A COMPUTER PROGRAM FOR CAPTURE-RECAPTURE STUDIES OF ANIMAL POPULATIONS:

A Fortran listing for the stochastic model of G. M. Jolly

E. G. White

TUSSOCK GRASSLANDS AND MOUNTAIN LANDS INSTITUTE

Special Publication No. 8 — 1971

A COMPUTER PROGRAM FOR CAPTURE-RECAPTURE STUDIES OF ANIMAL POPULATIONS:

A Fortran listing for the stochastic model of G. M. Jolly

E. G. White

Special Publication No. 8 — 1971

Tussock Grasslands and Mountain Lands Institute,
P.O. Box 56,
Lincoln College,
Christchurch,
New Zealand.

TABLE OF CONTENTS

PROGRAM ABSTRACT

White (1971) briefly describes the versatility of a Fortran computer program which analyses capture-recapture data from animal populations, using the stochastic model of G. M. Jolly (1965). The listing and a fuller description are now given.

The program is applicable to studies in which marked animals are individually recognisable, although White (1971) suggests two ways in which to adapt data from studies in which animals are not individually recognisable. The program is in five sections which store all the capture-recapture records and then organise any given selections of these data into trellis tables and the tables of derived population estimates. Data can be used from studies with or without intermediate recapture samples (described by Jolly (1965)), and any selection of consecutive data can be analysed separately for any sub-population. To permit such a selection, the migration of marked animals (as well as unmarked) between sub-populations is accounted for, both in space and time, but in such applications there should be no repeated to-and-fro migration since a basic assumption of Jolly's model is that all emigration is permanent. Certain restrictions can be placed upon the analysis of data and animals can be added to or withdrawn from the population during a study.

The program is written in Fortran IV and was compiled and tested on an 8K IBM 1130 computer using the IBM 1130 Monitor Fortran.

USER INFORMATION

A. Detailed Program Description

1. Composition - there are two linked programs:

NDATA reads the control card, then reads and stores all data, NOGHS performs all analyses and prints results, operating through three subroutines.

Subroutine NSUB1 selects the appropriate data for a given analysis.

Subroutine NSUB2 constructs and prints the trellis tables from the selected data.

Subroutine JOLLY tests the value of the selected data and calculates and prints the population estimates.

2. Restrictions - see Program Abstract and Section D, part 6.

3. Precision - Standard precision

Real variables and constants - two words

Integers - one word.

4. Program Requirements - Disk Monitor Programming system

1130-OS-005 version 2

5. System Configuration -

IBM 1131 model 2B CPU 8K (Disk)
IBM 1442 model 6 card read punch
IBM 1132 line printer

6. Timings -

Compile and store sample program	.10 hours
Store data for sample problem	.18 hours
Execute sample problem	.02 hours

B. Input Description

All input numbers are integer.

As many as four classes of attributes (e.g. capture site, species, age and sex) can be assigned to individual animals to permit separate analyses of different sub-populations, both singly and in combination. First, each different attribute within each class is ascribed a number. Thus if a population study area is sub-divided into say three adjoining sites, the numbers 1 to 3 can be used to distinguish the data associated with the three sites. Second, these data are selected for analysis in as many combinations as desired by assigning to each selection its own combination number (see example below) and adapting the program as described in Section D, part 3. The selections for each class are hereon referred to as combinations of sites or plots (KOMPL); of species (KOMSP); of age or stage of development (KOMST); and of sex (KOMSX). Hence, two of the three sub-areas above might be analysed singly by assigning the KOMPL values 1 and 2 respectively, then combined for KOMPL = 3, and all 3 sub-areas included when KOMPL = 4. Note that one of the three sub-areas in this example is not analysed singly, i.e. Within a class, any given attribute may be analysed as a member of any given combinations of attributes, whether it is analysed singly or not.

1. Data card format

Columns 1-2 A number belonging to one of the four classes of attributes and representing a single unit, such as a given plot. The number can vary from 1 to 98 and must be right justified.

Column 3 A number belonging to another of the four classes of attributes and representing a single unit, such as a given species. The number may vary from 1 to 9.

Column 4 A number belonging to another of the four classes of attributes and representing a single unit, such as a given stage of development. The number may vary from 1 to 9.

If any of the three classes in the above columns are not used, each must still be represented by a number and it is simplest to assign the number 1. If all three classes are each assigned the number 1, or if the first and third classes are each assigned only the numbers 1 and 2, the listed program will perform all possible selections of analyses without needing modification as described in Section D, part 3.

Column 5 A number belonging to another of the four classes of attributes and representing a single unit such as one of the sexes. The number may vary from 1 to 2. If no use is made of this class of attributes, a number is still necessary and the number 1 must be assigned.

Columns 6-8 The number representing the sample. The number can vary from 1 to 499 but 500 is added to the number of an intermediate recapture sample to distinguish it from a sample including newly released animals. Thus, if newly marked animals are released in the 5th and 7th samples but the 6th sample consists entirely of recaptured animals, the sequence of numbers on the respective cards would be 005, 506, 007. The number must be right justified.

The numbers in columns 1 to 8 classify all animals listed to the right of column 8.

Columns 9-10 Not used.

Columns 11-15 The number representing a given animal. The animal may be either a newly marked one or a recapture record. The digits in the first two columns represent the group of animals to which the animal belonged when it was first marked. Thus, if additional animals are newly marked in every sample, this group number is the same as the sample number when the animal was first marked. However, if say the 2nd, 4th and 6th samples consist entirely of recaptured animals, the group number of the 7th sample is 04 since only the 1st, 3rd, 5th and 7th samples include newly marked animals. The first two digits are right justified. The following digits in the right three columns represent the number of the animal within the group. The number can vary from 1 to 999 and is right justified. Thus animal number 20 in the 4th group of marks is listed as 04020. If an animal is permanently removed from the population before being marked and released, the first two digits are zeroes and the right three comprise a number that can vary from 1 to 300 and be repeated for different animals. This number is also right justified. The five digit number may therefore vary from 00001 to 00300 for permanently removed animals that are not marked and released, and from 01001 to 31999 for animals which are either recaptured or newly marked and released, irrespective of any permanent removals of such animals subsequently. The limit of 31999 is set by the 16-bit word length in the IBM 1130 but the limit could be expanded to 6 digits (allowing 4 digits for 9999 newly marked animals per marking time) in a computer with an 18-bit word length. (See Section D, part 1).

Columns 16-17 Not used.

Columns 18-22 The number representing another animal, different to that listed in columns 11 to 15, or referring to the permanent removal from the population of the recaptured animal listed in columns 11 to 15. Where the number refers to a different animal (including the permanent removal of an unmarked animal), the five digits are of the same composition as in columns 11 to 15. Where the recaptured animal listed in columns 11 to 15 is permanently removed, the two digits in columns 18 and 19 are zeroes and the right three digits comprise a number that can vary from 301 to 999 and be repeated for different animals. This number is right justified. The five digit number may therefore vary from 00001 to 00999 for permanently removed animals and from 01001 to 31999 for animals which are recaptured or newly marked and released, irrespective of any permanent removals of such animals subsequently. Columns 18 to 22 are not used if the animal listed in columns 11 to 15 is the last item of data conforming to the classification in columns 1 to 8.

Columns 23-24 Not used.

Columns 25-29 The number representing another animal, different to those listed in columns 11 to 15 and 18 to 22, or referring to the permanent removal from the population of the recaptured animal listed in columns 18 to 22. The composition of the five digits follows the same rules as listed for columns 18 to 22. Columns 25 to 29 are not used if an animal represented in columns 11 to 15 or 18 to 22 is the last item of the data conforming to the classification in columns 1 to 8.

Columns 30-80 Not used, but available for an extended format in the listing of animals (see Section D, part 5).

2. Last card format

Columns 1-2 The number 99

Columns 3-80 Not used

3. Control card format

Column 1 Not used

Columns 2-4 The number of the most recent sample to be included in current selections of data for analysis. e.g. If 20 samples have been obtained but a separate analysis of the first 10 samples is wanted, this number is 010. The number may vary from 1 to 499, i.e. 500 is not added to the number of an intermediate recapture sample. The sample can be either a marking-time sample or an intermediate recapture sample. All sample numbers must be right justified.

Column 5 Not used.

Columns 6-8 The number of the earliest sample to be included in current selections of data for analysis. This sample must be a marking-time sample. Hence, if there are intermediate recapture samples, none of these can be selected as the first sample in a given analysis. Sample numbers are as specified for columns 2 to 4.

Column 9 Not used.

Columns 10-12 The number representing the value (or least value in a series) of KOMPL to be included in current selections of data for analysis. All KOMPL numbers must be right justified.

Column 13 Not used.

Columns 14-16 The number representing the value (or least value in a series) of KOMSP to be included in the analysis of the first current value of KOMPL selected in columns 10 to 12. All KOMSP numbers must be right justified.

Column 17 Not used.

Columns 18-20 The number representing the value (or least value in a series) of KOMST to be included in the analysis of the first current values of KOMPL and KOMSP selected respectively in columns 10 to 12 and 14 to 16. All KOMST numbers must be right justified.

Columns 21-80 Not used.

C. <u>Output Description</u>

1. Combination line format

Specifies the selection of data which is analysed in the subsequent two trellis tables and the table of derived estimates by listing the values of KOMPL, KOMSP, KOMST and KOMSX to which the analysis applies. The significance of each value depends on the selection of data associated with it in the subroutine NSUB1. (See Section D, part 3.)

2. First trellis table format

The format is unchanged from that of Jolly (1965) except that the immigration of marked animals is also accounted for. Such immigration may occur in space or time, as described by White (1971).

The first column in the table lists the number of immigrants in each sample, the second the number of animals in each sample (Jolly's n_i) and the third the number of marked animals released from each sample (Jolly's s_i).

NRI - the number of marked animals that are released from sample i and caught subsequently. (Jolly's R_i)

3. Second trellis table format

The format is unchanged from that of Jolly (1965).

NZI - the number of animals marked before time i which are not caught in the ith sample but which are caught subsequently (Jolly's Z_i).

4. Estimates table format

The table is unchanged from that of Jolly (1965) except for a regrouping of the columns and the addition of Jolly's estimate of the probability of capture at time i for a reason discussed by White (1971). Asterisks are printed to mark estimates which cannot be calculated when there are intermediate recapture samples that do not include unmarked animals.

TIME	Sample number (Jolly's i)
PPTN RECAP	Proportion of recaptured animals in each sample (Jolly's α_i)
PROB CAP	Estimate of the probability of capture for each sample (Jolly's $\hat{p}_i$)

NO MARKED	Estimate of the number of marked animals released into the population and still alive at the time of each sample (Jolly's $\hat{M}_i$).
TOTAL POP	Estimate of population size at the time of each sample (Jolly's $\hat{N}_i$)
SE POP	Standard error of each estimate of population size (Jolly's $(V(\hat{N}_i))^{\frac{1}{2}}$)
SE ESTPOP	Standard error of estimation for each estimate of population size (Jolly's $(V(\hat{N}_i \mid N_i))^{\frac{1}{2}}$)
NO BORN	Estimate of the number of new animals joining the population in the interval between one sample of newly marked animals and the next and still alive when that next sample is taken (Jolly's $\hat{B}_i$)
SE BIRTHS	Standard error of each estimate of the number born (Jolly's $(V(\hat{B}_i))^{\frac{1}{2}}$)
SURV RATE	Estimated probability of survival that an animal alive at the moment of releasing one sample will survive until the time of capturing the next sample (Jolly's $\hat{\phi}_i$)
SE SURV	Standard error of each estimate of the probability of survival (Jolly's $(V(\hat{\phi}_i))^{\frac{1}{2}}$)
SE ESTSURV	Standard error of estimation for each estimate of the probability of survival (Jolly's $\left(V(\hat{\phi}_i) - \frac{\hat{\phi}_i^2(1-\hat{\phi}_i)}{\hat{M}_{i+1}}\right)^{\frac{1}{2}}$)

D. Modifying the Program

1. Increasing the maximum sample size

The maximum sample size in the listing is 999 newly marked animals plus any number of recaptured animals for each set of attributes that is always analysed separately. i.e. A given series of numbers may be ascribed to more than one group of newly marked animals in the same sample provided the recapture histories in the trellis tables are always computed separately for each group. Hence, since KOMSX is automatically analysed as two separate attributes (KOMSX = 1, KOMSX = 2) which are cumulatively stored and then analysed as a summation (KOMSX = 3), it is possible to extend the maximum sample size in the summation to 1998 newly marked animals. Even if no real dichotomy such as sex exists in the data, the animals can be arbitrarily placed in two groups to maximise sample size, and attention paid only to the analyses of KOMSX = 3, ignoring the first two arbitrary sets of analyses as meaningless. For further enlargement, it is necessary to use a computer capable of handling 6-digit numbers (an 18-bit word length or longer) to accommodate a 2-digit sample number plus 9999 newly marked animals per sample.

