N) .EQ. 'E') THEN
4380 DO 500 T=1, NOOFT(N)
4381 TQUANJ(J, N, T) = TQUANI(I, N, T, MOFJ(J)) / JPERI(I)
4382 CONTINUE
4383 ENDF
4384 CONTINUE
4385
4386 C COMPUTE COST OF INPUTS IN EACH J
4387 ENDF
4388 CALL COSTJ(INOW, JNOW, WKSNOW, MNOW, DAYNOW)
4389 ENDF
4390
4391 CONTINUE
4392
4393 CONTINUE
4394
4395 CONTINUE
4396
4397 CALL WRTOU(MBEGUN, LASTJ, WI)
4398 IF (QOUTQ(1) .EQ. 'Y') CALL WOUTN1(MBEGUN, LASTJ, WI)
4399 IF (QOUTQ(2) .EQ. 'Y') CALL WOUTN2(MBEGUN, LASTJ, WI)
4400 IF (QOUTQ(3) .EQ. 'Y') CALL WOUTN3(MBEGUN, LASTJ, WI)
4401 IF (QOUTQ(4) .EQ. 'Y') CALL WOUTN4(MBEGUN, LASTJ, WI)
4402 IF (QOUTQ(5) .EQ. 'Y') CALL WOUTN5(MBEGUN, LASTJ, WI)
4403 IF (QOUTQ(6) .EQ. 'Y') CALL WOUTN6(MBEGUN, LASTJ, WI)
4404
4405 CONTINUE
4406 WRITE(WI, 30000)
4407
4408 C OPTION TO SEE OUTPUT GENERATED FOR LAST BATCH
4409 WRITE(WI, 40000)
4410
4411
4412 READ (RI, 1000) QUEST
4413 IF (QUEST .EQ. 'Y') THEN
4414 CALL WRTOU(MBEGUN, LASTJ, WI)
4415 IF (QOUTQ(1) .EQ. 'Y') CALL WOUTN1(MBEGUN, LASTJ, WI)
4416 IF (QOUTQ(2) .EQ. 'Y') CALL WOUTN2(MBEGUN, LASTJ, WI)
4417 IF (QOUTQ(3) .EQ. 'Y') CALL WOUTN3(MBEGUN, LASTJ, WI)
4418 IF (QOUTQ(4) .EQ. 'Y') CALL WOUTN4(MBEGUN, LASTJ, WI)
4419 IF (QOUTQ(5) .EQ. 'Y') CALL WOUTN5(MBEGUN, LASTJ, WI)
4420 IF (QOUTQ(6) .EQ. 'Y') CALL WOUTN6(MBEGUN, LASTJ, WI)
4421 ENDF
4422
4423 30000 FORMAT(' /// ***** COMPUTATIONS COMPLETED *****')
4424
4425 40000 FORMAT(' /// DO YOU WANT TO SEE THE COMPUTATIONS FOR THE LAST ',
4426 2, ' BATCH?')
4427
4428 RETURN
4429 ENDF
4430
4431
4432 SUBROUTINE NUMBER(INOW, JNOW, WKSNOW, MNOW, DAYNOW)
4433
4434 C ORGNSM:
4435
4436 C SUBROUTINE VARIES WITH ORGANISM
4437 C VARIABLES MUST BE RETURNED:
4438 C NUMBER OR ORGANISMS IN J: ONOOFJ(I)
4439 C NUMBER OF ORGANISM UP TO J: ONOUPJ(J)
4440 C AGE OF ORGANISM UP TO END J: AGEUPJ(J)
4441 C CURRENT ORGANISM: BIVALVES
4442
4443 C STRUCTURE: COMPUTES AND RETURNS THE NUMBER OF ORGANISMS REMAINING AT
4444 THE END OF EACH PERIOD, ONOOFJ, AND THE NUMBER OF AVERAGE
4445 ORGANISMS DURING EACH PERIOD, ONOOFJ
4446
4447 C 1. PARAMETERS
4448 INCLUDE (IPAR1)
4449
4450 C 2. ARGUMENTS AND FUNCTIONS
4451 INTEGER INOW, JNOW, MNOW
4452 REAL AGENO, WKSNOW, DAYNOW
4453
4454 C 3. COMMON BLOCKS
4455 INCLUDE (ICMAIN)
4456 INCLUDE (ICORG)
4457 INCLUDE (ICBLKO)
4458
4459 C 4. LOCAL VARIABLES
4460 INTEGER D, DCNTR, I, J, M, VALUE
4461 REAL DAY, DMK, TNO, WKS
4462
4463 C NAME DESCRIPTION OF LOCAL VARIABLES
4464
4465 C AGENO FUNCTION: RELATES AGE/SURVIVAL, CALLED EACH DAY
4466 C D LOOP COUNTER FOR DAYS
4467 C DAY TRANSLATE ARGUMENT TO LOCAL DAYS
4468 C DAYNOW ARGUE: ELAPSED DAYS TO END OF CURRENT J
4469 C DCNTR NO. OF DAYS LOOPED THROUGH IN PERIOD
4470 C DMK TRANSLATE DAY LOOP INTO WEEKS
4471 C I TRANSLATE INOW INTO CURRENT STAGE
4472
4473
4474

```



```

C INOW ARGUE: STAGE THAT CURRENT PERIOD IS IN
C J TRANSLATES JNOW FOR CURRENT PERIOD
C M ARGUE: CURRENT PERIOD BEING CALCULATED
C M TRANSLATE MNOW FOR CURRENT MONTH
C MNOW ARGUE: MONTH THAT CURRENT J IS IN
C TNO SUM OF ALL ORGANISMS EXISTING EACH DAY OF PERIOD
C VALUE VALUE TO DECREASE REIMANN SUM. 1ST TERM OF SUM
C WKS TRANSLATE WKSNOV FOR ELAPSED WEEKS
C WKSNOV ARGUE: ELAPSED WEEKS FOR CURRENT J
C
C TRANSLATE LOCAL VARIABLES
  I = INOW
  J = JNOW
  WKS = WKSNOV
  M = MNOW
  DAY = DAYNOW
C
C ALGORITHM DEPENDS ON TYPE OF ORGANISM
IF (OTYPE .EQ. 1 .OR. OTYPE .EQ. 2) THEN
  WHEN I=CONDITIONING, ASSUME NO DEATH RATE AND ANIMALS ALL AT
  TWO YEARS OLD SO AS NOT TO EXCEED MAXIMUM OF AGE/AGE EQUATION
  IF (I .EQ. 1) THEN
    AGEUPJ(J) = 365. * 2.
    ONOOFJ(J) = NOO11
    ONOUPJ(J) = NOO11
  C
  ALL OTHER STAGES, COMPUTE NUMBER OF ORGANISMS SURVIVING AT EACH4504
  DAY OF THE PERIOD BY CALLING FUNCTION AGENO. THEN FIND AVERAGE4505
  SURVIVAL OF THE PERIOD BY SUMMING USING THE TRAPEZOIDAL RULE
  ELSEIF (I .GT. 1) THEN
    DCNTR = 0
    TNO = 0.
    DO 100 D = (DAY-DAYSJ(I)-DAYSII+1.), (DAY-DAYSII+.49999)
      DMK = FLOAT(D) / 7.
      ONOUPJ(J) = AGENO(DMK)
      TNO = TNO + ONOUPJ(J)
      DCNTR = DCNTR + 1
    100 CONTINUE
  C
  COMPUTE NUMBER DURING DAY BEFORE PERIOD BEGINS
  DMK = (DAY-DAYSJ(I)-DAYSII) / 7.
  VALUE = AGENO(DMK)
  C
  INCREASE TOTAL BY 1/2 VALUE, DECREASE BY 1/2 LAST SUM TERM
  TNO = TNO + ((VALUE - ONOUPJ(J)) / 2.)
  C
  COMPUTE AVERAGE VALUE WHERE DELTA T IS ALWAYS = 1
  ONOOFJ(J) = TNO / FLOAT(DCNTR)
  C
  AGE OF ANIMAL AT END OF PERIOD DOESNT INCLUDE CONDITIONING
  AGEUPJ(J) = DAY - DAYSII
  ENDF
ELSE
  STOP 'ORGANISM TYPE INCORRECT IN SUBR. NUMBRB'
ENDF
RETURN
END

```

```

4475
4476
4477
4478
4479
4480
4481
4482
4483
4484
4485
4486
4487
4488
4489
4490
4491
4492
4493
4494
4495
4496
4497
4498
4499
4500
4501
4502
4503
4504
4505
4506
4507
4508
4509
4510
4511
4512
4513
4514
4515
4516
4517
4518
4519
4520
4521
4522
4523
4524
4525
4526
4527
4528
4529
4530
4531
4532
4533
4534
4535
4536
4537
4538

```

```

C ARGUE: STAGE THAT CURRENT PERIOD IS IN
C TRANSLATES JNOW FOR CURRENT PERIOD
C CURRENT PERIOD BEING CALCULATED
C MONTH THAT CURRENT J IS IN
C SUM OF ALL ORGANISMS EXISTING EACH DAY OF PERIOD
C VALUE TO DECREASE REIMANN SUM. 1ST TERM OF SUM
C TRANSLATE WKSNOV FOR ELAPSED WEEKS
C ELAPSED WEEKS FOR CURRENT J
C
C TRANSLATE LOCAL VARIABLES
  I = INOW
  J = JNOW
  WKS = WKSNOV
  M = MNOW
  DAY = DAYNOW
C
C ALGORITHM DEPENDS ON TYPE OF ORGANISM
IF (OTYPE .EQ. 1 .OR. OTYPE .EQ. 2) THEN
  WHEN I=CONDITIONING, ASSUME NO DEATH RATE AND ANIMALS ALL AT
  TWO YEARS OLD SO AS NOT TO EXCEED MAXIMUM OF AGE/AGE EQUATION
  IF (I .EQ. 1) THEN
    AGEUPJ(J) = 365. * 2.
    ONOOFJ(J) = NOO11
    ONOUPJ(J) = NOO11
  C
  ALL OTHER STAGES, COMPUTE NUMBER OF ORGANISMS SURVIVING AT EACH4504
  DAY OF THE PERIOD BY CALLING FUNCTION AGENO. THEN FIND AVERAGE4505
  SURVIVAL OF THE PERIOD BY SUMMING USING THE TRAPEZOIDAL RULE
  ELSEIF (I .GT. 1) THEN
    DCNTR = 0
    TNO = 0.
    DO 100 D = (DAY-DAYSJ(I)-DAYSII+1.), (DAY-DAYSII+.49999)
      DMK = FLOAT(D) / 7.
      ONOUPJ(J) = AGENO(DMK)
      TNO = TNO + ONOUPJ(J)
      DCNTR = DCNTR + 1
    100 CONTINUE
  C
  COMPUTE NUMBER DURING DAY BEFORE PERIOD BEGINS
  DMK = (DAY-DAYSJ(I)-DAYSII) / 7.
  VALUE = AGENO(DMK)
  C
  INCREASE TOTAL BY 1/2 VALUE, DECREASE BY 1/2 LAST SUM TERM
  TNO = TNO + ((VALUE - ONOUPJ(J)) / 2.)
  C
  COMPUTE AVERAGE VALUE WHERE DELTA T IS ALWAYS = 1
  ONOOFJ(J) = TNO / FLOAT(DCNTR)
  C
  AGE OF ANIMAL AT END OF PERIOD DOESNT INCLUDE CONDITIONING
  AGEUPJ(J) = DAY - DAYSII
  ENDF
ELSE
  STOP 'ORGANISM TYPE INCORRECT IN SUBR. NUMBRB'
ENDF
RETURN
END

```

```

4539
4540
4541
4542
4543
4544
4545
4546
4547
4548
4549
4550
4551
4552
4553
4554
4555
4556
4557
4558
4559
4560
4561
4562
4563
4564
4565
4566
4567
4568
4569
4570
4571
4572
4573
4574
4575
4576
4577
4578
4579
4580
4581
4582
4583
4584
4585
4586
4587
4588
4589
4590
4591
4592
4593
4594
4595
4596
4597
4598
4599
4600
4601
4602

```

```

SUBROUTINE SIZE(INOW,JNOW,WKSNOV,MNOW,DAYNOW)
C
C ORGNM:
C
C SUBROUTINE VARIES WITH ORGANISM
C VARIABLES THAT MUST BE RETURNED TO ORGNM:
C SIZE AT END OF PERIOD : SIZEOFJ(J)
C AVERAGE SIZE IN PERIOD : SIZEOFJ(J)
C CURRENT ORGANISM: BIVALVES
C
C STRUCTURE: COMPUTES AND RETURNS SIZE OF ORGANISM DURING EACH PERIOD
              AND AT END OF PERIOD USING THE FUNCTION AGESZ
C
C VARIABLE DECLARATIONS:
C 1. PARAMETERS
  INCLUDE (IPAR1)
  INTEGER INOW, JNOW, MNOW
  REAL DAYNOW, AGESZ, WKSNOV
C 2. ARGUMENTS AND FUNCTIONS
C 3. COMMON BLOCKS
  INCLUDE (ICMAIN)
  INCLUDE (ICORG)
  INCLUDE (ICBLKO)
  LOCAL VARIABLES
  INTEGER D, DCNTR, I, J, M, VALUE
  REAL DAY, DMK, TSZE, WKS
C
C NAME DESCRIPTION OF LOCAL AND ARGUMENT VARIABLES
C AGESZ FUNCTION: RELATES AGE/AGE/AGE. CALLED EACH DAY OF PERIOD
C D LOOP COUNTER FOR DAYS
C DAYNOW TRANSLATE DAYNOW FOR ELAPSED DAYS OF PERIOD
C DCNTR LAST DAY OF PERIOD LOOPED THROUGH
C I TRANSLATE INOW INTO STAGE OF CURRENT PERIOD
C J TRANSLATE JNOW FOR CURRENT PERIOD
C M TRANSLATE MNOW FOR MONTH THAT CURRENT PERIOD OCCURS IN
C MNOW ARGUE: MONTH THAT CURRENT PERIOD OCCURS IN
C TSZE SUM OF SIZE FOR ALL DAYS OVER THE PERIOD
C VALUE VALUE OF 1ST TERM TO SUBTRACT OFF OF THE REIMANN SUM
C WKS TRANSLATE WKSNOV FOR ELAPSED WEEKS
C WKSNOV ARGUE: ELAPSED WEEKS TO END OF CURRENT PERIOD
C
C TRANSLATE ARGUMENTS INTO LOCAL VARIABLES
  I = INOW
  J = JNOW
  WKS = WKSNOV
  M = MNOW
  DAY = DAYNOW

```

