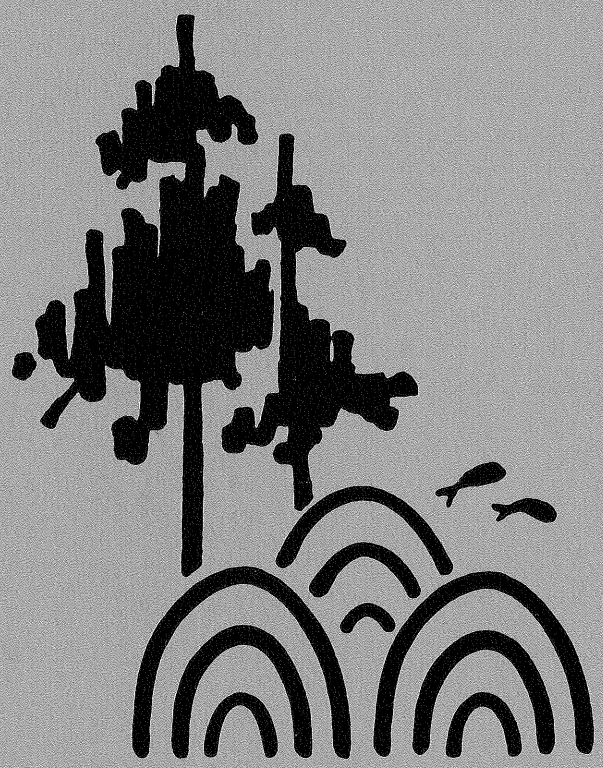
A Computer Program Package For Use With The Southern Pine Seed Orchard Inventory-Monitoring System



Publication No. FWS-2-82
School of Forestry and Wildlife Resources
Virginia Polytechnic Institute and State University
Blacksburg, Virginia 24061

1982

A COMPUTER PROGRAM PACKAGE FOR USE WITH THE SOUTHERN PINE SEED ORCHARD INVENTORY-MONITORING SYSTEM

by
Scott A. Merkle
Peter P. Feret
David L. Bramlett
Donald L. Queijo

Publication No. FWS-2-82
School of Forestry and Wildlife Resources
Virginia Polytechnic Institute and State University
Blacksburg, Virginia 24061
1982

ACKNOWLEDGEMENTS

The development of the Computerized Inventory-Monitoring System was funded by the U.S.D.A. Forest Service, Southeastern Forest Experiment Station.

The authors wish to acknowledge the Weyerhaeuser Company for its assistance with the development of the computerized IMS. Individual thanks are due to Mary Winslow, Ron Campbell, Michael Greenwood and Steven Duke.

AUTHORS

The authors are respectively Graduate Research Assistant and Associate Professor, Department of Forestry, VPI & SU, Blacksburg, VA, Project Leader USDA Forest Service, Southeastern Forest Experiment Station, Macon, GA, and Computer Programmer, Department of Forestry, VPI & SU Blacksburg, VA.

PROGRAM AVAILABILITY

The IMS computer package is currently on file at the Virginia Tech Computing Center. Those desiring tape copies of the package should send a nonlabeled, 1600 BPI tape to:

Dr. Peter P. Feret
Forestry Department
228 Cheatham Hall
VPI & SU
Blacksburg, VA 24061

TABLE OF CONTENTS

INTRODUCTION

Southern pine seed orchards covering more than 10,000 acres currently produce over 160,000 pounds of improved seed having a potential of 1 billion seedlings annually. By the year 2000, annual seed production is expected to reach 500,000 pounds. Seed orchards not only represent potential for improved growth, wood quality and pest resistance, but they also represent a large capital investment in orchard establishment and equipment and the significant annual costs of orchard maintenance, protection and harvesting. Since the immediate goal of the seed orchard investment is the annual production of cones and seed, the task of the seed orchard manager could be greatly lightened by a system that will forecast annual cone and seed crops and monitor production efficiency. Such a system has been developed by Bramlett and Godbee (1982). In the Inventory- Monitoring System (IMS) a set of sample trees are chosen from the seed orchard population. Then, based on the survival of cones on tagged sample branches in each sample tree, the expected number of cones and seed from the orchard can be predicted as early as 18 months prior to cone harvest. Bramlett and Godbee (1982) detailed various procedures for the selection of sample trees as well as methods for choosing sample branches and conducting flower, conelet and cone counts. Besides providing guidelines for data collection in the orchard, the authors also defined the variables used in the calculation of predicted cone and seed yields, including cone efficiency, seed potential, seed efficiency, extraction efficiency and germination efficiency. They showed how to compute (or update) the values of these variables and how to apply them in models to calculate predicted bushels of cones, predicted pounds of seed, predicted number of seedlings and other predicted values for the orchard. In addition to these predicted values, Bramlett and Godbee (1982) also demonstrated how the IMS can be used by the orchard manager to evaluate orchard productivity, identify the factors reducing yields, and formulate corrective action, including fertilization and pest management.

Because a great deal of record-keeping and repetitive mathematical operations are involved in the IMS, it is ideally suited to computerization. A computerized version of the IMS not only has the advantage of efficient data storage and manipulation, but it also makes possible the application of more sophisticated mathematical models as they become available, and facilitates the utilization of productivity data accumulated from year to year to improve the accuracy of the system.

The computerized IMS is an interactive package designed to be used by the seed orchard manager. It is currently accessed by CMS on the IBM 370 mainframe at Virginia Tech, but can be modified for use on other computers. To use the IMS package, the user collects data from seed orchard sample trees under the guidelines specified by Bramlett and Godbee (1982). It should be noted that the programs are set up to handle the sampling design described in that paper for "Estimating Orchard and Clone Productivity." The user employs the CMS editor to enter the data into formatted data files which are stored on the CMS disk. The IMS programs then access these files, as directed by the user, to compute the predicted and actual values for the variables noted above (e.g. for cone efficiency, seed efficiency) for each clone and for the whole orchard. These values are written by the programs to the user's CMS disk from which the user can access them.

Although this publication contains printed copies of the programs that make up the IMS package, it is primarily a user's guide and therefore describes the system at the user level, with little detail concerning the mechanics of the programs. In the user's guide are short descriptins of each of the programs of the IMS package and more detailed descriptions of the user-created data files utilized by the IMS programs and the output generated by the IMS programs. In the part of the user's guide which is concerned with creation of the data files, column by column data entry instructions are provided.

Also included in the user's guide are a user's flow chart of the IMS programs and files, and samples of each of the user-created data files and program-generated output files.

The printed copies of the IMS components contained in this publication include one CMS exec file and 3 Fortran programs. The Fortran programs are fully documented within the code, with a block of comments at the head of each subroutine describing its mechanics.

USER'S GUIDE FOR THE COMPUTERIZED INVENTORY-MONITORING SYSTEM

PROGRAMS WITHIN THE COMPUTERIZED INVENTORY-MONITORING SYSTEM

GOIMS

GOIMS is a CMS exec file which forms the major interactive part of the system. It asks the user questions concerning what kind of program run he desires to make ("Predictions" or "Actuals") and which data files he wants the program to employ in making these runs ("Observations," " Actual Flowers and Cones" or "Actual Cone Analysis"). It also tells the user under what file names he can find the results of a run on the project disk and gives him the option of having the system help him create new data files. The GOIMS exec accesses the various programs within the system as needed. It is initiated by typing "GOIMS."

IMSYS

IMSYS, the main program of the computerized IMS, is written in Fortran and is accessed by the GOIMS exec when the user requests a "Predictions" run. It utilizes strobilus survival data from sample trees and clonal estimates for seed efficiency, extraction efficiency, etc., supplied by the user in an "Observations" data file, to compute clonal cone and seed yield predictions and to update these predictions throughout the cone cycle. IMSYS automatically puts these predicted values on the CMS disk from which the user can access them.

AVGACTS

AVGACTS is a Fortran program accessed by the GOIMS exec when needed for an "Actuals" run. AVGACTS utilizes actual cone analysis data and actual flower-and cone-count data from sample trees, supplied by the user in specified data files, to compute "Actual" (<u>not</u> "Predictions") clonal averages for efficiency values, such as cone efficiency and seed efficiency. AVGACTS automatically puts these clonal average values on the CMS disk from which the user can access them.

NEWOBV

NEWOBV is a Fortran program accessed by the GOIMS exec when the user requests the system to help him create the basis for a new user "Observations" data file to eventually be used by IMSYS. NEWOBV utilizes, as specified by the user, a clonal averages file created by AVGACTS and an existing "Observations" file already on the CMS disk to construct the skeleton for a new user "Observations" file which the user can simply fill in with sample tree flower count data as they are obtained.

USER-CREATED DATA FILES UTILIZED BY THE COMPUTERIZED INVENTORY-MONITORING SYSTEM

Files used by IMSYS

IMSYS makes use of three user-created data files, CLONTAB, SYSTAT and DATA__ to make the "Predictions" runs referred to in the GOIMS exec.

CLONTAB (FN FT FM[1] = CLONTAB DATA A)

This data file contains one line of data for each clone. It can be used by IMSYS for multiple cone crops--it doesn't have to be created for each cone crop. CLONTAB once created is automatically accessed by IMSYS when needed. It should contain the following information for each clone:

Column	
1-4	Orchard name (alphanumeric, left-justified).
5-19	Clone name (alphanumeric, left-justified).
20-21	Index number assigned to clone. Any integer can be assigned to any clone from 1 up to the number of clones in the table (right-justified).
22-25	Estimated clonal cones-per-bushel (right-justified).
26-30	Estimated clonal seed-per-pound (right-justified).

Figure 1 is a sample CLONTAB data set.

---

[1] FN FT FM = Filename Filetype Filemode as used in CMS.

SYSTAT (FN FT FM = SYSTAT DATA A)

This data file contains only one line of data. It needs to be reset before makng any INITIAL run for a cone crop. SYSTAT, like CLONTAB, is automatically accessed by IMSYS when needed. It should contain the following information (all right-justified):

Column

1-3	Number of clone means in CLONTAB.
4-6	Zero (0) (for any INITIAL run).
7-9	Highest index number in CLONTAB.
10-12	Zero (0) (For an INITIAL run for an even cone year).
13-15	Zero (0) (For an INITIAL run for an odd cone year).
16-18	Last two digits of 1st observation year for the cone crop (Flowering Year).
19-21	Last two digits of 2nd observation year for the cone crop (Cone Harvest Year).

The zeros placed in this file by the user in columns 6, 12, and 15 are simply place-holders. As the program (IMSYS) executes, it will use these spaces to count the number of records it processes on an INITIAL run. Therefore, whenever an INITIAL run is made for an even-year cone crop (e.g. the 1980 crop) the counter in columns 10-12 will have to be reset to 0, and whenever an INITIAL run is made for an odd-year cone crop (e.g. 1981) the counter in columns 16-18 will have to be reset to 0.

Figure 2 is a sample SYSTAT file.

DATA__ (FN FT FM = DATA__ DATA A)

The actual name of this file depends on the harvest year for the cone crop being monitored. Thus for the 1980 cone crop, this file would be called DATA80, for the 1981 crop, DATA81, etc. DATA__ is the "Observations" data file referred to in the GOIMS exec. It contains one line of data

for each sample tree. For the first two cone crops to be processed by the system, it must be completely created by the user. All data in this data set should be sorted by clone, so that all sample ramets of one clone appear together, and so that the clone names appear in the same order as they do in CLONTAB. It should be noted that the format of this file is similar to but not identical to the sample tree data sheet (Figure 1) in Bramlett and Godbee (1982). Sample tree data to be used by the program must be entered according to the following format. In DATA__ the following information is entered for each sample tree:

Column

1-2	Last 2 digits of cone year
3-6	Orchard name (alphanumeric, left-justified).
7-19	Clone name (alphanumeric, left-justified).

All entries from this point are right-justified.

20-21	Row number of sample tree location.
22-24	Column number of sample tree location.
25-26	Last 2 digits of year sample tree was grafted.
27-29	Number of ramets of the same clone AND same age as the sample tree. If there is a clone with more age classes than there are sample trees, or if for some other reason all age classes are not represented by the sample trees, unrepresented trees should be grouped into the nearest age class, so that the total number of ramets in the clone is represented.
30-33	Total number of female strobili that emerge on the sample tree at anthesis.
34-36	Total number of female strobili that emerge on the sample branches of the sample tree at anthesis (this will be called the INITIAL observation).
37-39	Number of surviving female strobili counted on the sample branches at the time of the first count (called the FIRST MARCH OBSERVATION).
40-42	Number of surviving female strobili counted on the sample branches at the time of the second count (called the FIRST JUNE observation).
43-45	Number of surviving female strobili counted on the sample branches at the time of the third count (called the FIRST OCTOBER observation).
46-48	Number of surviving female strobili counted on the

sample branches at the time of fourth count (called the SECOND MARCH observation).

49-51 Number of surviving female strobili counted on the sample branches at the time of the fifth count (called the SECOND JUNE observation).

52-54 Number of surviving cones counted on the sample branches at harvest time (called the FINAL observation).

55-57 Estimated clonal seed potential

58-59 Estimated clonal cone efficiency.

60-61 Estimated clonal seed efficiency.

62-63 Estimated clonal extraction efficiency.

64-65 Estimated clonal germination efficiency.

DATA__ has a size limit of 255 records (255 sample trees). Selection of the sample trees to be used for this data file should be conducted according to Bramlett and Godbee's (1982) instructions for "Estimating Orchard and Clone Productivity." Likewise the sample (female strobili) counts and periodic sample branch counts should be conducted as detailed by Bramlett and Godbee (1982). Estimates for clonal seed potential, seed efficiency, extraction efficiency and germination efficiency should be supplied by the user from the most recent cone analysis data available. Bramlett et al. (1977) detailed the cone analysis procedure employed to obtain these data. If no cone analysis data is available, best available estimates for these values should be used. The same can be done for estimated clonal cone efficiency.

DATA__ is ready to be used by IMSYS for an INITIAL "Predictions" run when columns 1 through 36 and columns 55 through 65 are filled-in with values for each sample tree. An INITIAL run must be made before "Predictions" runs using any of the other strobili counts can be made. However, once an INITIAL run has been made, runs using the other strobili counts to update the predictions can be made in any order. For example, once SECOND MARCH strobili counts are placed in columns 40-42, a SECOND MARCH "Predictions" run can be made without making any of the preceding "Predictions" runs (FIRST MARCH, FIRST JUNE, FIRST OCTOBER) or even having values in those columns in DATA__. Thus the user has the option of making any of the six strobili counts following INITIAL that he wants to, in order to have IMSYS update the predictions.

See Files Created by NEWOBV for information on how the system can aid in the creation of DATA__ files for subsequent cone crops.

Figure 3 is a sample DATA__ data set called DATA82, since it is for a 1982 cone crop.

Files used by AVGACTS

Following the end of the first cycle (harvest of first monitored crop) the user has the option of employing two additional programs that will make information gained from the analysis of one cone and seed crop automatically available for use in predicting subsequent cone and seed crops. These programs are named AVGACTS and NEWOBV. AVGACTS makes use of two user-created data files, ACTFLC__ and ACTCAS__, to make the "Actuals" run referred to in the GOIMS exec.

ACTFLC__ (FN FT FM = ACTFLC__ DATA A)

This is the "actual flowers and cones" data set referred to in the GOIMS exec. As with DATA__, the full name of this file depends on the harvest year being monitored. Thus for the 1980 cone crop, the file would be called ACTFLC80, for the 1981 crop, ACTFLC81, etc. AVGACTS uses the data in ACTFLC__ to calculate the average actual cone efficiency for each sample clone. Data in this data set should be sorted by clone such that the clone names appear in the same order as they did in the DATA__ file for the same cone year. ACTFLC__ should contain the following data for <u>each sample tree</u>:

Column	
1-2	Last 2 digits of cone year.
3-10	Clone name (left-justified).
11	Blank

All entries from this point on are right-justified.

12-13 Row location of sample tree.
14 Blank
15-17 Column location of sample tree.
18 Blank
19-22 Total number of female strobili counted on the sample tree at anthesis.
23 Blank
24-27 Total number of living cones harvested from sample tree.

NOTE: If it is impossible or undesirable to count the total number of cones harvested from a sample tree, the following alternative values can be substituted for the whole tree counts, and will be used by the program to estimate "Actual" clonal cone efficiencies:

Column

19-10 Total number of female strobili counted on the sample branches of the sample tree at anthesis (same as the INITIAL observation in DATA__).
24-27 Total number surviving cones counted on the sample branches of the sample tree at harvest time (same as the FINAL observation in DATA__).

Figure 4 is a sample ACTFLC__ data set.

ACTCAS__ (FN FT FM = ACTCAS__ DATA A)

This is the "actual cone analysis" data set referred to in the GOIMS exec. As with ACTFLC__, the full name of this file depends on the harvest year for the cone crop being monitored (ACTCAS80, ACTCAS81 etc.). Data obtained from cone analysis of sample cones from each clone will be put in this data set. AVGACTS will use it to compute an average seed potential, seed efficiency, etc. for each clone. Data in this data file should be sorted by clone such that the clone names appear in the same order as they did in ACTFLC__

file for the same cone year. Bramlett et al. (1977) details the cone analysis procedure employed to obtain the data to be entered in this file.

IMPORTANT NOTE: If there is no cone analysis data for a particular clone (possibly because the clone produced no cones to analyze), the clone name should still be entered into this data file, at least once, and zeros should be filled in for number of fertile scales, number of extracted seed, etc. This procedure is necessary in order to keep a one-to-one correspondence between the clone names in ACTFLC__ and ACTCAS__.

ACTCAS__ should contain the following data for <u>each sample cone</u>:

<u>Column</u>

1-2	Last 2 digits of cone year.
3-10	Clone Name (left-justified).
11	Blank

All entries from this point on are right-justified.

12-13	Row location of sample trees that cone came from.
14	Blank
15-17	Column location of sample tree that cone came from.
18-20	Blank
21-22	Cone identification number (1, 2, 3, etc.) if more than one cone comes from a sample tree.
23	Blank
24-26	Number of fertile scales counted on cone.
27	Blank
28-30	Number of extracted seed from cone.
31	Blank
32-34	Total number of seed in cone (extracted + dissected)
35	Blank
36-38	Total number of filled seed from cone.
39	Blank
40-42	Total number of germinated seed from cone.

Figure 5 is a sample ACTCAS__ data set.

Files used by NEWOBV

NEWOBV uses two additional data files but neither has to be specially created by the user for NEWOBV to use. See "Files Created by NEWOBV" for information on these two files.

OUTPUT GENERATED BY THE COMPUTERIZED INVENTORY-MONITORING SYSTEM

Files created by IMSYS

IMSYS generates two output files that are written to the project disk for the user to access: OUTPUT and CONPROD.

OUTPUT (FN FT FM = OUTPUT DATA A)

This file contains, for any "Predictions" run requested of the system (INITIAL, FIRST MARCH, FIRST JUNE, FIRST OCTOBER, SECOND MARCH, SECOND JUNE or FINAL), all the predictive information generated by IMSYS for each clone, including : Predicted values for cone efficiency, seed potential, seed efficiency, extraction efficiency, germination efficiency, total seed, extracted seed, bushels of cones, total cones, pounds extracted seed, number of seedlings, and seed orchard-to-nursery efficiency. It also contains overall seed orchard predictions for these values.

Figures 6 and 7 are sample OUTPUT files.

CONPROD (FN FT FM = CONPROD GUIDE A)

This file is a shortened version of OUTPUT, containing only the following information for each clone for a given run (INITIAL, FIRST MARCH, etc.): Predicted bushels of cones, predicted number of cones, and number of ramets of the clone in the orchard.

Figure 8 shows two sample CONPROD files.

Files created by AVGACTS

CLNAVGS (FN FT FM = CLNAVGS DATA A)

AVGACTS generates one output file named CLNAVGS that is written to the project disk for the user to access. It contains output from the "Actuals" run performed by AVGACTS, including "Actual" clonal averages for seed potential, seed efficiency, extraction efficiency, germination efficiency, cone efficiency, and seed orchard-to-nursery efficiency.

Figure 9 is a sample CLNAVGS file.

Files created by NEWOBV

DATA__ (FN FT FM = DATA__ DATA A)

NEWOBV generates one data file that is both named and, to a certain extent, designed by the user.[2] This file is the basis for a new user "Observations" file DATA__, a file discussed earlier under the "User-Created Data Files" section. With the use of NEWOBV, the user no longer has to completely type the DATA__ file for each new cone crop. Instead, NEWOBV will create a skeleton of a DATA__ file for the user

[2] Note on naming files: Any of the user-created data files described in the user's guide except CLONTAB and SYSTAT can be named by the user. DATA82, ACTFLC82 and ACTCAS82 were used in the guide as recommended file names for systems that only need to monitor a single orchard. If, for example, two "Observations" files were needed for a system monitoring the 1982 cone crops in both a slash pine seed orchard and a loblolly pine seed orchard, they might be named SLASH82 and LOB82.

already containing most of the data of a complete DATA__ file <u>except</u> the flower count observations, which the user can simply fill in as he obtains the data. NEWOBV uses a DATA__ file from a previous crop specified by the user as the source for the data that will appear in columns 3-29 of the new DATA__ file (orchard, clone name, row, column, year grafted, number of ramets in same age class). Then, NEWOBV uses the clonal averages for seed potential, seed efficiency, etc. created by the latest run of AVGACTS (and stored on the CMS disk as CLNAVGS) to create updated clonal estimates for these values by averaging them with the estimates found in the old DATA__ file. These updated clonal estimates are then inserted into columns 55-65 of the new DATA__ file. This leaves columns 30-54 ready to be filled-in with flower-count observations by the user as he obtains them.

Cone year (columns 1-2) for the newly-created DATA__ file is automatically incremented by 2 by NEWOBV. This is because we have made the assumption that the actual cone analysis data from a given cone crop (say 1981) will not be available in the time to be of much use in predicting the next year's crop (1982) but should be ready just in time to coincide with the first flower counts for the following year's crop (1983). In other words, we assume that DATA81 and "Actual" data from the 1981 crop will be used to create the DATA83 file, DATA82 and "Actual" data from the 1982 crop will be used to create the DATA84 file, etc.