2. Adapting array sizes to the data

Several array sizes are inter-related and the following relationships must be kept when adapting array sizes:

JPOP and N — The data in columns 6 to 29 (N) of those data cards which conform to the same classification in columns 1 to 8 occupy a unit area (JPOP) in core. The listed maximum size of a JPOP unit is set by the storage capacity of a disk sector, but in any modifications the size must always remain a multiple of the size of N.
(N.B. When changing the size of JPOP, the change applies also to statement 19.)

NCOMB and N — The area of core used for given selections of data for analysis (NCOMB) sets the limit to the amount of data which can be analysed in any one selection of data. While maximising the size of NCOMB may often be desirable, its size must always remain a multiple of the size of N. For a given core capacity, NCOMB may be maximised by minimising the size of those arrays which limit the number of samples in the analyses, i.e. all arrays with a size of 24 words in the program listing.

(N.B. When changing the size of NCOMB, the change applies not only to Common but to statement numbers 4 and 130 + 002 in NSUB1, and 34008 + 001 in NOGHS.)

NHOLD and NIJ — The size of the trellis table array NHOLD in which the sums of the first two sets of KOMSX values are accumulated must always exceed by 4 the size of the trellis table array NIJ in which the first two sets of KOMSX values are in turn individually accumulated.

IQQQ and Common — The size of the dummy array (IQQQ) in the program NDATA must equal the total size of the various integer arrays listed in Common, beginning NHOLD and ending NAIJX.

RQQ and Common — The size of the dummy array (RQQ) in the program NDATA and the subroutines NSUB1 and NSUB2 must equal the total size of the various real arrays listed in Common, beginning ESTM and ending SENO.

LS and ESTNO — The maximum number of intermediate recapture sequences (LS) is attained if no two samples of newly marked animals constitute successive samples. (One intermediate recapture sequence may of course be comprised of any number of recapture samples.) Therefore, if the program is to retain generality, the size of LS should always remain one half of the maximum size ascribed to the output format of the various identically sized arrays such as the estimates of population numbers (ESTNO).

(N.B. When changing the size of LS, the change applies also to statement 1221 + 003.)

3. Programming to select given data

Selections of data according to the classes of attributes assigned to animals are programmed at the end of the program NOGHS and in the subroutine NSUB1. It has already been stated (p.10) that KOMSX increments automatically to the maximum value of 3, and this occurs at earlier points in NOGHS. Limits to the other three classes of attributes are set by the expressions in the last three IF statements in NOGHS. The order of incrementing should be noted. The selection of combinations of data within these limits is performed by NSUB1 according to the statements defining each set of combination numbers ascribed to the four classes of attributes in the manner shown on p.3. As an example, the program listing shows the selection of nine combinations (3 plots x 1 species x 3 stages) for each value of KOMSX in the series of statements 1031 + 003 and 35 to 38. This listing must be modified according to requirements and clearly the combination numbering in the sample problem output on p.15 results from a different listing for it exceeds the limits of the listed selection (KOMPL 12 > 3; KOMSP 4 > 1). Note that statements 120 on are excluded from modification in order to retain the level of program versatility, and also that the third plot and third stage combinations are simply cumulative combinations of the first two plots and stages respectively. Further note that by the use of statement 34 in NSUB1 the program can exit after a current selection of combinations is complete, even though a final IF statement in NOGHS may reserve a higher limit to include other possible selections in NSUB1 (compare the limit of KOMPL = 5 in NOGHS with the selected values of KOMPL = 1, 2 and 3 in NSUB1).

4. Setting restrictions on analyses

White (1971) refers to restrictions that can be placed upon the analysis of data for the table of estimates, pointing out that in a series of data selections, some selections may be less worthy of analysis than others. In the subroutine JOLLY, there are tests which set the chosen minima that are acceptable for the R_i and m_i values of Jolly (1965). The minima in the listing are each given as 7.0 (see statements 25222 and 6222). However, analyses of data are not rejected on the grounds of occasional low values but on the relative number of values below the minima. The final acceptance or rejection of data is determined by the expression at statement 225 where the value 1.04 is listed. Lower values are less selective than higher ones but the selectivity of a given value depends on the number of samples being analysed and the nature of the recapture program. A value of less than 1.00 is generally non-selective and must be used if there are no more than four samples in the analysis. The listed value 1.04 is too low for most data (it accepts data when only three or four samples out of 20 completely satisfy the minima tests above) but it has been used with data in which there were a relatively large number of intermediate recapture samples, each of which has the same effect as an unacceptable R_i or m_i value. If a value is chosen to screen out population estimates calculated from poorer selections of data, it should be remembered that the trellis tables will still be printed for such selections.

5. Optimising the format of input data

In the program listing, only three animals can be entered on a data card whereas all the space in columns 9 to 80 is available. The optimal choice of format depends on the total amount of data to be computed relative to its distribution between the combinations of selected attributes that are to be analysed. When there is more space utilised per card, there is proportionately less utilised by the classification groupings in columns 1 to 8, but there will also be more waste space in core when some cards have columns without entries. Therefore, if all data are to be computed collectively without selected combinations of attributes, the optimal format is a maximal number of animal entries per card. If, however, data tend to be minimal in any of the selected classification groupings given in columns 1 to 8, the optimal format may be fewer animal entries per card. The decision is best made according to the weakest selections of data, remembering that if the size of NCOMB tends to be limiting, optimisation of the format is important.

6. Setting the initial population size

In estimating the variance of population size it is necessary to have an estimate of population size at the time of releasing the first sample. Since an estimate cannot be calculated, Jolly (1965) suggests that if initially the population numbers appear from subsequent estimates to be reasonably stable, this value could be put equal to the estimate immediately following. Otherwise, in the presence of a trend in subsequent estimates, it would be better if the first value be guessed to comply with the trend. In the program listing, the initial estimate for a given selection of data is always put equal to the estimate from the next sample which includes newly marked animals. A modification to the program would therefore be necessary to substitute an estimate complying with a trend.

E. Sample Problem Output

A sample problem output is given on p.15 for a combination analysis of 2 species of acridid grasshoppers, *Paprides nitidus* Hutton and *Sigaus australis* (Hutton), which are common alpine species in the Craigieburn Range in Canterbury, New Zealand. The combination line values are those ascribed to the particular selection of data as follows:

KOMPL 12	A combined selection of adjacent plots on a south aspect (the example given by White (1971) was for otherwise comparable data from an adjoining north aspect).
KOMSP 4	A combined selection of the two species named above.
KOMST 1	A single selection of adults only.
KOMSX 3	The combined selection of both sexes.

The samples in the analysis were obtained during the 1969-70 season (earlier samples were obtained in the 1968-69 season), beginning with the first spring sample (sample 35) and ending with the final autumn sample (sample 53). The adults overwinter and so account for the immediate classification of marked "immigrants" in the first and some subsequent samples of the analysis. i.e. Immigration occurs in time. The numerous

KOMPL 12 KOMSP 4 KOMST 1 KOMSX 3

10	65	65	35																		
8	10	10	2	36																	
6	8	8	1	1	37																
3	5	5	2	0	0	38															
16	156	156	1	0	1	0	39														
10	37	37	1	1	0	0	25	40													
13	157	157	2	0	0	0	17	6	41												
6	43	43	0	0	0	0	9	4	24	42											
3	106	106	3	1	0	0	7	0	9	8	43										
3	42	42	0	0	0	0	4	1	10	0	24	44									
6	107	107	0	0	0	0	2	0	4	1	8	9	45								
4	39	39	1	0	0	0	1	0	2	0	3	5	23	46							
3	78	78	0	0	0	0	1	0	3	1	3	2	4	8	47						
2	31	31	0	0	0	0	0	0	1	1	2	2	7	3	13	48					
2	76	76	0	0	0	1	0	2	3	0	1	1	2	1	4	9	49				
1	26	26	0	0	0	0	1	1	2	1	0	0	3	0	1	0	16	50			
0	33	33	0	0	0	0	0	0	1	0	1	0	0	0	1	0	2	1	51		
0	7	7	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	4	2	52	
0	4	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	53
	NRI		13	3	1	1	67	14	59	12	42	19	39	12	20	9	18	5	3	1	

	35																		
	2	36																	
	1	2	37																
	2	2	2	38															
	1	1	2	2	39														
	1	2	2	2	27	40													
	2	2	2	2	19	25	41												
	0	0	0	0	9	13	37	42											
	3	4	4	4	11	11	20	28	43										
	0	0	0	0	4	5	15	15	39	44									
	0	0	0	0	2	2	6	7	15	24	45								
	1	1	1	1	2	2	4	4	7	12	35	46							
	0	0	0	0	1	1	4	5	8	10	14	22	47						
	0	0	0	0	0	0	1	2	4	6	13	16	29	48					
	0	0	0	1	1	3	6	6	7	8	10	11	15	24	49				
	0	0	0	0	1	2	4	5	5	5	8	8	9	9	25	50			
	0	0	0	0	0	0	1	1	2	2	2	2	3	3	5	6	51		
	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	5	7	52	
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	53
NZI	11	12	11	10	50	39	61	45	48	43	47	37	28	13	6	5	1	0	

TIME	PPTN RECAP	PROB CAP	NO MARKED	TOTAL POP	SE POP	SE ESTPOP	NO BORN	SE BIRTHS	SURV RATE	SE SURV	SE ESTSURV
35									0.5948	0.3133	0.3074
36	******	******	38.66	********	********	********	********	********	2.1000	2.1841	2.1954
37	******	******	98.00	********	********	********	********	********	0.5480	0.6993	0.6976
38	******	******	57.00	********	********	********	********	********	0.4213	0.3796	0.3742
39	0.0128	0.0791	25.28	1972.11	1408.74	1408.58	-524.08	952.43	0.8876	0.1797	0.1781
40	******	******	159.14	********	********	********	********	********	0.7613	0.1766	0.1735
41	0.1592	0.1941	128.77	808.73	172.24	170.34	106.97	111.73	0.9800	0.2337	0.2335
42	******	******	255.58	********	********	********	********	********	0.5412	0.1445	0.1412
43	0.2641	0.1977	141.57	535.94	107.59	105.79	293.44	124.40	0.6608	0.1168	0.1124
44	******	******	145.10	********	********	********	********	********	0.9586	0.1973	0.1967
45	0.2242	0.1690	141.97	632.96	140.12	139.12	131.48	130.64	0.8345	0.1998	0.1983
46	******	******	187.75	********	********	********	********	********	0.8672	0.2600	0.2588
47	0.2820	0.1322	166.30	589.60	158.45	157.59	43.86	53.57	0.5643	0.1685	0.1652
48	******	******	125.44	********	********	********	********	********	0.6190	0.2084	0.2040
49	0.3157	0.3042	78.88	249.81	66.78	65.51	221.11	183.10	0.4293	0.1482	0.1417
50	******	******	56.20	********	********	********	********	********	1.0664	0.7397	0.7405
51	0.1818	0.0983	61.00	335.50	241.46	241.25	********	********	0.1590	0.1309	0.1250
52	******	******	14.00	********	********	********					
53	0.5000										

intermediate recapture samples are reflected by the various smaller tallies in the trellis tables and are further indicated by the asterisked samples in the table of estimates.

OPERATING INSTRUCTIONS

A. Compiling and Storing the Program

Each section of the program is compiled and stored using the following card order:

1. //JOB
2. //FOR
3. *IOCS(1132 PRINTER,CARD,DISK)   (Exclude from subroutines)
4. *ONE WORD INTEGERS
5. *LIST ALL
6. (Program or subroutine cards)
7. //DUP
8. *STORE   WS   UA   NDATA   (or other name)

B. Storing Data and Executing Analyses

1. Creating the file and executing analyses.

   Put console switch 5 on. The card order is as follows:

   1. //JOB
   2. //XEQ NDATA   1
   3. *LOCALNOGHS,NSUB2,JOLLY,NSUB1
   4. (Control card)
   5. (Data cards)
   6. (Last card)
   7. //DUP
   8. *STOREDATA   WS   UA   NFILE 720

   Note that when the available computing time is insufficient to complete a series of analyses, it may be desirable to program only the final analysis when creating the file in order that the execution be completed and the data stored. Other analyses can be subsequently carried out following the operating instructions under C. below.