```

C ALGORITHM DEPENDS ON TYPE OF ORGANISM
  IF (OTYPE .EQ. 1 .OR. OTYPE .EQ. 2) THEN
C
C   CONDITIONING STAGE, ASSUME SIZE CONSTANT AND ALL AT 3 YEARS
  IF (I .EQ. 1) THEN
    DWK = AGEUPJ(J) / 7.
    SZEUPJ(J) = AGESZ(DWK)
    SZEUFJ(J) = SZEUPJ(J)
  ELSE
C
C   OTHERWISE, COMPUTE SIZE AT EACH DAY OF PERIOD BY CALLING
C   FUNCTION AGESZ, THEN FIND AVG. SIZE OF PERIOD BY SUMMING
C   USING THE TRAPEZOIDAL RULE
    ELSEIF (I .GT. 1) THEN
      DCNTR = 0
      TSZE = 0
      DO 100 D = (DAY-DAYSJ(I)-DAYSII+1.), (DAY-DAYSII+.4999)
        DWK = FLOAT(D) / 7.
        SZEUPJ(J) = AGESZ(DWK)
        TSZE = TSZE + SZEUPJ(J)
        DCNTR = DCNTR + 1
      CONTINUE
    100
C
C   COMPUTE SIZE DURING DAY BEFORE PERIOD BEGINS
    DWK = (DAY-DAYSJ(I)-DAYSII) / 7.
    VALUE = AGESZ(DWK)
C
C   INCREASE TOTAL BY 1/2 VALUE, DECREASE BY 1/2 LAST SUM TERM
    TSZE = TSZE + ((VALUE - SZEUPJ(J)) / 2.)
C
C   COMPUTE AVERAGE SIZE DURING J, WHERE DELTA T ALWAYS = 1
    SZEUFJ(J) = TSZE / FLOAT(DCNTR)
  ENDIF
ELSE
  STOP 'ORGANISM TYPE WRONG IN SUBROUTINE SIZE'
ENDIF
RETURN
END

SUBROUTINE QUANT1(INOW, JNOW, WKSNOW, MNOW, DAYNOW)
RETURN
END

SUBROUTINE QUANT2(INOW, JNOW, WKSNOW, MNOW, DAYNOW)
RETURN
END

SUBROUTINE QUANT3(INOW, JNOW, WKSNOW, MNOW, DAYNOW)
RETURN
END

SUBROUTINE QUANT4(INOW, JNOW, WKSNOW, MNOW, DAYNOW)
RETURN
END

SUBROUTINE QUANT5(INOW, JNOW, WKSNOW, MNOW, DAYNOW)
-----
C
C   ORGNISM:
C
C   SUBROUTINE VARIES WITH ORGANISM
C   VARIABLES RETURNED (ONLY IF OPTION TO COMPUTE Q FOR STAGES CHOSEN):
C   Q OF INPUT 5, IN J, OF TYPE T, CURRENT TIME: TQUANJ(J,5,T)
C   CURRENT ORGANISM: BIVALVES
C
C   STRUCTURE: COMPUTES THE QUANTITY OF EACH TYPE, T, OF INPUT 5,
C   USED IN EACH PERIOD IN ANY PARTICULAR M
C   ASSUMES THAT NO ALGAE IS PURCHASED FROM OUTSIDE SOURCES
4603
4604
4605
4606
4607
4608
4609
4610
4611
4612
4613
4614
4615
4616
4617
4618
4619
4620
4621
4622
4623
4624
4625
4626
4627
4628
4629
4630
4631
4632
4633
4634
4635
4636
4637
4638
4639
4640
4641
4642
4643
4644
4645
4646
4647
4648
4649
4650
4651
4652
4653
4654
4655
4656
4657
4658
4659
4660
4661
4662
4663
4664
4665
4666
C
C   NAME
C
C   AGENO: RELATES AGE/SURVIVAL. CALLED EACH DAY OF PERIOD
C   AGESZ: RELATES AGE/SIZE. CALLED EACH DAY OF PERIOD
C   CELLS: CELLS CONSUMED GIVEN FILTERING RATE AND REQ.
C   CDWK: CELLS(10,000) ALGAE EATEN/BIVALVE EACH DAY OF PERIOD
C   D: LOOP COUNTER FOR DAYS
C   DAY: TRANSLATE DAYNOW FOR ELAPSED DAYS
C   DAYNOW: ARGUE: ELAPSED DAYS TO END OF PERIOD
C   DCNTR: USED TO KEEP TRACK OF NUMBER OF DAYS LOOPED THRU A PERIOD
C   DMK: FOR EACH DAY OF PERIOD, TRANSLATE INTO ELAPSED WEEKS
C   FCELLS: CELLS REQUIRED CONSIDERING FILTERING RATE/STAGE
C   I: TRANSLATE INOW FOR CURRENT STAGE
C   J: ARGUE: CURRENT STAGE THAT J IS IN
C   JNOW: TRANSLATE JNOW FOR CURRENT PERIOD
C   M: ARGUE: CURRENT PERIOD
C   MNOW: TRANSLATE MNOW FOR CURRENT MONTH
C   NODWK: ARGUE: MONTH THAT CURRENT PERIOD IS IN
C   NODWK: NUMBER OF BIVALVE EXISTING AT EACH DAY OF PERIOD
C   SZEUPJ: SIZE OF BIVALVE AT EACH DAY OF PERIOD
C   T: LOOP COUNTER FOR TYPES OF INPUT N
C   TCELLS: SUM OF TCDWK OVER ENTIRE PERIOD
C   TCDWK: CELLS EATEN FOR ALL BIVALVES AT THAT DAY OF PERIOD
C   WKS: TRANSLATE WKSNOW FOR ELAPSED WEEKS
4667
4668
4669
4670
4671
4672
4673
4674
4675
4676
4677
4678
4679
4680
4681
4682
4683
4684
4685
4686
4687
4688
4689
4690
4691
4692
4693
4694
4695
4696
4697
4698
4699
4700
4701
4702
4703
4704
4705
4706
4707
4708
4709
4710
4711
4712
4713
4714
4715
4716
4717
4718
4719
4720
4721
4722
4723
4724
4725
4726
4727
4728
4729
4730

```

```

C WKSNOW ARGUE: ELAPSED WEEKS TO END OF PERIOD
C -----
C TRANSLATE ARGUMENTS INTO LOCAL VARIABLES
  I = JNOW
  J = JNOW
  WKS = WKSNOW
  M = MNOW
  DAY = DAYNOW
C ALGORITHM USED SHOULD DEPEND ON ORGANISM TYPE
IF (OTYPE.EQ.1 .OR. OTYPE.EQ.2) THEN
C CONDITIONING STAGE REQUIRES DIFFERENT ALGORITHM
IF (I.EQ.1) THEN
  STOP 'ERROR SUBR. QUANT5. CANT COMPUTE N5 FOR I=1'
C OTHERWISE, COMPUTE TOTAL CELLS USED OVER PERIOD BY SUMMING
C OVER THE PERIOD THE AMOUNT REQUIRED EACH DAY OF THE PERIOD.
C USE THE TRAPEZOIDAL RULE
ELSEIF (I.GT.1) THEN
  DCNTR = 0
  TCELLS = 0.
  DO 100 D = (DAY-DAYSJ(I)-DAYSII+1.), (DAY-DAYSII+.499)
    DWK = FLOAT(D) / 7.
C USE THE ALGORITHM CHOSEN IN READN5 TO FIND CELLS/BIVALV
IF (QEQ5.EQ.1) THEN
  SIZEDWK = AGESZ(DWK)
  CDWK = ((SIZEDWK)**(1/EQ5B(1))) *
2 (EXP((-EQ5A(1)) / EQ5B(1)))
ELSEIF (QEQ5.EQ.2) THEN
  CDWK = ((DWK)**(1/EQ5B(2))) *
2 (EXP((-EQ5A(2)) / EQ5B(2)))
ENDIF
C TOTAL CELLS/DAY IS FUNCTION OF NUMBER OF ANIMALS AT THAT
C DAY. ALSO MULTIPLY BY 10000. FOR PROPER EQUATION
NODWK = AGENO(DWK)
TCDWK = CDWK * NODWK * 10000.
TCELLS = TCELLS + TCDWK
DCNTR = DCNTR + 1
100 CONTINUE
C DETERMINE QUANTITY OF DAY BEFORE PERIOD BEGINS, TO MORE
C CLOSELY APPROXIMATE THE INTEGRAL
DWK = (DAY-DAYSJ(I)-DAYSII) / 7.
IF (QEQ5.EQ.1) THEN
  SIZEDWK = AGESZ(DWK)
  VALUE = ((SIZEDWK)**(1/EQ5B(1))) *
2 (EXP((-EQ5A(1)) / EQ5B(1)))
ELSEIF (QEQ5.EQ.2) THEN
  VALUE = ((DWK)**(1/EQ5B(2))) *
2 (EXP((-EQ5A(2)) / EQ5B(2)))
ENDIF
NODWK = AGENO(DWK)
VALUE = VALUE * NODWK * 10000.
C INCREASE SUM BY 1/2 VALUE, DECREASE BY 1/2 LAST SUM TERM
TCELLS = TCELLS + ((VALUE - TCDWK) / 2.)
C FILTERING RATE_100%, THEN INCREASE AMOUNT/PERIOD
4731 -----
4732
4733
4734
4735
4736
4737
4738
4739
4740
4741
4742
4743
4744
4745
4746
4747
4748
4749
4750
4751
4752
4753
4754
4755
4756
4757
4758
4759
4760
4761
4762
4763
4764
4765
4766
4767
4768
4769
4770
4771
4772
4773
4774
4775
4776
4777
4778
4779
4780
4781
4782
4783
4784
4785
4786
4787
4788
4789
4790
4791
4792
4793
4794
-----
FCELLS = TCELLS/FILTER(I)
DO 200 I=1,NOOFT(5)
C COMPUTE CELLS EACH TYPE, BASED ON 100% USED/STAGE
  CELLS = FCELLS * N5REQ(I,T)
C COMPUTE QUANT/TYPED MUST PAY FOR SINCE IS REDUCED BY QUANT4802
  AVAIL. FROM SEAWATER. ASSUME SAME % EACH TYPE IN THE SEA4803
  TQUANJ(J,5,T) = CELLS - (CELLS * SEACEL(I,M))
4804
4805
4806
4807
4808
4809
4810
4811
4812
4813
4814
4815
4816
4817
4818
4819
4820
4821
4822
4823
4824
4825
4826
4827
4828
4829
4830
4831
4832
4833
4834
4835
4836
4837
4838
4839
4840
4841
4842
4843
4844
4845
4846
4847
4848
4849
4850
4851
4852
4853
4854
4855
4856
4857
4858
FCELLS = TCELLS/FILTER(I)
DO 200 I=1,NOOFT(5)
C COMPUTE CELLS EACH TYPE, BASED ON 100% USED/STAGE
  CELLS = FCELLS * N5REQ(I,T)
C COMPUTE QUANT/TYPED MUST PAY FOR SINCE IS REDUCED BY QUANT4802
  AVAIL. FROM SEAWATER. ASSUME SAME % EACH TYPE IN THE SEA4803
  TQUANJ(J,5,T) = CELLS - (CELLS * SEACEL(I,M))
4804
4805
4806
4807
4808
4809
4810
4811
4812
4813
4814
4815
4816
4817
4818
4819
4820
4821
4822
4823
4824
4825
4826
4827
4828
4829
4830
4831
4832
4833
4834
4835
4836
4837
4838
4839
4840
4841
4842
4843
4844
4845
4846
4847
4848
4849
4850
4851
4852
4853
4854
4855
4856
4857
4858
SUBROUTINE QUANT6(INOW,JNOW,WKSNOW,MNOW,DAYNOW)
RETURN
END
SUBROUTINE COSTJ(INOW,JNOW,WKSNOW,MNOW,DAYNOW)
-----
C ORGN5M:
C -----
C SUBROUTINE DOES NOT VARY WITH ORGANISM
C VARIABLES DO NOT CHANGE
C STRUCTURE: COMPUTES COSTS OF PERIOD J
C COSTS ACCUMULATED IN J
C COSTS ACCUMULATED UP TO THE END OF J
C COSTS PER ORGANISM UP TO THE END OF J
C 1. PARAMETERS
C 1. INCLUDE (IPAR1)
C 2. ARGUMENTS
C 2. INTEGER INOW, JNOW, MNOW
C 2. REAL WKSNOW, DAYNOW
C 3. COMMON BLOCKS
C 3. INCLUDE (ICMAIN)
C 3. INCLUDE (ICBLK1)
C 3. INCLUDE (ICBLK0)
C 3. INCLUDE (ICUNIT)
C 4. LOCAL VARIABLES

```