Figure 10 is a sample NEWOBV-created DATA__ file called DATA84.

EXAMPLES OF USER-CREATED FILES

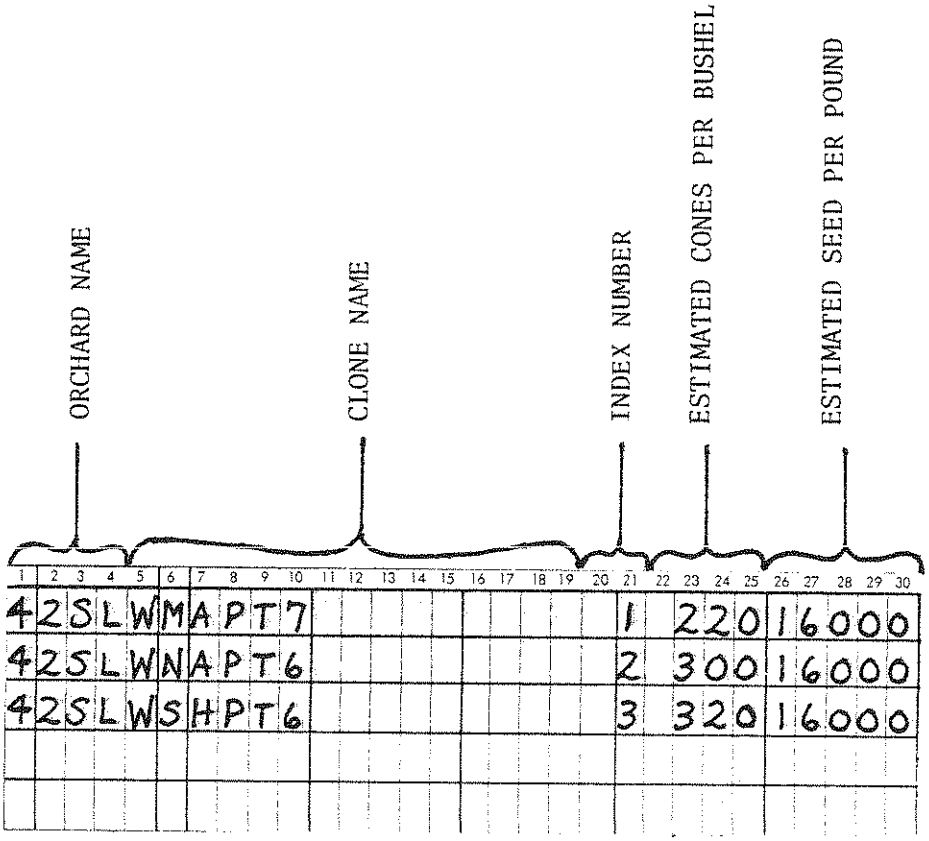


Figure 1. Sample CLONTAB. In this sample clone table there are three clones, all in the same orchard, assigned index numbers 1, 2 and 3.

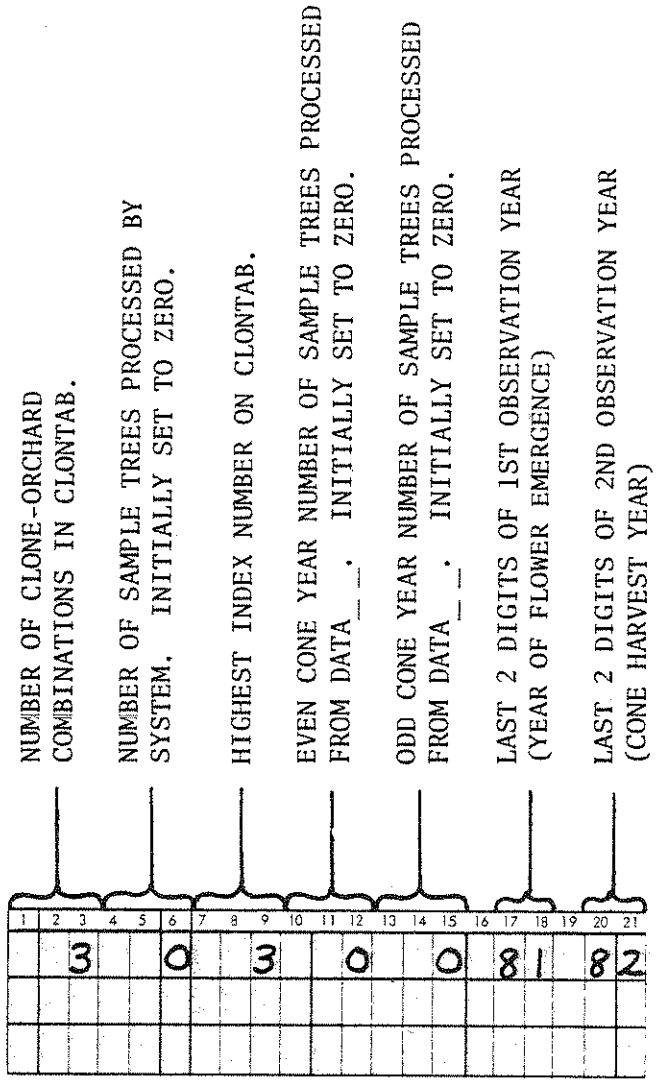


Figure 2. Sample SYSTAT. In this sample system status record, note that all the "counters" have been zeroed-out and that the year columns are set to process the 1982 cone crop.

CONE YEAR	ORCHARD NAME	CLONE NAME	ROW	COLUMN	YEAR GRAFTED	NUMBER OF RAMETS IN SAME AGE CLASS	TOTAL FEMALE STROBILI (WHOLE TREE)	INITIAL = TOTAL FEMALE STROBILI (SAMPLE BRANCHES)	1ST MARCH = TOTAL SURVIVING FEMALE STROBILI (SAMPLE BRANCHES)	1ST JUNE	1ST OCTOBER	2ND MARCH	2ND JUNE	FINAL	ESTIMATED SEED POTENTIAL	ESTIMATED CONE EFFICIENCY	ESTIMATED SEED EFFICIENCY	ESTIMATED EXTRACTION EFFICIENCY	ESTIMATED GERMINATION EFFICIENCY
82	42	5LWMAPT7	11	32	73	12	65	25	22	20	20				110	60	70	90	95
82	42	5LWMAPT7	26	108	73	12	80	40	38	37	30				110	60	70	90	95
82	42	5LWMAPT7	36	27	75	10	50	32	32	30	25				110	60	70	90	95
82	42	5LWNAPT6	15	30	75	17	70	35	30	28	25				130	60	70	90	95
82	42	5LWNAPT6	19	67	75	17	75	40	38	35	35				130	60	70	90	95
82	42	5LWNAPT6	27	23	75	17	60	30	30	18	15				130	60	70	90	95
82	42	5LWSHPT6	45	57	73	15	120	94	90	87	82				120	60	70	90	95
82	42	5LWSHPT6	15	15	73	15	101	87	80	80	71				120	60	70	90	95
82	42	5LWSHPT6	56	114	73	15	86	39	29	29	28				120	60	70	90	95

Figure 3. Sample DATA_ _ (DATA82). In this sample user "Observations" data set for a 1982 cone crop there are three clones, each represented by three sample trees. Note that clone WMAPT7 has sample trees representing two age classes. It has two sample trees to represent the 12 ramets grafted in 1973 and one sample tree to represent the 10 ramets grafted in 1975. This sample DATA82 has sample branch flower count data through the "FIRST OCTOBER" observation, with the rest of the flower counts yet to be filled in. THe efficiency estimates are just general estimates, not obtained from actual cone analysis data.

CONE YEAR (1–2)	CLONE NAME (3–10)	ROW (11–14)	COLUMN (15–18)	TOTAL FEMALE STROBILI (WHOLE TREE) (19–22)	TOTAL LIVE CONES HARVESTED (23–27)
82	WMAPT7	11	32	65	35
82	WMAPT7	26	108	80	60
82	WMAPT7	36	27	50	36
82	WNAPT6	15	30	70	51
82	WNAPT6	19	67	75	64
82	WNAPT6	27	23	60	32
82	WSHPT6	45	57	120	74
82	WSHPT6	15	15	101	81
82	WSHPT6	56	114	86	66

Figure 4. Sample ACTFLC_ _ (ACTFLC82) This "actual flowers and cones" data set contains the same nine sample trees as the sample DATA_ _ data set, DATA82. Note that the "Total Female Strobili (Whole Tree)" counts are the same on ACTFLC82 and DATA82.

CONE YEAR	CLONE NAME	ROW	COLUMN	CONE NUMBER	FERTILE SCALES	EXTRACTED SEED	TOTAL SEED	FILLED SEED	GERMINATED SEED
82	WMAPT7	11	32	1	51	90	93	70	69
82	WMAPT7	11	32	2	56	101	104	82	82
82	WMAPT7	11	32	3	55	97	98	68	65
82	WNAPT6	15	30	1	60	106	110	100	99
82	WNAPT6	15	30	2	58	105	107	77	61
82	WNAPT6	15	30	3	58	99	103	74	73
82	WSHPT6	45	57	1	62	86	112	88	88
82	WSHPT6	45	57	2	65	90	118	91	90
82	WSHPT6	45	57	3	64	83	109	82	77

Figure 5. Sample ACTCAS_ _ (ACTCAS82). This sample "actual cone analysis" data set has cone analysis data for three clones, each represented by three cones sampled from one tree of each clone.

EXAMPLES OF COMPUTERIZED IMS OUTPUT

```
 INITIAL RUN FOR CONE YR. 82 ACTUAL YR. 81
=========================================
 RESULTS FOR CLONE WMAPT7        ORCHARD 42SL YEAR 82
  CONE EFF. (PCE)=  0.60      SEED EFF. (PSE)=  0.70
  SEED POT. (PSP)=  110.0      EXT. EFF. (PEE)=  0.90
  GERM. EFF. (PGE)=  0.95     NO. RAMETS (NR)= 22
  AVG. FEM. FLOWERS/RAMET (NF)=   62.27
  CONES PER BU.=    220      SEED PER LB.= 16000
 PREDICTED SEED (PTS=NR*NF*PCE*PSP*PSE)                    63293.96
 PREDICTED NO. EXT. SEED (PES=PTS*PEE)                     56964.53
 PREDICTED BUSHELS OF CONES (PBU=(NR*NF*PCE)/CONBU)           3.74
 PREDICTED CONES (PCO=(NR*NF*PCE))                          822.00
 PREDICTED LBS EXTRACTED SEED (PLB=(PTS*PEE)/SEED             3.56
 PREDICTED NO. SEEDLINGS (PVS=PES*PGE)                     54116.28
 PREDICTED ORCH. TO NURS. EFF.                                35.91 %
=========================================
 RESULTS FOR CLONE WNAPT6        ORCHARD 42SL YEAR 82
  CONE EFF. (PCE)=  0.60      SEED EFF. (PSE)=  0.70
  SEED POT. (PSP)=  130.0      EXT. EFF. (PEE)=  0.90
  GERM. EFF. (PGE)=  0.95     NO. RAMETS (NR)= 17
  AVG. FEM. FLOWERS/RAMET (NF)=   68.33
  CONES PER BU.=    323      SEED PER LB.= 16000
 PREDICTED SEED (PTS=NR*NF*PCE*PSP*PSE)                    63426.89
 PREDICTED NO. EXT. SEED (PES=PTS*PEE)                     57084.18
 PREDICTED BUSHELS OF CONES (PBU=(NR*NF*PCE)/CONBU)           2.16
 PREDICTED CONES (PCO=(NR*NF*PCE))                          697.00
 PREDICTED LBS EXTRACTED SEED (PLB=(PTS*PEE)/SEED             3.57
 PREDICTED NO. SEEDLINGS (PVS=PES*PGE)                     54229.95
 PREDICTED ORCH. TO NURS. EFF.                                35.91 %
=========================================
 RESULTS FOR CLONE WSHPT6        ORCHARD 42SL YEAR 82
  CONE EFF. (PCE)=  0.60      SEED EFF. (PSE)=  0.70
  SEED POT. (PSP)=  120.0      EXT. EFF. (PEE)=  0.90
  GERM. EFF. (PGE)=  0.95     NO. RAMETS (NR)= 15
  AVG. FEM. FLOWERS/RAMET (NF)=  102.33
  CONES PER BU.=    416      SEED PER LB.= 16000
 PREDICTED SEED (PTS=NR*NF*PCE*PSP*PSE)                    77363.87
 PREDICTED NO. EXT. SEED (PES=PTS*PEE)                     69627.44
 PREDICTED BUSHELS OF CONES (PBU=(NR*NF*PCE)/CONBU)           2.21
 PREDICTED CONES (PCO=(NR*NF*PCE))                          921.00
 PREDICTED LBS EXTRACTED SEED (PLB=(PTS*PEE)/SEED             4.35
 PREDICTED NO. SEEDLINGS (PVS=PES*PGE)                     66146.00
 PREDICTED ORCH. TO NURS. EFF.                                35.91 %
=========================================
=========================================
=========================================
 RESULTS FOR ORCHARD 42SL
 PREDICTED ORCHARD SEED PRODUCTION (SUM PTS)     204084.7
 PREDICTED NO. OF BUSHELS OF CONES (SUM PBU)          8.1
 PREDICTED NO. OF CONES (SUM PCO)                  2440.0
 PREDICTED NO. OF SEEDLINGS (SUM PVS)            174492.2
 PREDICTED LBS. OF SEED (SUM PLB)                    11.5
 PREDICTED ORCH. TO NURS. EFF.                                35.91 %
```

Figure 6. Sample OUTPUT from INITIAL run. The results printed on this sample OUTPUT file were computed by IMSYS in an INITIAL "Predictions" run, using the INITIAL flower counts on DATA82.

```
 RESULTS FOR OCTOBER CONE YR. 82 ACTUAL YR. 81
=========================================
 RESULTS FOR CLONE WMAPT7        ORCHARD 42SL YEAR 82
  CONE EFF. (PCE)=  0.38      SEED EFF. (PSE)=  0.70
  SEED POT. (PSP)=  110.0      EXT. EFF. (PEE)=  0.90
  GERM. EFF. (PGE)=  0.95     NO. RAMETS (NR)= 22
  AVG. FEM. FLOWERS/RAMET (NF)=   62.27
  CONES PER BU.=    220      SEED PER LB.= 16000
 PREDICTED SEED (PTS=NR*NF*PCE*PSP*PSE)                    40437.78
 PREDICTED NO. EXT. SEED (PES=PTS*PEE)                     36393.98
 PREDICTED BUSHELS OF CONES (PBU=(NR*NF*PCE)/CONBU)            2.39
 PREDICTED CONES (PCO=(NR*NF*PCE))                           525.17
 PREDICTED LBS EXTRACTED SEED (PLB=(PTS*PEE)/SEED              2.27
 PREDICTED NO. SEEDLINGS (PVS=PES*PGE)                     34574.27
 PREDICTED ORCH. TO NURS. EFF.                                22.94 %
=========================================
 RESULTS FOR CLONE WNAPT6        ORCHARD 42SL YEAR 82
  CONE EFF. (PCE)=  0.50      SEED EFF. (PSE)=  0.70
  SEED POT. (PSP)=  130.0      EXT. EFF. (PEE)=  0.90
  GERM. EFF. (PGE)=  0.95     NO. RAMETS (NR)= 17
  AVG. FEM. FLOWERS/RAMET (NF)=   68.33
  CONES PER BU.=    323      SEED PER LB.= 16000
 PREDICTED SEED (PTS=NR*NF*PCE*PSP*PSE)                    52478.17
 PREDICTED NO. EXT. SEED (PES=PTS*PEE)                     47230.33
 PREDICTED BUSHELS OF CONES (PBU=(NR*NF*PCE)/CONBU)            1.79
 PREDICTED CONES (PCO=(NR*NF*PCE))                           576.68
 PREDICTED LBS EXTRACTED SEED (PLB=(PTS*PEE)/SEED              2.95
 PREDICTED NO. SEEDLINGS (PVS=PES*PGE)                     44868.79
 PREDICTED ORCH. TO NURS. EFF.                                29.71 %
=========================================
 RESULTS FOR CLONE WSHPT6        ORCHARD 42SL YEAR 82
  CONE EFF. (PCE)=  0.60      SEED EFF. (PSE)=  0.70
  SEED POT. (PSP)=  120.0      EXT. EFF. (PEE)=  0.90
  GERM. EFF. (PGE)=  0.95     NO. RAMETS (NR)= 15
  AVG. FEM. FLOWERS/RAMET (NF)=  102.33
  CONES PER BU.=    416      SEED PER LB.= 16000
 PREDICTED SEED (PTS=NR*NF*PCE*PSP*PSE)                    77638.12
 PREDICTED NO. EXT. SEED (PES=PTS*PEE)                     69874.25
 PREDICTED BUSHELS OF CONES (PBU=(NR*NF*PCE)/CONBU)            2.22
 PREDICTED CONES (PCO=(NR*NF*PCE))                           924.26
 PREDICTED LBS EXTRACTED SEED (PLB=(PTS*PEE)/SEED              4.37
 PREDICTED NO. SEEDLINGS (PVS=PES*PGE)                     66380.50
 PREDICTED ORCH. TO NURS. EFF.                                36.04 %
=========================================
=========================================
=========================================
 RESULTS FOR ORCHARD 42SL
 PREDICTED ORCHARD SEED PRODUCTION (SUM PTS)      170554.1
 PREDICTED NO. OF BUSHELS OF CONES (SUM PBU)           6.4
 PREDICTED NO. OF CONES (SUM PCO)                   2026.1
 PREDICTED NO. OF SEEDLINGS (SUM PVS)             145823.5
 PREDICTED LBS. OF SEED (SUM PLB)                      9.6
 PREDICTED ORCH. TO NURS. EFF.                                29.56 %
```

Figure 7. Sample OUTPUT from FIRST OCTOBER run. The results printed on this sample OUTPUT file were computed by IMSYS in a FIRST OCTOBER "Predictions" run, using the FIRST OCTOBER flower counts on DATA82.

CONE PRODUCTION GUIDE ----- INITIAL CONE YR. 82 ACTUAL YR. 81

CLONE ID	BUSHELS OF CONES	TOTAL CONES	RAMETS
WMAPT7	3.74	822.00	22
WNAPT6	2.16	697.00	17
WSHPT6	2.21	921.00	15

CONE PRODUCTION GUIDE ----- OCTOBER CONE YR. 82 ACTUAL YR. 81

CLONE ID	BUSHELS OF CONES	TOTAL CONES	RAMETS
WMAPT7	2.39	525.17	22
WNAPT6	1.79	576.68	17
WSHPT6	2.22	924.26	15

Figure 8. Sample CONPROD from INITIAL and FIRST OCTOBER runs. The results printed on these sample cone production reports are simply abbreviated versions of the results printed on OUTPUT for the INITIAL and FIRST OCTOBER runs.

****CLONAL AVERAGES****

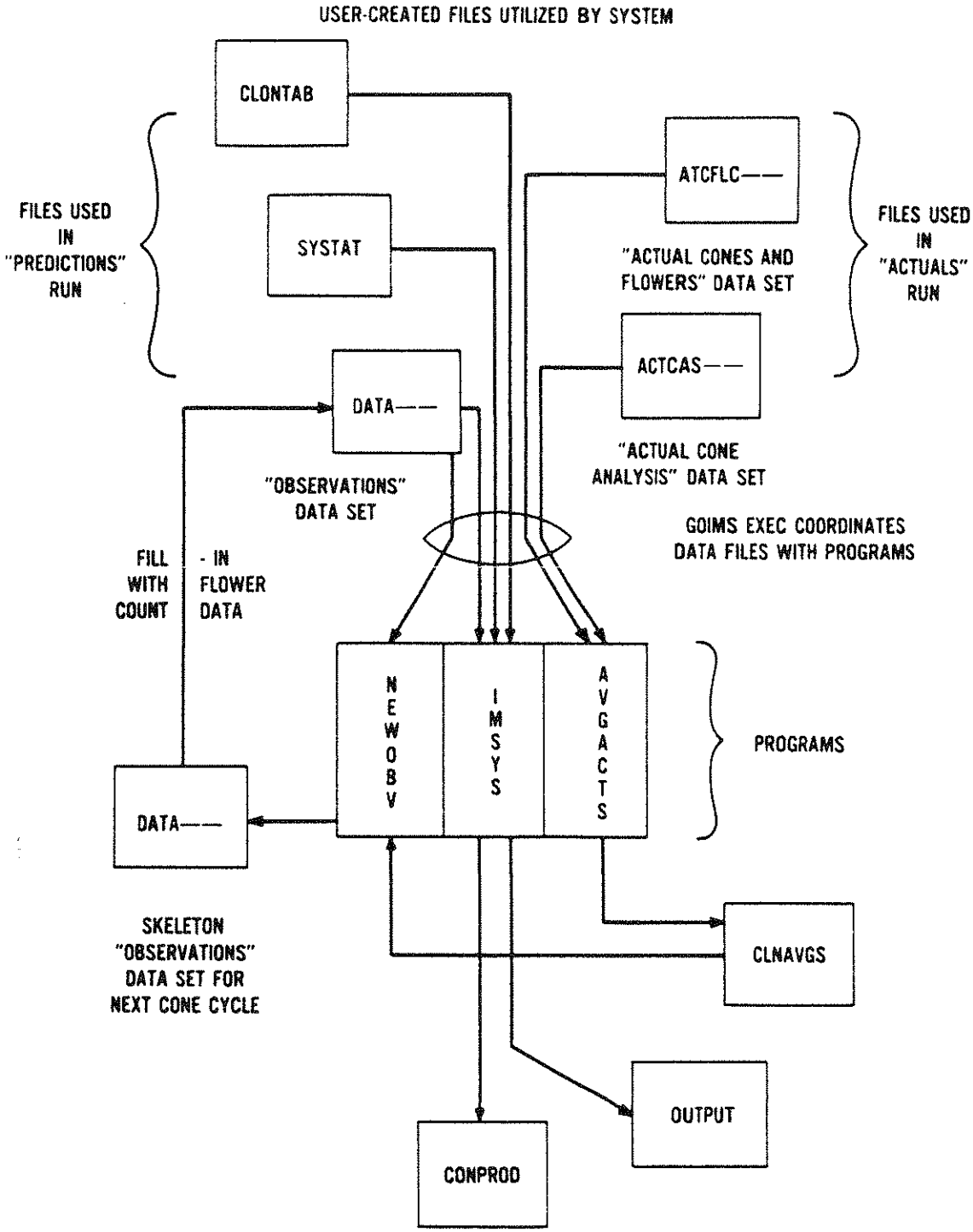
CLONE	SP	CE	SE	EE	.GE	SONE
WMAPT7	108.00	0.67	0.68	0.98	0.98	0.44
WNAPT6	117.33	0.72	0.71	0.97	0.93	0.46
WSHPT6	127.33	0.72	0.68	0.76	0.98	0.37

Figure 9. Sample CLNAVGS. The clonal averages for seed potential, cone efficiency, seed efficiency, extraction efficiency, germination efficiency and seed orchard-to-nursery efficiency were computed by AVGACTS in an "Actuals" run.

```
8442SLWMAPT7      11 3273 12      10963699496
8442SLWMAPT7      2610873 12      10963699496
8442SLWMAPT7      36 2775 10      10963699496
8442SLWNAPT6      15 3075 17      12366709394
8442SLWNAPT6      19 6775 17      12366709394
8442SLWNAPT6      27 2375 17      12366709394
8442SLWSHPT6      45 5773 15      12366698396
8442SLWSHPT6      15 1573 15      12366698396
8442SLWSHPT6      5611473 15      12366698396
```

Figure 10. Sample DATA_ _ (DATA84). This sample DATA_ _ file resembles the user-created DATA82 file shown in Figure 3. DATA84, however, was created by NEWOBV, using DATA82 and CLNAVGS (see Figure 9). Note that the efficiency estimates in DATA84 have been updated by averaging the old efficiency estimates in DATA82 with the actual clonal averages for these efficiency values from CLNAVGS. NEWOBV leaves the spaces in the center of DATA84 blank for the user to fill in with flower count data as they become available.