2. Updating the file and executing analyses

   Put console switch 5 on if stored data is to be deleted.

   The card order is as follows:

   1. //JOB
   2. //XEQ NDATA 2
   3. *LOCALNOGHS,NSUB2,JOLLY,NSUB1
   4. *FILES(100,NFILE)
   5. (Control card)
   6. (Data cards)
   7. (Last card)

   The note under Section B, part 1, also applies to updating the file.

C. Using Stored Data and Executing Analyses

Console switch 5 is off. The card order is identical to that given above in Section B, part 2, except that no data cards are needed.

SYSTEM MATERIAL

A program listing is given on pp 18-24 and flow diagrams are given on pp 25-32. In the latter, the following terminology has been used to describe the various arrays, files, and tables associated with the analyses of KOMSX values:

FEMALE	-	KOMSX	=	1
MALE	-	KOMSX	=	2
PRIME	-	KOMSX	=	1 or 2
JOINT SEX	-	KOMSX	=	3

NDATA

```
      DIMENSION IA(4),JA(4),N(4)
      COMMON JPOP(312),NA(4),IQQQ(342),NEXP,NX,RQQ(288),IMIG(24)
      COMMON NOW,NBGIN,NBNOW,KOMPL,KOMSP,KOMST,LS(12)
      DEFINE FILE 100(720,317,U,NEXP)
    1 FORMAT (I2,3I1,I3,3(2X,I5))
    2 FORMAT (5I4)
    3 FORMAT (' DISK FULL',I6,' RECORDS')
C-----MAKE EACH DISK RECORD IDENTIFICATION ZERO (SWITCH 5 ON) FOR FIRST
C-----DATA CARDS ONLY
      CALL DATSW(5,NSW1)
      GO TO (9,12),NSW1
    9 DO 10 I=1,4
   10 NA(I)=0
      DO 11 I=1,720
   11 WRITE (100'I) NA
   12 CONTINUE
      READ (2,2) NOW,NBGIN,KOMPL,KOMSP,KOMST
C-----IF PRIOR RECORDS LEFT ON DISK WHEN NBGIN IS INCREMENTED, EFFECTIVE
C-----CAPACITY OF NCOMB IS REDUCED
      NBNOW=NOW-NBGIN
      DO 1012 J=1,4
 1012 JA(J)=10000
   13 READ (2,1) IA,N
      IF(IA(1)-99) 16,28,28
   16 DO 1016 J=1,4
      IF (IA(J)-JA(J)) 17,1016,17
 1016 CONTINUE
      GO TO 19
   17 NEXT=1
 1017 READ (100'NEXT) NA,KOUNT,JPOP
      DO 2018 J=1,4
      IF (NA(J)-IA(J)) 18,2018,18
   18 IF (NA(1)-1) 23,1018,1018
 1018 NEXT=NEXT+1
      GO TO 1017
 2018 CONTINUE
   19 IF (KOUNT-312) 21,20,20
   20 NEXT=NEXT+1
      GO TO 1017
   21 DO 22 K=1,4
      LOT=KOUNT+K
   22 JPOP(LOT)=N(K)
      KOUNT=KOUNT+4
      WRITE (100'NEXT) NA,KOUNT,JPOP
      GO TO 1026
C-----THE EXPRESSION IN STATEMENT 23 MUST BE CHANGED IF THE FILE SIZE IS
C-----CHANGED
   23 IF (NEXT-720) 1024,24,24
   24 WRITE (3,3) NEXT
      CALL EXIT
 1024 DO 25 K=1,4
   25 JPOP(K)=N(K)
      KOUNT=4
      DO 26 J=1,4
   26 NA(J)=IA(J)
      WRITE (100'NEXT) NA,KOUNT,JPOP
 1026 DO 27 J=1,4
   27 JA(J)=NA(J)
      GO TO 13
   28 CONTINUE
      CALL LINK (NOGHS)
      END
```

NOGHS

```
      COMMON JPOP(312),NA(4),NHOLD(28),NIJ(24),NRI(24),NRIX(24),NZI(24),
     1NZIX(24),NAIJ(24),NMIY(24),NNIY(24),NSIY(24),NREP,NP,NNIX(24),
     2NSIX(24),NMIX(24),NAIJX(24),NEXP,NX,ESTM(24),RECAP(24)
     3,ESTNO(24),ESURV(24),EBORN(24),ESTP(24),VEST(24),SEEST(24),SESUR
     4(24),SEESS(24),SEBOR(24),SENO(24),IMIG(24)
      COMMON NOW,NBGIN,NBNOW,KOMPL,KOMSP,KOMST,LS(12),NCOMB(1584)
C-----NOTE WELL...THE SIZE OF NCOMB MUST BE A MULTIPLE OF THE ARRAY N
C-----IN PROGRAM NDATA
      DEFINE FILE 100(720,317,U,NEXP),200(24,30,U,NREP),300(24,25,U,NP),
     1400(24,25,U,NX)
 1004 FORMAT (//////,' KOMPL',I3,3X,'KOMSP',I2,3X,'KOMST',I2,3X,'KOMSX',
     1I2)
    8 FORMAT (/,I4,2I4,1X,24I4)
 4008 FORMAT (/,7X,'NRI',3X,24I4)
14008 FORMAT (/,13X,24I4)
24008 FORMAT (/,7X,'NZI',3X,24I4)
34008 FORMAT (//////,I17,/)
      NUMBR=1584
      KALC=0
      KOMSX=1
      NREC=24
C-----START SELECTING DATA IN CURRENT CLASS COMBINATION
   29 CALL NSUB1(KOMSX,KOM4,NREC,NUMBR)
      GO TO (276,33,271,288),KOM4
   33 WRITE (3,1004) KOMPL,KOMSP,KOMST,KOMSX
C-----START ANALYSIS
      CALL NSUB2 (NREC,NUMBR,KOMSX,NTWO)
C-----START JOLLY FORULAE
      NAG=0
      CALL JOLLY (NTWO,NAG)
      KALC=KALC+1
  271 IF (KOMSX-2) 272,273,273
  272 KOMSX=2
      GO TO 29
  273 KOMSX=3
      IF (KALC-2) 276,274,276
  274 DO 1274 J=1,24
 1274 IMIG(J)=0
      WRITE (3,1004) KOMPL,KOMSP,KOMST,KOMSX
      NB1=NBNOW+1
      DO 275 NREC=1,NB1
      NFIVE=NREC+3
      READ (200'NREC) NHOLD
      WRITE (3,8) (NHOLD(N),N=1,NFIVE)
  275 IMIG(NREC)=NHOLD(1)
      WRITE (3,4008) (NRIX(J),J=1,NBNOW)
      NAGGX=1
      WRITE (3,34008) NBGIN
      DO 1275 NTIME=1,NBNOW
      NTONE=NTIME+1
      READ (400'NAGGX) NAIJX
      WRITE (3,14008) (NAIJX(J),J=1,NTONE)
 1275 NAGGX=NAGGX+1
      WRITE (3,24008) (NZIX(J),J=1,NBNOW)
      NAG=1
      CALL JOLLY (NTWO,NAG)
  276 KALC=0
      KOMSX=1
      IF (KOMST-3) 278,280,280
  278 KOMST=KOMST+1
      GO TO 29
  280 KOMST=1
      IF (KOMSP-1) 282,284,284
  282 KOMSP=KOMSP+1
      GO TO 29
  284 KOMSP=1
      IF (KOMPL-5) 286,288,288
  286 KOMPL=KOMPL+1
      GO TO 29
  288 CALL EXIT
      END
```

NSUB1

```
      SUBROUTINE NSUB1(KOMSX,KOM4,NREC,NUMBR)
      COMMON JPOP(312),NA(4),NHOLD(28),NIJ(24),NRI(24),NRIX(24),NZI(24),
     1NZIX(24),NAIJ(24),NMIY(24),NNIY(24),NSIY(24),NREP,NP,NNIX(24),
     2NSIX(24),NMIX(24),NAIJX(24),NEXP,NX,RQQ(288),IMIG(24)
      COMMON NOW,NBGIN,NBNOW,KOMPL,KOMSP,KOMST,LS(12),NCOMB(1584)
    4 FORMAT (/////,' NCOMB FULL       EXCEEDS 1584    KOMPL',I3,'  KOMS
     1P',I3,'  KOMST',I3,'  KOMSX',I3)
      IF (KOMSX-1) 10029,10029,30
10029 DO 1029 J=1,28
 1029 NHOLD(J)=0
      DO 2029 J=1,NREC
      NNIX(J)=0
      NSIX(J)=0
      NMIX(J)=0
      NZIX(J)=0
 2029 NRIX(J)=0
   30 DO 1030 I=1,NUMBR
 1030 NCOMB(I)=0
      NUMBR=0
      NEXT=0
      KOM4=1
   31 NEXT=NEXT+1
      READ (100'NEXT) NA,KOUNT
      NPLOT=NA(1)
      IF (NPLOT-1) 137,1031,1031
 1031 NSPEC=NA(2)
      NSTAG=NA(3)
      NSEX=NA(4)
      GO TO (35,135,235,34),KOMPL
   34 KOM4=4
      GO TO 137
   35 IF (NPLOT-2) 36,31,31
  135 IF (NPLOT-2) 31,36,31
  235 IF (NPLOT-2) 36,36,31
   36 GO TO (37,38,120),KOMST
   37 IF (NSTAG-2) 120,31,31
   38 IF (NSTAG-2) 31,120,31
  120 GO TO (121,123),KOMSX
  121 IF (NSEX-2) 130,31,31
  123 IF (NSEX-2) 31,130,31
  130 KOM4=2
      NUMBR=NUMBR+KOUNT
      IF (NUMBR-1584) 133,133,131
  131 WRITE (3,4) KOMPL,KOMSP,KOMST,KOMSX
      KOM4=3
      NUMBR=NUMBR-KOUNT
      GO TO 137
  133 NK=NUMBR-KOUNT
      READ(100'NEXT) NA,KOUNT,JPOP
      DO 134 NTRY=1,KOUNT
      NT=NK+NTRY
  134 NCOMB(NT)=JPOP(NTRY)
      GO TO 31
  137 RETURN
      END
```