```

4859 INTEGER I, J, M, N, T
4860 REAL WKS, DAY, CSTOFJ
4861
4862 -----
4863 C NAME DESCRIPTION OF LOCAL VARIAB AND ARGUMENTS
4864 C CSTOFJ COST OF TYPE T IN J (WRITTEN OVER)
4865 C DAY TRANSLATE DAYNOW FOR ELAPSED DAYS
4866 C DAYNOW ARGUE: ELAPSED DAYS TO END OF PERIOD
4867 C I TRANSLATE INOW FOR CURRENT STAGE
4868 C INOW ARGUE: CURRENT STAGE THAT J IS IN
4869 C J TRANSLATE JNOW FOR CURRENT PERIOD
4870 C JNOW ARGUE: CURRENT PERIOD
4871 C M TRANSLATE MNOW FOR CURRENT MONTH
4872 C MNOW ARGUE: MONTH THAT CURRENT PERIOD IS IN
4873 C N LOOP COUNTER FOR INPUTS
4874 C T LOOP COUNTER FOR TYPES
4875 C WKS TRANSLATE WKSNOW FOR ELAPSED WEEKS
4876 C WKSNOW ARGUE: ELAPSED WEEKS TO END OF PERIOD
4877 -----
4878
4879 C TRANSLATE ARGUMENTS TO LOCAL VARIABLE
4880 I = INOW
4881 J = JNOW
4882 WKS = WKSNOW
4883 M = MNOW
4884 DAY = DAYNOW
4885
4886
4887 C COMPUTE COST OF EACH INPUT, N, ONE AT A TIME
4888 CSUPJ(J) = 0.
4889 CSOFJ(J) = 0.
4890 DO 100 N=1,NOOFN
4891 IF (OPTION.EQ.1 .AND. J.EQ.1) THEN
4892 CSOFJ(J) = CSOP1
4893 CSUPJ(J) = CSOP1
4894 CSNOFJ(J,N) = 0.
4895 CSNUPJ(J,N) = 0.
4896 ENDIF
4897
4898 C COST OF EACH TYPE,T, OF EACH INPUT,N, DURING A PERIOD,J
4899 IF (I.GE.BEGIN) THEN
4900 DO 200 T=1,NOOFT(N)
4901 CSNOFJ(J,N) = 0.
4902 CSTOFJ = TQUANJ(J,N,T) * PRICET(N,T,MOFJ(J))
4903 CSNOFJ(J,N) = CSNOFJ(J,N) + CSTOFJ
4904 CONTINUE
4905
4906 C IE, LET THIS VARIABLE EQUAL CALCULATIONS FOR LAST DAY
4907 COST OF ALL INPUTS USED IN A PERIOD
4908 CSOFJ(J) = CSOFJ(J) + CSNOFJ(J,N)
4909 IF (J.EQ.1) THEN
4910 CSNUPJ(J,N) = CSNOFJ(J,N)
4911 CSUPJ(J) = CSOFJ(J)
4912 ELSEIF (J.NE.1) THEN
4913 CSNUPJ(J,N) = CSNOFJ(J,N) + CSNUPJ(J-1,N)
4914 CSUPJ(J) = CSOFJ(J) + CSUPJ(J-1)
4915 ENDIF
4916 ENDIF
4917
4918 C 100 CONTINUE
4919
4920 -----
4921
4922 C COMPUTE COST PER ORGANISM
4923 AVERAGE PER DAY OVER THE PERIOD
4924 CSOFJ(J) = (CSOFJ(J) / ONOOFJ(J)) / DAYSJ(IOFJ(J))
4925 UP TO END OF PERIOD
4926 IF (CSOUPJ(J).EQ.0) THEN
4927 CSOUPJ(J) = 0.
4928 ELSE
4929 CSOUPJ(J) = CSUPJ(J) / ONOUPJ(J)
4930 ENDIF
4931 RETURN
4932 END
4933
4934 C SUBROUTINE WRTOU(MBEGUN, LASTJ, WRT)
4935 -----
4936 C ORGNSM:
4937 -----
4938 C SUBROUTINE DOES NOT VARY WITH ORGANISM
4939 C VARIABLES DO NOT CHANGE
4940
4941 C STRUCTURE: WRITES OUT INFORMATION COMPUTED FOR A BATCH BEGUN ON THE
4942 FIRST OF EACH MONTH OF THE PRODUCTION SEASON.
4943 THIS INFORMATION INCLUDES:
4944 -STAGE AND PERIOD INFORMATION
4945 -QUANTITY OF EACH TYPE OF EACH INPUT USED IN EACH PERIOD
4946 -COST OF EACH INPUT USED DURING EACH PERIOD AND UP TO A
4947 PERIOD
4948 -COST OF PRODUCING AN ORGANISM UP TO THE END OF A PERIOD
4949
4950 C 1. PARAMETERS
4951 INCLUDE (IPAR1)
4952
4953 C 2. ARGUMENTS
4954 INTEGER LASTJ, WRT, WRT2
4955 CHARACTER * 10, MBEGUN
4956
4957 C 3. COMMON BLOCKS
4958 INCLUDE (ICMAIN)
4959 INCLUDE (ICBLK1)
4960 INCLUDE (ICBLKO)
4961 INCLUDE (ICUNIT)
4962
4963 C 4. LOCAL VARIABLES
4964 INTEGER I, J, M, N, T, W
4965
4966 C NAME DESCRIPTION OF LOCAL VARIABLES AND ARGUMENTS
4967 -----
4968 C I LOOP COUNTER FOR STAGE
4969 C J LOOP COUNTER FOR PERIODS
4970 C LASTJ ARGUE: LAST J PERIOD SUCCESSFULLY COMPUTED
4971 C M LOOP COUNTER FOR MONTHS
4972 C MBEGUN MONTH THAT CURRENT BATCH WAS BEGUN
4973 C N LOOP COUNTER FOR INPUTS
4974 C T LOOP COUNTER FOR TYPES
4975 C WRT ARGUMENT: UNIT NUMBER
4976 C WRT2 TRANSLATE ARGUMENT UNIT NUMBER TO LOCAL WRITE VARIABLE
4977 ARGUE: TRANSLATE UNIT WRITE TO NEXT LEVEL SUBROUTINE
4978 -----
4979
4980 C 100 CONTINUE
4981
4982 -----
4983
4984
4985
4986
4987
4988
4989
4990
4991
4992

```

```

4987 W = WRT
4988 WRT2 = WRT
4989
4990 C FIRST PAGE INFORMATION FOR OUTPUT FILE
4991 IF (MBEGUN.EQ.NAMEN(M1)) THEN
4992 WRITE(W,10000) NAME, DATE, NAMEF1, NAME, NAMEO(OTYPE), NAME,
4993 ONOBEG, NAMES, OPTION, NAMEI(BEGINI)
4994 2 IF (OPTION.EQ.1) WRITE(W,13000) BEGINI-1
4995
4996 C PRICE MATRIX FOR EACH INPUT
4997 WRITE(W,16000) (NAMEN(M), M=BEGINM,ENDM)
4998 DO 100 N=1,NOOFN
4999 IF (NOOFT(N).EQ.1) THEN
5000 WRITE(W,16100) NAMEN(N), NAMENU(N),
5001 (PRICET(N,1,M), M=BEGINM,ENDM)
5002 2 ELSEIF (NOOFT(N).GT.1) THEN
5003 DO 150 T=1,NOOFT(N)
5004 WRITE(W,16100) NAMET(N,T), NAMENU(N),
5005 (PRICET(N,T,M), M=BEGINM,ENDM)
5006 2
5007 CONTINUE
5008 ENDF
5009 CONTINUE
5010 ENDF
5011
5012 C HEADINGS FOR EACH NEW MONTH BEGUN
5013 WRITE(W,20000) NAME, DATE, MBEGUN
5014
5015 C AGE, SIZE AND TIME
5016 WRITE(W,32000) NAME, NAME, NAME
5017 WRITE(W,90000)
5018 DO 300 J=1,LASTJ
5019 IF (PASEDM(J).EQ.0) THEN
5020 WRITE(W,33000) J, IOFJ(J), NAMEI(IOFJ(J)), MOFJ(J), DAYUPJ(J),
5021 (AGUPJ(J), ONOUPJ(J), ONOUPJ(J), SZEOPJ(J), SZEOPJ(J))
5022 2 ELSEIF (PASEDM(J).EQ.1) THEN
5023 WRITE(W,33100) J, IOFJ(J), NAMEI(IOFJ(J)), MOFJ(J),
5024 DAYUPJ(J), NAME, MBEGUN
5025 2 ENDF
5026 CONTINUE
5027
5028 C TOTAL COSTS
5029 WRITE(W,35000) NAME, NAME
5030 WRITE(W,93000)
5031 DO 350 J=1,LASTJ
5032 IF (PASEDM(J).EQ.0) THEN
5033 WRITE(W,36000) J, IOFJ(J), CSOFJ(J), CSOOFJ(J),
5034 CSOUPJ(J)
5035 2 ELSEIF (PASEDM(J).EQ.1) THEN
5036 WRITE(W,42100) J, IOFJ(J)
5037 ENDF
5038 CONTINUE
5039
5040 C COSTS IN A PERIOD, J OF EACH INPUT
5041 WRITE(W,40000)
5042 WRITE(W,41000) (NAMEN(N), N=1,NOOFN)
5043 WRITE(W,95000)
5044 DO 400 J = BEGINJ(BEGINI), LASTJ
5045 IF (PASEDM(J).EQ.0) THEN
5046 WRITE(W,42000) J, IOFJ(J), (CSNOFJ(J,N), N=1,NOOFT(N))
5047 ELSEIF (PASEDM(J).EQ.1) THEN
5048 WRITE(W,42100) J, IOFJ(J)
5049 ENDF
5050
5051
5052
5053
5054
5055
5056
5057
5058
5059
5060
5061
5062
5063
5064
5065
5066
5067
5068
5069
5070
5071
5072
5073
5074
5075
5076
5077
5078
5079
5080
5081
5082
5083
5084
5085
5086
5087
5088
5089
5090
5091
5092
5093
5094
5095
5096
5097
5098
5099
5100
5101
5102
5103
5104
5105
5106
5107
5108
5109
5110
5111
5112
5113
5114

```

```

4      T46, 'IN PERIOD/DAY', T61, 'AT END PERIOD'
36000 FORMAT( ' , I3, T11, I1, T16, G14.7, T31, G14.7, T46, G14.7, T61, G14.7)
33000 FORMAT( ' , I2, T8, I1, T11, A10, T26, I2, T31, G14.7, T46, G14.7, T61,
2      G14.7, T76, G14.7, T91, G14.7, T106, G14.7)
33100 FORMAT( ' , I2, T8, I1, T11, A10, T26, I2, T31, G14.7, T46, 'TIME PAST ',
2      'MONTHS OF ',
3      'PRODUCTION. CANT PRODUCE ', A, ' OF THIS AGE IF BEGUN IN ', A)
40000 FORMAT('0// ' , 'COST OF INPUTS IN A PERIOD: '// '+', 27(' _ '))
41000 FORMAT( ' , 'PERIOD', T9, 'STAGE', T16, 6(A10, 5X))
42000 FORMAT( ' , I3, T11, I1, 4(T16, 7(G14.7, 1X)) / ' ')
42100 FORMAT( ' , I3, T11, I1, T16, 'TIME PAST PRODUCTION MONTHS')
50000 FORMAT('0// ' , 'COST OF INPUTS UP TO END OF A PERIOD'// '+', 37(' _ '))
70000 FORMAT('0// ' , 'QUANTITY OF INPUT USED PER PERIOD: '// '+', 34(' _ '))
71000 FORMAT( ' // ' , A10, 1X, A, ' : // ' , 17(' - '))
72000 FORMAT( ' , 'PERIOD', T9, 'STAGE', T16, 10(A10, 5X))
72100 FORMAT( ' , 'PERIOD', T9, 'STAGE')
73000 FORMAT( ' , I3, T11, I1, T16, 10(G14.7, 1X))
90000 FORMAT( ' , 3(' - '), T6, 4(' - '), T11, 14(' - '), T26, 4(' - '), T31, 14(' - '),
2      T46, 14(' - '), T61, 14(' - '), T76, 14(' - '), T91, 14(' - '), T106, 14(' - '))
93000 FORMAT( ' , 6(' - '), T9, 6(' - '), T16, 14(' - '), T31, 14(' - '), T46, 14(' - '),
2      T61, 14(' - '))
95000 FORMAT( ' , 6(' - '), T9, 6(' - '), T16, 14(' - '), T31, 14(' - '), T46, 14(' - '),
2      T61, 14(' - '), T76, 14(' - '), T91, 14(' - '))

RETURN
END

SUBROUTINE WOUTN1(MBEGUN, LASTJ, WRT)
RETURN
END

SUBROUTINE WOUTN2(MBEGUN, LASTJ, WRT)
RETURN
END

SUBROUTINE WOUTN3(MBEGUN, LASTJ, WRT)
RETURN
END

SUBROUTINE WOUTN4(MBEGUN, LASTJ, WRT)
RETURN
END

SUBROUTINE WOUTN5(MBEGUN, LASTJ, WRT)
-----
C
C ORGNSM:
C -----
C SUBROUTINE VARIES WITH ORGANISM
C VARIABLES CHANGE WITH ORGANISM
C CURRENT ORGANISM: BIVALVES
C STRUCTURE: WRITE INFORMATION INTO A SEPARATE DATA FILE THAT WILL BE
C USED FOR ADDITIONAL ANALYSIS BY A STATISTICAL PROGRAM
C NOTE: THE USE OF THIS SUBROUTINE REQUIRES THAT THE
C NUMBER OF ALGAE STAGES BE SET AT 6
C 1. PARAMETERS
C 2. ARGUMENTS
C 3. COMMON BLOCKS
C 4. LOCAL VARIABLES
C NAME DESCRIPTION OF LOCAL VARIABLES AND ARGUMENTS
-----
C J LOOP COUNTER FOR PERIODS
C LASTJ ARGUE: LAST PERIOD THAT COMPUTATIONS WERE DONE FOR
C MBEGUN ARGUE: MONTH IN WHICH CURRENT BATCH IS BEGUN
C WRT ARGUE: UNIT NUMBER FOR WRITING
C W UNIT NUMBER TO WRITE
-----
C WRITE INFORMATION SPECIFIC TO THIS PROGRAM
DO 100 J = 1, LASTJ
WRITE(W, 10000) J, MOFJ(J), DAYUFJ(J), AGEUFJ(J), ONOOFJ(J),
2 ONOUPJ(J), SZEUFJ(J), SZEUPJ(J), CSOFJ(J),
3 CSUPJ(J), CSOOFJ(J), CSOUPJ(J), GSNOFJ(J, 5),
4 GSNUPJ(J, 5), TQUANJ(J, 5, 5), TQUANJ(J, 5, 6),
5 QUAOFJ(J, 5), QUAOFJ(J, 6)
100 CONTINUE