USER'S FLOW CHART FOR COMPUTERIZED
INVENTORY-MONITORING SYSTEM



LITERATURE CITED

Bramlett, D. L. and J. F. Godbee, Jr. 1982. Inventory-monitoring systems for southern pine seed orchards. Georgia Forest Research Paper No. 28. Georgia Forestry Commission, Macon, GA. 11 p.

Bramlett, D. L., E. W. Belcher Jr., G. L. Debarr, G. D. Hertel, R. P. Karrfalt, C. W. Lantz, T. Miller, K. D. Ware, and H. O. Yates III. 1977. Cone analysis of southern pines, a guidebook. USDA Forest Service General Technical Report SE-13. Southeastern Forest Expriment Station, Asheville NC, and Southeastern Area, State and Private Forestry, Atlanta, GA. 28 p.

IMS PROGRAMS

GOIMS

```
&CONTROL ERROR
GLOBAL TXTLIB FORTXLIB
*GLOBAL MACLIB WATLIB
CP TERMINAL LINESIZE 132
FILEDEF 04 TERMINAL
FILEDEF 06 TERMINAL
FILEDEF 19 TERMINAL
-MSG
&BEGTYPE


WELCOME TO THE "INVENTORY MONITORING SYSTEM."  WHEN READY TO BEGIN
YOUR SESSION, ENTER A NULL LINE.
&END
&READ ARGS
&IF &INDEX EQ 0 &SKIP 1
&GOTO -MSG
-PRED?
&BEGTYPE


DO YOU INTEND TO MAKE A "PREDICTIONS" RUN DURING THIS SESSION? (YES|NO)


&END
&READ ARGS
&IF &INDEX EQ 0 &1 = NO
&IF &1 EQ NO &GOTO -ACTS?
&IF &1 EQ N &GOTO -ACTS?
&IF &1 EQ YES &GOTO -PRO1
&IF &1 EQ Y &GOTO -PRO1
&TYPE *** ERROR *** INVALID RESPONSE. TRY AGAIN!
&GOTO -PRED?
-PRO1
&BEGTYPE
ENTER "OBSERVATIONS" DATA SET NAME. ENTER (FN FT FM)
&END
-LABEL1
&READ ARGS
&IF &INDEX EQ 3 &SKIP 3
&TYPE *** ERROR *** WRONG FORMAT.
&TYPE                SHOULD BE: FN FT FM, PLEASE REENTER.
&GOTO -LABEL1
&UDFN = &1
&UDFT = &2
&UDFM = &3
FI 03 DISK &UDFN &UDFT &UDFM ( PERM LRECL 80 BLKSIZE 800 RECFM FB )
FI 08 DISK SYSTAT DATA A1 ( LRECL 80 BLOCK 80 RECFM F
FI 01 DISK CLONTAB DATA A1 ( PERM LRECL 40 BLKSIZE 400 RECFM FB )
FI 02 DISK PCETAB DATA A1 ( LRECL 130 BLOCK 130 RECFM F )
FI 12 DISK ORCHDAT1 DATA A ( LRECL 130 BLOCK 130 RECFM FB )
FI 09 DISK ORCHDAT2 DATA A1 (LRECL 130 BLOCK 130 RECFM FB)
FI 11 DISK CONPROD GUIDE A1 ( LRECL 80 BLOCK 80 RECFM FB )
FI 10 DISK OUTPUT DATA A1 ( LRECL 80 BLOCK 80 RECFM FB )
LOAD IMSYS ( CLEAR NOMAP START )
*WATFIV MAIN
&BEGTYPE
            PREDICTIONS CALCULATED.

     PREDICTIVE REPORT EXISTS ON DISK FILE "OUTPUT DATA A1".
     CONE PRODUCTION GUIDE EXISTS ON DISK FILE "CONPROD GUIDE A1".
```

```
&END
-ACTS?
&BEGTYPE


DO YOU INTEND TO MAKE AN "ACTUALS" RUN DURING THIS SESSION ? (YES|NO)
&END
&READ ARGS
&IF &INDEX EQ 0 &1 = NO
&IF &1 EQ NO &GOTO -DONE
&IF &1 EQ N &GOTO -DONE
&IF &1 EQ YES &GOTO -PRO2
&IF &1 EQ Y &GOTO -PRO2
&TYPE *** ERROR *** INVALID RESPONSE. TRY AGAIN!
&GOTO -ACTS?
-PRO2
&BEGTYPE


ENTER "ACTUAL FLOWERS AND CONES" DATA SET NAME (FN FT FM).
&END
-LABEL2
&READ ARGS
&IF &INDEX EQ 3 &SKIP 3
&TYPE *ERROR* WRONG FORMAT.
&TYPE SHOULD BE FN FT FM, REENTER
&GOTO -LABEL2
&CRFN = &1
&CRFT = &2
&CRFM = &3
FI 11 DISK &CRFN &CRFT &CRFM (LRECL 80 BLOCK 80 RECFM FB)
&BEGTYPE


ENTER "ACTUAL CONE ANALYSIS" DATA SET NAME (FN FT FM).
&END
-LABEL3
&READ ARGS
&IF &INDEX EQ 3 &SKIP 3
&TYPE *ERROR* WRONG FORMAT.
&TYPE SHOULD BE FN FT FM, REENTER.
&GOTO -LABEL3
&CAFN = &1
&CAFT = &2
&CAFM = &3
FI 13 DISK &CAFN &CAFT &CAFM ( LRECL 80 BLOCK 80 RECFM FB)
FI 18 DISK CLNAVGS DATA A1 ( LRECL 80 BLOCK 80 RECFM FB )
FI 08 DISK SLAVE DATA A1 (LRECL 80 BLOCK 80 RECFM FB )
&BEGTYPE

            CALCULATING CLONAL AVERAGES FROM ACTUAL CONE ANALYSIS DATA.


&END
LOAD AVGACTS ( CLEAR NOMAP START )
&BEGTYPE


            CLONAL AVERAGES COMPUTED --- REPORT OF RESULTS EXISTS
            ON DISK FILE "CLNAVGS DATA A1".
```

```
&END
-NEWOBV
&BEGTYPE


DO YOU WISH TO USE THE CALCULATED CLONAL AVERAGES AS A BASIS
FOR A NEW USER "OBSERVATIONS" DATA SET? (YES|NO)
&END
&READ ARGS
&IF &INDEX EQ 0 &1 = NO
&IF &1 EQ NO &GOTO -DONE
&IF &1 EQ N &GOTO -DONE
&IF &1 EQ YES &GOTO -PRO3
&IF &1 EQ Y &GOTO -PRO3
&TYPE *** ERROR *** INVALID RESPONSE. TRY AGAIN!
&GOTO -NEWOBV
-PRO3
&BEGTYPE


WHICH EXISTING "OBSERVATIONS" DATA SET DO YOU WISH IT MODELED
AFTER? (ENTER FN FT FM)
&END
-LABEL4
&READ ARGS
&IF &INDEX EQ 3 &SKIP 3
&TYPE *** ERROR *** WRONG FORMAT.
&TYPE                SHOULD BE: FN FT FM, PLEASE REENTER.
&GOTO -LABEL4
&ODFN = &1
&ODFT = &2
&ODFM = &3
FI 03 DISK &ODFN &ODFT &ODFM (PERM LRECL 80 BLKSIZE 800 RECFM FB )
&BEGTYPE


WHAT DO YOU WISH TO CALL THE NEWLY CREATED "OBSERVATIONS"
DATA SET ? (ENTER FILENAME ONLY)
&END
-LABEL5
&READ ARGS
&IF &INDEX EQ 1 &SKIP 3
&TYPE *** ERROR *** WRONG FORMAT.
&TYPE                PLEASE REENTER, "FILENAME" ONLY.
&GOTO -LABEL5
&NDFN = &1
FI 10 DISK &1 DATA A1 ( LRECL 80 BLOCK 800 RECFM FB )
FI 18 DISK CLNAVGS DATA A1 (LRECL 80 BLOCK 80 RECFM FB )
&BEGTYPE

            CREATING NEW USER "OBSERVATIONS" DATA SET.


&END
LOAD NEWOBV ( CLEAR NOMAP START )
&BEGTYPE


             NEW "OBSERVATIONS" DATA SET CREATED AND STORED ON
             DISK UNDER THE FILENAME YOU SPECIFIED ABOVE.
&END
```

```
-DONE
&BEGTYPE

            THE SESSION IS OVER --- RETURNING USER TO CMS.

&END
&EXIT
```