NSUB2

```
      SUBROUTINE NSUB2 (NREC,NUMBR,KOMSX,NTWO)
      COMMON JPOP(312),NA(4),NHOLD(28),NIJ(24),NRI(24),NRIX(24),NZI(24),
     1NZIX(24),NAIJ(24),NMIY(24),NNIY(24),NSIY(24),NREP,NP,NNIX(24),
     2NSIX(24),NMIX(24),NAIJX(24),NEXP,NX,ROQ(288),IMIG(24)
      COMMON NOW,NBGIN,NBNOW,KOMPL,KOMSP,KOMST,LS(12),NCOMB(1584)
    5 FORMAT (' ANIMAL NO.',I6,1X,'RECORDED MORE THAN ONCE, KK=',I3,5X,'
     1OR IF KK=0, ERROR IN PROGRAM')
    6 FORMAT (' ERROR IN DISK SEARCH OPERATION, ANIMAL NO.',I6,1X,'NOT
     1RELOCATED, KK=',I3)
 1008 FORMAT (/,I4,2I4,1X,24I4)
 4008 FORMAT (/,7X,'NRI',3X,24I4)
14008 FORMAT (/,13X,24I4)
24008 FORMAT (/,7X,'NZI',3X,24I4)
34008 FORMAT (//////,I17,/)
      NREC=NOW-NBGIN+1
      DO 137 J=1,NREC
      IMIG(J)=0
      NRI(J)=0
      NZI(J)=0
      NNIY(J)=0
      NSIY(J)=0
      NMIY(J)=0
  137 NAIJ(J)=0
      NREL=0
      NREC=1
      NAGGR=1
      NAGGX=1
      DO 220 NTIME=NBGIN,NOW
      NBT=NTIME-NBGIN+1
      NEWM=0
      IMMIG=0
      NMI=0
      MW=0
      NW=0
      MREC=0
      NCHEK=1
      NT=1
      DO 138 J=1,NREC
  138 NAIJX(J)=0
      DO 140 J=1,NBNOW
  140 NIJ(J)=0
  141 NREAL=NCOMB(NT)
      IF (NREAL-500) 1142,1142,142
  142 NREAL=NREAL-500
 1142 IF (NREAL-NTIME) 189,143,189
  143 DO 188 I=1,3
      NTI=NT+I
      N2=NCOMB(NTI)
      IF (N2-1) 189,144,144
  144 IF (N2-1000) 183,183,145
  145 KK=0
      JTIME=NBGIN-1
      LTIME=1
  146 IF (NCOMB(LTIME)-500) 148,148,147
  147 LREAL=NCOMB(LTIME)-500
      GO TO 149
  148 LREAL=NCOMB(LTIME)
  149 IF (LREAL-NBGIN) 171,150,150
C-----THE FOREGOING TEST ASSUMES NBGIN IS ALWAYS A MARKING TIME.
C-----WHEN NBGIN IS MADE GREATER THAN 1, THIS TEST PROCEEDS
C-----TO OVERLOOK  EARLIER RECORDS AND LISTS AS AN IMMIGRATION THE FIRST
C-----RECAPTURE OF ANY ANIMAL THAT WAS MARKED PRIOR TO NBGIN.
  150 IF (LREAL-NREAL) 154,154,171
  154 DO 168 L=1,3
      LTL=LTIME+L
      IF (NCOMB(LTL)-N2) 168,156,168
  156 IF (NCOMB(LTIME)-500) 158,158,163
  158 JREL=N2/1000
      IF (NREL-JREL) 161,163,1161
C-----THE TEST (NREL-JREL) DETERMINES FROM THE CURRENT (NTIME) RECORDS
C-----THEMSELVES WHETHER OR NOT THE MARKING TIME (JREL) OF A GIVEN
C-----ANIMAL WAS PRIOR TO THE MOST RECENT MARKING TIME (NREL) OF ANIMALS
C-----SO FAR TESTED.              IF ANY MARKED IMMIGRANTS AT NTIME ARE
C-----LISTED BEFORE NREL HAS BEEN UPDATED TO THE MOST RECENT MARKING
C-----TIME, THEY ARE TEMPORARILY CLASSIFIED AS NEWLY MARKED (NEWM) UNTIL
C-----THE UPDATING OF NREL CORRECTLY RECLASSIFIES THEM (SEE 161 + 001).
C-----THE TEST (NREL-JREL) FAILS IN THE SAME UNLIKELY AND UNIMPORTANT
C-----CIRCUMSTANCES LISTED SECOND FOR NCHEK BELOW.
  161 NREL=JREL
      IMMIG=NEWM
      NEWM=0
      GO TO 163
 1161 NCHEK=NCHEK+1
  163 IF (LREAL-NTIME) 169,165,171
  165 KK=KK+1
      NTEST=NCOMB(NT)
      IF (KK-1) 167,168,167
  167 WRITE (3,5) N2,KK
C-----THE ABOVE ERROR MESSAGE IS NOT PRINTED WHEN KOMSX=3
  168 CONTINUE
      GO TO 171
  169 IF (LREAL-JTIME) 171,171,170
  170 JTIME=LREAL
  171 IF (LTIME+3-NUMBR) 172,174,174
  172 LTIME=LTIME+4
      GO TO 146
```

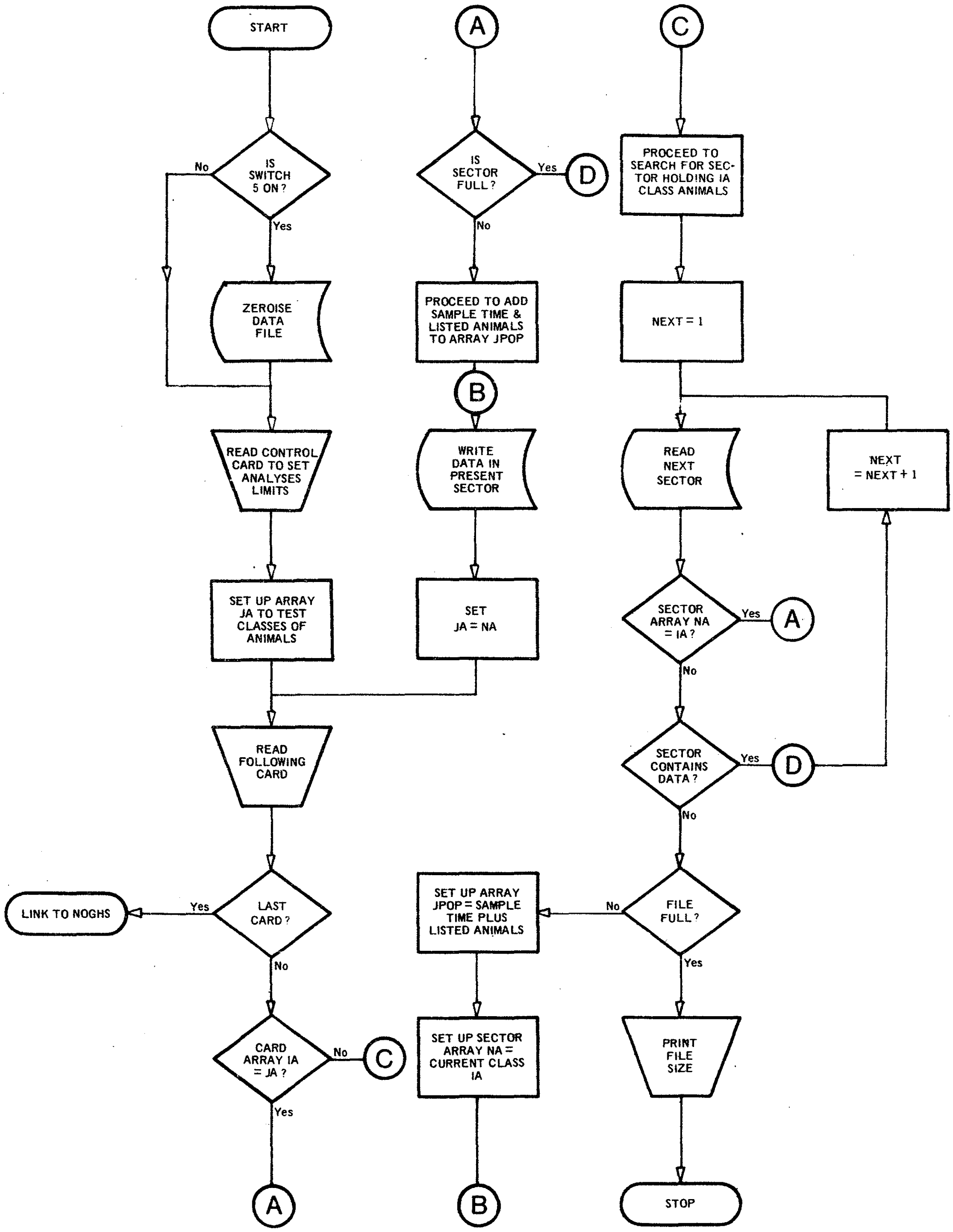
```
  174 IF (KK-1) 175,176,188
  175 WRITE (3,6) N2,KK
  176 IF (JTIME-NBGIN+1) 188,178,177
  177 J=JTIME-NBGIN+1
      NIJ(J)=NIJ(J)+1
      NMI=NMI+1
      GO TO 188
  178 IF (NTEST-500) 179,179,180
  179 IF (NCHEK-2) 1180,180,180
C-----NCHEK FOR IMMIGRANTS FAILS IF OTHER ERRORS ARE PRINTED.
C-----NCHEK ALSO FAILS FOR THE IMMIGRANTS WITH THE MOST RECENT MARKING
C-----TIME IF NO NEWLY MARKED ANIMALS ARE PRESENT AT NTIME AND MORE
C-----RECENT MARKINGS ARE ABSENT IN PREVIOUS VALUES OF NTIME.  THE LIKE-
C-----LIHOOD AND IMPORTANCE OF SUCH AN ERROR IS SLIGHT.
  180 IMMIG=IMMIG+1
      GO TO 188
 1180 NEWM=NEWM+1
      GO TO 188
  183 IF (N2-300) 185,185,186
  185 NW=NW+1
      GO TO 188
  186 MW=MW+1
  188 NCHEK=1
  189 NT=NT+4
      IF (NT-NUMBR) 141,191,191
  191 NNI=NEWM+NMI+IMMIG+NW
      NSI=NNI-MW-NW
      NNIY(NBT)=NNI
      NSIY(NBT)=NSI
      NNIX(NBT)=NNIX(NBT)+NNI
      NSIX(NBT)=NSIX(NBT)+NSI
      IF (NTIME-NBGIN-1) 1193,2193,2193
 1193 WRITE (3,1008)IMMIG,NNI,NSI,NTIME
      GO TO 201
 2193 NMIY(NBT)=NMI
      NMIX(NBT)=NMIX(NBT)+NMI
      NT1=NTIME-NBGIN
      WRITE (3,1008) IMMIG,NNI,NSI,(NIJ(J),J=1,NT1),NTIME
  201 IMIG(NBT)=IMMIG
      IF (KOMSX-2) 205,204,204
  204 READ (200'NREC) NHOLD
  205 NHOLD(1)=NHOLD(1)+IMMIG
      NHOLD(2)=NHOLD(2)+NNI
      NHOLD(3)=NHOLD(3)+NSI
      IF (NTIME-NBGIN-1) 208,1205,1205
 1205 NPLUS=NTIME-NBGIN+3
      DO 206 K=4,NPLUS
  206 NHOLD(K)=NHOLD(K)+NIJ(K-3)
      NHOLD(NPLUS+1)=NTIME
      GO TO 209
  208 NHOLD(4)=NTIME
  209 WRITE (200'NREC) NHOLD
      DO 210 J=1,28
  210 NHOLD(J)=0
      IF (NTIME-NBGIN-1) 220,1210,1210
 1210 IF (KOMSX-1) 212,212,211
  211 READ (400'NAGGX) NAIJX
  212 NAIJ(1)=NIJ(1)
      NAIJX(1)=NAIJX(1)+NIJ(1)
      IF (NTIME-NBGIN-2) 1214,213,213
  213 NTWO=NTIME-1-NBGIN
      DO 214 I=1,NTWO
      NAIJ(I+1)=NAIJ(I)+NIJ(I+1)
      NAIJX(I+1)=NAIJX(I+1)+NAIJ(I+1)
      NZI(I)=NZI(I)+NAIJ(I)
      NZIX(I)=NZIX(I)+NAIJ(I)
  214 CONTINUE
 1214 WRITE (300'NAGGR) (NAIJ(J),J=1,NT1),NTIME
      WRITE (400'NAGGX) (NAIJX(J),J=1,NT1),NTIME
      NAGGR=NAGGR+1
      NAGGX=NAGGX+1
      DO 215 I=1,NBNOW
  215 NRI(I)=NRI(I)+NIJ(I)
  220 NREC=NREC+1
      DO 216 I=1,NBNOW
  216 NRIX(I)=NRIX(I)+NRI(I)
      WRITE (3,4008) (NRI(J),J=1,NBNOW)
      NAGGR=1
      NAGGX=1
      WRITE (3,34008) NBGIN
      DO 1220 NTIME=1,NBNOW
      NTONE=NTIME+1
      READ (300'NAGGR) NAIJ
      WRITE (3,14008) (NAIJ(J),J=1,NTONE)
 1220 NAGGR=NAGGR+1
      WRITE (3,24008) (NZI(J),J=1,NBNOW)
      RETURN
      END
```