W = WRT

WRITE INFORMATION SPECIFIC TO THIS PROGRAM
DO 100 J = 1, LASTJ
WRITE(W, 10000) J, MOFJ(J), DAYUFJ(J), AGEUFJ(J), ONOOFJ(J),
2 ONOUPJ(J), SZEUFJ(J), SZEUPJ(J), CSOFJ(J),
3 CSUPJ(J), CSOOFJ(J), CSOUPJ(J), GSNOFJ(J, 5),
4 GSNUPJ(J, 5), TQUANJ(J, 5, 5), TQUANJ(J, 5, 6),
5 QUAOFJ(J, 5), QUAOFJ(J, 6)
100 CONTINUE

10000 FORMAT('0', I2, 1X, I2, T10, 8E12.5 / ' ', T10, 8E12.5)

```

```

5243 RETURN
5244 END
5245
5246 SUBROUTINE MOUTNG(MBEGUN, LASTJ, WRT)
5247 RETURN
5248 END
5249
5250 REAL FUNCTION AGENO(WEEKS)
5251
5252
5253
5254
5255
5256
5257
5258
5259
5260
5261
5262
5263
5264
5265
5266
5267
5268
5269
5270
5271
5272
5273
5274
5275
5276
5277
5278
5279
5280
5281
5282
5283
5284
5285
5286
5287
5288
5289
5290
5291
5292
5293
5294
5295
5296
5297
5298
5299
5300
5301
5302
5303
5304
5305
5306

C FUNCTION RETURNED TO SUBROUTINES QUANT5, SIZE
C CURRENT ORGANISM: BIVALVES
C STRUCTURE: RELATES SIZE TO AGE IN TERMS OF WEEKS
C 1. PARAMETERS
C INCLUDE (IPAR1)
C 2. ARGUMENTS
C REAL WEEKS
C 3. COMMON BLOCKS
C INCLUDE (ICMAIN)
C INCLUDE (ICBLKO)
C NAME DESCRIPTION OF LOCAL VARIABLES AND ARGUMENTS
C WEEKS ARGUE: ELAPSED WEEKS UP TO THE END OF PERIOD J
C
C ALGORITHM USED SHOULD DEPEND ON ORGANISM TYPE
C IF (OTYPE .EQ. 1 .OR. OTYPE .EQ. 2) THEN
C   AGESZ = (EQSZA * WEEKS) + (EQSZB * WEEKS * WEEKS)
C ELSE
C   STOP 'ERROR WITH ORGANISM TYPE IN FUNCTION AGESZ'
C ENDIF
C RETURN
C END
C
C BLOCK DATA BLK1
C
C ORGNSM:
C
C SUBROUTINE VARIES WITH ORGANISM
C VARIABLES MUST BE DATA INITIALIZED:
C NOOFT(N)
C OPQOK
C OPNOK
C OPNSOK(N)
C OPTOK(N)
C CURRENT ORGANISM: BIVALVES
C STRUCTURE: INITIALIZE VARIABLES IN COMMON BLOCKS, ICBLK1
C 1. PARAMETERS
C INCLUDE (IPAR1)
C INCLUDE (ICUNIT)
C 3. COMMON BLOCKS
C INCLUDE (ICBLK1)
C 4. LOCAL VARIABLES
C INTEGER I, J, K, M, N, T
C NAME DESCRIPTION OF LOCAL VARIABLES
C I LOOP COUNTER FOR STAGES
C J LOOP COUNTER FOR PERIODS

```

```

5307
5308
5309
5310
5311
5312
5313
5314
5315
5316
5317
5318
5319
5320
5321
5322
5323
5324
5325
5326
5327
5328
5329
5330
5331
5332
5333
5334
5335
5336
5337
5338
5339
5340
5341
5342
5343
5344
5345
5346
5347
5348
5349
5350
5351
5352
5353
5354
5355
5356
5357
5358
5359
5360
5361
5362
5363
5364
5365
5366
5367
5368
5369
5370

```

```

C K LOOP COUNTER FOR PROGRAM RUNNING OPTIONS
C M LOOP COUNTER FOR MONTHS
C N LOOP COUNTER FOR INPUTS
C T LOOP COUNTER FOR TYPES OF INPUT N
-----
C VARIABLES THAT SHOULD NOT CHANGE
C -----
C NAME OF MONTHS
2 DATA (NAMEM(M), M=1,12) / 'JANUARY', 'FEBRUARY', 'MARCH', 'APRIL',
3 'MAY', 'JUNE', 'JULY', 'AUGUST', 'SEPTEMBER', 'OCTOBER', 'NOVEMBER', 'DECEMBER' /
-----
C UNIT NUMBERS FOR ALL BUT OUTPUT FILES/ AND TO INITIATE VARIABLES
C -----
DATA WF, RI, WI, RF / 3,5,6,8 /
DATA (QOUTQN(N), N=1,NOOFN) / NOOFN * 'N' /
-----
C VARIABLES THAT CHANGE WITH CAPABILITIES OF THE PROGRAM'S OPTIONS
C -----
C NUMBER OF TYPES,T, OF EACH INPUT,N
DATA (NOOFT(N), N=1,NOOFN) / 1,1,1,1,1,1 /
-----
C OPTION INFORMATION: 0=AVAILABLE, 1=NOT AVAILABLE
C OPTIONS 1-3 FOR RUNNING PROGRAM
DATA (OPOOK(K), K=1,3) / 0,0,1 /
-----
C TO COMPUTE QUANTITY FOR EACH INPUT,N, IN EACH STAGE,J
DATA (OPNOK(I,1), I=1,NOOFI) / NOOFI * 1 /
DATA (OPNOK(I,2), I=1,NOOFI) / NOOFI * 1 /
DATA (OPNOK(I,3), I=1,NOOFI) / NOOFI * 1 /
DATA (OPNOK(I,4), I=1,NOOFI) / NOOFI * 1 /
DATA (OPNOK(I,5), I=1,NOOFI) / 1,0,0,0 /
DATA (OPNOK(I,6), I=1,NOOFI) / NOOFI * 1 /
-----
C TO COMPUTE THE PRICE OF INPUT,N
DATA (OPNSOK(N), N=1,NOOFN) / 1,1,1,1,0,1 /
-----
C TO CHANGE THE NUMBER OF TYPES,T, OF EACH INPUT,N
DATA (OPTOK(N), N=1,NOOFN) / 1,1,0,1,0,1 /
-----
C TO PRINT OUT ADDITIONAL INFORMATION ABOUT PRODUCTION INPUTS
DATA (OUTQN(N), N=1,NOOFN) / 1,1,1,1,0,1 /
-----
END
-----
BLOCK DATA BLKO
C -----
C ORGNSM:
C -----
C SUBROUTINE VARIES WITH ORGANISM
C VARIABLES THAT MUST BE DATA INITIALIZED:
C NAMEM(N)
C NAMEU(N)
C NAMEI(I)
-----
5371 C NAMEI(N,T)
5372 C NAMEO(O)
5373 C CURRENT ORGANISM: BIVALVES
5374 C
5375 C STRUCTURE: INITIALIZE VARIABLES IN COMMON BLOCK, ICBLKO
5376 C
5377 C 1. PARAMETERS
5378 C INCLUDE (IPAR1)
5379 C 3. COMMON BLOCKS
5380 C INCLUDE (ICBLKO)
5381 C 4. LOCAL VARIABLES
5382 C INTEGER EQ, I, N, O, T
5383 C
5384 C NAME DESCRIPTION OF LOCAL VARIABLES
5385 C -----
5386 C EQ LOOP COUNTER FOR FEEDING EQUATIONS, INPUT 5
5387 C I LOOP COUNTER FOR STAGES
5388 C N LOOP COUNTER FOR INPUTS
5389 C O LOOP COUNTER FOR ORGANISM TYPES
5390 C T LOOP COUNTER FOR TYPES OF INPUT N
5391 C -----
5392 C
5393 C VARIABLES REQUIRED BUT THAT CHANGE WITH ORGANISM:
5394 C -----
5395 C DATA NAME / 'BIVALVE' /
5396 C
5397 C DATA (NAMEM(N), N=1,NOOFN) / 'ELECTRICITY', 'FUEL OIL', 'LABOR',
5398 C 'MATERIALS', 'ALGAE', 'CULCH' /
5399 C
5400 C DATA (NAMEU(N), N=1,NOOFN) / 'KWHR', 'LITRE', 'HOUR', 'UNIT',
5401 C 'CELL', 'LITRE' /
5402 C
5403 C DATA (NAMEI(I), I=1,NOOFI) / 'CONDITION', 'SPAWN', 'LARVAE',
5404 C 'SETTING', 'JUVENILE' /
5405 C
5406 C DATA (NAMEO(O), O=1,NOOFO) / 'OYSTERS', 'CLAMS' /
5407 C
5408 C
5409 C
5410 C VARIABLES RELATING TO ORGANISM DEPENDENT FUNCTIONS:
5411 C -----
5412 C DATA EQNO(1) / POPULATION = POPULATION(INITIAL) * [(WEEKS+1)**A] /
5413 C
5414 C DATA EQNOAF / -.255068 /
5415 C
5416 C DATA (EQQ5(EQ), EQ=1,2) /
5417 C 2 'NATURAL LOG(SIZE IN MM) = A + B * NAT LOG(10,000 CELLS)',
5418 C 2 'NATURAL LOG(AGE IN WKS) = A + B * NAT LOG(10,000 CELLS)' /
5419 C
5420 C DATA (EQQ5AF(EQ), EQ=1,2) / -.631048, -.330940 /
5421 C DATA (EQQ5BF(EQ), EQ=1,2) / .338426, .344505 /
5422 C
5423 C DATA EQSQZ(1) / 'SIZE = (A * AGE) + (B * AGE SQUARED)' /
5424 C
5425 C DATA EQSZAF / .737 /
5426 C DATA EQSZBF / -.00252 /
5427 C
5428 C ORGNSM:
5429 C -----
5430 C SUBROUTINE VARIES WITH ORGANISM
5431 C VARIABLES THAT MUST BE DATA INITIALIZED:
5432 C NAMEM(N)
5433 C NAMEU(N)
5434 C NAMEI(I)
-----
5435 C NAMEI(N,T)
5436 C NAMEO(O)
5437 C CURRENT ORGANISM: BIVALVES
5438 C
5439 C STRUCTURE: INITIALIZE VARIABLES IN COMMON BLOCK, ICBLKO
5440 C
5441 C 1. PARAMETERS
5442 C INCLUDE (IPAR1)
5443 C 3. COMMON BLOCKS
5444 C INCLUDE (ICBLKO)
5445 C 4. LOCAL VARIABLES
5446 C INTEGER EQ, I, N, O, T
5447 C
5448 C NAME DESCRIPTION OF LOCAL VARIABLES
5449 C -----
5450 C EQ LOOP COUNTER FOR FEEDING EQUATIONS, INPUT 5
5451 C I LOOP COUNTER FOR STAGES
5452 C N LOOP COUNTER FOR INPUTS
5453 C O LOOP COUNTER FOR ORGANISM TYPES
5454 C T LOOP COUNTER FOR TYPES OF INPUT N
5455 C -----
5456 C
5457 C VARIABLES REQUIRED BUT THAT CHANGE WITH ORGANISM:
5458 C -----
5459 C DATA NAME / 'BIVALVE' /
5460 C
5461 C DATA (NAMEM(N), N=1,NOOFN) / 'ELECTRICITY', 'FUEL OIL', 'LABOR',
5462 C 'MATERIALS', 'ALGAE', 'CULCH' /
5463 C
5464 C DATA (NAMEU(N), N=1,NOOFN) / 'KWHR', 'LITRE', 'HOUR', 'UNIT',
5465 C 'CELL', 'LITRE' /
5466 C
5467 C DATA (NAMEI(I), I=1,NOOFI) / 'CONDITION', 'SPAWN', 'LARVAE',
5468 C 'SETTING', 'JUVENILE' /
5469 C
5470 C DATA (NAMEO(O), O=1,NOOFO) / 'OYSTERS', 'CLAMS' /
5471 C
5472 C
5473 C
5474 C VARIABLES RELATING TO ORGANISM DEPENDENT FUNCTIONS:
5475 C -----
5476 C DATA EQNO(1) / POPULATION = POPULATION(INITIAL) * [(WEEKS+1)**A] /
5477 C
5478 C DATA EQNOAF / -.255068 /
5479 C
5480 C DATA (EQQ5(EQ), EQ=1,2) /
5481 C 2 'NATURAL LOG(SIZE IN MM) = A + B * NAT LOG(10,000 CELLS)',
5482 C 2 'NATURAL LOG(AGE IN WKS) = A + B * NAT LOG(10,000 CELLS)' /
5483 C
5484 C DATA (EQQ5AF(EQ), EQ=1,2) / -.631048, -.330940 /
5485 C DATA (EQQ5BF(EQ), EQ=1,2) / .338426, .344505 /
5486 C
5487 C DATA EQSQZ(1) / 'SIZE = (A * AGE) + (B * AGE SQUARED)' /
5488 C
5489 C DATA EQSZAF / .737 /
5490 C DATA EQSZBF / -.00252 /
5491 C
5492 C ORGNSM:
5493 C -----
5494 C SUBROUTINE VARIES WITH ORGANISM
5495 C VARIABLES THAT MUST BE DATA INITIALIZED:
5496 C NAMEM(N)
5497 C NAMEU(N)
5498 C NAMEI(I)
-----
END

```