IMSYS

```
C$JOB WATFIV,FREE
C
C                           PROGRAM "IMSYS"
C
C-----------------------------------------------------------------------C
C                           ROUTINE "MAIN"                              C
C     ROUTINE "MAIN" DIRECTS EXECUTION OF THE SUBROUTINES WHICH         C
C     ACCOMPLISH THE VARIOUS TASKS REQUIRED OF THE SYSTEM. A GENERAL    C
C     OUTLINE OF THESE TASKS FOLLOWS:                                   C
C        1.) QUERY USER FOR OBSERVATION YEAR AND MONTH.                 C
C        2.) ACCESS SYSTAT FILE(08) TO INITIALIZE SYSTEM COUNTERS.      C
C        3.) READ LAST RUN'S PREDICTED CONE EFFICIENCY TABLE (STORED    C
C            ON DISK FILE 02 (PCETAB)) INTO CURRENT RUN'S PREDICTED     C
C            CONE EFFICIENCY TABLE IF OTHER THAN INITIAL RUN.           C
C        4.) CALL SUBROUTINE "SPDAT" TO INPUT CURRENT MONTH'S           C
C            OBSERVATIONS.                                              C
C        5.) CALL SUBROUTINE DAVER TO PERFORM THE REQUIRED              C
C            CALCULATIONS.                                              C
C        6.) SAVE THE PCETAB TABLE BY WRITING ITS CONTENTS TO DISK      C
C            FILE 02.                                                   C
C        7.) STOP                                                       C
C-----------------------------------------------------------------------C

      INTEGER DESC(6),CLONE(3)
      COMMON CLONE,LOC,ICONYR,IYR,INDEX,ICONBU,ISEDLB,NORM,I1ST,I2ND
      COMMON INDEX2(255),ICOL2(255),IROW2(255),ASUM2(255,16),
     1CSUM2(255,4),DSUM2(255,4),NFR2(255),IYRGF2(255)
      DIMENSION IREP(4)
      DATA IREP/'Y','N'/

C
C------->QUERY USER FOR CONE HARVEST YEAR(EVEN OR ODD) AND OBSERVATION
C------->MONTH(1,2,3,4,5,6,OR 7).
C
      WRITE(19,1000)
1000  FORMAT(' ENTER WHETHER CONE HARVEST YEAR EVEN OR ODD.          '/
     1'       ENTER:      "1" FOR "ODD YEAR"                         '/
     2'              OR   "2" FOR "EVEN YEAR"  ')
      READ(04,11) IYRT
      IYR=IYRT

10    WRITE(19,2000)
2000  FORMAT(' ENTER CODE OF CURRENT OBSERVATION MONTH:              '/
     1'         1  =  INITIAL OBSERVATION                            '/
     2'         2  =  1ST MARCH OBSERVATION                          '/
     3'         3  =  1ST JUNE OBSERVATION                           '/
     4'         4  =  OCTOBER OBSERVATION                            '/
     5'         5  =  2ND MARCH OBSERVATION                          '/
     6'         6  =  2ND JUNE OBSERVATION                           '/
     7'         7  =  FINAL OBSERVATION    ')
      READ(04,11) IMON
11    FORMAT(I1)
      IF(IMON.LT.1.OR.IMON.GT.7) GO TO 12
      GO TO 14
12    WRITE(04,13)
13    FORMAT(' INVALID OBSERVATION MONTH --- REENTER.')
      GO TO 10
C

C------->READ SYSTEM STATUS RECORD FROM UNIT 08(SYSTAT)
C--------> NOC  = NO. OF CLONE-ORCH. COMBINATIONS ON 01(CLONTAB)
```

```
C---------> NORM = NO. OF ROW-COLUMNS WRITTEN TO 02(PCETAB) BY SYSTEM
C---------> IND  = HIGHEST INDEX ON FILE 01(CLONTAB)
C---------> NOE  = EVEN YR. NO. OF ROW COLUMNS FROM USER INPUT(03)
C---------> NOD  = ODD YR. NO. OF ROW-COLUMNS FROM USER INPUT(03)
C---------> I1ST-->I2ND = OBSERVATION YEARS INCLUDED IN THE STUDY
C
14    CONTINUE
      REWIND 08
      READ(08,200) NOC,NORM,IND,NOE,NOD,I1ST,I2ND
200   FORMAT(8I3)
C

C------->SET "NOO7" TO THE NUMBER OF RECORDS TO BE PROCESSED IN
C------->ORCHDAT1 (FILE 12) FOR ODD CONE YEAR OR ORCHDAT2 (FILE 09)
C------->FOR EVEN CONE YEAR AS DETERMINED BY THE SYSTEM DURING INPUT
C------->OF USERS OBSERVATION DATA ( A ONE TO ONE CORRESPONDENCE
C------->EXISTS). SET NOO7=NOD FOR ODD CONE YEAR OR NOO7=NOE FOR EVEN
C------->CONE YEAR.
C
      IF(IYRT-1) 50,60,50
60    NOO7=NOD
      GO TO 70
50    NOO7=NOE

C
C------->READ LAST RUN'S PREDICTED CONE EFFICIENCY TABLE INTO CURRENT
C------->RUN'S PREDICTED CONE EFFICIENCY TABLE IF OTHER THAN INITIAL
C------->RUN.
C
70    IF (IMON-1) 9005,83,80
80    REWIND 02
      DO 100 I=1,NORM
      READ(02,205) INDEX2(I),IROW2(I),ICOL2(I),NFR2(I),IYRGF2(I)
      READ(02,206) (ASUM2(I,J),J=1,16)
      READ(02,206) (CSUM2(I,J),J=1,4),(DSUM2(I,J),J=1,4)
100   CONTINUE
205   FORMAT(I5,2I3,2I4)
206   FORMAT(16F8.2)
83    CONTINUE

81    ISTP=0

C
C------->CALL SPDAT TO INPUT AND VERIFY USER INPUT DATA
C
      CALL SPDAT(IMON,ISTP,IYRT,NOO7,NOC,IND,NOE,NOD)
      IF(ISTP) 9004,9004,9005
9004  WRITE(19,9034)
      WRITE(19,9033)
      WRITE(19,9034)
9033  FORMAT(' USER DATA UPDATE COMPLETE --- CALCULATIONS BEGIN. ')
9034  FORMAT( // )

C
C------->CALL DAVER TO PERFORM CALCULATIONS ON INPUT DATA
C
      CALL DAVER(IMON,NOO7)

C
C------->SAVE CONTENTS OF PCETAB TABLE BY WRITING TO DISK 02(PCETAB)
C
```

```
      REWIND 02
      DO 300 I=1,NORM
      WRITE(02,205)INDEX2(I),IROW2(I),ICOL2(I),NFR2(I),IYRGF2(I)
      WRITE(02,206) (ASUM2(I,J),J=1,16)
      WRITE(02,206)(CSUM2(I,J),J=1,4),(DSUM2(I,J),J=1,4)
300   CONTINUE
C
C-------->GIVE USER THE CHANCE TO EXECUTE THE SYSTEM AGAIN
C
      WRITE(19,9034)
365   WRITE(19,375)
375   FORMAT(' ARE THERE ADDITIONAL OBSERVATIONS ON YOUR INPUT FILE  '/
     1' YOU WISH TO PROCESS? (Y/N) ')
      READ(04,376) IANS
      WRITE(19,9034)
376   FORMAT(A1)
      IF(IANS.EQ.IREP(1)) GO TO 10
      IF(IANS.NE.IREP(2)) GO TO 365
9005  CONTINUE
      STOP
      END
C-----------------------------------------------------------------------C
C                           SUBROUTINE "SPDAT"                          C
C                                                                       C
C     SUBROUTINE "SPDAT" IS CALLED BY ROUTINE "MAIN" TO INPUT THE       C
C     USER'S CURRENT OBSERVATION DATA FROM DISK FILE 03 , BUILD AND     C
C     UPDATE THE ORCHARD DATA TABLE AND INITIATE CONSTRUCTION OF THE    C
C     INITIAL PREDICTED CONE EFFICIENCY TABLE.                          C
C-----------------------------------------------------------------------C
      SUBROUTINE SPDAT(IMON,ISTP,IYRT,NOO7,NOPC,INDL,NOE,NOD)
      INTEGER CLONE(3)
      COMMON CLONE,LOC,ICONYR,IYR,INDEX,ICONBU,ISSED,NORM,I1ST,I2ND
      COMMON INDEX2(255),ICOL2(255),IROW2(255),ASUM2(255,16),
     1CSUM2(255,4),DSUM2(255,4),NFR2(255),IYRGF2(255)
      DIMENSION ICYRR7(255),LOCC7(255),ICLCC1(255),ICLCC2(255),
     1ICLCC3(255),IRW7(255),ICL7(255),IYRGF7(255),NRAMT7(255),NF7(255),
     2IBRCT7(255),IMAR1(255),IJUN1(255),IOCT(255),IMAR2(255),
     3IJUN2(255),IFINAL(255),IESP(255),IECE(255),IESE(255),IEEE(255),
     4IEGE(255),IASP(255),IASE(255),IAEE(255),IAGE(255)
      DIMENSION IREST(17),IZERO(6)
      INTEGER LOCC,CLCC(3)
      REWIND 03
      IK7=0
      NORMH=NORM
      NEW8=0
      DO 9060 IKL=1,6
9060  IZERO(IKL)=0
      IF(IMON-1) 1000,1000,2000
C
C-------->INPUT INITIAL OBSERVATION DATA
C
C-------->READ INITIAL OBSERVATION DATA. BUILD STARTING ORCHARD DATA
C-------->TABLE AND WRITE IT TO DISK FILE 12 (ORCHDAT1) FOR ODD CONE
C-------->YEAR OR DISK FILE 09 (ORCHDAT2) FOR EVEN CONE YEAR. BUILD
C-------->STARTING PREDICTED CONE EFFICIENCY TABLE BY CALLING SUBROUTINE
C-------->"FINDRO". UPDATE SYSTEM STATUS RECORD (DISK FILE 08 (SYSTAT))
C-------->TO REFLECT THE NUMBER OF RECORDS READ FROM FILE 03 (NOD OR NOE)
C-------->AND NUMBER OF RECORDS WRITTEN TO PCETAB (NORM).
C
1000  READ(03,1002,END=5005) ICONYR,LOC,(CLONE(K),K=1,3),IROW,ICOL,
     1IYRGRF,NRAMET,NF,IBRCNT,IESTSP,IESTCE,IESTSE,IESTEE,IESTGE
```

```
1002  FORMAT(I2,A4,3A4,2I3,I2,I3,I4,I3,18X,I3,4I2)
C-------->CALL SUBROUTINE "FNDCL" TO CHECK CLONE VALIDITY
      CALL FNDCL(IFND,NOPC,ISTP)
C-------->IFND=0 MEANS CLONE NOT IN CLONE FILE
C-------->IFND=1 MEANS CLONE WAS FOUND AT THE LOC AND SO PROCEED
      IF(IFND) 1000,1000,1113
1113  CALL FINDRO(IROW,ICOL,NEW8,NORMH,NRAMET,IYRGRF)
      IK7=IK7+1
      IF(IYR.EQ.2) GO TO 1114
      WRITE(12,1003) ICONYR,LOC,(CLONE(I),I=1,3),IROW,ICOL,IYRGRF,
     1NRAMET,NF,IBRCNT,(IZERO(K),K=1,6),IESTSP,IESTCE,IESTSE,IESTEE,
     2IESTGE,(IZERO(K),K=1,4)
1003  FORMAT(I2,4A4,21I4)
      GO TO 1000
1114  WRITE(09,1003) ICONYR,LOC,(CLONE(I),I=1,3),IROW,ICOL,IYRGRF,
     1NRAMET,NF,IBRCNT,(IZERO(K),K=1,6),IESTSP,IESTCE,IESTSE,IESTEE,
     2IESTGE,(IZERO(K),K=1,4)
      GO TO 1000

C-------->UPDATE THE SYSTEM STATUS RECORD ( DISK FILE 08 (SYSTAT)).
5005  NOO7=IK7
      IF(NOO7) 9077,9076,9077
9077  IF(IYRT-1) 9074,9075,9074
C-------->WRITE NEW SYSTEM STATUS RECORD FOR ODD YEAR
9075  REWIND 08
      WRITE(08,200) NOPC,NORM,INDL,NOE,NOO7,I1ST,I2ND
9076  RETURN
C-------->WRITE NEW SYSTEM STATUS RECORD FOR EVEN YEAR
9074  REWIND 08
      WRITE(08,200) NOPC,NORM,INDL,NOO7,NOD,I1ST,I2ND
      RETURN

C-------->IF OBV. MONTH OUT OF RANGE PRINT ERROR MSG. AND ABORT RUN
2000  IF(IMON-7) 2003,2003,2004
2004  WRITE(19,2005)
2005  FORMAT(' ERROR ON MONTH --- RUN IS TERMINATED.')
200   FORMAT(8I3)
      STOP

C-------->FOR OBSERVATIONS 2 - 7 READ THE VALUES OF THE LAST PROGRAM
C-------->RUN'S ORCHARD DATA TABLE ( STORED AUTOMATICALLY ON DISK FILE
C-------->12 (ORCHDAT1) FOR ODD CONE YEAR OR DISK FILE 09 (ORCHDAT2) FOR
C-------->EVEN CONE YEAR AFTER EACH OBSERVATION MONTH'S USER DATA UPDATE)
C-------->INTO THE CURRENT RUN'S ORCHARD DATA TABLE AND THEN BRANCH TO
C-------->THE CURRENT MONTH'S USER INPUT MODULE.
C
2003  IF(IYR.EQ.2) GO TO 2006
      REWIND 12
      DO 2100 I=1,NOO7
      READ(12,1003) ICYRR7(I),LOCC7(I),ICLCC1(I),ICLCC2(I),ICLCC3(I),
     1IRW7(I),ICL7(I),IYRGF7(I),NRAMT7(I),NF7(I),IBRCT7(I),IMAR1(I),
     2IJUN1(I),IOCT(I),IMAR2(I),IJUN2(I),IFINAL(I),IESP(I),IECE(I),
     3IESE(I),IEEE(I),IEGE(I),IASP(I),IASE(I),IAEE(I),IAGE(I)
2100  CONTINUE
      GO TO 2999
2006  REWIND 09
      DO 2200 I=1,NOO7
      READ(09,1003) ICYRR7(I),LOCC7(I),ICLCC1(I),ICLCC2(I),ICLCC3(I),
     1IRW7(I),ICL7(I),IYRGF7(I),NRAMT7(I),NF7(I),IBRCT7(I),IMAR1(I),
     2IJUN1(I),IOCT(I),IMAR2(I),IJUN2(I),IFINAL(I),IESP(I),IECE(I),
     3IESE(I),IEEE(I),IEGE(I),IASP(I),IASE(I),IAEE(I),IAGE(I)
```

```
2200  CONTINUE
C
C------->FOR OBSERVATIONS 2-7 BRANCH TO THE APPROPRIATE MODULE TO INPUT
C------->THE USER'S CURRENT OBSERVATION DATA AND ENTER EACH NEW OBSER-
C------->VATION VALUE INTO THE ORCHARD DATA TABLE.
C

2999  INON=IMON-1
      GO TO (3000,4000,5000,6000,7000,8000),INON

C
C-------->INPUT 1ST MARCH OBSERVATION DATA
C
3000  READ(03,3002,END=9000) ICONYR,LOC,(CLONE(K),K=1,3),IROW,ICOL,MOBV
3002  FORMAT(I2,A4,3A4,2I3,12X,I3)
      CALL FNDCL(IFND,NOPC,ISTP)
      IF(IFND) 3000,3000,3013
3013  DO 4050 KK=1,NOO7
      IF(ICONYR-ICYRR7(KK)) 4050,4052,4050
4052  IF(LOCC7(KK).NE.LOC) GO TO 4050
      IF(CLONE(1).NE.ICLCC1(KK).OR.CLONE(2).NE.ICLCC2(KK).OR.CLONE(3)
     1.NE.ICLCC3(KK)) GO TO 4050
      IF(IROW-IRW7(KK)) 4050,4055,4050
4055  IF(ICOL-ICL7(KK)) 4050,4056,4050
4056  IMAR1(KK)=MOBV
      GO TO 3000
4050  CONTINUE
      WRITE(19,4051) (CLONE(I),I=1,3),LOC,IROW,ICOL
4051  FORMAT(' ROW AND COL. MISSING: CLONE',3A4,' LOC',A4,' ROW',I4,
     1' COL',I4)
      ISTP=1
      GO TO 3000
C
C------->INPUT 1ST JUNE OBSERVATION DATA
C
4000  READ(03,4002,END=9000) ICONYR,LOC,(CLONE(K),K=1,3),IROW,ICOL,MOBV
4002  FORMAT(I2,A4,3A4,2I3,15X,I3)
      CALL FNDCL(IFND,NOPC,ISTP)
      IF(IFND) 4000,4000,4013
4013  DO 4150 KK=1,NOO7
      IF(ICONYR-ICYRR7(KK)) 4150,4152,4150
4152  IF(LOCC7(KK).NE.LOC) GO TO 4150
      IF(CLONE(1).NE.ICLCC1(KK).OR.CLONE(2).NE.ICLCC2(KK).OR.CLONE(3)
     1.NE.ICLCC3(KK)) GO TO 4150
      IF(IROW-IRW7(KK)) 4150,4155,4150
4155  IF(ICOL-ICL7(KK)) 4150,4156,4150
4156  IJUN1(KK)=MOBV
      GO TO 4000
4150  CONTINUE
      WRITE(19,4051) (CLONE(I),I=1,3),LOC,IROW,ICOL
      ISTP=1
      GO TO 4000
C
C------->INPUT OCTOBER OBSERVATION DATA
C
5000  READ(03,5002,END=9000) ICONYR,LOC,(CLONE(K),K=1,3),IROW,ICOL,MOBV
5002  FORMAT(I2,A4,3A4,2I3,18X,I3)
      CALL FNDCL(IFND,NOPC,ISTP)
      IF(IFND) 5000,5000,5013
5013  DO 4250 KK=1,NOO7
      IF(ICONYR-ICYRR7(KK)) 4250,4252,4250
```

```
4252  IF(LOCC7(KK).NE.LOC) GO TO 4250
      IF(CLONE(1).NE.ICLCC1(KK).OR.CLONE(2).NE.ICLCC2(KK).OR.CLONE(3)
     1.NE.ICLCC3(KK)) GO TO 4250
      IF(IROW-IRW7(KK)) 4250,4255,4250
4255  IF(ICOL-ICL7(KK)) 4250,4256,4250
4256  IOCT(KK)=MOBV
      GO TO 5000
4250  CONTINUE
      WRITE(19,4051) (CLONE(I),I=1,3),LOC,IROW,ICOL
      ISTP=1
      GO TO 5000
C
C------->INPUT 2ND MARCH OBSERVATION DATA
C
6000  READ(03,6002,END=9000) ICONYR,LOC,(CLONE(K),K=1,3),IROW,ICOL,MOBV
6002  FORMAT(I2,A4,3A4,2I3,21X,I3)
      CALL FNDCL(IFND,NOPC,ISTP)
      IF(IFND) 6000,6000,6013
6013  DO 4350 KK=1,NOO7
      IF(ICONYR-ICYRR7(KK)) 4350,4352,4350
4352  IF(LOCC7(KK).NE.LOC) GO TO 4350
      IF(CLONE(1).NE.ICLCC1(KK).OR.CLONE(2).NE.ICLCC2(KK).OR.CLONE(3)
     1.NE.ICLCC3(KK)) GO TO 4350
      IF(IROW-IRW7(KK)) 4350,4355,4350
4355  IF(ICOL-ICL7(KK)) 4350,4356,4350
4356  IMAR2(KK)=MOBV
      GO TO 6000
4350  CONTINUE
      WRITE(19,4051) (CLONE(I),I=1,3),LOC,IROW,ICOL
      ISTP=1
      GO TO 6000
C
C------->INPUT 2ND JUNE OBSERVATION DATA
C
7000  READ(03,7002,END=9000) ICONYR,LOC,(CLONE(K),K=1,3),IROW,ICOL,MOBV
7002  FORMAT(I2,A4,3A4,2I3,24X,I3)
      CALL FNDCL(IFND,NOPC,ISTP)
      IF(IFND) 7000,7000,7013
7013  DO 4450 KK=1,NOO7
      IF(ICONYR-ICYRR7(KK)) 4450,4452,4450
4452  IF(LOCC7(KK).NE.LOC) GO TO 4450
      IF(CLONE(1).NE.ICLCC1(KK).OR.CLONE(2).NE.ICLCC2(KK).OR.CLONE(3)
     1.NE.ICLCC3(KK)) GO TO 4450
      IF(IROW-IRW7(KK)) 4450,4455,4450
4455  IF(ICOL-ICL7(KK)) 4450,4456,4450
4456  IJUN2(KK)=MOBV
      GO TO 7000
4450  CONTINUE
      WRITE(19,4051) (CLONE(I),I=1,3),LOC,IROW,ICOL
      ISTP=1
      GO TO 7000
C
C------->INPUT FINAL OBSERVATION DATA
C
8000  READ(03,8002,END=9000) ICONYR,LOC,(CLONE(K),K=1,3),IROW,ICOL,MOBV
8002  FORMAT(I2,A4,3A4,2I3,27X,I3)
      CALL FNDCL(IFND,NOPC,ISTP)
      IF(IFND) 8000,8000,8013
8013  DO 4550 KK=1,NOO7
      IF(ICONYR-ICYRR7(KK)) 4550,4552,4550
4552  IF(LOCC7(KK).NE.LOC) GO TO 4550
```

```
      IF(CLONE(1).NE.ICLCC1(KK).OR.CLONE(2).NE.ICLCC2(KK).OR.CLONE(3)
     1.NE.ICLCC3(KK)) GO TO 4550
      IF(IROW-IRW7(KK)) 4550,4555,4550
4555  IF(ICOL-ICL7(KK)) 4550,4556,4550
4556  IFINAL(KK)=MOBV
      GO TO 8000
4550  CONTINUE
      WRITE(19,4051) (CLONE(I),I=1,3),LOC,IROW,ICOL
      ISTP=1
      GO TO 8000

C------->OUTPUT A COPY OF THE CURRENT ORCHARD DATA TABLE TO DISK FILE 12
C------->(ORCHDAT1) FOR ODD CONE YEAR OR DISK FILE 09 (ORCHDAT2) FOR
C------->EVEN CONE YEAR FOR IMMEDIATE PROCESSING BY SUBROUTINE "DAVER"
C------->AND ACCESS BY FUTURE SYSTEM RUNS.
9000  IF(IYR.EQ.2) GO TO 9100
      REWIND 12
      DO 9006 I=1,NOO7
      WRITE(12,1003) ICYRR7(I),LOCC7(I),ICLCC1(I),ICLCC2(I),
     1ICLCC3(I),IRW7(I),ICL7(I),IYRGF7(I),NRAMT7(I),NF7(I),
     2IBRCT7(I),IMAR1(I),IJUN1(I),IOCT(I),IMAR2(I),IJUN2(I),IFINAL(I)
     3,IESP(I),IECE(I),IESE(I),IEEE(I),IEGE(I),
     4IASP(I),IASE(I),IAEE(I),IAGE(I)
9006  CONTINUE
      RETURN
9100  REWIND 09
      DO 9106 I=1,NOO7
      WRITE(09,1003) ICYRR7(I),LOCC7(I),ICLCC1(I),ICLCC2(I),
     1ICLCC3(I),IRW7(I),ICL7(I),IYRGF7(I),NRAMT7(I),NF7(I),
     2IBRCT7(I),IMAR1(I),IJUN1(I),IOCT(I),IMAR2(I),IJUN2(I),IFINAL(I)
     3,IESP(I),IECE(I),IESE(I),IEEE(I),IEGE(I),
     4IASP(I),IASE(I),IAEE(I),IAGE(I)
9106  CONTINUE
      RETURN
      END
C-----------------------------------------------------------------------C
C                            SUBROUTINE "FNDCL"                         C
C                                                                       C
C     SUBROUTINE "FNDCL" IS CALLED BY SUBROUTINE "SPDAT" TO             C
C     VERIFY THAT EACH CLONE ENCOUNTERED IN THE USER'S INPUT DATA       C
C     IS A VALID CLONE, INCLUDED IN A PREVIOUSLY DEFINED OBSERVATION    C
C     GROUP. THE VALID CLONES COMPRISING THE OBSERVATION GROUP ARE      C
C     DEFINED BY THE USER'S ENTRIES IN DISK FILE 01 ( CLONTAB ).        C
C-----------------------------------------------------------------------C

      SUBROUTINE FNDCL(IFND,NOPC,ISTP)
      INTEGER CLONE(3)
      COMMON CLONE,LOC,ICONYR,IYR,INDEX,ICONBU,ISEED,NORM,I1ST,I2ND
      COMMON INDEX2(255),ICOL2(255),IROW2(255),ASUM2(255,16),
     1CSUM2(255,4),DSUM2(255,4),NFR2(255),IYRGF2(255)
      INTEGER CLONT(3),LOCT
C------->SET IFND=1 IF CLONE THERE; OTHERWISE SET IFND=0
      IFND=0
      REWIND 01
      DO 10 I=1,NOPC
      READ(01,500) LOCT,(CLONT(K),K=1,3),IND,ICONBU,ISEDLB
500   FORMAT(A4,3A4,I5,I4,I5)
      IF(LOCT.NE.LOC) GO TO 10
      IF(CLONT(1).NE.CLONE(1).OR.CLONT(2).NE.CLONE(2).OR.CLONT(3)
     1.NE.CLONE(3)) GO TO 10
      IFND=1
```

```
      INDEX=IND
      RETURN
10    CONTINUE
      WRITE(19,600) LOC,(CLONE(I),I=1,3)
600   FORMAT(' NEED TO ADD LOC ',A4,' CLONE ',3A4)
      ISTP=1
      RETURN
      END
C-----------------------------------------------------------------------C
C                            SUBROUTINE "FINDRO"                        C
C                                                                       C
C     SUBROUTINE "FINDRO" IS CALLED BY SUBROUTINE "SPDAT" TO BUILD      C
C     THE STARTING PREDICTED CONE EFFICIENCY TABLE SEGMENT FOR EACH     C
C     NEW CLONE ENCOUNTERED IN THE USER'S INPUT DATA (INIT. RUN ONLY).  C
C-----------------------------------------------------------------------C

      SUBROUTINE FINDRO(IROW,ICOL,NEW8,NORMH,NFR,IYRGRF)
      INTEGER CLONE(3)
      COMMON CLONE,LOC,ICONYR,IYR,INDEX,ICONBU,ISEED,NORM,I1ST,I2ND
      COMMON INDEX2(255),ICOL2(255),IROW2(255),ASUM2(255,16),
     1CSUM2(255,4),DSUM2(255,4),NFR2(255),IYRGF2(255)
      DIMENSION IFR(2),AR(16),CR(4),DR(4),ASUM(16),
     1CSUM(4),DSUM(4),NOCL(10),ITOT(10),ITTOT(10)
      IKT=0
      REWIND 15
C------->INITIALIZE ARRAY ASUM(16) TO 0.
      DO 3000 K=1,16
      ASUM(K)=0.
3000  CONTINUE
C------->INITIALIZE ARRAYS CSUM(4) AND DSUM(4) TO 0.
      DO 3002 K=1,4
      CSUM(K)=0.
3002  DSUM(K)=0.
      IF(NORM) 999,985,999
999   DO 10 I=1,NORM
      IF(INDEX2(I)-INDEX) 10,12,10
12    IF(I-NORMH) 2012,2012,1000
2012  IKT=IKT+1
1000  IF(IROW2(I)-IROW) 10,13,10
13    IF(ICOL2(I)-ICOL) 10,14,10
10    CONTINUE
      IF(IKT) 985,985,993
993   CALL FBNDRO(IROW,ICOL,NEW8,NORMH,NFR,IYRGRF,IKT)
      GO TO 14
985   NORM=NORM+1
      NEW8=1
      INDEX2(NORM)=INDEX
      IROW2(NORM)=IROW
      ICOL2(NORM)=ICOL
      DO 100 I=1,16
100   ASUM2(NORM,I)=ASUM(I)
      DO 202 I=1,4
      CSUM2(NORM,I)=CSUM(I)
      DSUM2(NORM,I)=DSUM(I)
202   CONTINUE
      NFR2(NORM)=NFR
      IYRGF2(NORM)=IYRGRF
      WRITE(10,800) INDEX,IROW,ICOL
800   FORMAT(' NEW ROW AND COL. FOR CLONE NO. ',I4,' ROW',I4,' COL',I4)
14    CONTINUE
      RETURN
```

```
      END
C-----------------------------------------------------------------------C
C                            SUBROUTINE "FBNDRO"                        C
C                                                                       C
C     SUBROUTINE "FBNDRO" IS CALLED BY SUBROUTINE "FINDRO" TO ADD       C
C     ALL OBSERVATIONS WITHIN A GIVEN CLONE ID TO THE PREDICTED         C
C     CONE EFFICIENCY TABLE.                                            C
C-----------------------------------------------------------------------C

      SUBROUTINE FBNDRO(IROW,ICOL,NEW8,NORMH,NFR,IYRGRF,IKT)
      INTEGER CLONE(3)
      COMMON CLONE,LOC,ICONYR,IYR,INDEX,ICONBU,ISEED,NORM,I1ST,I2ND
      COMMON INDEX2(255),ICOL2(255),IROW2(255),ASUM2(255,16),
     1CSUM2(255,4),DSUM2(255,4),NFR2(255),IYRGF2(255)
      DIMENSION IFR(2),AR(16),CR(4),DR(4)
     1,ASUM(16),CSUM(4),DSUM(4),AS(10,16)
     2,CS(10,4),NOCL(10),DS(10,4),ITOT(10),ITTOT(10)
C------->SET NEW8=1 IF ROW AND COL ADDED
      REWIND 15
      IFST=1
      KLAS=1
      DO 4000 I=1,10
      DO 4000 K=1,16
      AS(I,K)=0.
4000  CONTINUE
      DO 4001 I=1,10
      DO 4001 K=1,4
      CS(I,K)=0.
      DS(I,K)=0.
4001  CONTINUE
      DO 4002 I=1,10
      ITTOT(I)=0
      ITOT(I)=0
4002  CONTINUE
      IGTOT=0
      DO 3000 K=1,16
      ASUM(K)=0.
3000  CONTINUE
      DO 3002 K=1,4
      CSUM(K)=0.
3002  DSUM(K)=0.
      ITOT2=0
999   DO 10 I=1,IKT
      IFR2=IYRGF2(I)
      IFR(2)=NFR2(I)
      DO 940 K=1,12
940   AR(K)=ASUM2(I,K)
      DO 941 K=1,4
      CR(K)=CSUM2(I,K)
      DR(K)=DSUM2(I,K)
941   CONTINUE
2012  IF(IFST-1) 1301,1301,1302
1301  IFST=2
      JJ=1
      NOCL(KLAS)=IFR(2)
      ITTOT(1)=IFR(1)
      IGTOT=IFR(1)
      GO TO 1303
1302  DO 1304 JJ=1,KLAS
      IF(IFR(2)-NOCL(JJ)) 1304,1303,1304
1304  CONTINUE
```

```
      KLAS=KLAS+1
      JJ=KLAS
      NOCL(KLAS)=IFR(2)
      ITTOT(KLAS)=IFR(1)
      IGTOT=IGTOT+IFR(1)
1303  ITOT(JJ)=ITOT(JJ)+1
      DO 1012 K=1,16
      AS(JJ,K)=AS(JJ,K)+AR(K)
1012  CONTINUE
      DO 1011 K=1,4
      CS(JJ,K)=CS(JJ,K)+CR(K)
1011  DS(JJ,K)=DS(JJ,K)+DR(K)
10    CONTINUE
985   NORM=NORM+1
      NEW8=1
      IF(IKT) 2010,2000,2010
2010  DO 20 J=1,KLAS
      DO 20 JJ=1,4
      DS(J,JJ)=DS(J,JJ)/ITOT(J)
20    CS(J,JJ)=CS(J,JJ)/ITOT(J)
      DO 26 J=1,KLAS
      DO 26 JJ=1,16
26    AS(J,JJ)=AS(J,JJ)/ITOT(J)
      DO 22 I=1,4
      DO 22 IKMM=1,KLAS
      CSUM(I)=CSUM(I)+CS(IKMM,I)*ITTOT(IKMM)
22    DSUM(I)=DSUM(I)+DS(IKMM,I)*ITTOT(IKMM)
      DO 23 I=1,16
      DO 23 IKMM=1,KLAS
      ASUM(I)=ASUM(I)+AS(IKMM,I)*ITTOT(IKMM)
23    CONTINUE
      DO 24 I=1,16
      ASUM(I)=ASUM(I)/IGTOT
24    CONTINUE
      DO 25 I=1,4
      CSUM(I)=CSUM(I)/IGTOT
25    DSUM(I)=DSUM(I)/IGTOT
2000  INDEX2(NORM)=INDEX
      IROW2(NORM)=IROW
      ICOL2(NORM)=ICOL
      DO 100 I=1,16
100   ASUM2(NORM,I)=ASUM(I)
      DO 200 I=1,4
      CSUM2(NORM,I)=CSUM(I)
200   DSUM2(NORM,I)=DSUM(I)
      NFR2(NORM)=NFR
      IYRGF2(NORM)=IYRGRF
      WRITE(10,222) INDEX,IROW,ICOL
222   FORMAT(' NEW ROW AND COL. FOR CLONE NO. ',I4,' ROW',I4,' COL',I4)
14    RETURN
      END
C-----------------------------------------------------------------------C
C                          SUBROUTINE "DAVER"                           C
C                                                                       C
C     SUBROUTINE "DAVER" IS CALLED BY ROUTINE "MAIN"TO CALCULATE THE    C
C     AVERAGE NUMBER OF FLOWERS PER SAMPLE TREE FOR EACH CLONE.         C
C     "DAVER" ALSO CALLS SUBROUTINE "PCESTR" ON THE INITIAL RUN AND     C
C     SUBROUTINE "PCEMON" ON SUBSEQUENT RUNS TO OBTAIN INDIVIDUAL       C
C     SAMPLE TREE VALUES FOR PREDICTED SEED POTENTIAL, CONE             C
C     EFFICIENCY, SEED EFFICIENCY, EXTRACTION EFFICIENCY AND            C
C     GERMINATION EFFICIENCY, AND TRANSFORMS THESE INTO CLONAL AND      C