JOLLY

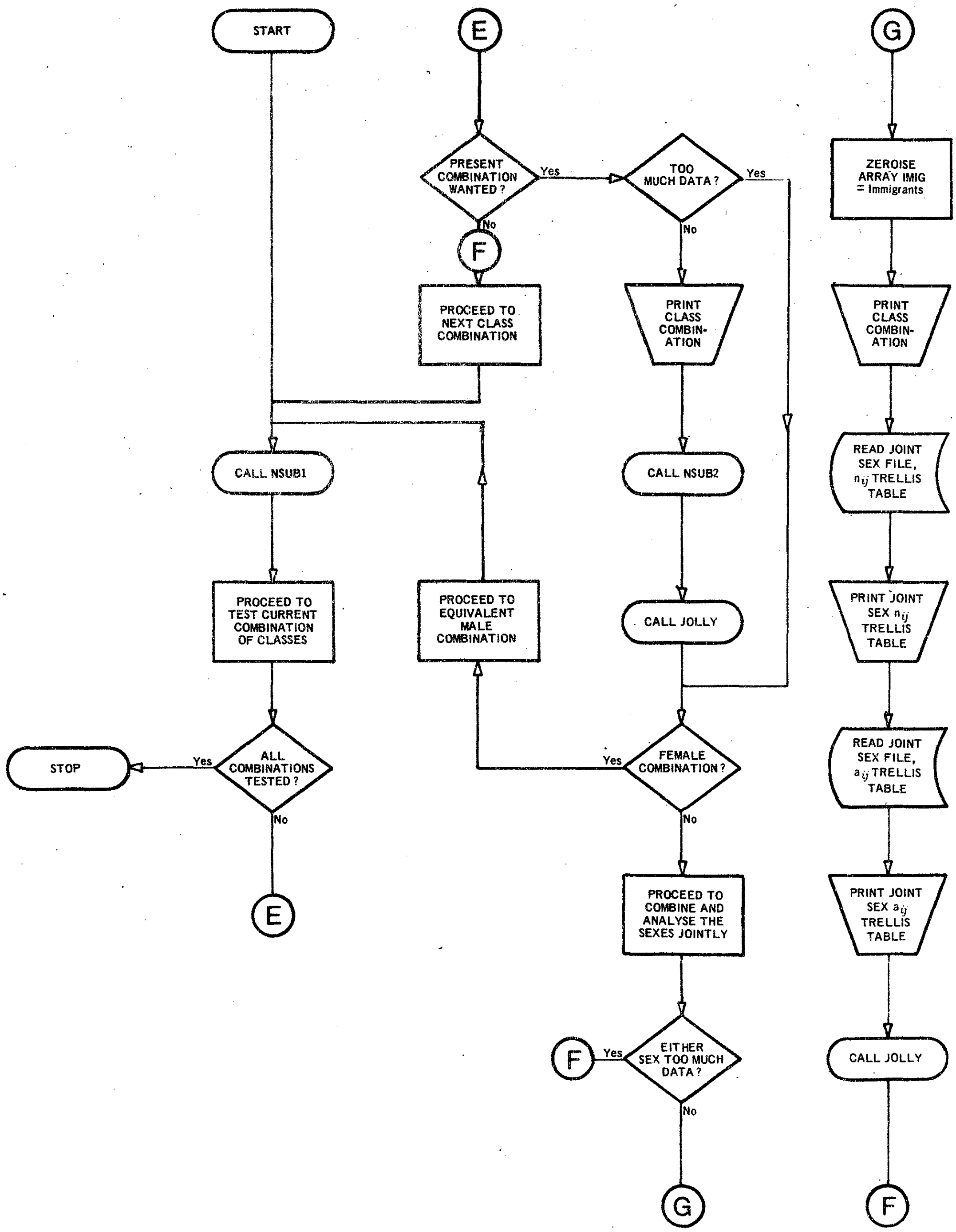
```
      SUBROUTINE JOLLY (NTWO,NAG)
      COMMON JPOP(312),NA(4),NHOLD(28),NIJ(24),NRI(24),NRIX(24),NZI(24),
     1NZIX(24),NAIJ(24),NMIY(24),NNIY(24),NSIY(24),NREP,NP,NNIX(24),
     2NSIX(24),NMIX(24),NAIJX(24),NEXP,NX,ESTM(24),RECAP(24)
     3,ESTNO(24),ESURV(24),EBORN(24),ESTP(24),VEST(24),SEEST(24),SESUR
     4(24),SEESS(24),SEBOR(24),SENO(24),IMIG(24)
      COMMON NOW,NBGIN,NBNOW,KOMPL,KOMSP,KOMST,LS(12)
 5008 FORMAT (////,' TIME  PPTN RECAP  PROB CAP  NO MARKED  TOTAL POP  S
     1E POP  SE ESTPOP   NO BORN  SE BIRTHS  SURV RATE  SE SURV  SE ESTS
     2URV')
 6008 FORMAT (I4,2(5X,F6.4),2X,F8.2,2X,5(2X,F8.2),1X,3(4X,F6.4))
 7008 FORMAT (I4,85X,3(4X,F6.4))
 8008 FORMAT (I4,2(5X,F6.4),2X,F8.2,2X,3(2X,F8.2),/,I4,5X,F6.4,///)
      NB1=NBNOW+1
      IF (NAG-1) 1221,2220,2220
 2220 DO 221 JTIME=1,NB1
      NNIY(JTIME)=NNIX(JTIME)
      NSIY(JTIME)=NSIX(JTIME)
      NMIY(JTIME)=NMIX(JTIME)
      NRI(JTIME)=NRIX(JTIME)
  221 NZI(JTIME)=NZIX(JTIME)
 1221 RNO=0.
      K=0
      L=0
      DO 2221 M=1,12
 2221 LS(M)=1
      L111=1
      ESTM(1)=0.
      DO 226 JTIME=2,NB1
      RNNI=NNIY(JTIME)
      RNSI=NSIY(JTIME)
      RNMI=NMIY(JTIME)
      RIMIG=IMIG(JTIME)
      IF (JTIME-NB1) 12221,22221,226
12221 RNRI=NRI(JTIME)
      RNZI=NZI(JTIME-1)
      ESTM(JTIME)=RNSI*RNZI/RNRI+RNMI
C-----FOLLOWING TEST FOR MARKING TIMES ONLY LIKELY TO FAIL WHEN SAMPLE
C-----SIZE VERY SMALL AND TABLES NOT PRINTED
22221 IF(RNSI-RNMI-RIMIG) 223,223,2222
 2222 RECAP(JTIME)=RNMI/RNNI
      IF (L-1) 3222,4222,4222
 3222 L=JTIME
 4222 IF (L111-1) 223,15222,5222
 5222 K=K+1
      LS(K)=L111
      L111=1
15222 IF (JTIME-NB1) 25222,226,226
25222 IF (RNRI-7.0) 1223,1223,6222
 6222 IF (RNMI-7.0) 1223,1223,224
  223 RECAP(JTIME)=111111.
      ESTP(JTIME)=111111.
      ESTNO(JTIME)=111111.
      SENO(JTIME)=111111.
      SEEST(JTIME)=111111.
      ESURV(JTIME)=111111.
      SESUR(JTIME)=111111.
      SEESS(JTIME)=111111.
      L111=L111+1
      RNO=RNO+1.0
      GO TO 225
 1223 RNO=RNO+1.0
  224 ESTNO(JTIME)=ESTM(JTIME)/RECAP(JTIME)
      ESTP (JTIME)=RNNI/ESTNO(JTIME)
      VEST(JTIME)=ESTNO(JTIME)*(ESTNO(JTIME)-RNNI)*(((ESTM(JTIME)-RNMI+
     1RNSI)/ESTM(JTIME))*((1./RNRI)-(1./RNSI))+((1.-RECAP(JTIME))/RNMI))
      SEEST(JTIME)=SQRT(VEST(JTIME))
  225 IF (((NBNOW-2)/RNO)-1.04) 260,1225,1225
C-----FINAL VALUE IN THE FOREGOING TEST DETERMINES WHETHER SUFFICIENT
C-----SAMPLING TIMES HAVE ADEQUATE DATA (SEE STATEMENTS 25222 AND 6222)
C-----TO PRINT TABLE.
 1225 EBORN(JTIME)=111111.
      SEBOR(JTIME)=111111.
  226 CONTINUE
C-----IF POPULATION NOS RATHER UNSTABLE, ESTIMATE N1 FROM TREND, NOT =N2
      ESTNO(1)=ESTNO(L)
      ESURV(1)=ESTM(2)/NSIY(1)
      DO 228 JTIME=1,NTWO
      JTADD=JTIME+1
      RNSI=NSIY(JTIME)
      RNMI=NMIY(JTIME)
      RNRI=NRI(JTIME)
      TNSI=NSIY(JTADD)
      TNMI=NMIY(JTADD)
      TNRI=NRI(JTADD)
      VITEM=(((ESTM(JTADD)-TNMI)*(ESTM(JTADD)-TNMI+TNSI))/ESTM(JTADD)**2
     1)*((1./TNRI)-(1./TNSI))
      VFORM=((ESTM(JTIME)-RNMI)/(ESTM(JTIME)-RNMI+RNSI))*((1./RNRI)-(1./
     1RNSI))
      IF(JTIME-2)1227,227,227
  227 RNZI=NZI(JTIME-1)
      ESURV(JTIME)=ESTM(JTADD)/((RNSI*RNZI/RNRI)+RNSI)
 1227 SESUR(JTIME)=SQRT(ESURV(JTIME)**2*(VITEM+VFORM+((1.-ESURV(JTIME))/
     1ESTM(JTADD))))
      SEESS(JTIME)=SQRT(SESUR(JTIME)**2-((ESURV(JTIME)**2*(1.-ESURV(JTIM
     1E)))/ESTM(JTADD)))
  228 CONTINUE
```

```
      K=0
      DO 230 JTIME=2,NTWO
      IF (ESTNO(JTIME)-111111.) 229,5229,5229
  229 IF (ESTNO(JTIME+1)-111111.) 1229,2229,2229
 1229 L111=1
      GO TO 4229
 2229 K=K+1
      L111=LS(K)
      IF (L111-1) 230,230,4229
 4229 JTADD=JTIME+L111
      RNNI=NNIY(JTIME)
      TNNI=NNIY(JTADD)
      TNMI=NMIY(JTADD)
      RNSI=NSIY(JTIME)
      TNSI=NSIY(JTADD)
      RNMI=NMIY(JTIME)
      RNRI=NRI(JTIME)
      TNRI=NRI(JTADD)
      TSURV=1
      J2=JTADD-1
      DO 34229 ITIME=JTIME,J2
34229 TSURV=TSURV*ESURV(ITIME)
      EBORN(JTIME)=ESTNO(JTADD)-TSURV*(ESTNO(JTIME)-RNNI+RNSI)
      IF (NBNOW-JTADD) 230,44229,44229
44229 XTRA=(ESTNO(JTADD)*(ESTNO(JTADD)-TNNI)*((1.-RECAP(JTADD))/TNMI))+
     1TSURV**2*ESTNO(JTIME)*(ESTNO(JTIME)-RNNI)*((1.-RECAP(JTIME))/RNMI)
      VITEM=(((ESTM(JTADD)-TNMI)*(ESTM(JTADD)-TNMI+TNSI))/ESTM(JTADD)**2
     1)*((1./TNRI)-(1./TNSI))
      VFORM=((ESTM(JTIME)-RNMI)/(ESTM(JTIME)-RNMI+RNSI))*((1./RNRI)-(1./
     1RNSI))
      SEBOR(JTIME)=SQRT(EBORN(JTIME)**2*VITEM+(VFORM*(((TSURV*RNSI*(1.-R
     1ECAP(JTIME)))/RECAP(JTIME))**2))+(((ESTNO(JTIME)-RNNI)*(ESTNO(JTAD
     2D)-EBORN(JTIME))*(1.-RECAP(JTIME))*(1.-TSURV))/(ESTM(JTIME)-RNMI+R
     3NSI))+XTRA)
      GO TO 230
 5229 IF (JTIME-2) 230,6229,230
 6229 K=1
  230 CONTINUE
      EBORN(1)=ESTNO(L)-ESURV(1)*(ESTNO(1)-NNIY(1)+NSIY(1))
      DO 238 JTIME=L,NBNOW
      TOTAL=ESTNO(L)*((ESTNO(L)-EBORN(1))/ESTNO(1))
      IF (ESTNO(JTIME)-111111.) 231,238,238
  231 IF (L-2) 232,232,233
  232 M=0
      GO TO 1233
  233 M=1
 1233 SUM=0
      J1=JTIME-1
      DO 237 I=1,J1
      K=M
      IF (ESTNO(I)-111111.) 1231,237,237
 1231 TERMA=EBORN(I)
      IF (JTIME-I) 236,236,2231
 2231 I1=I+1
      MM=0
      DO 6235 J=I1,J1
      IF (ESTNO(J)-111111.) 234,6235,6235
  234 MM=MM+1
      IF (ESTNO(J+1)-111111.) 2235,1235,1235
 1235 K=K+1
      L111=LS(K)
      IF (MM-1) 3235,11235,3235
11235 M=K
      GO TO 3235
 2235 L111=1
 3235 JL111=J+L111
      IF (SUM) 5235,4235,5235
 4235 TOTAL=TOTAL*((ESTNO(JL111)-EBORN(J))/ESTNO(J))
 5235 TERMA=TERMA*((ESTNO(JL111)-EBORN(J))/ESTNO(J))
 6235 CONTINUE
  236 SUM=SUM+TERMA*TERMA/EBORN(I)
  237 CONTINUE
      SENO(JTIME)=SQRT((VEST(JTIME))+ESTNO(JTIME)-SUM-TOTAL*TOTAL/ESTNO(
     11))
  238 CONTINUE
      WRITE (3,5008)
      WRITE (3,7008) NBGIN,ESURV(1),SESUR(1),SEESS(1)
      DO 253 JTIME=2,NTWO
      NTIME=JTIME+NBGIN-1
  253 WRITE (3,6008) NTIME,RECAP(JTIME),ESTP(JTIME),ESTM(JTIME),ESTNO(JT
     1IME),SENO(JTIME),SEEST(JTIME),EBORN(JTIME),SEBOR(JTIME),ESURV
     2(JTIME),SESUR(JTIME),SEESS(JTIME)
      JTNOW=NOW-1
      WRITE (3,8008) JTNOW,RECAP(NBNOW),ESTP(JTIME),ESTM(NBNOW),ESTNO(NB
     1NOW),SENO(NBNOW),SEEST(NBNOW),NOW,RECAP(NB1)
  260 RETURN
      END
```