```

0001 C FILE NAME: ICN5$
0002 C ORGNSM: VARIABLES SPECIFIC TO ORGANISM
0003 C CURRENT ORGANISM: BIVALVES
0004 C TO INCLUDE COMMONS USED IN PRICES SUBROUTINE
0005 C VARIABLE NAMES ARE IDENTICAL IN ALL PRICES SUBROUTINES.
0006 C PARAMETERS USED IN PRICES
0007 INTEGER NUMGOD, MAXLAB
0008 PARAMETER (NUMGOD=2, MAXLAB=6)
0009 COMMON / CNS /
0010 I BEGPRO, ENDPRO, NUMLAB, NUMSTG,
0011 R BLBLIF, COMEFF, HRCOM, HRPMP, HRSKOM, NUMTUB,
0012 R PKWHR, PMEFF, PFGPM, POIL, WATAIR, WATAUT, WATBLB,
0013 R ALLAB(MAXT), BLBRM(MAXT), CALIN(MAXT), TCELL(MAXT), CPERML(MAXT),
0014 R DAYSTG(MAXT), DISSTG(MAXT), HRSAUT(MAXT), HRSBLB(MAXT), HZOTNK(MAXT),
0015 R INTNK(MAXT), LFOOD(MAXT), LITWHR(MAXT), NUMAIR(MAXT), NUMTNK(MAXT),
0016 R STGRM(MAXT), TALLAB(MAXT), TALLAL(MAXT), TCCALIN(MAXT), TTKWHR(MAXT),
0017 R TKWHR(MAXT), TLTRE(MAXT), TOLL(MAXT), TTKWHL(MAXT), TTKWHR(MAXT),
0018 R TTOIL(MAXT), TTOILL(MAXT), NUTLTR(MAXT), TUBTNK(MAXT),
0019 R ALFOOU(MAXT), TCCELL(MAXT),
0020 R HRLABW(6), PLABOR(6), TCLAY(6), TLBY(6, MAXT), TTLATY(6, MAXT),
0021 R HRLABS(6, MAXT), PGOOD(5), TGOY(5, MAXT), TTGOY(5, MAXT),
0022 R CINF(4, MAXT), TCINF(4, MAXT), TCINPL(4, MAXT),
0023 R HRSAIR(MAXM, MAXT), TCLITR(MAXT),
0024 C NAWALG(MAXT), MONAVA(MAXT),
0025 C DATE2, QWRT$5
0026 INTEGER BEGPRO, ENDPRO, NUMLAB, NUMSTG
0027 REAL BLBLIF, COMEFF, HRCOM, HRPMP, HRSKOM, NUMTUB, PKWHR,
0028 R PMEFF, PFGPM, POIL, WATAIR, WATAUT, WATBLB,
0029 R ALLAB, BLBRM, CALIN, TCELL, CPERML, DAYSTG, DISSTG, HRSAUT,
0030 R HRSBLB, HZOTNK, INTNK, LFOOD, LITWHR, NUMAIR, NUMTNK,
0031 R STGRM, TALLAB, TALLAL, TCCALIN, TTKWHR, TLTRE, TOLL,
0032 R TTKWHL, TTKWHR, TTOIL, TTOILL, NUTLTR, TUBTNK,
0033 R ALFOOU, TCCELL,
0034 R HRLABW, PLABOR, TGLATY, TLBY, TTLATY,
0035 R PGOOD, TGOY, TTGOY, CINF, TCINF, TCINPL, HRLABS,
0036 R HRSAIR, TCLITR
0037 CHARACTER*10 NAWALG, MONAVA
0038 CHARACTER*40 DATE2
0039 CHARACTER*1 QWRT$5

```

```

0001 C FILE NAME: ICMAIN
0002 C ORGNSM: VARIABLES DON'T CHANGE
0003 C TO INCLUDE COMMONS NECESSARY TO RUN BIVALVE PROGRAM SKELETON
0004 COMMON / CMAIN /
0005 I ANYNOK, BEGNI, BEGIM, BEGINM, BEGINJ, OTYPE, ENDJ,
0006 I ENDM, IOFJ, JPERI, MOFJ, M1, M2, OPTION, PASEDM, TOTALI,
0007 R AGEUPJ, ONOBEG, ONOOFJ, ONOUPJ, CSOOFJ, CSOUPJ, CSNOFJ,
0008 R CSNUPJ, CSOFJ, CSOPI, CSUPJ,
0009 R DAYUPJ, DAYOPI, DAYSJ, DAYSJ, PRICET, SZEOPJ, SZEUPJ,
0010 R TQUANI, TQUANJ, WKSUPJ,
0011 C DATE, NAMEF1, NAMES, NQBYM, N$BYM, OPNNEW, QOPNS, QOUTQN
0012 INTEGER
0013 I ANYNOK, BEGNI, BEGIM, BEGINM, BEGINJ, OTYPE, ENDJ,
0014 I ENDM, IOFJ, JPERI, MOFJ, M1, M2, OPTION, PASEDM, TOTALI
0015 REAL
0016 R AGEUPJ, ONOBEG, ONOOFJ, ONOUPJ, CSOOFJ, CSOUPJ, CSNOFJ,
0017 R CSNUPJ, CSOFJ, CSOPI, CSUPJ, DAYUPJ,
0018 R DAYOPI, DAYSJ, DAYSJ, PRICET, SZEOPJ, SZEUPJ,
0019 R TQUANI, TQUANJ, WKSUPJ
0020 CHARACTER * 50 DATE, NAMES
0021 CHARACTER * 10 NAMEF1
0022 CHARACTER * 1 NQBYM, N$BYM, QOPNS, OPNNEW, QOUTQN
0023 DIMENSION
0024 I ANYNOK(NOOFN), BEGNI(NOOFI), IOFJ(MAXJI), JPERI(NOOFI),
0025 I MOFJ(MAXJI), PASEDM(MAXJI),
0026 R AGEUPJ(MAXJI), ONOOFJ(MAXJI), ONOUPJ(MAXJI), CSOOFJ(MAXJI),
0027 R CSOUPJ(MAXJI), CSNOFJ(MAXJI, NOOFN), CSNUPJ(MAXJI, NOOFN),
0028 R CSOFJ(MAXJI),
0029 R CSUPJ(MAXJI), DAYUPJ(MAXJI), DAYSJ(NOOFI),
0030 R DAYOPI(NOOFI), ENDJ(NOOFI), PRICET(NOOFN, MAXT, MAXM),
0031 R SZEOPJ(MAXJI), SZEUPJ(MAXJI),
0032 R TQUANI(NOOFI, NOOFN, MAXT, MAXM), TQUANJ(MAXJI, NOOFN, MAXT),
0033 C NQBYM(NOOFN), N$BYM(NOOFN), OPNNEW(NOOFI, NOOFN), QOPNS(NOOFN),
0034 C QOUTQN(NOOFN)
0035

```

```

0001 C FILE NAME: ICBLKI
0002 C ORGNSM: VARIABLES DON'T CHANGE
0003 C USED FOR INCLUSION OF COMMON BLOCKS VARIABLES INITIALIZED IN BLOCK
0004 C DATA SUBPROGRAM
0005 COMMON / CBLKI /
0006 I NOOFT, OPTOK, OPNSOK, OPQOK, OPNOK, OUTQN,
0007 C NAMEM
0008 INTEGER NOOFT, OPTOK, OPNSOK, OPQOK, OPNOK, OUTQN
0009 CHARACTER * 10 NAMEM
0010 DIMENSION
0011 I NOOFT(NOOFN), OPTOK(NOOFN), OPNSOK(NOOFN), OPQOK(3),
0012 I OPNOK(NOOFI, NOOFN), OUTQN(NOOFN),
0013 C NAMEM(MAXM)
0014

```

```

0001 C FILE NAME: ICUNIT
0002 C ORGNSM: VARIABLES DON'T CHANGE WITH ORGANISM, BUT WITH MACHINE CAPABIT0002
0003 C ALL UNIT NUMBER FOR READ, WRITE, AND OUTPUT FILES
0004 COMMON / CUNIT /
0005 I RF, RI, WF, WI, WO, WON
0006 INTEGER RF, RI, WF, WI, WO, WON
0007 DIMENSION WON(NOOFN)
0008
0009

```

```

0001 C FILE NAME: ICBLKO
0002 C ORGNSM: VARIABLE DON'T CHANGE
0003 C FOR INCLUSION OF COMMON BLOCK DATA INITIALIZED, ORGANISM SPECIFIC
0004 COMMON / CBLKO /
0005 R EQNOA, EQNOAF, EQQ5A, EQQ5AF, EQQ5B, EQQ5BF,
0006 R EQSZA, EQSZAF, EQSZB, EQSZBF,
0007 C EQNO, EQQ5, EQSZE,
0008 C NAME, NAMEO, NAMEI, NAMEM, NAMEF, NAMEMU
0009 REAL EQNOA, EQNOAF, EQQ5A, EQQ5AF, EQQ5B, EQQ5BF,
0010 R EQSZA, EQSZAF, EQSZB, EQSZBF
0011 CHARACTER * 5 NAMEMU
0012 CHARACTER * 10 NAME, NAMEO, NAMEI, NAMEM, NAMEF
0013 CHARACTER * 62 EQNO, EQQ5, EQSZE
0014 DIMENSION
0015 R EQQ5A(2), EQQ5AF(2), EQQ5B(2), EQQ5BF(2),
0016 C EQNO(1), EQQ5(2), EQSZE(1),
0017 C NAMEO(NOOFI), NAMEI(NOOFI), NAMEM(NOOFN),
0018 C NAMEF(NOOFN, MAXT), NAMEMU(NOOFN)

```

```

C FILE NAME: ICORG
C ORGNSM: VARIABLES SPECIFIC TO ORGANISM
C CURRENT ORGANISM: BIVALVE
C TO INCLUDE COMMONS SPECIFIC TO ORGANISM
COMMON / CORG /
I QEQQ5,
R DAYS11, FILTER, NOO11, N5AVAL, N5REQ, QUAOFJ, SEACEL,
C NSABYM, SEABYM
INTEGER QEQQ5
REAL DAYS11, FILTER, NOO11, N5AVAL, N5REQ, QUAOFJ, SEACEL
CHARACTER*1 NSABYM, SEABYM
DIMENSION
R FILTER(MAX1), N5AVAL(MAX1,MAXM),
R NSREQ(MAX1,MAXT), QUAOFJ(MAXJI,MAXT), SEACEL(MAXI,MAXM)
0001
0002
0003
0004
0005
0006
0007
0008
0009
0010
0011
0012
0013
0014