```

```
C     OVERALL ORCHARD ESTIMATES FOR THESE VARIABLES.  "DAVER" ALSO     C
C     USES THESE EFFICIENCY VALUES TO COMPUTE CLONAL AND OVERALL       C
C     ORCHARD VALUES FOR PREDICTED TOTAL SEED, EXTRACTED SEED, BUSHELS C
C     OF CONES, TOTAL CONES, POUNDS EXTRACTED SEED, TOTAL SEEDLINGS    C
C     AND SEED ORCHARD TO NURSERY EFFICIENCY.  FINALLY, "DAVER" SENDS  C
C     THESE RESULTS AS OUTPUT TO TWO DISK FILES.  ONE OF THESE         C
C     ("OUTPUT") CONTAINS ALL COMPUTED RESULTS.  THE OTHER ("CONPROD") C
C     IS A SIMPLIFIED CONE PRODUCTION REPORT.                          C
C-----------------------------------------------------------------------C

      SUBROUTINE DAVER(IMON,NOO7)
      INTEGER HCLON(3),HLOC,HCONYR,HESTCE
      COMMON HCLON,HLOC,HCONYR,IYR,INDEX,ICONBU,ISEDLB,NORM,I1ST,I2ND
      COMMON INDEX2(255),ICOL2(255),IROW2(255),ASUM2(255,16),
     1CSUM2(255,4),DSUM2(255,4),NFR2(255),IYRGF2(255)
      DIMENSION IAGE(10),NORAM(10),MONTO(6),SUMT(4),IBR(10),INF(10)
     1,IRO(10),ICO(10),ESTSE(10),ESTCE(10),ESTSP(10),ESTEE(10),ESTGE(10)
     2,AAGE(10),ACTCE(10),ACTSE(10),ACTSP(10),ACTEE(10),ACTGE(10)
     3,AMONTO(10),QCEU(10),QSEU(10),QSPU(10),QSEE(10),BAGE(10),ITTOT(10)
     4,ITOT(10),QSUM(10),NFF(10),QSGE(10),ITOTCE(10),ITTOTX(10)
      INTEGER LOC,CLONE(3),AGECL,PTREES,RAMETS,GTOTCE
      REAL ACTCE,ACTSE,ACTSP,ACTEE,ACTGE
      REAL DUM
      OPC1=0.
      OPC2=0.
      OPC3=0.
      OPC4=0.
      ADDCL=0.
      ADDC1=0.
      OPES=0.
      OPTS=0.
      OPBU=0.
      PCOS=0.
      OPVS=0.
      OPLB=0.
      INON2=0
      ANONE=0.
      ONONE=0.
      PCEU=0.
      PSEU=0.
      PSPU=0.
      PSEE=0.
      PSGE=0.

C------->INITIALIZE COUNTER "IK7" TO COUNT RECORDS READ FROM ORCHDAT1
C------->(FILE 12) OR ORCHDAT2 (FILE 09).
      IK7=1

      IF(IK7-NOO7) 8188,8188,500
8188  IF(IYR.EQ.2) GO TO 8288
C------->REWIND ORCHDAT1 (FILE 12) TO PREPARE FOR PROCESSING.
      REWIND 12
C------->READ 1ST RECORD FROM ORCHDAT1 (FILE 12) AND COMPUTE REQUIRED
C------->OUTPUT HEADER INFORMATION.
      READ(12,833) ICONYR,LOC,(CLONE(K),K=1,3),IROW,ICOL,IYRGRF,NRAMET,
     1NF,IBRCNT,(MONTO(K),K=1,5),IOBV7,IESTSP,IESTCE,IESTSE,IESTEE,
     2IESTGE
833   FORMAT(I2,4A4,17I4)
      GO TO 8289
C------->REWIND ORCHDAT2 (FILE 09) FOR PROCESSING.
8288  REWIND 09
```

```
C------->READ 1ST RECORD FROM ORCHDAT2 (FILE 09) AND COMPUTE REQUIRED
C------->OUTPUT HEADER INFORMATION.
      READ(09,833) ICONYR,LOC,(CLONE(K),K=1,3),IROW,ICOL,IYRGRF,NRAMET,
     1NF,IBRCNT,(MONTO(K),K=1,5),IOBV7,IESTSP,IESTCE,IESTSE,IESTEE,
     2IESTGE
8289  IACTYR=ICONYR-1
      ITOT2=NRAMET

C-------> PROVIDE SPACING BETWEEN CONSECUTIVE RUN REPORTS
      WRITE(10,1199)
      WRITE(10,1199)
      WRITE(11,1199)
      WRITE(11,1199)


C
C-------->PRINT OUTPUT HEADERS FOR CURRENT OBSERVATION MONTH
C
      GO TO (9001,9002,9003,9004,9005,9006,9007),IMON

C-------->HEADER FOR 1ST MARCH OBSERVATION RESULTS
9002  WRITE(10,9052) ICONYR,IACTYR
9052  FORMAT(' RESULTS FOR 1ST MAR. CONE YR. ',I2,' ACTUAL YR. ',I2)
      WRITE(11,9152) ICONYR,IACTYR
9152  FORMAT(' CONE PRODUCTION GUIDE ----- 1ST MARCH  CONE YR. ',I2,' AC
     1TUAL YR. ',I2)
      GO TO 9060

C-------->HEADER FOR 1ST JUNE OBSERVATION RESULTS
9003  WRITE(10,9053) ICONYR,IACTYR
9053  FORMAT(' RESULTS FOR 1ST JUNE CONE YR. ',I2,' ACTUAL YR. ',I2)
      WRITE(11,9153) ICONYR,IACTYR
9153  FORMAT(' CONE PRODUCTION GUIDE ----- 1ST JUNE   CONE YR. ',I2,' AC
     1TUAL YR. ',I2)
      GO TO 9060

C-------->HEADER FOR OCTOBER OBSERVATION RESULTS
9004  WRITE(10,9054) ICONYR,IACTYR
9054  FORMAT(' RESULTS FOR OCTOBER CONE YR. ',I2,' ACTUAL YR. ',I2)
      WRITE(11,9154) ICONYR,IACTYR
9154  FORMAT(' CONE PRODUCTION GUIDE ----- OCTOBER    CONE YR. ',I2,' AC
     1TUAL YR. ',I2)
      GO TO 9060

C-------->HEADER FOR 2ND MARCH OBSERVATION RESULTS
9005  WRITE(10,9055) ICONYR,ICONYR
9055  FORMAT(' RESULTS FOR 2ND MARCH CONE YR. ',I2,' ACTUAL YR. ',I2)
      WRITE(11,9155) ICONYR,ICONYR
9155  FORMAT(' CONE PRODUCTION GUIDE ----- 2ND MARCH  CONE YR. ',I2,' AC
     1TUAL YR. ',I2)
      GO TO 9060

C-------->HEADER FOR 2ND JUNE OBSERVATION RESULTS
9006  WRITE(10,9056) ICONYR,ICONYR
9056  FORMAT(' RESULTS FOR 2ND JUNE   CONE YR. ',I2,' ACTUAL YR.',I2)
      WRITE(11,9156) ICONYR,ICONYR
9156  FORMAT(' CONE PRODUCTION GUIDE ----- 2ND JUNE   CONE YR. ',I2,' AC
     1TUAL YR. ',I2)
      GO TO 9060

C------->HEADER FOR FINAL OBSERVATION RESULTS
```

```
9007  WRITE(10,9057) ICONYR,ICONYR
9057  FORMAT(' FINAL RESULTS FOR CONE YR. ',I2, 'ACTUAL YR. ',I2)
      WRITE(11,9157) ICONYR,ICONYR
9157  FORMAT(' CONE PRODUCTION GUIDE ----- FINAL      CONE YR. ',I2,' AC
     1TUAL YR. ',I2)
      GO TO 9060

C------->HEADER FOR INITIAL OBSERVATION RESULTS
9001  WRITE(10,9051) ICONYR,IACTYR
9051  FORMAT(' INITIAL RUN FOR CONE YR. ',I2,' ACTUAL YR. ',I2)
      WRITE(11,9151) ICONYR,IACTYR
9151  FORMAT(' CONE PRODUCTION GUIDE ----- INITIAL    CONE YR. ',I2,' AC
     1TUAL YR. ',I2)
9060  WRITE(11,1199)
1199  FORMAT( ' ' )
      WRITE(11,1200)
1200  FORMAT('    CLONE ID       BUSHELS OF CONES     TOTAL CONES     RA
     1METS')
      WRITE(11,1201)
1201  FORMAT('    --------       ----------------     -----------     --
     1----')


C
C------->BEGIN CALCULATIONS ON CURRENT MONTHS OBSERVATION DATA
C
      NORAM(1)=NRAMET
      ISET2=0
      ISET=0
C------->KTC COUNTS THE NUMBER OF RECORDS READ FOR EACH CLONE ID
      KTC=1
C-------> AAGE = AGE OF RAMET
C-------> IBR = INITIAL SAMPLE BRANCH COUNT
C-------> NF = TOTAL NUMBER OF FLOWERS ON SAMPLE TREE
      AAGE(1)=ICONYR-IYRGRF
      IBR(KTC)=IBRCNT
      INF(KTC)=NF
      IRO(KTC)=IROW
      ICO(KTC)=ICOL
      NFF(KTC)=NF
      ESTSE(KTC)=IESTSE/100.
      ESTCE(KTC)=IESTCE/100.
      ESTSP(KTC)=IESTSP
      ESTEE(KTC)=IESTEE/100.
      ESTGE(KTC)=IESTGE/100.

      IF(IMON-1) 2040,2040,2041

C
C------->FOR OBSERVATIONS 2 - 7
C
C-------> AMONTO = THE NUMBER OF FLOWERS COUNTED ON THE SAMPLE BRANCHES
C-------> OF A SAMPLE RAMET IN AN OBSERVATION MONTH
2041  IF(IMON-7) 5039,5038,5038
5038  AMONTO(KTC)=IOBV7
      GO TO 5040
5039  AMONTO(KTC)=MONTO(IMON-1)
5040  IF(IBRCNT) 1050,1050,1052
C-------->FOR ZERO BRANCH COUNT
1050  AMONTO(KTC)=0.
      GO TO 2040
```

```
C------->FOR POSITIVE BRANCH COUNT
C-------> AMONTO NOW BECOMES THE PERCENTAGE OF FLOWERS SURVIVING
C-------> TILL THE CURRENT OBSERVATION MONTH
1052  AMONTO(KTC)=(AMONTO(KTC)/IBR(KTC))


C
C------->FOR ALL OBSERVATION MONTHS
C
2040  HCLON(1)=CLONE(1)
      HCLON(2)=CLONE(2)
      HCLON(3)=CLONE(3)
      HLOC=LOC
      HCONYR=ICONYR
      CALL FINDCL

C
C------->INCREMENT RECORDS READ COUNTER(IK7) AND BEGIN A LOOP TO PROCESS
C------->ALL REMAINING RECORDS EXISTING ON DISK FILE 12 (ORCHDAT1) OR
C------->DISK FILE 09 (ORCHDAT2). WHEN IK7>NOO7 ALL RECORDS CONTAINED IN
C------->ORCHDAT1 OR ORCHDAT2 HAVE BEEN INITIALLY PROCESSED AND THE
C------->LOOP IS EXITED.
C

10    IK7=IK7+1
      IF(IK7-NOO7) 8189,8189,500
8189  IF(IYR.EQ.2) GO TO 8190
      READ(12,833) ICONYR,LOC,(CLONE(K),K=1,3),IROW,ICOL,IYRGRF,
     1NRAMET,NF,IBRCNT,(MONTO(K),K=1,5),IOBV7,IESTSP,IESTCE,IESTSE,
     2IESTEE,IESTGE
      GO TO 8199
8190  READ(09,833) ICONYR,LOC,(CLONE(K),K=1,3),IROW,ICOL,IYRGRF,
     1NRAMET,NF,IBRCNT,(MONTO(K),K=1,5),IOBV7,IESTSP,IESTCE,IESTSE,
     2IESTEE,IESTGE
C------->SET STATUS FLAGS ISET AND ISET2 TO ZERO
C------->ISET2 IS SET TO 1 WHEN  ALL RECORDS OF A GIVEN CLONE HAVE BEEN
C------->PROCESSED.
C------->ISET IS SET TO 1 WHEN ALL RECORDS IN FILE 12 (ORCHDAT1) OR FILE
C------->09 (ORCHDAT2) HAVE BEEN PROCESSED.
C------->HAVE BEEN PROCESSED.
8199  ISET2=0
      ISET=0
C
C------->IF CURRENT CLONE NAME(CLONE) AND LOCATION(LOC) ARE THE SAME
C------->AS LAST READ CLONE NAME(HCLON) AND LOCATION(HLOC) CONTINUE
C------->BUILDING TABLE WITH DATA CONTAINED ON THE CURRENT RECORD.
C------->IF NAME AND/OR LOCATION DIFFER GO SUM LAST CLONE ENTRIES BY
C------->AGE CLASS.
C
      IF(CLONE(1).NE.HCLON(1).OR.CLONE(2).NE.HCLON(2).OR.CLONE(3).NE.
     1HCLON(3)) GO TO 200
      IF(LOC.NE.HLOC) GO TO 200
      ITOT2=ITOT2+NRAMET
      IF(IMON-1) 110,110,120
120   KTC=KTC+1

      IF(IMON-7) 5020,1063,1063

C------>FOR OBSERVATIONS 2 - 7
5020  IF(IMON-7) 5022,5021,1063
C-------> AMONTO = THE NUMBER OF FLOWERS COUNTED ON THE SAMPLE BRANCHES
```

```
C-------> OF A SAMPLE RAMET IN AN OBSERVATION MONTH
5022  AMONTO(KTC)=MONTO(IMON-1)
      GO TO 5033
5021  AMONTO(KTC)=IOBV7
5033  IF(IBRCNT) 1060,1060,1062
C-------->FOR ZERO BRANCH COUNT
1060  AMONTO(KTC)=0.
      GO TO 1063
C-------->FOR POSITIVE BRANCH COUNT
C-------->AMONTO NOW BECOMES THE PERCENTAGE OF FLOWERS SURVIVING UNTIL
C-------->THE CURRENT OBSERVATION MONTH
1062  AMONTO(KTC)=(AMONTO(KTC)/IBRCNT)

C
C-------->FOR OBSERVATIONS 2 - 7
C
C-------> NF = TOTAL NUMBER OF FLOWERS ON SAMPLE RAMET
C-------> IBR = INITIAL SAMPLE BRANCH COUNT
C-------> AAGE = AGE OF RAMET=CONE YEAR-YEAR GRAFTED
1063  NFF(KTC)=NF
      INF(KTC)=NF
      IBR(KTC)=IBRCNT
      IRO(KTC)=IROW
      ICO(KTC)=ICOL
      NORAM(KTC)=NRAMET
      AAGE(KTC)=ICONYR-IYRGRF
      GO TO 10


C
C-------->FOR INITIAL OBSERVATION ONLY
C
C-------> KTC IS INCREMENTED TO COUNT THE NUMBER OF RECORDS READ
C-------> FOR THE CURRENT CLONE
110   KTC=KTC+1
      IBR(KTC)=IBRCNT
      INF(KTC)=NF
      IRO(KTC)=IROW
      ICO(KTC)=ICOL
      ESTCE(KTC)=IESTCE/100.
      ESTSE(KTC)=IESTSE/100.
      ESTSP(KTC)=IESTSP
      ESTEE(KTC)=IESTEE/100.
      ESTGE(KTC)=IESTGE/100.
      NFF(KTC)=NF
      NORAM(KTC)=NRAMET
      AAGE(KTC)=(ICONYR-IYRGRF)
      GO TO 10
C
C-------> CODE FOR ZERO INITIAL BRANCH COUNT CASE FOR CONE EFFICIENCY
C-------> GTOTCE = TOTAL NUMBER OF RAMETS TO BE USED AS A DIVISOR
C-------> AGECL = AGE CLASS
C-------> PTREES = NUMBER OF SAMPLE TREES IN AGE CLASS
C-------> KTC = NUMBER OF SAMPLE TREES IN CURRENT CLONE
200   GTOTCE=0
      AGECL=1
      PTREES=1
      IF(IBR(1).EQ.0) PTREES=0
      RAMETS=NORAM(1)
      IF(IBR(1).EQ.0) RAMETS=0
      MKAGE=IAGE(1)
```

```
      DO 501 I=2,KTC
      IF(MKAGE.NE.IAGE(I)) GO TO 100
      IF(MKAGE.EQ.IAGE(I).AND.IBR(I).EQ.O) GO TO 501
      PTREES=PTREES+1
      RAMETS=NORAM(I)
      GO TO 501
100   ITOTCE(AGECL)=PTREES
      ITTOTX(AGECL)=RAMETS
      GTOTCE=GTOTCE+RAMETS
      PTREES=1
      IF(IBR(I).EQ.0) PTREES=0
      RAMETS=NORAM(I)
      IF(IBR(I).EQ.0) RAMETS=0
      MKAGE=IAGE(I)
      AGECL=AGECL+1
501   CONTINUE
      ITOTCE(AGECL)=PTREES
      ITTOTX(AGECL)=RAMETS
      GTOTCE=GTOTCE+RAMETS


C
C------->BEGIN SUMMING PROCESS FOR THE CURRENT CLONE
C
C------->IF IMON=1(INITIAL OBV.), WRITE INIT. SUM INFO. TO HEADER
      IF(IMON-1) 201,201,202
201   IF(ISET-1) 2030,2031,2030
C------->ISET2=1 MEANS BEEN THRU SUMMING PROCESS
2031  IF(ISET2-1) 2030,285,2030
C
C------->BEGIN SUMMING PROCESS FOR INITIAL OBSERVATION DATA
C
2030  ISET2=1
C-------> NOCLAS = COUNTER FOR NUMBER OF AGE CLASSES IN CURRENT CLONE
      NOCLAS=1
      I=1
C-------> ITOT = COUNTER FOR NUMBER OF SAMPLE TREES IN AN AGE CLASS
      ITOT(NOCLAS)=1
C-------> ITTOT = NUMBER OF RAMETS IN AN AGE CLASS
      ITTOT(NOCLAS)=NORAM(1)
C-------> GTOT = NUMBER OF RAMETS IN THE CURRENT CLONE
      GTOT=NORAM(1)
      BAGE(NOCLAS)=AAGE(1)
      IF(KTC-1) 9670,9660,9670
9670  IL=I+1
      DO 9620 L=IL,KTC
      DO 9622 KK=1,NOCLAS
      IF(AAGE(L).EQ.BAGE(KK)) GO TO 9623
9622  CONTINUE
      NOCLAS=NOCLAS+1
      ITOT(NOCLAS)=1
      ITTOT(NOCLAS)=NORAM(L)
      BAGE(NOCLAS)=AAGE(L)
C-------> GTOT = THE TOTAL NUMBER OF RAMETS OF CURRENT CLONE OF ALL AGES
      GTOT=GTOT+NORAM(L)
      GO TO 9620
9623  ITOT(NOCLAS)=ITOT(NOCLAS)+1
9620  CONTINUE
9660  TOTAGE=0.
      DO 9624 K=1,NOCLAS
9624  TOTAGE=TOTAGE+BAGE(K)*ITTOT(K)
```

```
C-------> AGE = AVERAGE AGE OF RAMETS WITHIN CURRENT CLONE GROUP
C-------> WEIGHTED BY THE NUMBER OF RAMETS IN EACH AGE CLASS
      AGE=TOTAGE/GTOT
      ITOTR=GTOT+.0001
      DO 9735 IKM=1,NOCLAS
      QSUM(IKM)=0.
9735  CONTINUE
      DO 5502 IKM=1,KTC
      DO 9731 IKMM=1,NOCLAS
      IF(AAGE(IKM).EQ.BAGE(IKMM)) GO TO 9732
9731  CONTINUE
      GO TO 5502
C-------> QSUM = TOTAL NUMBER OF FLOWERS ON SAMPLE RAMETS OF ONE AGE
C-------> CLASS
9732  QSUM(IKMM)=QSUM(IKMM)+NFF(IKM)
5502  CONTINUE
      DO 9733 IKMM=1,NOCLAS
C-------> NOW,QSUM = AVERAGE NUMBER OF FLOWERS PER RAMET IN AN
C-------> AGE CLASS
      QSUM(IKMM)=QSUM(IKMM)/ITOT(IKMM)
9733  CONTINUE
      SUMNF=0.
      DO 9734 IKMM=1,NOCLAS
C-------> SUMNF = NUMBER OF FLOWERS FOR THE CLONE WITHIN THE ORCHARD
      SUMNF=SUMNF+QSUM(IKMM)*ITTOT(IKMM)
9734  CONTINUE
C-------> CALCULATE AVERAGE NUMBER OF FLOWERS PER RAMET ACROSS ALL AGE
C-------> CLASSES FOR THE CURRENT CLONE=SUMNF
      SUMNF=SUMNF/GTOT
      DO 9705 IKM=1,NOCLAS
      QCEU(IKM)=0.
      QSEU(IKM)=0.
      QSPU(IKM)=0.
      QSEE(IKM)=0.
      QSGE(IKM)=0.
9705  CONTINUE
      DO 5002 IKM=1,KTC
      DO 9701 IKMM=1,NOCLAS
      IF(AAGE(IKM).EQ.BAGE(IKMM)) GO TO 9702
9701  CONTINUE
      GO TO 5002
C
C-------> SEND ESTIMATES FOR CONE EFFICIENCY, SEED EFFICIENCY,
C-------> SEED POTENTIAL, EXTRACTION EFFICIENCY AND
C-------> GERMINATION EFFICIENCY TO SUBROUTINE PCESTR.
C-------> PCESTR CALCULATES AND RETURNS PREDICTED (UPDATED)
C-------> CONE EFFICIENCY, SEED EFFICIENCY, SEED POTENTIAL,
C-------> EXTRACTION EFFICIENCY AND GERMINATION EFFICIENCY.
C
9702  CALL PCESTR(IRO(IKM),ICO(IKM),ESTCE(IKM),ESTSE(IKM),
     1ESTSP(IKM),ESTEE(IKM),ESTGE(IKM),AAGE(IKM),QCEU,QSEU,QSPU,QSGE,
     2QSEE,IKMM)
5002  CONTINUE
C
C-------> COMPUTE AGE CLASS (WITHIN-CLONE) AVERAGES FOR PREDICTED
C-------> CONE EFFICIENCY, SEED EFFICIENCY, SEED POTENTIAL,
C-------> EXTRACTION EFFICIENCY AND GERMINATION EFFICIENCY.
C
      DO 9703 IKMM=1,NOCLAS
      QCEU(IKMM)=QCEU(IKMM)/ITOT(IKMM)
      QSEU(IKMM)=QSEU(IKMM)/ITOT(IKMM)
```

```
      QSPU(IKMM)=QSPU(IKMM)/ITOT(IKMM)
      QSEE(IKMM)=QSEE(IKMM)/ITOT(IKMM)
9703  QSGE(IKMM)=QSGE(IKMM)/ITOT(IKMM)
      PCEU=0.
      PSEU=0.
      PSPU=0.
      PSEE=0.
      PSGE=0.
      DO 9704 IKMM=1,NOCLAS
      PCEU=PCEU+QCEU(IKMM)*ITTOT(IKMM)
      PSEU=PSEU+QSEU(IKMM)*ITTOT(IKMM)
      PSPU=PSPU+QSPU(IKMM)*ITTOT(IKMM)
      PSEE=PSEE+QSEE(IKMM)*ITTOT(IKMM)
9704  PSGE=PSGE+QSGE(IKMM)*ITTOT(IKMM)
C
C-------> COMPUTE CLONAL AVERAGES FOR PREDICTED CONE
C-------> EFFICIENCY, SEED EFFICIENCY, SEED POTENTIAL,
C-------> EXTRACTION EFFICIENCY AND GERMINATION
C-------> EFFICIENCY.
C
      PCEU=PCEU/GTOT
      PSEU=PSEU/GTOT
      PSPU=PSPU/GTOT
      PSEE=PSEE/GTOT
      PSGE=PSGE/GTOT
      GO TO 304

C
C-------->BEGIN SUMMING PROCESS FOR FOR OBSERVATIONS 2 - FINAL(7)
C

202   IF(ISET-1) 2032,2033,2032
2033  IF(ISET2-1) 2032,285,2032
2032  ISET2=1
C-------> NOCLAS = THE NUMBER OF DIFFERENT AGE CLASSES ENCOUNTERED
C-------> WITHIN THE CURRENT CLONE ID
      NOCLAS=1
      I=1
C-------> ITOT = COUNTER FOR NUMBER OF SAMPLE TREES IN AN AGE CLASS
C-------> ITTOT = NUMBER OF RAMETS IN AN AGE CLASS FOR CURRENT CLONE
C-------> GTOT = NUMBER OF RAMETS IN CURRENT CLONE
      ITOT(NOCLAS)=1
      ITTOT(NOCLAS)=NORAM(1)
      GTOT=NORAM(1)
      BAGE(NOCLAS)=AAGE(1)
      IF(KTC-1) 2052,2051,2052
2052  IL=I+1
      DO 2053 L=IL,KTC
      DO 2054 KK=1,NOCLAS
      IF(AAGE(L).EQ.BAGE(KK)) GO TO 2055
2054  CONTINUE
      NOCLAS=NOCLAS+1
      ITOT(NOCLAS)=1
      ITTOT(NOCLAS)=NORAM(L)
      BAGE(NOCLAS)=AAGE(L)
      GTOT=GTOT+NORAM(L)
      GO TO 2053
C-------> INCREMENT NUMBER OF SAMPLE RAMETS IN THE AGE CLASS BY ONE
2055  ITOT(NOCLAS)=ITOT(NOCLAS)+1
2053  CONTINUE
C-------> CALCULATE TOTAL AGE AND THEN AVERAGE AGE FOR CLONE
```

```
2051  TOTAGE=0.
      DO 2057 K=1,NOCLAS
2057  TOTAGE=TOTAGE+BAGE(K)*ITTOT(K)
      AGE=TOTAGE/GTOT
      ITOTR=GTOT+.0001
      DO 9635 IKM=1,NOCLAS
      QSUM(IKM)=0.
9635  CONTINUE
      DO 5602 IKM=1,KTC
      DO 9631 IKMM=1,NOCLAS
      IF(AAGE(IKM).EQ.BAGE(IKMM)) GO TO 9632
9631  CONTINUE
      GO TO 5602
9632  QSUM(IKMM)=QSUM(IKMM)+NFF(IKM)
5602  CONTINUE
C-------> CALCULATE QSUM FOR EACH AGE CLASS. QSUM(IKMM) = TOTAL
C-------> NUMBER OF FLOWERS FOR RAMETS OF EACH AGE CLASS IN
C-------> CURRENT CLONE ID
      DO 9633 IKMM=1,NOCLAS
C-------> NOW, QSUM = AVERAGE NUMBER OF FLOWERS PER SAMPLE RAMET
C-------> IN AN AGE CLASS
      QSUM(IKMM)=QSUM(IKMM)/ITOT(IKMM)
9633  CONTINUE
      SUMNF=0.
      DO 9634 IKMM=1,NOCLAS
C-------> SUMNF = NUMBER OF FLOWERS FOR THE WHOLE CURRENT CLONE
      SUMNF=SUMNF+QSUM(IKMM)*ITTOT(IKMM)
9634  CONTINUE
C-------> NOW, SUMNF = AVERAGE NUMBER OF FLOWERS PER RAMET FOR THE
C-------> CURRENT CLONE
      SUMNF=SUMNF/GTOT


C
C-------->FOR OBSERVATION MONTHS 2 - 7
C
600   DO 9605 IKM=1,NOCLAS
      QCEU(IKM)=0.
      QSEU(IKM)=0.
      QSPU(IKM)=0.
      QSEE(IKM)=0.
      QSGE(IKM)=0.
9605  CONTINUE
      DO 5005 IKM=1,KTC
      DO 9601 IKMM=1,NOCLAS
      IF(AAGE(IKM).EQ.BAGE(IKMM)) GO TO 9602
9601  CONTINUE
C
C-------> CALL PCEMON TO CALCULATE UPDATED PREDICTED CONE EFFICIENCY,
C-------> SEED EFFICIENCY, SEED POTENTIAL, EXTRACTION EFFICIENCY AND
C-------> GERMINATION EFFICIENCY FOR EACH RECORD (RAMET). DAVER SENDS
C-------> PCEMON AMONTO=CURRENT MONTH'S OBSERVED CONE EFFICIENCY FOR
C-------> EACH SAMPLE RAMET. AMONTO WILL BE RECEIVED BY PCEMON AS OBCE.
C
9602  CALL PCEMON(IRO(IKM);ICO(IKM),IMON,QCEU,AMONTO(IKM),
     1QSEU,QSPU,QSEE,QSGE,IKMM)
5005  CONTINUE
C
C-------> COMPUTE AGE CLASS (WITHIN-CLONE) AVERAGES FOR PREDICTED
C-------> CONE EFFICIENCY, SEED EFFICIENCY, SEED POTENTIAL,
C-------> EXTRACTION EFFICIENCY AND GERMINATION EFFICIENCY.
```

```
C
      DO 9603 IKMM=1,NOCLAS
      IF(ITOTCE(IKMM).EQ.0) GO TO 9617
      QCEU(IKMM)=QCEU(IKMM)/ITOTCE(IKMM)
9617  QSEU(IKMM)=QSEU(IKMM)/ITOT(IKMM)
      QSPU(IKMM)=QSPU(IKMM)/ITOT(IKMM)
      QSEE(IKMM)=QSEE(IKMM)/ITOT(IKMM)
9603  QSGE(IKMM)=QSGE(IKMM)/ITOT(IKMM)
      PCEU=0.
      PSEU=0.
      PSPU=0.
      PSEE=0.
      PSGE=0.
      DO 9604 IKMM=1,NOCLAS
      IF(ITTOTX(IKMM).EQ.0) GO TO 9618
      PCEU=PCEU+QCEU(IKMM)*ITTOTX(IKMM)
9618  PSEU=PSEU+QSEU(IKMM)*ITTOT(IKMM)
      PSPU=PSPU+QSPU(IKMM)*ITTOT(IKMM)
      PSEE=PSEE+QSEE(IKMM)*ITTOT(IKMM)
9604  PSGE=PSGE+QSGE(IKMM)*ITTOT(IKMM)
      IF(GTOTCE.EQ.0) GO TO 9700
      PCEU=PCEU/GTOTCE
      GO TO 9707
9700  PCEU=0
C
C-------> COMPUTE CLONAL AVERAGES FOR PREDICTED CONE
C-------> EFFICIENCY, SEED EFFICIENCY, SEED POTENTIAL,
C-------> EXTRACTION EFFICIENCY AND GERMINATION
C-------> EFFICIENCY.
C
9707  PSEU=PSEU/GTOT
      PSPU=PSPU/GTOT
      PSEE=PSEE/GTOT
      PSGE=PSGE/GTOT
      GO TO 304


C
C-------->COMPUTE  FINAL OUTPUT VALUES FOR THE CURRENT CLONE
C
C-------> PTS = PREDICTED TOTAL SEED
C-------> PES = PREDICTED EXTRACTED SEED
C-------> PCO = PREDICTED NUMBER OF CONES
C-------> PBU = PREDICTED BUSHELS OF CONES
C-------> PLB = PREDICTED POUNDS OF SEED
C-------> PVS = PREDICTED VIABLE SEED (PREDICTED NUMBER OF SEEDLINGS)
C-------> ANONE = PREDICTED ORCHARD-TO-NURSERY EFFICIENCY
C-------> OPC1 = ORCHARD SUM PREDICTED CONE EFFICIENCY
C-------> OPC2 = ORCHARD SUM PREDICTED SEED EFFICIENCY
C-------> OPC3 = ORCHARD SUM PREDICTED EXTRACTION EFFICIENCY
C-------> OPC4 = ORCHARD SUM PREDICTED GERMINATION EFFICIENCY
C-------> OPES = ORCHARD SUM PREDICTED EXTRACTED SEED
C-------> OPTS = ORCHARD SUM PREDICTED TOTAL SEED
C-------> OPBU = ORCHARD SUM PREDICTED BUSHELS OF CONES
C-------> PCOS = ORCHARD SUM PREDICTED TOTAL CONES
C-------> OPVS = ORCHARD SUM PREDICTED VIABLE SEED (NUMBER OF SEEDLINGS)
C-------> OPLB = ORCHARD SUM PREDICTED POUNDS OF SEED
304   PTS=PCEU*ITOTR*SUMNF*PSEU*PSPU
      PES=PTS*PSEE
      PCO=ITOTR*SUMNF*PCEU
      PBU=PCO/ICONBU
```

```
      PLB=(PTS*PSEE)/ISEDLB
      PVS=PES*PSGE
      ANONE=(PCEU*PSEU*PSGE*PSEE)*100.
      OPC1=OPC1+PCEU
      OPC2=OPC2+PSEU
      OPC3=OPC3+PSGE
      OPC4=OPC4+PSEE
C-------> ADDCL COUNTS NUMBER OF CLONES
C-------> ADDC1 COUNTS NUMBER OF CLONES WITH NONZERO
C-------> FLOWER COUNTS.
      IF(SUMNF.EQ.0) GO TO 305
      ADDC1=ADDC1+1
305   ADDCL=ADDCL+1
      OPES=OPES+PES
      OPTS=OPTS+PTS
      OPBU=OPBU+PBU
      PCOS=PCOS+PCO
      OPVS=OPVS+PVS
      OPLB=OPLB+PLB


C
C-------> WRITE A CONE PRODUCTION REPORT RECORD
C
      WRITE(11,1202) HCLON(1),HCLON(2),HCLON(3),PBU,PCO,ITOTR
1202  FORMAT(4X,3A4,6X,F12.2,6X,F12.2,6X,I3)
C
C------->OUTPUT CALCULATED RESULTS FOR CURRENT CLONE
C
      WRITE(10,1357)
      WRITE(10,1361) HCLON(1),HCLON(2),HCLON(3),LOC,HCONYR
1361  FORMAT(' RESULTS FOR CLONE ',3A4,' ORCHARD ',A4,' YEAR ',I2)
      IF(SUMNF.EQ.0) GO TO 1608
      WRITE(10,1637) PCEU,PSEU
1637  FORMAT('  CONE EFF. (PCE)=',F6.2,4X,' SEED EFF. (PSE)=',F6.2)
      GO TO 1610
1608  WRITE(10,1609) PSEU
1609  FORMAT('  CONE EFF. (PCE)=  NO DATA',2X,' SEED EFF. (PSE)=',F6.2)
1610  WRITE(10,9637) PSPU,PSEE
9637  FORMAT('  SEED POT. (PSP)=',F7.1,4X,' EXT. EFF. (PEE)=',F6.2)
      WRITE(10,1639) PSGE,ITOTR
1639  FORMAT('  GERM. EFF. (PGE)=',F6.2,4X,'NO. RAMETS (NR)=',I3)
      WRITE(10,9638) SUMNF
      WRITE(10,1638) ICONBU,ISEDLB
9638  FORMAT('  AVG. FEM. FLOWERS/RAMET (NF)=',F8.2)
1638  FORMAT('  CONES PER BU.= ',I6,4X,' SEED PER LB.=',I6)

C
C------->OUTPUT ESTIMATED VALUES FOR:
C------->PREDICTED SEED, EXTRACTED SEED, BUSHELS OF CONES, CONES,
C------->LBS. EXTRACTED SEED AND NUMBER OF SEEDLINGS.
C
      WRITE(10,1362) PTS
1362  FORMAT(' PREDICTED SEED (PTS=NR*NF*PCE*PSP*PSE)',10X,F15.2)
      WRITE(10,1363) PES
1363  FORMAT(' PREDICTED NO. EXT. SEED (PES=PTS*PEE)',11X,F15.2)
      WRITE(10,1364) PBU
      WRITE(10,9987) PCO
1364  FORMAT(' PREDICTED BUSHELS OF CONES (PBU=(NR*NF*PCE)/CONBU)
     1,F12.2)
9987  FORMAT(' PREDICTED CONES (PCO=(NR*NF*PCE)) ',17X,F12.2)
```

```
      WRITE(10,1365) PLB
1365  FORMAT(' PREDICTED LBS EXTRACTED SEED (PLB=(PTS*PEE)/SEED
     1F12.2)
      WRITE(10,1366) PVS
      IF(SUMNF.EQ.0) GO TO 9555
      WRITE(10,9561) ANONE
      GO TO 1380
9555  WRITE(10,9562)
1366  FORMAT(' PREDICTED NO. SEEDLINGS (PVS=PES*PGE)',10X,F15.2)


C
C-------->IF ISET=1, ALL CLONES HAVE BEEN PROCESSED AND THE OVERALL
C-------->ORCHARD RESULTS WILL NEXT BE OUTPUT. IF ISET=0 GO PROCESS THE
C-------->NEXT CLONE.
C
1380  IF(ISET) 1383,1383,1384
1383  IF(HLOC.EQ.LOC) GO TO 205
1384  WRITE(10,1357)
      WRITE(10,1357)
      WRITE(10,1357)
1357  FORMAT('=====================================')
      WRITE(10,1359) LOC
1359  FORMAT(' RESULTS FOR ORCHARD ',A4)