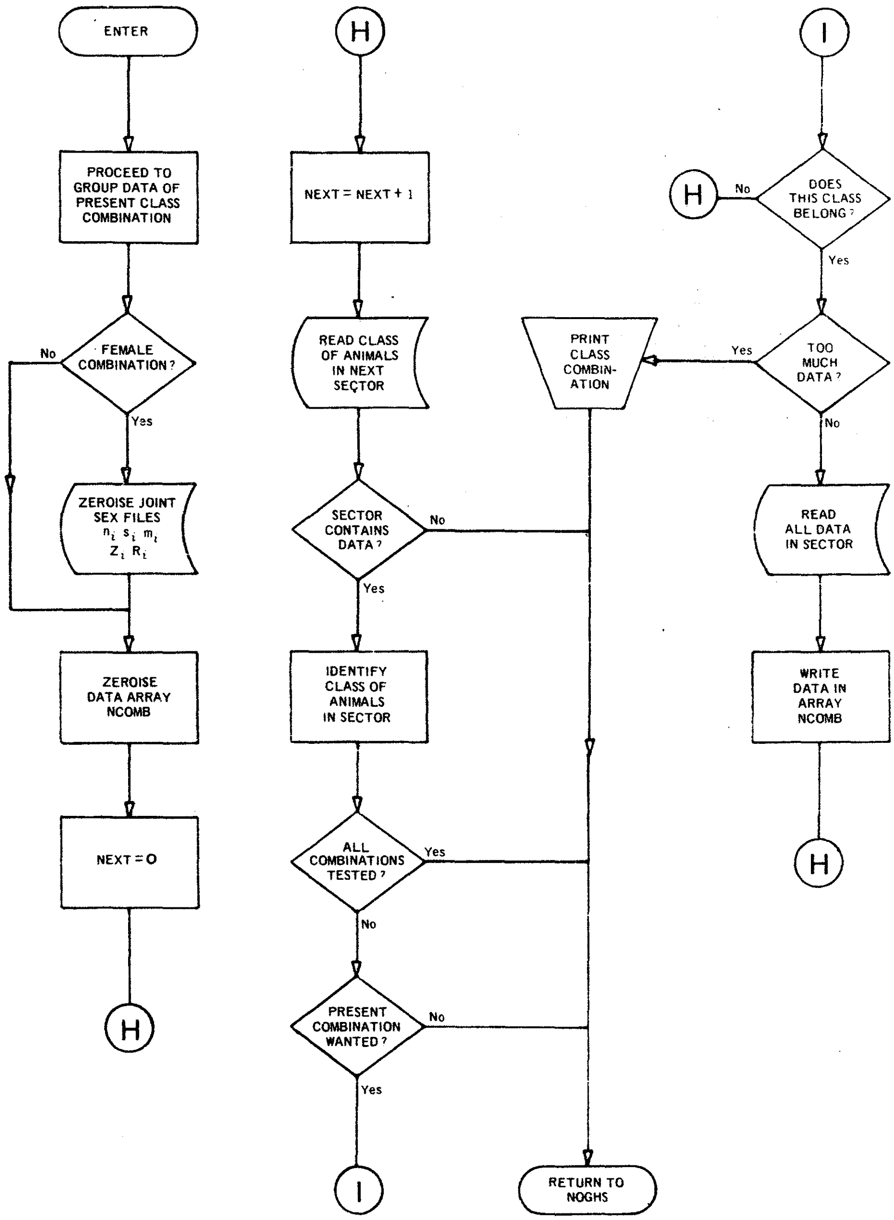
NDATA


START
IS SWITCH 5 ON?
No
Yes
ZEROISE DATA FILE
READ CONTROL CARD TO SET ANALYSES LIMITS
SET UP ARRAY JA TO TEST CLASSES OF ANIMALS
READ FOLLOWING CARD
LAST CARD?
Yes
LINK TO NOGHS
No
CARD ARRAY IA = JA ?
No
C
Yes
A
A
IS SECTOR FULL?
Yes
D
No
PROCEED TO ADD SAMPLE TIME & LISTED ANIMALS TO ARRAY JPOP
B
WRITE DATA IN PRESENT SECTOR
SET JA = NA
C
PROCEED TO SEARCH FOR SECTOR HOLDING IA CLASS ANIMALS
NEXT = 1
READ NEXT SECTOR
NEXT = NEXT + 1
SECTOR ARRAY NA = IA ?
Yes
A
No
SECTOR CONTAINS DATA?
Yes
D
No
FILE FULL?
No
SET UP ARRAY JPOP = SAMPLE TIME PLUS LISTED ANIMALS
SET UP SECTOR ARRAY NA = CURRENT CLASS IA
B
Yes
PRINT FILE SIZE
STOP

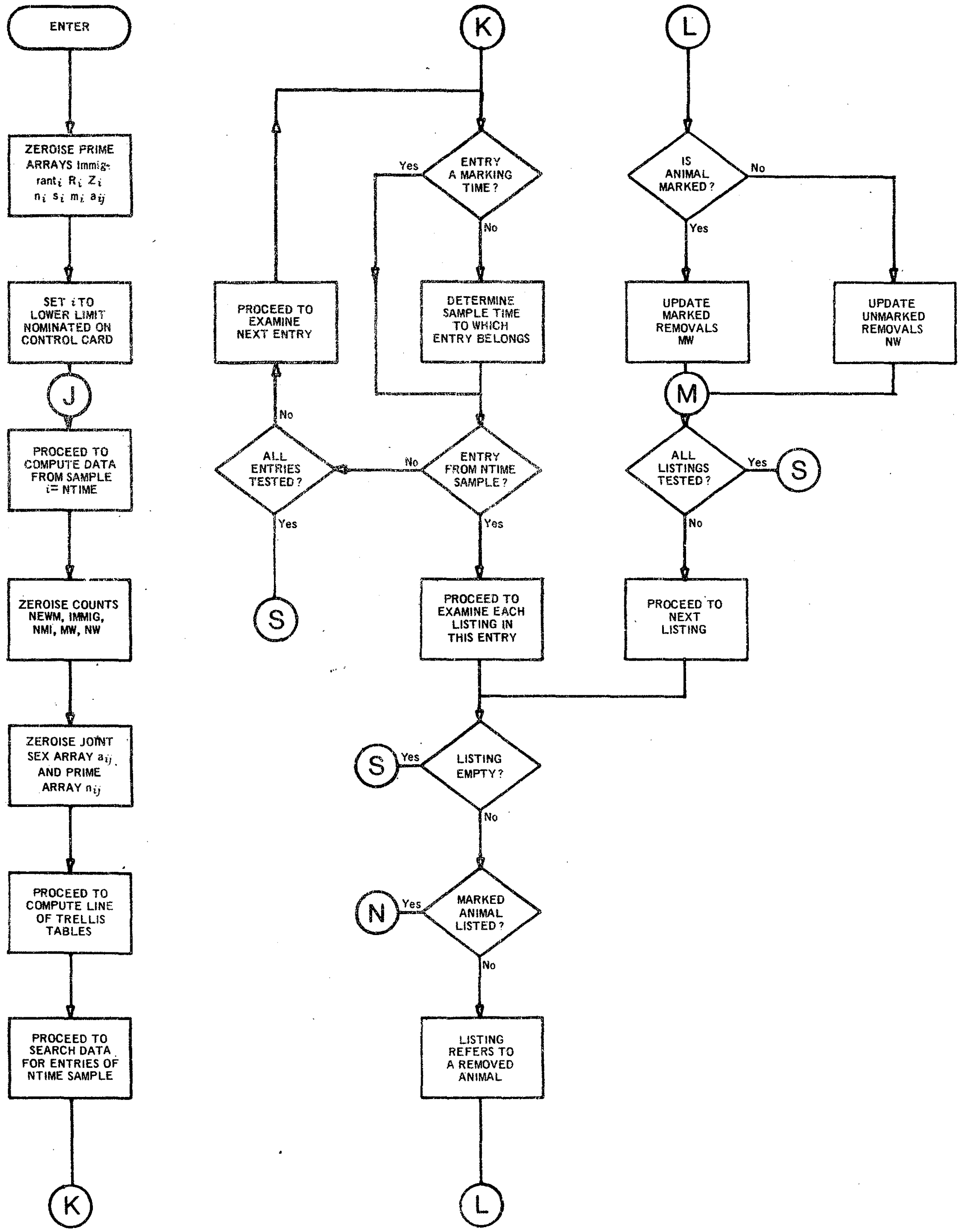
NOGHS

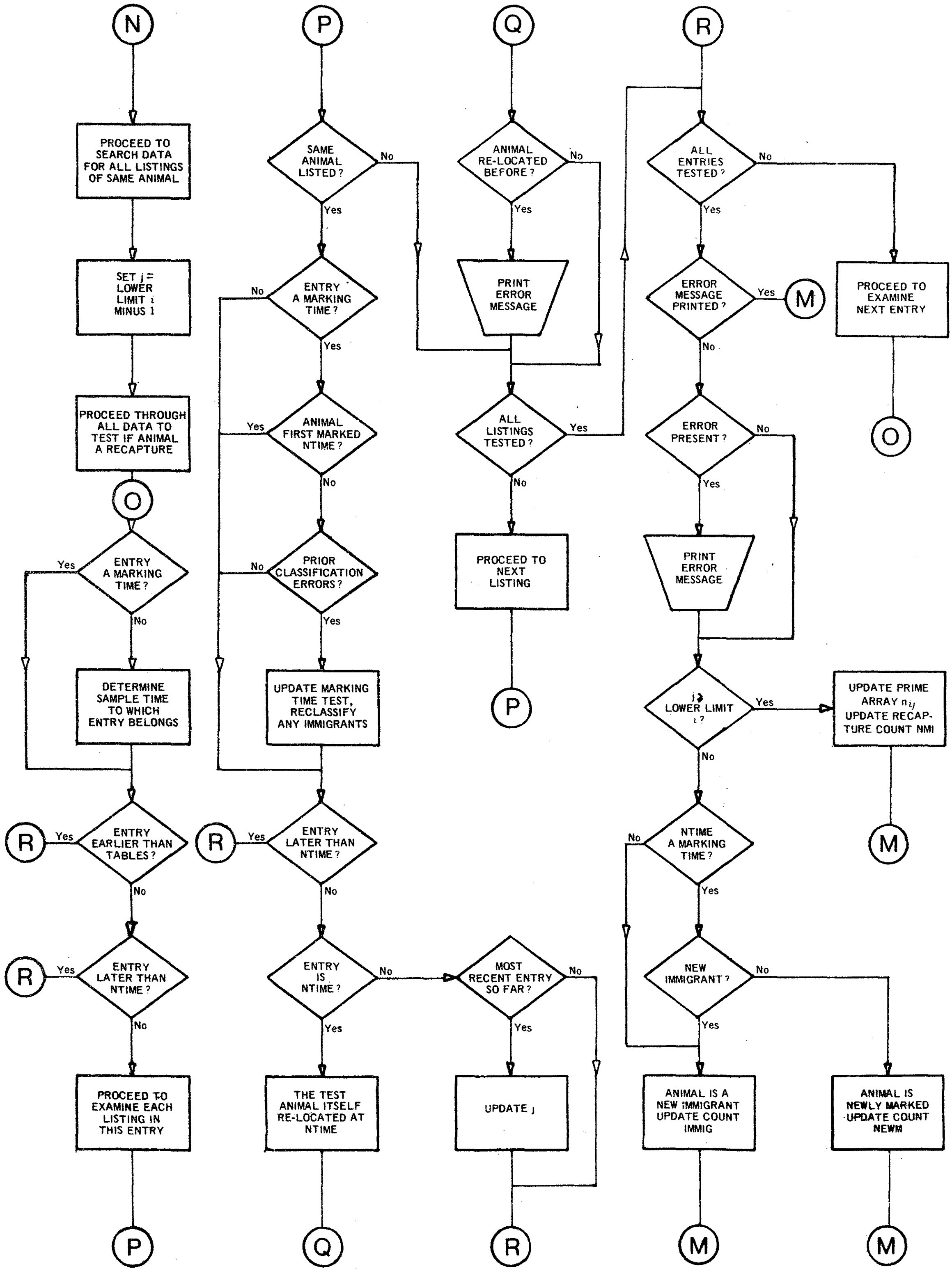

START
E
G
PRESENT COMBINATION WANTED ?
Yes
No
TOO MUCH DATA ?
Yes
No
ZEROISE ARRAY IMIG = Immigrants
F
PROCEED TO NEXT CLASS COMBINATION
PRINT CLASS COMBIN-ATION
PRINT CLASS COMBIN-ATION
READ JOINT SEX FILE, n_{ij} TRELLIS TABLE
CALL NSUB1
CALL NSUB2
PROCEED TO TEST CURRENT COMBINATION OF CLASSES
PROCEED TO EQUIVALENT MALE COMBINATION
CALL JOLLY
PRINT JOINT SEX n_{ij} TRELLIS TABLE
STOP
Yes
ALL COMBINATIONS TESTED ?
No
Yes
FEMALE COMBINATION ?
No
READ JOINT SEX FILE, a_{ij} TRELLIS TABLE
E
PROCEED TO COMBINE AND ANALYSE THE SEXES JOINTLY
PRINT JOINT SEX a_{ij} TRELLIS TABLE
F
Yes
EITHER SEX TOO MUCH DATA ?
No
CALL JOLLY
G
F