C FILE NAME : IPARI
C ORGNSM: VARIABLES DON'T CHANGE; PARAMETERS ARE SPECIFIC TO ORGANISM
C CURRENT ORGANISM: BIVALVE
C FOR INCLUSION OF PARAMETERS
INTEGER
I MAXI, MAXJ, MAXJI, MAXM, MAXN, MAXT,
I NOOFO, NOOFI, NOOFN
PARAMETER
I (MAXI = 6, MAXJ = 10, MAXM = 12, MAXN = 6, MAXT = 6,
I MAXJI = MAXI * MAXJ,
I NOOFO = 2, NOOFI = 5, NOOFN = 6)
0001
0002
0003
0004
0005
0006
0007
0008
0009
0010
0011
0012

```

APPENDIX B

This appendix contains the data used in the base run of the simulation model and the output for the base run. All or portions of this information is printed at the user's request. However, what appears in the following pages is an edited version of what is actually printed by the program. Data were not available to estimate all component costs of bivalve seed. Thus, to save space, only that information pertaining to algae and total feed costs is reported. In addition, the algae price computations are reported only for one month, although algae cost differences by month are reflected in the bivalve feed costs reported in subsequent sections of the output.

BIVALVE COST SIMULATION MODEL: INPUTS ENTERED

DATE: MARCH 30, 1988
 OPTION DATA ENTERED
 NAME OF FILE USED: 1201 8DATA
 OPTION CHOSEN: 1

COST UP TO FIRST STAGE OF INTEREST IS ENTERED
 USER CHOOSES INPUTS TO COMPUTE IN REMAINING STAGES
 THE FIRST STAGE OF INTEREST IS THE SPAWN STAGE

IF C, THE PROGRAM COMPUTES THE QUANTITY
 IF E, THE USERS ENTERS THE QUANTITY
 IF -, THE QUANTITY IS NEITHER ENTERED NOR COMPUTED

INPUT	CONDITION	SPAWN	LARVAE	SETTING	JUVENILE
ELECTRICITY	-	E	E	E	E
FUEL OIL	-	E	E	E	E
LABOR	-	E	E	E	E
MATERIALS	-	E	E	E	E
ALGAE	-	C	C	C	C
CULCH	-	E	E	E	E

THE NUMBER OF EACH SECTION FOR WHICH DATA MUST BE ENTERED.

SECTION 1: GENERAL DATA
 SECTION 2: BIVALVE DATA
 SECTION 7: ALGAE DATA

BIVALVE COST SIMULATION MODEL: ALGAE PRICE COMPUTATIONS

DATE: MARCH 30, 1988 INPUTS ENTERED

1. ALGAE THESIS RESULTS

*PRICE INFORMATION	PRICE	UNIT
2. OF ELECTRICITY	\$ 0.1000	PER KWHR
3. OF FUEL OIL	\$ 0.2990	PER LITR
4. OF LIGHTBULBS	\$ 2.0000	PER BULB
4. OF NUTRIENT SOL.	\$ 1.0000	PER LITR
5. # OF LABOR TYPES: 3		
6. PRICE OF TYPE1	\$ 6.7310	PER HR
6. PRICE OF TYPE2	\$ 5.5290	PER HR
6. PRICE OF TYPE3	\$ 4.8080	PER HR

*MISC. INFORMATION - NON STAGE DEPENDENT

8. WATT RATING OF AIR CONDIT.	1050.0	WATT
9. WATT RATING OF LITERBULBS	40.0	WATT
10. LIFETIME OF LITERBULBS	15000.0	HRS
11. WATT RATING OF AUTOCLAVE	2000.0	WATT
12. COMPRESSOR HORSEPOWER	1.0	HP
13. HOURS COMPRESSOR ON/DAY	24.0	HRS
14. TOTAL # COMPRESSOR TUBES	74.	TUBE
15. EFF. RATING OF COMPRESSOR	0.850	
16. EFF. RATING OF WATER PUMP	0.450	
17. GPM RATING OF WATER PUMP	40.0	GPM
18. HP RATING OF WATER PUMP	2.5	HP
19. HOURS OF LABOR/WEEK		

OF TYPE1	11.5	HRS
OF TYPE2	3.0	HRS
OF TYPE3	0.0	HRS

BIVALVE COST SIMULATION MODEL: ALGAE PRICE COMPUTATIONS (cont.)

DATE: MARCH 30, 1988
 OUTPUT GENERATED

DATE OF PRICE COMPUTATIONS: ALGAE THESIS RESULTS

BATCH BEGUN: APRIL

	1STAGE		2STAGE		3STAGE		4STAGE		5STAGE		6STAGE	
	APRIL	APRIL	APRIL	APRIL	APRIL	APRIL	MAY	MAY	MAY	MAY	MAY	MAY
ALGAE AVAILABLE FOR USE IN MONTH												
*TOTAL LITRES PRODUCED	0.2500E-01	4.170	70.00	3406.	0.1722E+05	0.6718E+05						
LITRES OF ALGAE AVAILABLE	0.0000E+00	3.170	0.0000E+00	1511.	9652.	0.6718E+05						
CELLS OF ALGAE PRODUCED	0.7500E+07	0.1251E+10	0.2100E+11	0.1362E+13	0.1119E+14	0.4031E+14						
*TOTAL COST OF STAGE	50.64	80.90	49.79	110.3	109.3	107.1						
TOTAL COST PER LITRE OF ALGAE	2025.	19.40	0.7113	0.3240E-01	0.6347E-02	0.1594E-02						
TOTAL COST PER CELL OF ALGAE	0.6752E-05	0.6467E-07	0.2371E-08	0.8099E-10	0.9764E-11	0.2656E-11						
TOTAL COST OF STAGE, LESS INNOCULATION	50.64	30.26	30.39	60.55	47.91	59.01						
*TOTAL COST OF ELECTRICITY	9.060	13.47	7.716	24.41	18.79	13.47						
TOTAL COST OF ELECTRICITY/LITRE OF ALG	362.4	3.231	0.1102	0.7166E-02	0.1091E-02	0.2005E-03						
TOTAL COST OF ELEC. LESS INNOCULATION	9.060	4.414	4.484	16.69	5.208	5.210						
*TOTAL KWHR USED	90.60	134.7	77.16	244.1	187.9	134.7						
TOTAL KWHR USED PER LITRE OF ALGAE	3624.	32.31	1.102	0.7166E-01	0.1091E-01	0.2005E-02						
TOTAL KWHR USED LESS INNOCULATION	90.60	44.14	44.84	166.9	52.08	52.10						
*TOTAL COST OF FUEL OIL	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00						
TOTAL COST OF OIL/LITRE OF ALGAE	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00						
TOTAL COST OF OIL LESS INNOCULATION	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00						
*TOTAL LITRES OF FUEL OIL	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00						
TOTAL LITRES OIL USED/LITRE OF ALGAE	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00						
TOTAL L. OIL USED LESS INNOCULATION	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00						
*TOTAL COST OF ALL LABOR TYPES	41.43	67.19	41.87	84.46	87.18	88.12						
TOTAL COST ALL LABOR/LITRE OF ALGAE	1657.	16.11	0.5982	0.2480E-01	0.5062E-02	0.1312E-02						
TOTAL COST ALL LABOR LESS INNOCULATN	41.43	25.76	25.76	42.59	40.19	49.80						
*TOTAL HOURS OF ALL LABOR TYPES	6.333	10.25	6.375	12.79	13.53	14.37						
TOTAL HRS ALL LABOR USED/LITRE ALGAE	253.3	2.458	0.9107E-01	0.3756E-02	0.7858E-03	0.2138E-03						
TTL HRS ALL LABOR LESS INNOCULATION	6.333	3.917	3.917	6.417	6.417	8.417						
*TOTAL COST OF ALL MATERIALS	0.1494	0.2380	0.2017	1.470	3.333	5.468						
TTL COST ALL MATERIALS/LITRE OF ALGAE	5.975	0.5708E-01	0.2882E-02	0.4315E-03	0.1935E-03	0.8139E-04						
TTL COST ALL MATER. LESS INNOCULATION	0.1494	0.8867E-01	0.1446	1.268	2.515	4.003						

BIVALVE COST SIMULATION MODEL: INPUT ENTERED (CONT.) MARCH 30, 1988
 GENERAL DATA

1. MONTH THAT PRODUCTION SEASON BEGINS: JANUARY (1)
 2. MONTH THAT PRODUCTION SEASON ENDS: DECEMBER (12)
 2 STAGE 3 STAGE 4 STAGE 5 STAGE

3. NUMBER OF DAYS IN EACH STAGE:
 2.00 12.00 5.00 371.00

4. NUMBER OF PERIODS IN EACH STAGE:
 1 3 1 53

5. FOR OPTION 1, THE NUMBER OF DAYS UP TO STAGE 2: 30.00
 6. FOR OPTION 1, THE COST UP TO STAGE 2: 0.00

INFORMATION PERTAINING TO PRICE

INPUT: ELECTRICITY
 7. PRICE COMPUTED BY PROGRAM?: N
 INPUT: FUEL OIL
 8. PRICE COMPUTED BY PROGRAM?: N
 INPUT: LABOR
 9. PRICE COMPUTED BY PROGRAM?: N
 INPUT: MATERIALS
 10. PRICE COMPUTED BY PROGRAM?: N
 INPUT: ALGAE
 11. PRICE COMPUTED BY PROGRAM?: Y
 INPUT: CULCH
 12. PRICE COMPUTED BY PROGRAM?: N

32. PRICE OF ALGAE PER CELL (COMPUTED BY THE PROGRAM):

	JANUARY	FEBRUARY	MARCH	APRIL	MAY	JUNE
	JULY	AUGUST	SEPTEMBER	OCTOBER	NOVEMBER	DECEMBER
PRIMARY1 :	0.6464135E-05	0.6464135E-05	0.6464135E-05	0.6751600E-05	0.7039068E-05	0.7215468E-05
	0.7509468E-05	0.7444134E-05	0.7215468E-05	0.7012933E-05	0.6751600E-05	0.6464135E-05
PRIMARY2 :	0.6208467E-07	0.6208467E-07	0.6208467E-07	0.6466979E-07	0.6725492E-07	0.6884125E-07
	0.7148515E-07	0.7089761E-07	0.6884125E-07	0.6701993E-07	0.6466979E-07	0.6208467E-07
PRIMARY3 :	0.2282785E-08	0.2282785E-08	0.2282785E-08	0.2371048E-08	0.2459312E-08	0.2513474E-08
	0.2603744E-08	0.2583683E-08	0.2513474E-08	0.2451288E-08	0.2371048E-08	0.2282785E-08
SECONDARY :	0.7488353E-10	0.7488353E-10	0.7488353E-10	0.7725728E-10	0.8099153E-10	0.8380865E-10
	0.8707120E-10	0.8792311E-10	0.8572570E-10	0.8297109E-10	0.7985458E-10	0.7624403E-10
TERTIARY :	0.9350308E-11	0.9350308E-11	0.9350308E-11	0.9511042E-11	0.9763898E-11	0.9954654E-11
	0.1017557E-10	0.1023326E-10	0.1008446E-10	0.9897940E-11	0.9686914E-11	0.9442431E-11
SOLAR :	0.2605360E-11	0.2605360E-11	0.2605360E-11	0.2624981E-11	0.2655846E-11	0.2679130E-11
	0.2706097E-11	0.2713139E-11	0.2694976E-11	0.2672208E-11	0.2646449E-11	0.2616606E-11

BIVALVE COST SIMULATION MODEL: INPUT ENTERED (CONT.) MARCH 30, 1988
 BIVALVE DATA

- 1. TYPE OF BIVALVE 1 =OYSTERS
- 2. NAME OF SYSTEM UPWELL
- 3. NUMBER DAYS IN STAGE CONDITION 30.000
- 4. NUMBER BIVALVES IN STAGE CONDITION 2.0000
- 5. NUMBER OF OYSTERS BATCH BEGUN WITH: 1000000.

EQUATION TO DETERMINE SURVIVAL:

POPULATION = POPULATION(INITIAL) * [(WEEKS+1)**A]
 6. PARAMETER A = -0.2550680

EQUATION TO EQUATE AGE TO SIZE:

SIZE = (A * AGE) + (B * AGE SQUARED)
 7. PARAMETER A = 0.7370000
 8. PARAMETER B = -0.2520000E-02

BIVALVE COST SIMULATION MODEL: INPUT ENTERED (CONT.) MARCH 30, 1988
 ALGAE DATA

- 1. AGE/ALGAE EQUATION CHOSEN: 1
 NATURAL LOG(SIZE IN MM) = A + B * NAT LOG(10,000 CELLS)
- 2. PARAMETER A = -0.6310480
- 3. PARAMETER B = 0.3384260

IN THE FOLLOWING, WHERE QUANTITY=0, QUANTITY IS ENTERED

4. FILTERING RATE	2 STAGE	3 STAGE	4 STAGE	5 STAGE
0.50	0.50	0.50	0.50	0.50
5. FRACTION OF EACH ALGAE TYPE REQUIRED PER STAGE:				
TYPE PRIMARY1	0.00E+00	0.00E+00	0.00E+00	0.00E+00
TYPE PRIMARY2	0.00E+00	0.00E+00	0.00E+00	0.00E+00
TYPE PRIMARY3	0.00E+00	0.00E+00	0.00E+00	0.00E+00
TYPE SECONDARY	0.00E+00	0.00E+00	0.00E+00	0.00E+00
TYPE TERTIARY	1.0	1.0	1.0	0.00E+00
TYPE SOLAR	0.00E+00	0.00E+00	0.00E+00	1.0
6. FRACTION FROM SEA VARIES BY MONTH?: N				
7. FRACTION OF DIET FROM SEA ALGAE IN EACH STAGE, EACH MONTH:				
JANUARY	0.00E+00	0.00E+00	0.00E+00	0.12
FEBRUARY	0.00E+00	0.00E+00	0.00E+00	0.12
MARCH	0.00E+00	0.00E+00	0.00E+00	0.12
APRIL	0.00E+00	0.00E+00	0.00E+00	0.12
MAY	0.00E+00	0.00E+00	0.00E+00	0.12
JUNE	0.00E+00	0.00E+00	0.00E+00	0.12
JULY	0.00E+00	0.00E+00	0.00E+00	0.12
AUGUST	0.00E+00	0.00E+00	0.00E+00	0.12
SEPTEMBER	0.00E+00	0.00E+00	0.00E+00	0.12
OCTOBER	0.00E+00	0.00E+00	0.00E+00	0.12
NOVEMBER	0.00E+00	0.00E+00	0.00E+00	0.12
DECEMBER	0.00E+00	0.00E+00	0.00E+00	0.12

BIVALVE COST SIMULATION MODEL: OUTPUT GENERATED

DATE: MARCH 30, 1988
 FILE: 1201 SDATA

BIVALVE TYPE: OYSTERS
 NUMBER BIVALVE BEGUN: .10000E+07
 SYSTEM USED: UPWELL

OPTION CHOSEN: 1, FIRST STAGE OF INTEREST IS THE SPAWN STAGE