      WRITE(10,1391) OPTS
      WRITE(10,1392) OPBU
      WRITE(10,9985) PCOS
      WRITE(10,1393) OPVS
      WRITE(10,1394) OPLB
C-------> OPC1 = ORCHARD AVERAGE PREDICTED CONE EFFICIENCY
C-------> OPC2 = ORCHARD AVERAGE PREDICTED SEED EFFICIENCY
C-------> OPC3 = ORCHARD AVERAGE PREDICTED EXTRACTION EFFICIENCY
C-------> OPC4 = ORCHARD AVERAGE PREDICTED GERMINATION EFFICIECY
      OPC1=OPC1/ADDC1
      OPC2=OPC2/ADDCL
      OPC3=OPC3/ADDCL
      OPC4=OPC4/ADDCL
C-------> ONONE = AVERAGE ORCHARD SEED ORCHARD-TO-NURSERY EFFICIENCY
      ONONE=(OPC1*OPC2*OPC3*OPC4)*100.
      WRITE(10,9561) ONONE
      ONONE=0.
      OPES=0.
      OPTS=0.
      OPBU=0.
      OPVS=0.
      PCOS=0.
      ADDCL=0.
      ADDC1=0.
      OPC1=0.
      OPC2=0.
      OPC3=0.
      OPC4=0.

205   IF(ISET) 280,280,285
1391  FORMAT(' PREDICTED ORCHARD SEED PRODUCTION (SUM PTS) ',F12.1)
1392  FORMAT(' PREDICTED NO. OF BUSHELS OF CONES (SUM PBU) ',F12.1)
9985  FORMAT(' PREDICTED NO. OF CONES (SUM PCO) ',11X,F12.1)
1393  FORMAT(' PREDICTED NO. OF SEEDLINGS (SUM PVS)',8X,F12.1)
1394  FORMAT(' PREDICTED LBS. OF SEED (SUM PLB)',12X,F12.1)
9561  FORMAT(' PREDICTED ORCH. TO NURS. EFF.',26X,F8.2,' %')
9562  FORMAT(' PREDICTED ORCH. TO NURS. EFF.',26X,' NO DATA')
```

```
280   IF(IMON-1) 210,210,220
220   SUMT(IMON)=0
      KTC=0.
      SUMNF=0.
      ITOTR=0
      TOTAGE=0.
      ITOT2=0
      HCONYR=ICONYR
      HCLON(1)=CLONE(1)
      HCLON(2)=CLONE(2)
      HCLON(3)=CLONE(3)
      HLOC=LOC
      HESTCE=IESTCE
      CALL FINDCL
      GO TO 120
210   SUMNF=0
      ITOTR=0
      TOTAGE=0.
      ITOT2=0
      KTC=0
      HCONYR=ICONYR
      HCLON(1)=CLONE(1)
      HCLON(2)=CLONE(2)
      HCLON(3)=CLONE(3)
      HLOC=LOC
      CALL FINDCL
      GO TO 110

500   ISET=1
      GO TO 200

285   RETURN
      END
C-----------------------------------------------------------------------C
C                           SUBROUTINE "PCESTR"                         C
C                                                                       C
C     SUBROUTINE "PCESTR" IS CALLED BY SUBROUTINE "DAVER" ON THE        C
C     INITIAL RUN TO COMPUTE PREDICTED CONE EFFICIENCY, SEED            C
C     POTENTIAL, SEED EFFICIENCY, EXTRACTION EFFICIENCY, AND            C
C     GERMINATION EFFICIENCY VALUES FOR EACH SAMPLE FROM THE USER'S     C
C     OBSERVATIONS FILE.  THESE VALUES ARE BOTH STORED IN THE "PCETAB"  C
C     TABLE AND RETURNED TO "DAVER."                                    C
C-----------------------------------------------------------------------C

      SUBROUTINE PCESTR(IRO,ICOL,EST,ESTSE,ESTSP,ESTEE,ESTGE,AGE,PCE1,
     1PSEU1,PSP1,PSG1,PSE1,IKMM)
      INTEGER CLONE(3)
      COMMON CLONE,LOC,ICONYR,IYR,INDEX,ICONBU,ISEDLB,NORMM,I1ST,I2ND
      COMMON INDEX2(255),ICOL2(255),IROW2(255),ASUM2(255,16),
     1CSUM2(255,4),DSUM2(255,4),NFR2(255),IYRGF2(255)
      DIMENSION PSEU1(10),PCE1(10),PSP1(10),PSG1(10),PSE1(10)
      REAL DUM

      IF(ICONYR-I1ST) 10,20,40
10    WRITE(19,12)
12    FORMAT(' ERROR ON YEAR --- RUN IS TERMINATED. ')
      STOP

40    IF(ICONYR-I2ND) 10,20,10
```

```
20    IF(EST.EQ.0.) GO TO 700
      PCE=EST
      GO TO 702
700   PCE=0.
702   PCE1(IKMM)=PCE1(IKMM)+PCE
      IF(ESTSE.EQ.0.) GO TO 704
      PSEU=ESTSE
      GO TO 705
704   PSEU=0.
705   PSEU1(IKMM)=PSEU1(IKMM)+PSEU
      PSP=ESTSP
      PSP1(IKMM)=PSP1(IKMM)+PSP
      PSG=ESTGE
      PSE=ESTEE
      PSG1(IKMM)=PSG1(IKMM)+PSG
      PSE1(IKMM)=PSE1(IKMM)+PSE

70    IF(MOD(IYR,2)) 17,18,17
C------->STORE INITIAL PCE FOR ODD YEAR
17    CALL PCETAB(IRO,ICOL,PCE,DUM,1,DUM,DUM,DUM,DUM,DUM,DUM,DUM,DUM,DUM
     1,DUM,DUM,DUM)
C------->STORE INITIAL PSEU AND PSP
      CALL PCETAB(IRO,ICOL,PSEU,PSP,19,PSE,PSG,DUM,DUM,DUM,DUM,DUM,DUM,D
     1UM,DUM,DUM,DUM)
      RETURN

C------->STORE INITIAL PCE FOR EVEN YR.
18    CALL PCETAB(IRO,ICOL,PCE,DUM,6,DUM,DUM,DUM,DUM,DUM,DUM,DUM,DUM,DUM
     1,DUM,DUM,DUM)
C------->STORE INITIAL PSEU,PSP FOR EVEN YR.
      CALL PCETAB(IRO,ICOL,PSEU,PSP,21,PSE,PSG,DUM,DUM,DUM,DUM,DUM,DUM,D
     1UM,DUM,DUM,DUM)
      RETURN
      END

C-----------------------------------------------------------------------C
C                          SUBROUTINE "PCEMON"                          C
C                                                                       C
C     SUBROUTINE "PCEMON" IS CALLED BY SUBROUTINE "DAVER" ON ALL        C
C     RUNS EXCEPT THE INITIAL RUN TO COMPUTE A PREDICTED CONE           C
C     EFFICIENCY UPDATE FOR EACH SAMPLE TREE BASED ON THE OBSERVED      C
C     CONE EFFICIENCY VALUE SENT BY "DAVER" FOR A SPECIFIC OBSERVATION C
C     MONTH (1ST MARCH, 1ST JUNE, 1ST OCTOBER, 2ND MARCH, 2ND JUNE OR   C
C     FINAL).  FOR A GIVEN OBSERVATION MONTH, "PCEMON" PLUGS THE        C
C     OBSERVED CONE EFFICIENCY VALUE INTO AN EQUATION BASED ON A        C
C     GENERALIZED CONE MORTALITY CURVE, TO CREATE A PREDICTED CONE      C
C     EFFICIENCY UPDATE FOR EACH SAMPLE TREE.  THIS VALUE AND VALUES    C
C     FOR PREDICTED SEED POTENTIAL, SEED EFFICIENCY, EXTRACTION         C
C     EFFICIENCY AND GERMINATION EFFICIENCY RETRIEVED FROM THE          C
C     "PCETAB" TABLE ARE RETURNED TO "DAVER."  THE VALUE FOR            C
C     PREDICTED CONE EFFICIENCY UPDATE IS ALSO STORED IN THE "PCETAB"   C
C     TABLE.                                                            C
C-----------------------------------------------------------------------C

      SUBROUTINE PCEMON(IRO,ICOL,IMON,PCEU1,OBCE,PSEU1,PSPU1,PSEE1,PSGE1
     1,IKMM)
      INTEGER CLONE(3)
      COMMON CLONE,LOC,ICONYR,IYR,INDEX,ICONBU,ISEDLB,NORMM,I1ST,I2ND
      COMMON INDEX2(255),ICOL2(255),IROW2(255),ASUM2(255,16),
     1CSUM2(255,4),DSUM2(255,4),NFR2(255),IYRGF2(255)
      DIMENSION PSPU1(10),PCEU1(10),PSEU1(10),PSEE1(10),PSGE1(10)
```

```
      REAL DUM

C-------->GET INITIAL PCE FOR EVEN OR ODD YEAR
      IF(MOD(IYR,2)) 22,23,22
22    CALL PCETAB(IRO,ICOL,PCE,DUM,3,DUM,DUM,DUM,DUM,DUM,DUM,DUM,DUM,DUM
     1,DUM,DUM,DUM)
      GO TO 24
C-------->GET INITIAL PCE FOR EVEN YEAR
23    CALL PCETAB(IRO,ICOL,PCE,DUM,10,DUM,DUM,DUM,DUM,DUM,DUM,DUM,DUM,DU
     1M,DUM,DUM,DUM)

C-------->CHECK TO BE SURE A VALID OBSERVATION MONTH HAS BEEN PASSED
C-------->OBSERVATIONS 2 - 7 ARE VALID.
24    IF(IMON-1) 10,10,20
20    IF(IMON-7) 25,25,10
10    WRITE(19,12)
12    FORMAT(' ERROR IN MONTH --- RUN IS TERMINATED. ')
      STOP

C-------->BRANCH TO THE MODULE FOR THE CURRENT OBSERVATION MONTH
25    INON=IMON-1
      GO TO (100,200,300,400,500,600),INON

C
C-------->CALCULATE AND ENTER PCEU FOR 1ST MARCH OBSERVATION
C
100   IF(OBCE.LE.0.) GO TO 701
      IF(PCE.EQ.0.) GO TO 701
      GO TO 702
701   PCEU=0.0
      GO TO 703
C--------> THIS EQUATION USES THE ORCHARD'S GENERALIZED CONE MORTALITY
C--------> CURVE TO CONVERT THE OBSERVED CONE EFFICIENCY (OBCE SENT FROM
C--------> DAVER) TO UPDATED PREDICTED CONE EFFICIENCY FOR THE FIRST
702   PCEU=PCE+.05-(1.00-OBCE)
      IF(PCEU.GT.1.00) PCEU=1.00
      IF(PCEU.LT.0.00) PCEU=0.00
703   PCEU1(IKMM)=PCEU1(IKMM)+PCEU

      IF(MOD(IYR,2)) 102,103,102
C-------->STORE FOR MONTH 1, PCEU (ODD)
102   CALL PCETAB(IRO,ICOL,PCEU,DUM,4,DUM,DUM,DUM,DUM,DUM,DUM,DUM,DUM,DU
     1M,DUM,DUM,DUM)
C-------->GET THE STORED PSEU AND PSP (ODD)
      CALL PCETAB(IRO,ICOL,PSEU,PSPU,20,PSEE,PSGE,DUM,DUM,DUM,DUM,DUM,DU
     1M,DUM,DUM,DUM,DUM)
      PSPU1(IKMM)=PSPU1(IKMM)+PSPU
      PSEU1(IKMM)=PSEU1(IKMM)+PSEU
      PSEE1(IKMM)=PSEE1(IKMM)+PSEE
      PSGE1(IKMM)=PSGE1(IKMM)+PSGE
      RETURN

103   CALL PCETAB(IRO,ICOL,PCEU,DUM,11,DUM,DUM,DUM,DUM,DUM,DUM,DUM,DUM,D
     1UM,DUM,DUM,DUM)
      CALL PCETAB(IRO,ICOL,PSEU,PSPU,22,PSEE,PSGE,DUM,DUM,DUM,DUM,DUM,DU
     1M,DUM,DUM,DUM,DUM)
      PSPU1(IKMM)=PSPU1(IKMM)+PSPU
      PSEU1(IKMM)=PSEU1(IKMM)+PSEU
      PSEE1(IKMM)=PSEE1(IKMM)+PSEE
      PSGE1(IKMM)=PSGE1(IKMM)+PSGE
      RETURN
```

```
C
C------->CALCULATE AND ENTER VALUES FOR 1ST JUNE OBSERVATION
C
200    IF(OBCE.LE.0.) GO TO 704
       IF(PCE.EQ.0.) GO TO 704
       GO TO 705
704    PCEU=0.0
       GO TO 706
C-------> THIS EQUATION USES THE ORCHARD'S GENERALIZED CONE MORTALITY
C-------> CURVE TO CONVERT THE OBSERVED CONE EFFICIENCY (OBCE) TO
C-------> UPDATED PREDICTED CONE EFFICIENCY FOR THE FIRST JUNE
C-------> OBSERVATION IN THE CONE CYCLE.
705    PCEU=PCE+.15-(1.00-OBCE)
       IF(PCEU.GT.1.00) PCEU=1.00
       IF(PCEU.LT.0.00) PCEU=0.00
706    PCEU1(IKMM)=PCEU1(IKMM)+PCEU

       IF(MOD(IYR,2)) 202,203,202
203    CALL PCETAB(IRO,ICOL,PCEU,DUM,12,DUM,DUM,DUM,DUM,DUM,DUM,DUM,DUM,D
      1UM,DUM,DUM,DUM)
       CALL PCETAB(IRO,ICOL,PSEU,PSPU,22,PSEE,PSGE,DUM,DUM,DUM,DUM,DUM,DU
      1M,DUM,DUM,DUM,DUM)
       PSPU1(IKMM)=PSPU1(IKMM)+PSPU
       PSEU1(IKMM)=PSEU1(IKMM)+PSEU
       PSEE1(IKMM)=PSEE1(IKMM)+PSEE
       PSGE1(IKMM)=PSGE1(IKMM)+PSGE
       RETURN

202    CALL PCETAB(IRO,ICOL,PCEU,DUM,7,DUM,DUM,DUM,DUM,DUM,DUM,DUM,DUM,DU
      1M,DUM,DUM,DUM)
       CALL PCETAB(IRO,ICOL,PSEU,PSPU,20,PSEE,PSGE,DUM,DUM,DUM,DUM,DUM,DU
      1M,DUM,DUM,DUM,DUM)
       PSPU1(IKMM)=PSPU1(IKMM)+PSPU
       PSEU1(IKMM)=PSEU1(IKMM)+PSEU
       PSEE1(IKMM)=PSEE1(IKMM)+PSEE
       PSGE1(IKMM)=PSGE1(IKMM)+PSGE
       RETURN

C
C------->CALCULATE AND INPUT VALUES FOR OCTOBER OBSERVATION MONTH
C
300    IF(OBCE.LE.0.) GO TO 707
       IF(PCE.EQ.0.) GO TO 707
       GO TO 708
707    PCEU=0.0
       GO TO 709
C-------> THIS EQUATION USES THE ORCHARD'S GENERALIZED CONE MORTALITY
C-------> CURVE TO CONVERT THE OBSERVED CONE EFFICIENCY (OBCE) TO
C-------> UPDATED PREDICTED CONE EFFICIENCY FOR THE OCTOBER OBSERVATION
C-------> IN THE CONE CYCLE.
708    PCEU=PCE+.20-(1.00-OBCE)
       IF(PCEU.GT.1.00) PCEU=1.00
       IF(PCEU.LT.0.00) PCEU=0.00
709    PCEU1(IKMM)=PCEU1(IKMM)+PCEU

       IF(MOD(IYR,2)) 302,303,302
303    CALL PCETAB(IRO,ICOL,PCEU,DUM,13,DUM,DUM,DUM,DUM,DUM,DUM,DUM,DUM,D
      1UM,DUM,DUM,DUM)
       CALL PCETAB(IRO,ICOL,PSEU,PSPU,22,PSEE,PSGE,DUM,DUM,DUM,DUM,DUM,DU
      1M,DUM,DUM,DUM,DUM)
```

```
      PSPU1(IKMM)=PSPU1(IKMM)+PSPU
      PSEU1(IKMM)=PSEU1(IKMM)+PSEU
      PSEE1(IKMM)=PSEE1(IKMM)+PSEE
      PSGE1(IKMM)=PSGE1(IKMM)+PSGE
      RETURN

302   CALL PCETAB(IRO,ICOL,PCEU,DUM,8,DUM,DUM,DUM,DUM,DUM,DUM,DUM,DUM,DU
     1M,DUM,DUM,DUM)
      CALL PCETAB(IRO,ICOL,PSEU,PSPU,20,PSEE,PSGE,DUM,DUM,DUM,DUM,DUM,DU
     1M,DUM,DUM,DUM,DUM)
      PSPU1(IKMM)=PSPU1(IKMM)+PSPU
      PSEU1(IKMM)=PSEU1(IKMM)+PSEU
      PSEE1(IKMM)=PSEE1(IKMM)+PSEE
      PSGE1(IKMM)=PSGE1(IKMM)+PSGE
      RETURN

C
C-------->CALCULATE AND INPUT VALUES FOR 2ND MARCH OBSERVATION MONTH
C
400   IF(OBCE.LE.0.) GO TO 710
      IF(PCE.EQ.0.) GO TO 710
      GO TO 711
710   PCEU=0.0
      GO TO 712
C--------> THIS EQUATION USES THE ORCHARD'S GENERALIZED CONE MORTALITY
C--------> CURVE TO CONVERT THE OBSERVED CONE EFFICIENCY (OBCE) TO
C--------> UPDATED PREDICTED CONE EFFICIENCY FOR THE SECOND MARCH
C--------> OBSERVATION IN THE CONE CYCLE.
711   PCEU=PCE+.25-(1.00-OBCE)
      IF(PCEU.GT.1.00) PCEU=1.00
      IF(PCEU.LT.0.00) PCEU=0.00
712   PCEU1(IKMM)=PCEU1(IKMM)+PCEU
      IF(MOD(IYR,2)) 402,403,402
403   CALL PCETAB(IRO,ICOL,PCEU,DUM,14,DUM,DUM,DUM,DUM,DUM,DUM,DUM,DUM,D
     1UM,DUM,DUM,DUM)
      CALL PCETAB(IRO,ICOL,PSEU,PSPU,22,PSEE,PSGE,DUM,DUM,DUM,DUM,DUM,DU
     1M,DUM,DUM,DUM,DUM)
      PSPU1(IKMM)=PSPU1(IKMM)+PSPU
      PSEU1(IKMM)=PSEU1(IKMM)+PSEU
      PSEE1(IKMM)=PSEE1(IKMM)+PSEE
      PSGE1(IKMM)=PSGE1(IKMM)+PSGE
      RETURN

402   CALL PCETAB(IRO,ICOL,PCEU,DUM,9,DUM,DUM,DUM,DUM,DUM,DUM,DUM,DUM,DU
     1M,DUM,DUM,DUM)
      CALL PCETAB(IRO,ICOL,PSEU,PSPU,20,PSEE,PSGE,DUM,DUM,DUM,DUM,DUM,DU
     1M,DUM,DUM,DUM,DUM)
      PSPU1(IKMM)=PSPU1(IKMM)+PSPU
      PSEU1(IKMM)=PSEU1(IKMM)+PSEU
      PSEE1(IKMM)=PSEE1(IKMM)+PSEE
      PSGE1(IKMM)=PSGE1(IKMM)+PSGE
      RETURN

C
C-------->CALCULATE AND INPUT VALUES FOR 2ND JUNE OBSERVATION MONTH
C
500   IF(OBCE.LE.0.) GO TO 713
      IF(PCE.EQ.0.) GO TO 713
      GO TO 714
713   PCEU=0.0
      GO TO 715
```

```
C-------> THIS EQUATION USES THE ORCHARD'S GENERALIZED MORTALITY
C-------> CURVE TO CONVERT THE OBSERVED CONE EFFICIENCY (OBCE) TO
C-------> UPDATED PREDICTED CONE EFFICIENCY FOR THE SECOND JUNE
C-------> OBSERVATION IN THE CONE CYCLE.
714    PCEU=PCE+.30-(1.00-OBCE)
       IF(PCEU.GT.1.00) PCEU=1.00
       IF(PCEU.LT.0.00) PCEU=0.00
715    PCEU1(IKMM)=PCEU1(IKMM)+PCEU

       IF(MOD(IYR,2)) 502,503,502
503    CALL PCETAB(IRO,ICOL,PCEU,DUM,17,DUM,DUM,DUM,DUM,DUM,DUM,DUM,DUM,D
      1UM,DUM,DUM,DUM)
       CALL PCETAB(IRO,ICOL,PSEU,PSPU,22,PSEE,PSGE,DUM,DUM,DUM,DUM,DUM,DU
      1M,DUM,DUM,DUM,DUM)
       PSPU1(IKMM)=PSPU1(IKMM)+PSPU
       PSEU1(IKMM)=PSEU1(IKMM)+PSEU
       PSEE1(IKMM)=PSEE1(IKMM)+PSEE
       PSGE1(IKMM)=PSGE1(IKMM)+PSGE
       RETURN

502    CALL PCETAB(IRO,ICOL,PCEU,DUM,15,DUM,DUM,DUM,DUM,DUM,DUM,DUM,DUM,D
      1UM,DUM,DUM,DUM)
       CALL PCETAB(IRO,ICOL,PSEU,PSPU,20,PSEE,PSGE,DUM,DUM,DUM,DUM,DUM,DU
      1M,DUM,DUM,DUM,DUM)
       PSPU1(IKMM)=PSPU1(IKMM)+PSPU
       PSEU1(IKMM)=PSEU1(IKMM)+PSEU
       PSEE1(IKMM)=PSEE1(IKMM)+PSEE
       PSGE1(IKMM)=PSGE1(IKMM)+PSGE
       RETURN

C
C-------> CALCULATE AND INPUT VALUES FOR FINAL OBSERVATION MONTH
C
600    IF(OBCE.LE.0.) GO TO 813
       IF(PCE.EQ.0.) GO TO 813
       GO TO 814
813    PCEU=0.0
       GO TO 815
C-------> THIS EQUATION USES THE ORCHARD'S GENERALIZED CONE MORTALITY
C-------> CURVE TO CONVERT THE OBSERVED CONE EFFICIENCY (OBCE) TO
C-------> UPDATED PREDICTED CONE EFFICIENCY FOR THE FINAL OBSERVATION
C-------> IN THE CONE CYCLE.
814    PCEU=PCE+.45-(1.00-OBCE)
       IF(PCEU.GT.1.00) PCEU=1.00
       IF(PCEU.LT.0.00) PCEU=0.00
815    PCEU1(IKMM)=PCEU1(IKMM)+PCEU

       IF(MOD(IYR,2)) 603,602,603
603    CALL PCETAB(IRO,ICOL,PCEU,DUM,16,DUM,DUM,DUM,DUM,DUM,DUM,DUM,DUM,D
      1UM,DUM,DUM,DUM)
       CALL PCETAB(IRO,ICOL,PSEU,PSPU,20,PSEE,PSGE,DUM,DUM,DUM,DUM,DUM,DU
      1M,DUM,DUM,DUM,DUM)
       PSPU1(IKMM)=PSPU1(IKMM)+PSPU
       PSEU1(IKMM)=PSEU1(IKMM)+PSEU
       PSEE1(IKMM)=PSEE1(IKMM)+PSEE
       PSGE1(IKMM)=PSGE1(IKMM)+PSGE
       RETURN

602    CALL PCETAB(IRO,ICOL,PCEU,DUM,18,DUM,DUM,DUM,DUM,DUM,DUM,DUM,DUM,D
      1UM,DUM,DUM,DUM)
       CALL PCETAB(IRO,ICOL,PSEU,PSPU,22,PSEE,PSGE,DUM,DUM,DUM,DUM,DUM,DU
```

```
     1M,DUM,DUM,DUM,DUM)
      PSPU1(IKMM)=PSPU1(IKMM)+PSPU
      PSEU1(IKMM)=PSEU1(IKMM)+PSEU
      PSEE1(IKMM)=PSEE1(IKMM)+PSEE
      PSGE1(IKMM)=PSGE1(IKMM)+PSGE
      RETURN
      END
C-----------------------------------------------------------------------C
C                           SUBROUTINE "PCETAB"                         C
C                                                                       C
C     SUBROUTINE "PCETAB" IS CALLED BY SUBROUTINES "PCESTR" AND
C     "PCEMON" TO STORE DATA IN, AND RETRIEVE DATA FROM THE             C
C     PREDICTED CONE EFFICIENCY TABLE.                                  C
C     SUBROUTINE "PCETAB" IS CALLED BY SUBROUTINE "PCESTR" TO           C
C     STORE STARTING VALUES IN THE PCETAB TABLE WHICH CONTAINS,         C
C     FOR EACH SAMPLE TREE, INITIAL ESTIMATES FOR PREDICTED CONE        C
C     EFFICIENCY, SEED POTENTIAL,SEED EFFICIENCY, EXTRACTION            C
C     EFFICIENCY AND GERMINATION EFFICIENCY.  "PCETAB" IS ALSO          C
C     CALLED BY SUBROUTINE "PCEMON" TO RETRIEVE THESE VALUES FOR        C
C     "PCEMON" TO USE IN ITS CALCULATIONS AND "PCETAB" IS CALLED        C
C     BY "PCEMON" TO STORE THE PREDICTED CONE EFFICIENCY UPDATES        C
C     THAT IT COMPUTES FOR EACH SAMPLE TREE IN THE "PCETAB" TABLE.      C
C-----------------------------------------------------------------------C

      SUBROUTINE PCETAB(IRO,ICOL,PCESUM,C2,ICOD,PCSSUM,CS2,PCPSUM,CP2,
     1PEESUM,CE2,PGSUM,CG2,ASE,ASP,AEE,AGG)
      INTEGER CLONE(3)
      COMMON CLONE,LOC,ICONYR,IYR,INDEX,ICONBU,ISEDLB,NORM,I1ST,I2ND
      COMMON INDEX2(255),ICOL2(255),IROW2(255),ASUM2(255,16),
     1CSUM2(255,4),DSUM2(255,4),NFR2(255),IYRGF2(255)
C-------->ASUM2 IS FOR INIT CE OF ODD AND EVEN YR. AND FOR
C-------->CE UPDATES FOR 1ST MARCH, 1ST JUNE, 1ST OCTOBER,
C-------->2ND MARCH, 2ND JUNE AND FINAL OF ODD AND EVEN YR.
C-------->CSUM2 IS FOR INIT SP AND SE OF ODD AND EVEN YR.
C-------->DSUM2 IS FOR INIT EE AND GE OF ODD AND EVEN YR.

C-------->IF AN INVALID CODE HAS BEEN PASSED TO "PCETAB" PRINT
C-------->A MESSAGE AND ABORT THE RUN.
      IF(ICOD-1) 10,40,20
20    IF(ICOD-24) 40,40,10
10    WRITE(19,111)
111   FORMAT(' ILLEGAL CODE IN PCETAB ROUTINE --- RUN IS TERMINATED. ')
      STOP

C-------->CALL FINDPO TO FIND THE POSITION OF THE CURRENT CLONE
C-------->IN THE PCETAB TABLE (FILE 02).  FINDPO IS SENT INDEX,
C-------->IRO AND ICOL. FINDPO RETURNS I.
40    CALL FINDPO(IRO,ICOL,I,INDL)

C-------->GO TO THE MODULE INDICATED BY THE OPERATION CODE (ICOD)

      GO TO (41,80,43,44,80,41,47,48,49,43,44,47,48,49,50,51,50,51,701,
     1702,701,702,80,80),ICOD

C-------->CODE  1  -- STORE INITIAL PCE FOR ODD YEAR
C-------->CODE  6  -- STORE INITIAL PCE FOR EVEN YEAR
41    IF(ICOD-1) 456,456,457
456   ASUM2(I,1)=PCESUM
      RETURN
457   ASUM2(I,2)=PCESUM
      RETURN
```

```
C------->CODE  3 -- RETRIEVE INITIAL PCE FOR ODD YEAR
C------->CODE 10 -- RETRIEVE INITIAL PCE FOR EVEN YEAR
43     IF(ICOD-3) 576,576,577
576    PCESUM=ASUM2(I,1)
       RETURN
577    PCESUM=ASUM2(I,2)
       RETURN

C------->CODE  4 -- STORE PCEU FOR 1ST MARCH OBV., ODD YEAR
C------->CODE 11 -- STORE PCEU FOR 1ST MARCH OBV., EVEN YEAR
44     IF(ICOD-4) 586,586,587
586    ASUM2(I,5)=PCESUM
       RETURN
587    ASUM2(I,11)=PCESUM
       RETURN

C------->CODE  7 -- STORE PCEU FOR 1ST JUNE OBV., ODD YEAR
C------->CODE 12 -- STORE PCEU FOR 1ST JUNE OBV., EVEN YEAR
47     IF(ICOD-7) 596,596,597
596    ASUM2(I,6)=PCESUM
       RETURN
597    ASUM2(I,12)=PCESUM
       RETURN

C------->CODE  8 -- STORE PCEU FOR OCTOBER OBV., ODD YEAR
C------->CODE 13 -- STORE PCEU FOR OCTOBER OBV., EVEN YEAR
48     IF(ICOD-8) 606,606,607
606    ASUM2(I,7)=PCESUM
       RETURN
607    ASUM2(I,13)=PCESUM
       RETURN

C------->CODE  9 -- STORE PCEU FOR 2ND MARCH OBV., ODD YEAR
C------->CODE 14 -- STORE PCEU FOR 2ND MARCH OBV., EVEN YEAR
49     IF(ICOD-9) 616,616,617
616    ASUM2(I,8)=PCESUM
       RETURN
617    ASUM2(I,14)=PCESUM
       RETURN

C------->CODE 15 -- STORE PCEU FOR 2ND JUNE OBV., ODD YEAR
C------->CODE 17 -- STORE PCEU FOR 2ND JUNE OBV., EVEN YEAR
50     IF(ICOD-15) 626,626,627
626    ASUM2(I,9)=PCESUM
       RETURN
627    ASUM2(I,15)=PCESUM
       RETURN

C------->CODE 16 -- STORE PCEU FOR FINAL OBV., ODD YEAR
C------->CODE 18 -- STORE PCEU FOR FINAL OBV., EVEN YEAR
51     IF(ICOD-16) 636,636,637
636    ASUM2(I,10)=PCESUM
       RETURN
637    ASUM2(I,16)=PCESUM
       RETURN


C------->CODE 19 -- STORE INITIAL PSEU AND PSP FOR ODD YEAR
C------->CODE 21 -- STORE INITIAL PSEU AND PSP FOR EVEN YEAR
```

```
701   IF(ICOD-19) 757,757,758
C------->STORE FOR ODD YR., INITIAL PSEU AND PSP
757   CSUM2(I,1)=PCESUM
      CSUM2(I,2)=C2
      DSUM2(I,1)=PCSSUM
      DSUM2(I,2)=CS2
      RETURN
C------->STORE FOR EVEN YR., INITIAL PSEU AND PSP AND PSE,PSG
758   CSUM2(I,3)=PCESUM
      CSUM2(I,4)=C2
      DSUM2(I,3)=PCSSUM
      DSUM2(I,4)=CS2
      RETURN

C------->CODE 20 --- GET STORED PSEU AND PSP FOR ODD YEAR
C------->CODE 22 --- GET STORED PSEU AND PSP FOR EVEN YEAR
702   IF(ICOD-20) 767,767,768
767   PCESUM=CSUM2(I,1)
      C2=CSUM2(I,2)
      PCSSUM=DSUM2(I,1)
      CS2=DSUM2(I,2)
      RETURN
768   PCESUM=CSUM2(I,3)
      C2=CSUM2(I,4)
      PCSSUM=DSUM2(I,3)
      CS2=DSUM2(I,4)
      RETURN
80    CONTINUE
      END
C-----------------------------------------------------------------------C
C                          SUBROUTINE "FINDCL"                          C
C                                                                       C
C     SUBROUTINE "FINDCL" IS CALLED BY SUBROUTINE "DAVER" TO            C
C     DETERMINE IF A PARTICULAR CLONE NAME ENCOUNTERED IN THE USER'S    C
C     "OBSERVATION" DATA IS A VALID CLONE CONTAINED IN THE STUDY GROUP C
C     DEFINED BY THE ENTRIES ON DISK FILE CLONTAB. IF FOUND THE INDEX   C
C     INTO CLONTAB IS RETURNED ALONG WITH VALUES FOR CONES PER BUSHEL   C
C     AND SEED PER LB. FOR THE PARTICULAR CLONE ID. IF NOT FOUND, AN    C
C     ERROR MSG. IS OUTPUT AND THE RUN TERMINATED.                      C
C-----------------------------------------------------------------------C

      SUBROUTINE FINDCL
      INTEGER CLONE(3)
      COMMON CLONE,LOC,ICONYR,IYR,INDEX,ICONBU,ISEDLB,NORM,I1ST,I2ND
      COMMON INDEX2(255),ICOL2(255),IROW2(255),ASUM2(255,16),
     1CSUM2(255,4),DSUM2(255,4),NFR2(255),IYRGF2(255)
      INTEGER ALOC,ACLON(3)
      REWIND 01
10    READ(01,500,END=25) ALOC,(ACLON(K),K=1,3),IND,ICONBU,ISEDLB
      IF(ALOC.NE.LOC) GO TO 10
      IF(ACLON(1).NE.CLONE(1).OR.ACLON(2).NE.CLONE(2).OR.ACLON(3)
     1.NE.CLONE(3)) GO TO 10
      INDEX=IND
      RETURN
25    WRITE(19,26) CLONE,LOC
26    FORMAT(' CLONE ',3A4,' AT LOC ',I4,' MISSING --- RUN IS TERMINATED
     1. ')
500   FORMAT(A4,3A4,I5,I4,I5)
      STOP
      END
C-----------------------------------------------------------------------C
```

```
C                           SUBROUTINE "FINDPO"                          C
C                                                                        C
C     SUBROUTINE "FINDPO " IS CALLED BY SUBROUTINE "PCETAB" TO FIND      C
C     THE POSITION OF THE CURRENT CLONE BEING PROCESSED WITHIN THE       C
C     PREDICTED CONE EFFICIENCY TABLE. IF FOUND THE INDEX INTO THE       C
C     TABLE IS RETURNED. IF NOT FOUND AN ERROR MSG IS OUTPUT AND THE     C
C     RUN TERMINATED.                                                    C
C-----------------------------------------------------------------------C

      SUBROUTINE FINDPO(IROW,ICOL,J,INDL)
      INTEGER CLONE(3)
      COMMON CLONE,LOC,ICONYR,IYR,INDEX,ICONBU,ISEDLB,NORM,I1ST,I2ND
      COMMON INDEX2(255),ICOL2(255),IROW2(255),ASUM2(255,16),
     1CSUM2(255,4),DSUM2(255,4),NFR2(255),IYRGF2(255)
C-------->TO FIND POSITION; RETURN J; SENT INDEX,IROW,ICOL
      DO 55 J=1,NORM
      IF(INDEX2(J)-INDEX) 55,56,55
56    IF(IROW2(J)-IROW) 55,57,55
57    IF(ICOL2(J)-ICOL) 55,58,55
55    CONTINUE
555   WRITE(19,556)
556   FORMAT(' INDEX ON ROW AND COL. NOT ON FILE 02 --- RUN IS TERMINATE
     1D. ')
      STOP
58    INDL=INDEX2(J)
      RETURN
      END
C$ENTRY
```