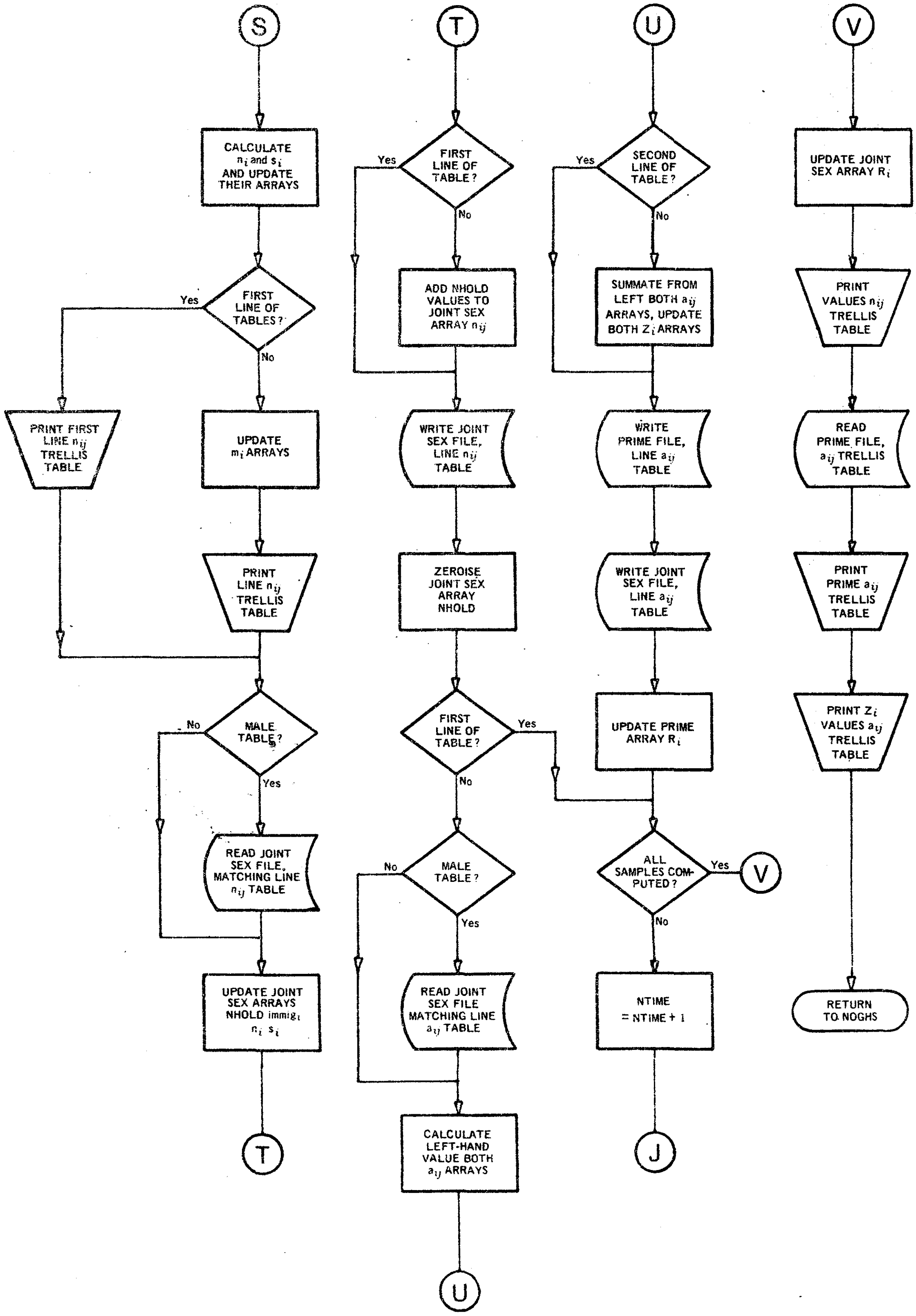
NSUB1


ENTER
PROCEED TO GROUP DATA OF PRESENT CLASS COMBINATION
FEMALE COMBINATION ?
No
Yes
ZEROISE JOINT SEX FILES
n_i s_i m_i
Z_i R_i
ZEROISE DATA ARRAY NCOMB
NEXT = O
H
H
NEXT = NEXT + 1
READ CLASS OF ANIMALS IN NEXT SECTOR
SECTOR CONTAINS DATA ?
No
Yes
IDENTIFY CLASS OF ANIMALS IN SECTOR
ALL COMBINATIONS TESTED ?
Yes
No
PRESENT COMBINATION WANTED ?
No
Yes
I
PRINT CLASS COMBIN-ATION
RETURN TO NOGHS
I
H
No
DOES THIS CLASS BELONG ?
Yes
TOO MUCH DATA ?
Yes
No
READ ALL DATA IN SECTOR
WRITE DATA IN ARRAY NCOMB
H