 (*NOTE: STAGE 1 INCLUDES ALL STAGES UP TO THAT STAGE)

PRICE OF INPUTS PER MONTH

MONTH	JANUARY		FEBRUARY		MARCH		APRIL		MAY		JUNE	
	JULY	AUGUST	SEPTEMBER	OCTOBER	NOVEMBER	DECEMBER	NOVEMBER	DECEMBER	NOVEMBER	DECEMBER	NOVEMBER	DECEMBER
ELECTRICITY/KWHR	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
FUEL OIL /LITRE	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
LABOR /HOUR	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
MATERIALS /UNIT	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
PRIMARY1 /CELL	0.6464135E-05	0.6464135E-05	0.6464135E-05	0.6464135E-05	0.6464135E-05	0.6464135E-05	0.6464135E-05	0.6464135E-05	0.6464135E-05	0.6464135E-05	0.6464135E-05	0.6464135E-05
PRIMARY2 /CELL	0.7509468E-05	0.7444134E-05	0.7215468E-05	0.7012933E-05	0.6751600E-05	0.651600E-05	0.625492E-07	0.6084125E-07	0.5884125E-07	0.5684125E-07	0.5484125E-07	0.5284125E-07
PRIMARY3 /CELL	0.7148515E-07	0.7089761E-07	0.6884125E-07	0.6701993E-07	0.6466979E-07	0.6208467E-07	0.595151E-07	0.569797E-07	0.544343E-07	0.518889E-07	0.493435E-07	0.467981E-07
SECONDARY /CELL	0.260374E-08	0.2282785E-08	0.2282785E-08	0.2371048E-08	0.2459312E-08	0.2513474E-08	0.2577616E-08	0.2641758E-08	0.2705900E-08	0.2770042E-08	0.2834184E-08	0.2898326E-08
TERTIARY /CELL	0.8707120E-10	0.8792311E-10	0.8877502E-10	0.8962693E-10	0.9047884E-10	0.9133075E-10	0.9218266E-10	0.9303457E-10	0.9388648E-10	0.9473839E-10	0.9559030E-10	0.9644221E-10
SOLAR /CELL	0.1017557E-10	0.1023326E-10	0.1029095E-10	0.1034864E-10	0.1040633E-10	0.1046402E-10	0.1052171E-10	0.1057940E-10	0.1063709E-10	0.1069478E-10	0.1075247E-10	0.1081016E-10
CULCH /LITRE	0.2706097E-11	0.2713139E-11	0.2720181E-11	0.2727223E-11	0.2734265E-11	0.2741307E-11	0.2748349E-11	0.2755391E-11	0.2762433E-11	0.2769475E-11	0.2776517E-11	0.2783559E-11

PRD	*STG	NAME OF STAGE	COST SIMULATION MODEL: OUTPUT GENERATED (CONT.)			MARCH 30, 1988			BATCH BEGUN: APRIL			SIZE BIVALV AT END PERIOD
			TIME UP TO	AGE AT	END OF PERIOD	AVG NO. BIVALV NO. BIVALV AT	END OF PERIOD	IN PERIOD	AVG SIZE	IN PERIOD		
			MNTH END OF PERIOD	END OF PERIOD	END OF PERIOD	IN PERIOD	END OF PERIOD	IN PERIOD	IN PERIOD	IN PERIOD		
1	1	CONDITION	4	30.00000	730.0000	2.000000	2.000000	2.000000	49.45227	49.45227	49.45227	49.45227
2	2	SPAWN	4	32.00000	2.000000	967734.0	967734.0	967734.0	0.1052085	0.1052085	0.1052085	0.2103657
3	3	LARVAE	5	36.00000	6.000000	892905.5	892905.5	892905.5	0.3939469	0.3939469	0.3939469	0.6298628
4	3	LARVAE	5	40.00000	10.00000	824216.0	824216.0	824216.0	0.7601836	0.7601836	0.7601836	1.047713
5	3	LARVAE	5	44.00000	14.00000	775672.3	775672.3	775672.3	1.249980	1.249980	1.249980	1.463920
6	4	SETTING	5	49.00000	19.00000	734727.4	734727.4	734727.4	1.676703	1.676703	1.676703	1.981861
7	5	JUVENILE	5	56.00000	26.00000	693416.0	693416.0	693416.0	2.272539	2.272539	2.272539	2.702662
8	5	JUVENILE	6	63.00000	33.00000	641096.2	641096.2	641096.2	3.010759	3.010759	3.010759	3.418421
9	5	JUVENILE	6	70.00000	40.00000	627751.0	627751.0	627751.0	3.744302	3.744302	3.744302	4.129142
10	5	JUVENILE	6	77.00000	47.00000	604251.8	604251.8	604251.8	4.473164	4.473164	4.473164	4.834821
11	5	JUVENILE	6	84.00000	54.00000	584536.4	584536.4	584536.4	5.125919	5.125919	5.125919	5.535461
12	5	JUVENILE	7	91.00000	61.00000	567632.6	567632.6	567632.6	5.845417	5.845417	5.845417	6.231061
13	5	JUVENILE	7	98.00000	68.00000	552892.6	552892.6	552892.6	6.560244	6.560244	6.560244	6.921621
14	5	JUVENILE	7	105.0000	75.00000	539864.1	539864.1	539864.1	7.198956	7.198956	7.198956	7.607141
15	5	JUVENILE	7	112.0000	82.00000	528219.8	528219.8	528219.8	7.904419	7.904419	7.904419	8.287622
16	5	JUVENILE	7	119.0000	89.00000	517716.3	517716.3	517716.3	8.605203	8.605203	8.605203	8.963061
17	5	JUVENILE	8	126.0000	96.00000	508166.7	508166.7	508166.7	9.229876	9.229876	9.229876	9.633462
18	5	JUVENILE	8	133.0000	103.0000	499426.1	499426.1	499426.1	9.921297	9.921297	9.921297	10.29882
19	5	JUVENILE	8	140.0000	110.0000	491378.7	491378.7	491378.7	10.60804	10.60804	10.60804	10.95914
20	5	JUVENILE	8	147.0000	117.0000	483931.8	483931.8	483931.8	11.21867	11.21867	11.21867	11.61442
21	5	JUVENILE	9	154.0000	124.0000	477009.6	477009.6	477009.6	11.89605	11.89605	11.89605	12.26466
22	5	JUVENILE	9	161.0000	131.0000	470548.7	470548.7	470548.7	12.56876	12.56876	12.56876	12.90986
23	5	JUVENILE	9	168.0000	138.0000	464496.7	464496.7	464496.7	13.16535	13.16535	13.16535	13.55002
24	5	JUVENILE	9	175.0000	145.0000	458809.6	458809.6	458809.6	13.82870	13.82870	13.82870	14.18514
25	5	JUVENILE	10	182.0000	152.0000	453449.7	453449.7	453449.7	14.48736	14.48736	14.48736	14.81521
26	5	JUVENILE	10	189.0000	159.0000	448384.1	448384.1	448384.1	15.06990	15.06990	15.06990	15.44025
27	5	JUVENILE	10	196.0000	166.0000	443585.7	443585.7	443585.7	15.71921	15.71921	15.71921	16.06024
28	5	JUVENILE	10	203.0000	173.0000	439029.6	439029.6	439029.6	16.36383	16.36383	16.36383	16.67520
29	5	JUVENILE	10	210.0000	180.0000	434695.1	434695.1	434695.1	16.93234	16.93234	16.93234	17.28513
30	5	JUVENILE	11	217.0000	187.0000	430563.1	430563.1	430563.1	17.56760	17.56760	17.56760	17.89000
31	5	JUVENILE	11	224.0000	194.0000	426617.7	426617.7	426617.7	18.12675	18.12675	18.12675	18.48984
32	5	JUVENILE	11	231.0000	201.0000	422844.1	422844.1	422844.1	18.75266	18.75266	18.75266	19.08464
33	5	JUVENILE	11	238.0000	208.0000	419229.0	419229.0	419229.0	19.37389	19.37389	19.37389	19.67441
34	5	JUVENILE	12	245.0000	215.0000	415761.1	415761.1	415761.1	19.91899	19.91899	19.91899	20.25912
35	5	JUVENILE	12	252.0000	222.0000	412430.1	412430.1	412430.1	20.53085	20.53085	20.53085	20.83881
36	5	JUVENILE	12	259.0000	229.0000	409226.6	409226.6	409226.6	21.06662	21.06662	21.06662	21.41344
37	5	JUVENILE	12	266.0000	236.0000	406141.7	406141.7	406141.7	21.66911	21.66911	21.66911	21.98305
38	5	JUVENILE	1	273.0000	243.0000	403167.8	403167.8	403167.8	22.19551	22.19551	22.19551	22.54761
39	5	JUVENILE	1	280.0000	250.0000	400298.1	400298.1	400298.1	22.78865	22.78865	22.78865	23.07112
40	5	JUVENILE	1	287.0000	257.0000	397526.6	397526.6	397526.6	23.37712	23.37712	23.37712	23.66161
41	5	JUVENILE	1	294.0000	264.0000	394846.6	394846.6	394846.6	23.88947	23.88947	23.88947	24.21104
42	5	JUVENILE	2	301.0000	271.0000	392253.4	392253.4	392253.4	24.46858	24.46858	24.46858	24.75545
43	5	JUVENILE	2	308.0000	278.0000	389742.0	389742.0	389742.0	25.04131	25.04131	25.04131	25.29480
44	5	JUVENILE	2	315.0000	285.0000	387307.6	387307.6	387307.6	25.54131	25.54131	25.54131	25.82912
45	5	JUVENILE	2	322.0000	292.0000	384946.1	384946.1	384946.1	26.03494	26.03494	26.03494	26.35840
46	5	JUVENILE	2	329.0000	299.0000	382654.3	382654.3	382654.3	26.59534	26.59534	26.59534	26.88266
47	5	JUVENILE	3	336.0000	306.0000	380427.6	380427.6	380427.6	27.07961	27.07961	27.07961	27.40186
48	5	JUVENILE	3	343.0000	313.0000	378263.6	378263.6	378263.6	27.63063	27.63063	27.63063	27.91602
49	5	JUVENILE	3	350.0000	320.0000	376158.6	376158.6	376158.6	28.10555	28.10555	28.10555	28.42513
50	5	JUVENILE	3	357.0000	327.0000	374110.3	374110.3	374110.3	28.64720	28.64720	28.64720	28.92921
51	5	JUVENILE	4	364.0000	334.0000	372115.6	372115.6	372115.6	29.11276	29.11276	29.11276	29.42825
52	5	JUVENILE	4	371.0000	341.0000	370172.0	370172.0	370172.0	29.64507	29.64507	29.64507	29.92226
53	5	JUVENILE	4	378.0000	348.0000	368277.6	368277.6	368277.6	30.10126	30.10126	30.10126	30.41121
54	5	JUVENILE	4	385.0000	355.0000	366430.0	366430.0	366430.0	30.62421	30.62421	30.62421	30.89513
55	5	JUVENILE	5	392.0000	362.0000	364626.7	364626.7	364626.7	31.07103	31.07103	31.07103	31.37401
56	5	JUVENILE	5	399.0000	369.0000	362866.6	362866.6	362866.6	31.58463	31.58463	31.58463	31.84785
57	5	JUVENILE	5	406.0000	376.0000	361147.4	361147.4	361147.4	32.02209	32.02209	32.02209	32.31665
58	5	JUVENILE	5	413.0000	383.0000	359467.6	359467.6	359467.6	32.52632	32.52632	32.52632	32.78041
59	5	JUVENILE	5	420.0000	390.0000	357825.6	357825.6	357825.6	32.95444	32.95444	32.95444	33.23914

PERIOD	STAGE	NAME OF STAGE	MMTH	COST OF PERIOD	COST UP TO END OF PERIOD	AVG COST IN PERIOD	BIV AT END	COST BIVALV AT END
1	1	CONDITION	4	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00
2	2	SPAWN	5	0.7474519E-02	0.7474519E-02	0.3861864E-08	0.7969341E-08	0.7969341E-08
3	3	LARVAE	5	0.4317153	0.4391898	0.1208737E-06	0.5143113E-06	0.5143113E-06
4	3	LARVAE	5	2.615227	3.054416	0.7932467E-06	0.3830176E-05	0.3830176E-05
5	3	LARVAE	5	7.857286	10.91170	0.2532412E-05	0.1444079E-04	0.1444079E-04
6	4	SETTING	5	23.55299	34.46469	0.6411356E-05	0.4816496E-04	0.4816496E-04
7	5	JUVENILE	5	18.45863	52.92332	0.3802835E-05	0.7859844E-04	0.7859844E-04
8	5	JUVENILE	6	38.54652	91.46985	0.8386842E-05	0.1426773E-03	0.1426773E-03
9	5	JUVENILE	6	68.17392	159.6438	0.1551432E-04	0.2594735E-03	0.2594735E-03
10	5	JUVENILE	6	108.8479	268.4915	0.3953188E-04	0.4521173E-03	0.4521173E-03
11	5	JUVENILE	6	161.7548	430.2461	0.3953188E-04	0.7473773E-03	0.7473773E-03
12	5	JUVENILE	7	230.2422	660.4883	0.5794545E-04	0.1179565E-02	0.1179565E-02
13	5	JUVENILE	7	311.4697	971.9580	0.8047794E-04	0.1779747E-02	0.1779747E-02
14	5	JUVENILE	7	407.9121	1379.870	0.1079404E-03	0.2584838E-02	0.2584838E-02
15	5	JUVENILE	7	520.3247	1900.195	0.1407219E-03	0.3634693E-02	0.3634693E-02
16	5	JUVENILE	7	649.3794	2549.574	0.1791879E-03	0.4971918E-02	0.4971918E-02
17	5	JUVENILE	8	797.7424	3347.317	0.2242634E-03	0.6643836E-02	0.6643836E-02
18	5	JUVENILE	8	962.2297	4309.543	0.2752387E-03	0.8700971E-02	0.8700971E-02
19	5	JUVENILE	8	1144.994	5454.535	0.3328808E-03	0.1118737E-01	0.1118737E-01