AVGACTS

```
C-----------------------------------------------------------------------C
C                            ROUTINE "AVGACTS"                          C
C                                                                       C
C     THE FOLLOWING PROGRAM USES TWO USER DEFINED INPUT FILES TO        C
C     TO CALCULATE A CLONAL AVERAGE FOR ACTUAL SEED POTENTIAL, SEED     C
C     EFFICIENCY, EXTRACTION EFFICIENCY, GERMINATION EFFICIENCY,        C
C     AND CONE EFFICIENCY FOR EACH CLONE IN THE USERS OBSERVATION       C
C     GROUP.                                                            C
C                                                                       C
C     INPUT TO THE PROGRAM:                                             C
C        1. ACTUAL FLOWERS AND CONES DATA (DISK FILE 11)                C
C        2. ACTUAL CONE ANALYSIS DATA (DISK FILE 13)                    C
C                                                                       C
C     OUTPUT:                                                           C
C        A DISK FILE CONTAINING THE CLONAL AVERAGES, ONE RECORD         C
C        PER CLONE. (DISK FILE 18 (CLNAVGS))                            C
C                                                                       C
C-----------------------------------------------------------------------C
      INTEGER CLN1,CLN2,CUR1,CUR2,YR,ROW,COL,CURROW,CURCOL
      INTEGER FERT,EXTRAC,TOTSD,FILL,GERM
      INTEGER ENDCLN,EOF,TOTFL,TOTCO,DONE,PTREES
      REAL ASP,ASE,AEE,AGE,CNT
      REAL TFERT,TEXTS,TTOTSD,TFILL,TGERM
      REAL XACTCE,GTOTCO,GTOTFL
      GTOTFL=0.0
      GTOTCO=0.0
      PTREES=0
      XACTCE=0.0
      DONE=0
C
C-------> AVERAGE THE FLOWERS AND CONES DATA BY CLONE DELETING ZERO
C-------> FLOWER COUNTS.
C-------> COMPUTE THE ACTUAL CONE EFFICIENCY FOR EACH CLONE ID
C-------> ENCOUNTERED IN THE USERS FLOWERS AND CONES DATA SET AND
C-------> WRITE A RECORD CONTAINING YR., CLONE ID, ROW, COL AND
C-------> ACTUAL CONE EFFICIENCY FOR EACH ON TEMPORARY DISK 08.
C
      READ(11,100,END=500) YR,CLN1,CLN2,ROW,COL,TOTFL,TOTCO
100   FORMAT(I2,2A4,1X,I2,1X,I3,1X,I4,1X,I4)
      CUR1=CLN1
      CUR2=CLN2
      IF(TOTFL.EQ.0) GO TO 200
      GTOTFL=GTOTFL+TOTFL
      GTOTCO=GTOTCO+TOTCO
      PTREES=PTREES+1
200   READ(11,100,END=500) YR,CLN1,CLN2,ROW,COL,TOTFL,TOTCO
      IF(CLN1.EQ.CUR1.AND.CLN2.EQ.CUR2) GO TO 400
210   IF(GTOTFL.EQ.0.0) GO TO 225
220   XACTCE=(GTOTCO/GTOTFL)
      IF(XACTCE.GT.1.00) XACTCE=1.00
      GO TO 230
225   XACTCE=2.0
C-------> WRITE A RECORD CONTAINING CURRENT CLONES ACTUAL CONE EFF.
230   WRITE(08,240) YR,CUR1,CUR2,ROW,COL,XACTCE
240   FORMAT(I2,2A4,I2,I3,F6.2)
      IF(DONE.EQ.1) GO TO 777
      PTREES=0
      GTOTFL=0.0
      GTOTCO=0.0
      CUR1=CLN1
      CUR2=CLN2
```

```
400   IF( TOTFL.EQ.0 ) GO TO 200
      GTOTFL=GTOTFL+TOTFL
      GTOTCO=GTOTCO+TOTCO
      PTREES=PTREES+1
      GO TO 200
500   DONE=1
      GO TO 210
C------->
C------->COMPUTE CLONAL AVERAGES FOR ACTUAL: SEED POTENTIAL, SEED
C------->EFFICIENCY, EXTRACTION EFFICIENCY, AND GERMINATION EFFICIENCY.
C------->READ THE PREVIOUSLY CALCULATED VALUE FOR ACTUAL CONE EFFICIENCY
C------->FROM DISK FILE 08(FOR THE CURRENT CLONE ID) AND WRITE A RECORD
C------->TO DISK FILE 18 CONTAINING THE CLONE ID AND ALL CORRESPONDING
C------->CLONAL AVERAGES.
C------->
777   REWIND 08
C------->OUTPUT HEADER FOR CLONAL AVERAGES FILE.
      WRITE(18,887)
      WRITE(18,888)
      WRITE(18,889)
887   FORMAT('              **** CLONAL AVERAGES ****')
888   FORMAT(' CLONE       SP     CE    SE    EE    GE   SONE ')
889   FORMAT(' -----      ----   ----  ----  ----  ----  ---- ')
      EOF=0
      ASP=0.0
      ASE=0.0
      AEE=0.0
      AGE=0.0
C-------> READ AN ACTUAL CONE ANALYSIS RECORD AND INITIALIZE AVERAGING
      READ(13,1000) YR,CLN1,CLN2,ROW,COL,FERT,EXTRAC,TOTSD,FILL,GERM
1000  FORMAT(I2,2A4,1X,I2,1X,I3,5X,5I4)
1100  CNT=1.0
      CUR1=CLN1
      CUR2=CLN2
C-------> INCREMENT WITHIN CLONE TOTALS FOR FERTILE SCALES, EXTRACTED
C-------> SEED, TOTAL SEED, FILLED SEED AND GERMINATED SEED.
      TFERT=FERT
      TEXTS=EXTRAC
      TTOTSD=TOTSD
      TFILL=FILL
      TGERM=GERM
C-------> READ ANOTHER ACTUAL CONE ANALYSIS RECORD AND CONTINUE
C-------> AVERAGING UNTIL CLONE ID CHANGES.
1500  READ(13,1000,END=1750) YR,CLN1,CLN2,ROW,COL,FERT,EXTRAC,TOTSD,FILL
     1,GERM
      IF(CLN1.NE.CUR1.OR.CLN2.NE.CUR2) GO TO 2200
      CNT=CNT+1
      TFERT=TFERT+FERT
      TEXTS=TEXTS+EXTRAC
      TTOTSD=TTOTSD+TOTSD
      TFILL=TFILL+FILL
      TGERM=TGERM+GERM
      GO TO 1500
1750  EOF=1
C-------> IF TOTAL FERTILE SCALES = ZERO OUTPUT ZEROS AND "NO DATA" MSG.
C-------> IF TOTAL FERTILE SCALES > ZERO CALCULATE THE CLONAL AVERAGES
C-------> AND OUTPUT THEM.
2200  IF(TFERT.EQ.0.0) GO TO 4500
C-------> ASP=CLONAL AVERAGE SEED POTENTIAL
C-------> ASE=CLONAL AVERAGE SEED EFFICIENCY
C-------> AEE=CLONAL AVERAGE EXTRACTION EFFICIENCY
```

```
C-------> AGE=CLONAL AVERAGE GERMINATION EFFICIENCY
      ASP=((2*TFERT)/CNT)
      ASE=((TFILL/ASP)/CNT)
      AEE=TEXTS/TTOTSD
      AGE=TGERM/TFILL
      TFERT=0.0
      TEXTS=0.0
      TTOTSD=0.0
      TFILL=0.0
      TGERM=0.0
C-------> GET THE AVERAGED ACTUAL CONE EFFICIENCY FOR CURRENT CLONE
4500  READ(08,4600) ACE
4600  FORMAT(15X,F6.2)
C------> WRITE THE CLONAL AVGS. OF CURRENT CLONE TO DISK
C------> FOR ZERO CONE EFF. WRITE ALL ZEROS AND "NO DATA" MESSAGE
      IF(ACE.EQ.2.0) GO TO 4780
      IF(ACE.EQ.0.0) GO TO 4790
      WRITE(18,4750) CUR1,CUR2,ASP,ASE,AEE,AGE,ACE
4750  FORMAT(2A4,2X,5F6.2)
      GO TO 4800
4780  WRITE(18,4781) CUR1,CUR2
4781  FORMAT(2A4,'    0.0   0.0   0.0   0.0   0.0 **NOTE** NO DATA - ZER
     1O FLOWERS PRODUCED')
      GO TO 4800
4790  WRITE(18,4791) CUR1,CUR2
4791  FORMAT(2A4,'    0.0   0.0   0.0   0.0   0.0 **NOTE** NO CAS DATA -
     1 NO CONES PRODUCED')
4800  IF(EOF.EQ.1) GO TO 5000
C-------->ZERO OUT CLONAL ACCUMULATORS
      ASP=0.0
      ASE=0.0
      AEE=0.0
      AGE=0.0
      TFERT=0.0
      TEXTS=0.0
      TTOTSD=0.0
      TFILL=0.0
      TGERM=0.0
      GO TO 1100
5000  CONTINUE
      END
```

NEWOBV

```
C-----------------------------------------------------------------------C
C                          ROUTINE "NEWOBV"                             C
C                                                                       C
C     THE FOLLOWING PROGRAM ALLOWS THE USER TO CREATE THE BASIS         C
C     FOR A NEW "OBSERVATIONS" DATA SET USING AN EXISTING               C
C     "OBSERVATIONS" DATA SET. THE NEW DATA SET IS MODELED AFTER        C
C     THE CHOSEN "OBSERVATIONS" DATA SET, INCREMENTING CONE YEAR        C
C     BY TWO(2) YEARS AND AVERAGING THE CLONAL AVERAGES COMPUTED BY     C
C     ROUTINE "AVGACTS" WITH THE CHOSEN DATA SETS ESTIMATES,            C
C     RESULTING IN A NEW SET OF ESTIMATES FOR SEED POTENTIAL, SEED      C
C     EFFICIENCY, EXTRACTION EFFICIENCY, GERMINATION EFFICIENCY, AND    C
C     CONE EFFICIENCY. THE "OBSERVATION" ENTRIES ARE LEFT BLANK IN      C
C     IN THE NEW FILE, AND ARE TO BE ENTERED BY THE USER AS THEY        C
C     BECOME AVAILABLE.                                                 C
C                                                                       C
C     INPUT:                                                            C
C        1. DISK FILE 03 --- THE DATA SET THAT THE NEW "OBSERVATIONS"   C
C           DATA SET IS TO BE MODELED AFTER.                            C
C        2. DISK FILE 18 --- THE DATA SET CREATED BY ROUTINE            C
C           "AVGACTS" CONTAINING THE ACTUAL CLONAL AVERAGES OF THE      C
C           OBSERVATION GROUP.                                          C
C                                                                       C
C     OUTPUT:                                                           C
C        DISK FILE 10 --- THE NEW "OBSERVATIONS" DATA SET, MODELED      C
C        AFTER THE CHOSEN "OBSERVATIONS" DATA SET.                      C
C                                                                       C
C-----------------------------------------------------------------------C
      INTEGER DIV,CLONE1,CLONE2,ASE,AEE,AGE,ACE,ASP
      REWIND 03
      REWIND 18
C-------->STRIP HEADER INFO OFF OF CLONE AVERAGES FILE
      DO 75 I=1, 3
      READ(18,50)
50    FORMAT(1X)
75    CONTINUE
C-------->READ A USER "OBSERVATIONS" RECORD (DISK FILE 13)
      READ(03,100,END=800) ICONYR,LOC,CLONE1,CLONE2,IROW,ICOL,IYRGRF,NRA
     1MET,IESTSP,IESTCE,IESTSE,IESTEE,IESTGE
100   FORMAT(I2,A4,2A4,4X,2I3,I2,I3,25X,I3,4I2)
      ICUR1=CLONE1
      ICUR2=CLONE2
C-------> READ A CLONAL AVERAGES RECORD (DISK FILE 18)
      READ(18,200) JCUR1,JCUR2,ASP,ASE,AEE,AGE,ACE
200   FORMAT(2A4,2X,I3,7X,I2,4X,I2,4X,I2,4X,I2)
C-------> AVERAGE CURRENT CLONES ACTUALS WITH OLD ESTIMATES
250   NCONYR=ICONYR+2
      IF(ASP.EQ.0) DIV=1
      IF(ASP.GT.0) DIV=2
      NESTSP=(IESTSP+ASP)/DIV
      NESTCE=(IESTCE+ACE)/DIV
      NESTSE=(IESTSE+ASE)/DIV
      NESTEE=(IESTEE+AEE)/DIV
      NESTGE=(IESTGE+AGE)/DIV
C-------> WRITE A NEW "OBSERVATIONS" DATA SET RECORD
      WRITE(10,100)NCONYR,LOC,CLONE1,CLONE2,IROW,ICOL,IYRGRF,NRAMET,NEST
     1SP,NESTCE,NESTSE,NESTEE,NESTGE
C-------> READ ANOTHER OLD "OBSERVATIONS" RECORD
400   READ(03,100,END=800)ICONYR,LOC,CLONE1,CLONE2,IROW,ICOL,IYRGRF,NRAM
     1ET,IESTSP,IESTCE,IESTSE,IESTEE,IESTGE
C-------> CONTINUE WRITING NEW "OBSERVATIONS" RECORDS UNTIL
C-------> NEW CLONE ID OCCURS.
```

```
      IF(CLONE1.NE.ICUR1.OR.CLONE2.NE.ICUR2) GO TO 500
      WRITE(10,100)NCONYR,LOC,CLONE1,CLONE2,IROW,ICOL,IYRGRF,NRAMET,NEST
     1SP,NESTCE,NESTSE,NESTEE,NESTGE
      GO TO 400
500   ICUR1=CLONE1
      ICUR2=CLONE2
C-------> READ ANOTHER CLONAL AVERAGES RECORD AND CONTINUE
C-------> PROCESSING.
      READ(18,200) JCUR1,JCUR2,ASP,ASE,AEE,AGE,ACE
      GO TO 250
800   CONTINUE
      END
```