20	5	JUVENILE	8	1346.435	6800.969	0.3974687E-03	0.1415722E-01	0.1415722E-01
21	5	JUVENILE	9	1556.416	8357.383	0.4661228E-03	0.1764254E-01	0.1764254E-01
22	5	JUVENILE	9	1794.602	10151.98	0.5448356E-03	0.2171741E-01	0.2171741E-01
23	5	JUVENILE	9	2052.225	12204.21	0.6311669E-03	0.2643918E-01	0.2643918E-01
24	5	JUVENILE	9	2329.478	14533.68	0.7253170E-03	0.3186667E-01	0.3186667E-01
25	5	JUVENILE	10	2604.325	17138.01	0.8204796E-03	0.3801095E-01	0.3801095E-01
26	5	JUVENILE	10	2918.578	20056.59	0.9298711E-03	0.4497571E-01	0.4497571E-01
27	5	JUVENILE	10	3252.629	23309.21	0.1047512E-02	0.5282313E-01	0.5282313E-01
28	5	JUVENILE	10	3606.510	26915.72	0.1173322E-02	0.6161642E-01	0.6161642E-01
29	5	JUVENILE	10	3980.225	30895.95	0.1308051E-02	0.7141972E-01	0.7141972E-01
30	5	JUVENILE	11	4331.574	35227.52	0.1437179E-02	0.8219957E-01	0.8219957E-01
31	5	JUVENILE	11	4740.824	39968.34	0.1587512E-02	0.9410876E-01	0.9410876E-01
32	5	JUVENILE	11	5169.527	45137.87	0.1746516E-02	0.1072130	0.1072130
33	5	JUVENILE	11	5617.547	50755.42	0.1914244E-02	0.1215781	0.1215781
34	5	JUVENILE	12	6016.117	56771.54	0.2067161E-02	0.1371048	0.1371048
35	5	JUVENILE	12	6496.813	63268.35	0.2250359E-02	0.1540095	0.1540095
36	5	JUVENILE	12	6996.082	70264.38	0.2442266E-02	0.1723580	0.1723580
37	5	JUVENILE	12	7513.711	77778.06	0.2642889E-02	0.1922169	0.1922169
38	5	JUVENILE	1	8014.852	85792.88	0.2839955E-02	0.2135657	0.2135657
39	5	JUVENILE	1	8566.055	94358.88	0.3057026E-02	0.2365496	0.2365496
40	5	JUVENILE	1	9134.746	103493.6	0.3282708E-02	0.2612339	0.2612339
41	5	JUVENILE	2	10323.35	113214.1	0.3516958E-02	0.2876840	0.2876840
42	5	JUVENILE	2	10942.60	123537.4	0.3759722E-02	0.3159648	0.3159648
43	5	JUVENILE	2	11578.04	146058.0	0.4010931E-02	0.3461406	0.3461406
44	5	JUVENILE	2	12229.26	158287.3	0.4270520E-02	0.3782755	0.3782755
45	5	JUVENILE	2	12895.91	171183.1	0.4538391E-02	0.4124326	0.4124326
46	5	JUVENILE	2	13577.59	184760.7	0.4814457E-02	0.4486746	0.4486746
47	5	JUVENILE	3	14273.87	199034.5	0.5098619E-02	0.4870635	0.4870635
48	5	JUVENILE	3	14984.34	214018.8	0.5390748E-02	0.5276604	0.5276604
49	5	JUVENILE	3	15708.59	229727.4	0.5690739E-02	0.5705256	0.5705256
50	5	JUVENILE	4	16570.00	246297.4	0.5998455E-02	0.6157185	0.6157185
51	5	JUVENILE	4	17326.07	263623.4	0.6361309E-02	0.6636314	0.6636314
52	5	JUVENILE	4	18094.64	281718.1	0.6686494E-02	0.7140064	0.7140064
53	5	JUVENILE	4	18875.22	300593.3	0.7019021E-02	0.7669003	0.7669003
54	5	JUVENILE	4	19898.57	320491.8	0.7358730E-02	0.8223686	0.8223686
55	5	JUVENILE	5	20711.18	341202.9	0.7796064E-02	0.8811012	0.8811012
56	5	JUVENILE	5	21534.45	362737.4	0.8153796E-02	0.9425476	0.9425476
57	5	JUVENILE	5	22367.88	385105.3	0.8518264E-02	1.006761	1.006761
58	5	JUVENILE	5	23210.93	408316.1	0.8889288E-02	1.073791	1.073791
59	5	JUVENILE	5			0.9266656E-02	1.143687	1.143687

QUANTITY OF INPUT USED PER PERIOD:

ALGAE	CELL :	SOLAR
TERTIARY		
0.7655265E+09	0.0000000E+00	0.0000000E+00
0.4421547E+11	0.0000000E+00	0.0000000E+00
0.2678467E+12	0.0000000E+00	0.0000000E+00
0.8047285E+12	0.0000000E+00	0.0000000E+00
0.2412253E+13	0.0000000E+00	0.0000000E+00
0.0000000E+00	0.6950191E+13	0.1438770E+14
0.0000000E+00	0.0000000E+00	0.2544629E+14
0.0000000E+00	0.0000000E+00	0.4062808E+14
0.0000000E+00	0.0000000E+00	0.6037587E+14
0.0000000E+00	0.8508278E+14	0.8058278E+14
0.0000000E+00	0.1150993E+15	0.1150993E+15
0.0000000E+00	0.1507383E+15	0.1507383E+15
0.0000000E+00	0.1922787E+15	0.1922787E+15
0.0000000E+00	0.2399691E+15	0.2399691E+15
0.0000000E+00	0.2940294E+15	0.2940294E+15
0.0000000E+00	0.3546556E+15	0.3546556E+15
0.0000000E+00	0.4220181E+15	0.4220181E+15
0.0000000E+00	0.4962647E+15	0.4962647E+15
0.0000000E+00	0.5775249E+15	0.5775249E+15
0.0000000E+00	0.6659065E+15	0.6659065E+15
0.0000000E+00	0.7615004E+15	0.7615004E+15
0.0000000E+00	0.8643780E+15	0.8643780E+15
0.0000000E+00	0.9745968E+15	0.9745968E+15
0.0000000E+00	0.1092197E+16	0.1092197E+16
0.0000000E+00	0.1217207E+16	0.1217207E+16
0.0000000E+00	0.1349637E+16	0.1349637E+16
0.0000000E+00	0.1489489E+16	0.1489489E+16
0.0000000E+00	0.16366750E+16	0.16366750E+16
0.0000000E+00	0.1791391E+16	0.1791391E+16
0.0000000E+00	0.1953383E+16	0.1953383E+16
0.0000000E+00	0.2122674E+16	0.2122674E+16
0.0000000E+00	0.2299207E+16	0.2299207E+16
0.0000000E+00	0.2482916E+16	0.2482916E+16
0.0000000E+00	0.2673726E+16	0.2673726E+16
0.0000000E+00	0.2871549E+16	0.2871549E+16
0.0000000E+00	0.3076293E+16	0.3076293E+16
0.0000000E+00	0.3287859E+16	0.3287859E+16
0.0000000E+00	0.3506136E+16	0.3506136E+16
0.0000000E+00	0.3731006E+16	0.3731006E+16
0.0000000E+00	0.3962351E+16	0.3962351E+16
0.0000000E+00	0.4200033E+16	0.4200033E+16
0.0000000E+00	0.4443929E+16	0.4443929E+16
0.0000000E+00	0.4693984E+16	0.4693984E+16
0.0000000E+00	0.4949761E+16	0.4949761E+16
0.0000000E+00	0.5211406E+16	0.5211406E+16
0.0000000E+00	0.5478656E+16	0.5478656E+16
0.0000000E+00	0.5751352E+16	0.5751352E+16
0.0000000E+00	0.6029335E+16	0.6029335E+16
0.0000000E+00	0.6312429E+16	0.6312429E+16
0.0000000E+00	0.6600459E+16	0.6600459E+16
0.0000000E+00	0.6893246E+16	0.6893246E+16
0.0000000E+00	0.7190613E+16	0.7190613E+16
0.0000000E+00	0.7492369E+16	0.7492369E+16
0.0000000E+00	0.7798338E+16	0.7798338E+16
0.0000000E+00	0.8108318E+16	0.8108318E+16
0.0000000E+00	0.8422130E+16	0.8422130E+16
0.0000000E+00	0.8739356E+16	0.8739356E+16

COST OF INPUTS UP TO END OF A PERIOD

ALGAE	CELL :	SOLAR
TERTIARY		
0.7474519E-02	0.7474519E-02	0.7474519E-02
0.4391898	0.4391898	0.4391898
3.054416	3.054416	3.054416
7.857286	7.857286	7.857286
34.46469	34.46469	34.46469
52.92332	52.92332	52.92332
91.46985	91.46985	91.46985
159.6438	159.6438	159.6438
268.4915	268.4915	268.4915
430.2461	430.2461	430.2461
660.4883	660.4883	660.4883
971.9580	971.9580	971.9580
1379.870	1379.870	1379.870
1900.195	1900.195	1900.195
2549.574	2549.574	2549.574
3347.317	3347.317	3347.317
4309.543	4309.543	4309.543
5454.535	5454.535	5454.535
6800.969	6800.969	6800.969
8357.383	8357.383	8357.383
10151.98	10151.98	10151.98
12204.21	12204.21	12204.21
14533.68	14533.68	14533.68
17138.01	17138.01	17138.01
20056.59	20056.59	20056.59
23309.21	23309.21	23309.21
26915.72	26915.72	26915.72
30895.95	30895.95	30895.95
35227.52	35227.52	35227.52
39968.34	39968.34	39968.34
45137.87	45137.87	45137.87
50755.42	50755.42	50755.42
56771.54	56771.54	56771.54
63268.35	63268.35	63268.35
70264.38	70264.38	70264.38
77778.06	77778.06	77778.06
85792.88	85792.88	85792.88
94358.88	94358.88	94358.88
103493.6	103493.6	103493.6
113214.1	113214.1	113214.1
123337.4	123337.4	123337.4
134480.0	134480.0	134480.0
146058.0	146058.0	146058.0
158287.3	158287.3	158287.3
171183.1	171183.1	171183.1
184760.7	184760.7	184760.7
199034.5	199034.5	199034.5
214018.8	214018.8	214018.8
229727.4	229727.4	229727.4
246297.4	246297.4	246297.4
263623.4	263623.4	263623.4
281718.1	281718.1	281718.1
300593.3	300593.3	300593.3
320491.8	320491.8	320491.8
341202.9	341202.9	341202.9
362737.4	362737.4	362737.4
385105.3	385105.3	385105.3
408316.1	408316.1	408316.1

COST OF INPUTS IN A PERIOD:

ALGAE	CELL :	SOLAR
TERTIARY		
0.7474519E-02	0.7474519E-02	0.7474519E-02
0.4317153	0.4317153	0.4317153
2.615227	2.615227	2.615227
7.857286	7.857286	7.857286
23.55299	23.55299	23.55299
18.45863	18.45863	18.45863
38.54652	38.54652	38.54652
68.17392	68.17392	68.17392
108.8479	108.8479	108.8479
161.7548	161.7548	161.7548
230.2422	230.2422	230.2422
311.4697	311.4697	311.4697
407.9121	407.9121	407.9121
520.3247	520.3247	520.3247
649.3794	649.3794	649.3794
797.7424	797.7424	797.7424
962.2297	962.2297	962.2297
1144.994	1144.994	1144.994
1346.435	1346.435	1346.435
1556.416	1556.416	1556.416
1794.602	1794.602	1794.602
2052.225	2052.225	2052.225
2329.478	2329.478	2329.478
2604.325	2604.325	2604.325
2918.578	2918.578	2918.578
3252.629	3252.629	3252.629
3606.510	3606.510	3606.510
3980.225	3980.225	3980.225
4331.574	4331.574	4331.574
4740.824	4740.824	4740.824
5169.527	5169.527	5169.527
5617.547	5617.547	5617.547
6016.117	6016.117	6016.117
6496.813	6496.813	6496.813
6996.082	6996.082	6996.082
7513.711	7513.711	7513.711
8014.852	8014.852	8014.852
8566.055	8566.055	8566.055
9134.746	9134.746	9134.746
9720.613	9720.613	9720.613
10323.35	10323.35	10323.35
10942.60	10942.60	10942.60
11578.04	11578.04	11578.04
12229.26	12229.26	12229.26
12895.91	12895.91	12895.91
13577.59	13577.59	13577.59
14273.87	14273.87	14273.87
14984.34	14984.34	14984.34
15708.59	15708.59	15708.59
16570.00	16570.00	16570.00
17326.07	17326.07	17326.07
18094.64	18094.64	18094.64
18875.22	18875.22	18875.22
19898.57	19898.57	19898.57
20711.18	20711.18	20711.18
21534.45	21534.45	21534.45
22367.88	22367.88	22367.88
23210.93	23210.93	23210.93

PRD	*STAGE	MNTH
2	2 SPAWN	5
3	3 LARVAE	5
4	3 LARVAE	5
5	3 LARVAE	5
6	4 SETTING	5
7	5 JUVENILE	5
8	5 JUVENILE	6
9	5 JUVENILE	6
10	5 JUVENILE	6
11	5 JUVENILE	6
12	5 JUVENILE	7
13	5 JUVENILE	7
14	5 JUVENILE	7
15	5 JUVENILE	7
16	5 JUVENILE	7
17	5 JUVENILE	8
18	5 JUVENILE	8
19	5 JUVENILE	8
20	5 JUVENILE	8
21	5 JUVENILE	9
22	5 JUVENILE	9
23	5 JUVENILE	9
24	5 JUVENILE	9
25	5 JUVENILE	10
26	5 JUVENILE	10
27	5 JUVENILE	10
28	5 JUVENILE	10
29	5 JUVENILE	11
30	5 JUVENILE	11
31	5 JUVENILE	11
32	5 JUVENILE	11
33	5 JUVENILE	11
34	5 JUVENILE	12
35	5 JUVENILE	12
36	5 JUVENILE	12
37	5 JUVENILE	12
38	5 JUVENILE	1
39	5 JUVENILE	1
40	5 JUVENILE	1
41	5 JUVENILE	1
42	5 JUVENILE	2
43	5 JUVENILE	2
44	5 JUVENILE	2
45	5 JUVENILE	2
46	5 JUVENILE	2
47	5 JUVENILE	3
48	5 JUVENILE	3
49	5 JUVENILE	3
50	5 JUVENILE	3
51	5 JUVENILE	4
52	5 JUVENILE	4
53	5 JUVENILE	4
54	5 JUVENILE	4
55	5 JUVENILE	5
56	5 JUVENILE	5
57	5 JUVENILE	5
58	5 JUVENILE	5
59	5 JUVENILE	5