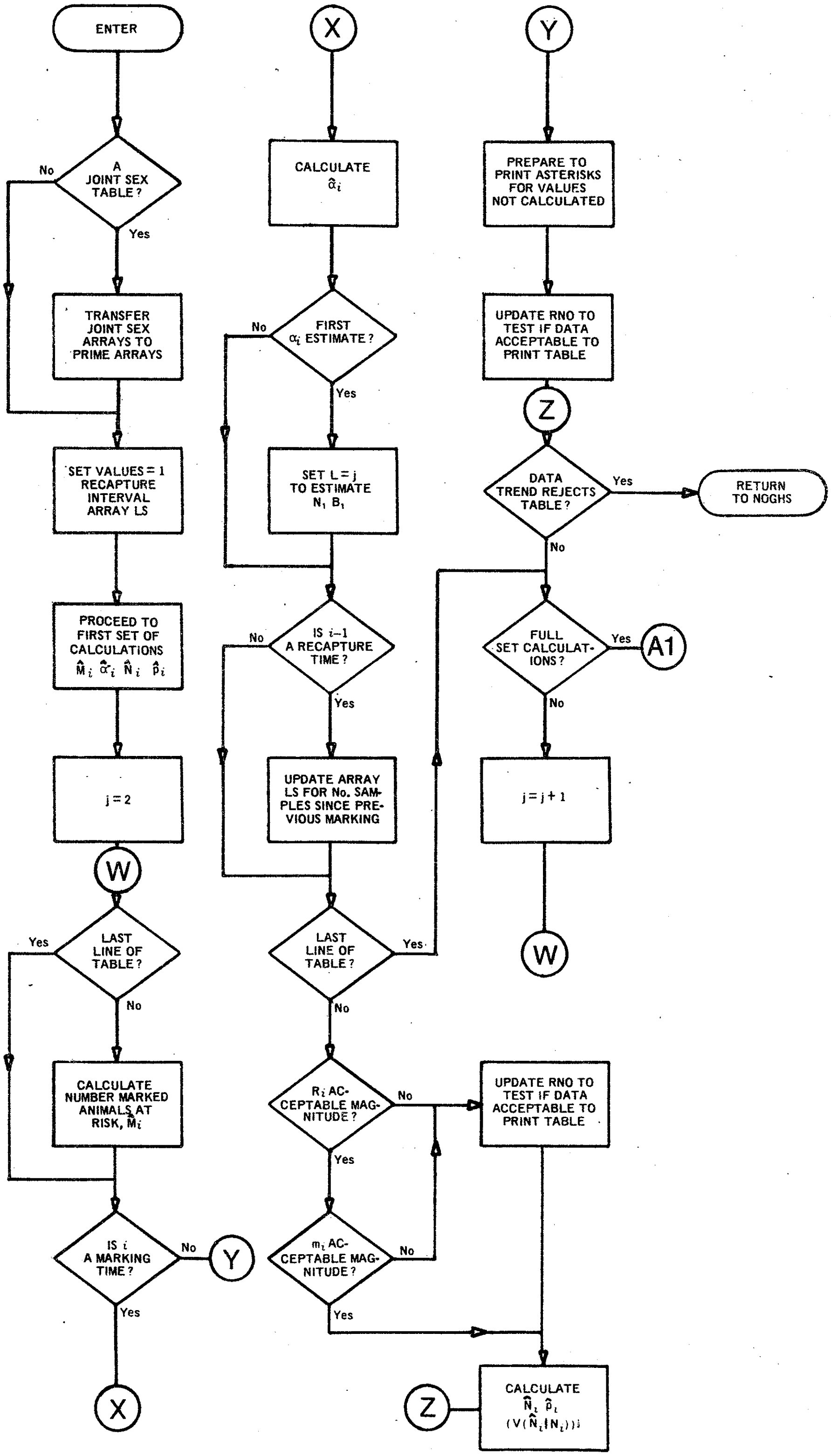
NSUB2

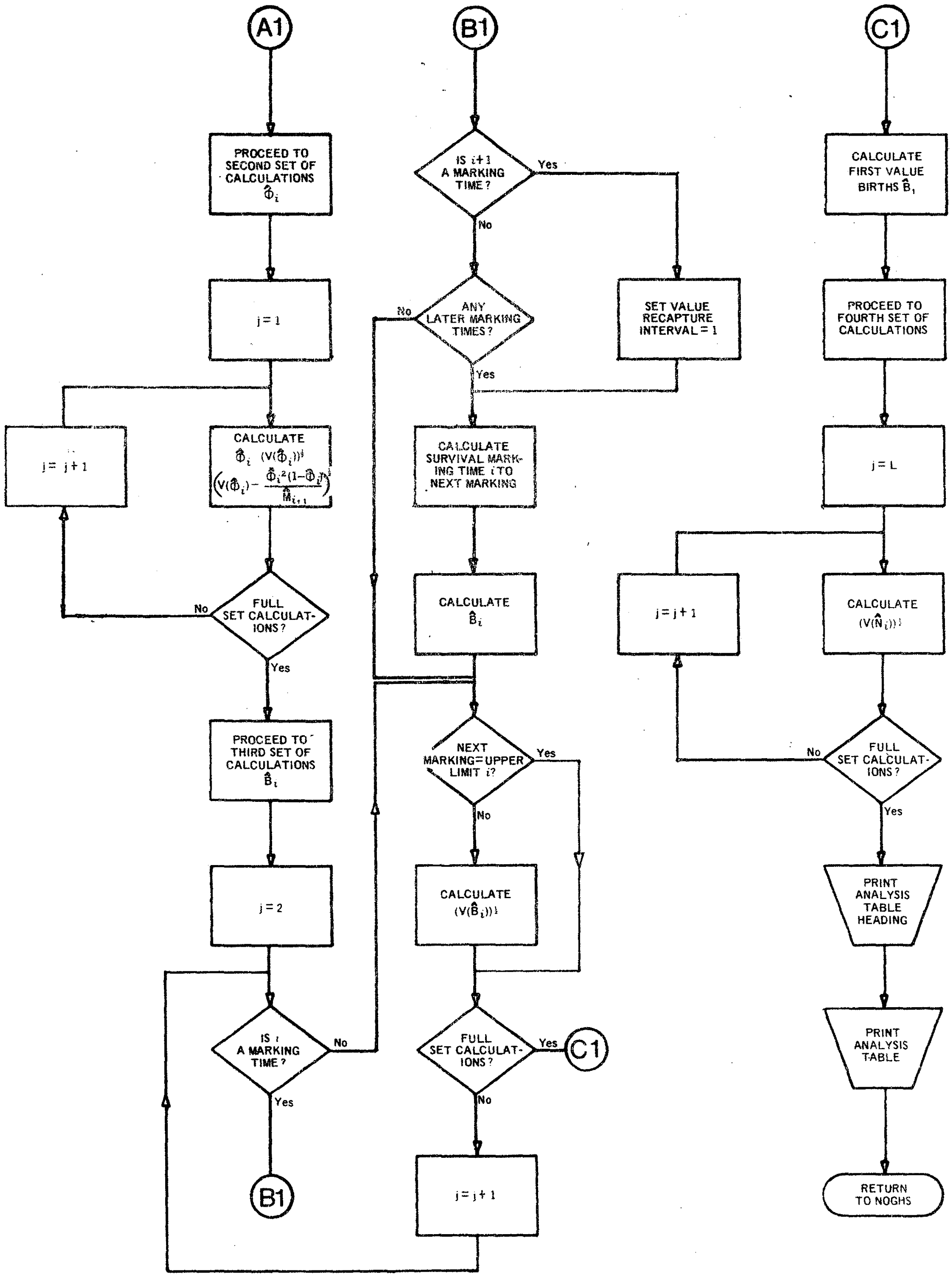

ENTER
ZEROISE PRIME ARRAYS Immigrant$_i$ R_i Z_i n_i s_i m_i a_{ij}
SET i TO LOWER LIMIT NOMINATED ON CONTROL CARD
J
PROCEED TO COMPUTE DATA FROM SAMPLE i= NTIME
ZEROISE COUNTS NEWM, IMMIG, NMI, MW, NW
ZEROISE JOINT SEX ARRAY a_{ij} AND PRIME ARRAY n_{ij}
PROCEED TO COMPUTE LINE OF TRELLIS TABLES
PROCEED TO SEARCH DATA FOR ENTRIES OF NTIME SAMPLE
K
K
ENTRY A MARKING TIME?
Yes
No
DETERMINE SAMPLE TIME TO WHICH ENTRY BELONGS
PROCEED TO EXAMINE NEXT ENTRY
No
ALL ENTRIES TESTED?
Yes
S
No
ENTRY FROM NTIME SAMPLE?
Yes
PROCEED TO EXAMINE EACH LISTING IN THIS ENTRY
S
Yes
LISTING EMPTY?
No
N
Yes
MARKED ANIMAL LISTED?
No
LISTING REFERS TO A REMOVED ANIMAL
L
L
IS ANIMAL MARKED?
No
Yes
UPDATE MARKED REMOVALS MW
UPDATE UNMARKED REMOVALS NW
M
ALL LISTINGS TESTED?
Yes
S
No
PROCEED TO NEXT LISTING


N
PROCEED TO SEARCH DATA FOR ALL LISTINGS OF SAME ANIMAL
SET j = LOWER LIMIT i MINUS 1
PROCEED THROUGH ALL DATA TO TEST IF ANIMAL A RECAPTURE
O
ENTRY A MARKING TIME ?
Yes
No
DETERMINE SAMPLE TIME TO WHICH ENTRY BELONGS
ENTRY EARLIER THAN TABLES ?
R
ENTRY LATER THAN NTIME ?
PROCEED TO EXAMINE EACH LISTING IN THIS ENTRY
P
SAME ANIMAL LISTED ?
ENTRY A MARKING TIME ?
ANIMAL FIRST MARKED NTIME ?
PRIOR CLASSIFICATION ERRORS ?
UPDATE MARKING TIME TEST, RECLASSIFY ANY IMMIGRANTS
ENTRY LATER THAN NTIME ?
ENTRY IS NTIME ?
THE TEST ANIMAL ITSELF RE-LOCATED AT NTIME
Q
ANIMAL RE-LOCATED BEFORE ?
PRINT ERROR MESSAGE
ALL LISTINGS TESTED ?
PROCEED TO NEXT LISTING
MOST RECENT ENTRY SO FAR ?
UPDATE j
ALL ENTRIES TESTED ?
ERROR MESSAGE PRINTED ?
M
ERROR PRESENT ?
j ≥ LOWER LIMIT i ?
NTIME A MARKING TIME ?
NEW IMMIGRANT ?
ANIMAL IS A NEW IMMIGRANT UPDATE COUNT IMMIG
ANIMAL IS NEWLY MARKED UPDATE COUNT NEWM
PROCEED TO EXAMINE NEXT ENTRY
UPDATE PRIME ARRAY n_{ij} UPDATE RECAPTURE COUNT NMI


S
CALCULATE n_i and s_i AND UPDATE THEIR ARRAYS
FIRST LINE OF TABLES?
Yes
No
PRINT FIRST LINE n_{ij} TRELLIS TABLE
UPDATE m_i ARRAYS
PRINT LINE n_{ij} TRELLIS TABLE
MALE TABLES?
No
Yes
READ JOINT SEX FILE, MATCHING LINE n_{ij} TABLE
UPDATE JOINT SEX ARRAYS NHOLD $immig_i$ n_i s_i
T
T
FIRST LINE OF TABLE?
Yes
No
ADD NHOLD VALUES TO JOINT SEX ARRAY n_{ij}
WRITE JOINT SEX FILE, LINE n_{ij} TABLE
ZEROISE JOINT SEX ARRAY NHOLD
FIRST LINE OF TABLE?
Yes
No
MALE TABLE?
No
Yes
READ JOINT SEX FILE MATCHING LINE a_{ij} TABLE
CALCULATE LEFT-HAND VALUE BOTH a_{ij} ARRAYS
U
U
SECOND LINE OF TABLE?
Yes
No
SUMMATE FROM LEFT BOTH a_{ij} ARRAYS, UPDATE BOTH Z_i ARRAYS
WRITE PRIME FILE, LINE a_{ij} TABLE
WRITE JOINT SEX FILE, LINE a_{ij} TABLE
UPDATE PRIME ARRAY R_i
ALL SAMPLES COMPUTED?
Yes
V
No
NTIME = NTIME + 1
J
V
UPDATE JOINT SEX ARRAY R_i
PRINT VALUES n_{ij} TRELLIS TABLE
READ PRIME FILE, a_{ij} TRELLIS TABLE
PRINT PRIME a_{ij} TRELLIS TABLE
PRINT Z_i VALUES a_{ij} TRELLIS TABLE
RETURN TO NOGHS

JOLLY


ENTER
A JOINT SEX TABLE ?
No
Yes
TRANSFER JOINT SEX ARRAYS TO PRIME ARRAYS
SET VALUES = 1 RECAPTURE INTERVAL ARRAY LS
PROCEED TO FIRST SET OF CALCULATIONS
$\hat{M}_i$ $\hat{\alpha}_i$ $\hat{N}_i$ $\hat{p}_i$
j = 2
W
LAST LINE OF TABLE ?
CALCULATE NUMBER MARKED ANIMALS AT RISK, $\hat{M}_i$
IS i A MARKING TIME ?
Y
X
CALCULATE $\hat{\alpha}_i$
FIRST α_i ESTIMATE ?
SET L = j TO ESTIMATE N_1 B_1
IS $i-1$ A RECAPTURE TIME ?
UPDATE ARRAY LS FOR No. SAMPLES SINCE PREVIOUS MARKING
LAST LINE OF TABLE ?
R_i ACCEPTABLE MAGNITUDE ?
m_i ACCEPTABLE MAGNITUDE ?
UPDATE RNO TO TEST IF DATA ACCEPTABLE TO PRINT TABLE
Z
CALCULATE $\hat{N}_i$ $\hat{p}_i$ $(V(\hat{N}_i|N_i))^{\frac{1}{2}}$
PREPARE TO PRINT ASTERISKS FOR VALUES NOT CALCULATED
UPDATE RNO TO TEST IF DATA ACCEPTABLE TO PRINT TABLE
DATA TREND REJECTS TABLE ?
RETURN TO NOGHS
FULL SET CALCULATIONS ?
A1
j = j + 1


A1
PROCEED TO SECOND SET OF CALCULATIONS $\hat{\Phi}_i$
j = 1
CALCULATE $\hat{\Phi}_i$ $(V(\hat{\Phi}_i))^{\frac{1}{2}}$ $\left(V(\hat{\Phi}_i) - \frac{\hat{\Phi}_i^2(1-\hat{\Phi}_i)}{\hat{M}_{i+1}}\right)^{\frac{1}{2}}$
j = j + 1
FULL SET CALCULATIONS?
No
Yes
PROCEED TO THIRD SET OF CALCULATIONS $\hat{B}_i$
j = 2
IS i A MARKING TIME?
No
Yes
B1
B1
IS $i+1$ A MARKING TIME?
Yes
No
ANY LATER MARKING TIMES?
No
Yes
SET VALUE RECAPTURE INTERVAL = 1
CALCULATE SURVIVAL MARKING TIME i TO NEXT MARKING
CALCULATE $\hat{B}_i$
NEXT MARKING=UPPER LIMIT i?
Yes
No
CALCULATE $(V(\hat{B}_i))^{\frac{1}{2}}$
FULL SET CALCULATIONS?
Yes
No
C1
j = j + 1
C1
CALCULATE FIRST VALUE BIRTHS $\hat{B}_1$
PROCEED TO FOURTH SET OF CALCULATIONS
j = L
CALCULATE $(V(\hat{N}_i))^{\frac{1}{2}}$
j = j + 1
FULL SET CALCULATIONS?
No
Yes
PRINT ANALYSIS TABLE HEADING
PRINT ANALYSIS TABLE
RETURN TO NOGHS

ACKNOWLEDGEMENTS

The author gratefully acknowledges the guidance of Dr G. A. F. Seber in clarifying certain aspects of capture-recapture theory, and Miss E. E. Emerson for her invaluable assistance in the writing of the program.

REFERENCES

JOLLY, G. M. 1965 Explicit estimates from capture-recapture data with both death and immigration - stochastic model. Biometrika, 52 (1 and 2): 225-247

WHITE, E. G. 1971 A versatile Fortran computer program for the capture-recapture stochastic model of G. M. Jolly. J. Fish. Res. Bd Canada 28: 443-445

ADDENDUM TO

Special Publication No. 8 (1971)

of the

Tussock Grasslands and Mountain Lands Institute

entitled

A COMPUTER PROGRAM FOR CAPTURE-RECAPTURE STUDIES OF ANIMAL POPULATIONS: A FORTRAN LISTING FOR THE STOCHASTIC MODEL OF G. M. JOLLY

E. G. White

Issued October, 1971

Tussock Grasslands and Mountain Lands Institute,
P. O. Box 56,
Lincoln College,
Canterbury,
New Zealand

PROGRAM LISTING FOR THE IBM 360 COMPUTER

The program listing for the stochastic capture-recapture model of G. M. Jolly as given on Pp. 18-24 is for the IBM 1130 computer. A Fortran listing of the same program is now given for the IBM 360 computer.

The program is written in Fortran IV and was compiled and tested on an IBM 360/44 computer.

USER INFORMATION

The user information given on Pp. 1-14 applies to the present listing with the following modifications:

A. Detailed Program Description

1. Composition - subroutine NOGHS replaces program NOGHS

2. Program requirements - any IBM 360 programming system (Disk). The listed program requires 19,212 four-byte words of core on the IBM 360/44.

B. Input Description

1. Control card format:

Columns 1-20 As detailed on Pp. 6-7

Columns 21-23	Not used
Column 24	The number representing the value of NSW1, which in the IBM 1130 program was set by the position of console switch 5. The number 1 is used to zeroise the file NA when first using the file or deleting stored data; the number 2 is used when executing analyses of stored data or storing further data for analysis.
Columns 25-80	Not used

OPERATING INSTRUCTIONS

The compilation and execution of the program should be standardised according to the model of IBM 360 computer being used. The following four files must be defined:

FILE 1, 250 records of size 1268 bytes, data set reference no. 1

FILE 2, 36 records of size 168 bytes, data set reference no. 2

FILE 3, 36 records of size 148 bytes, data set reference no. 3

FILE 4, 36 records of size 148 bytes, data set reference no. 4

Data set reference numbers are used in a direct access manner.

SYSTEM MATERIAL

A. Program Listing

The present program listing is basically the same as that given for the IBM 1130 on Pp. 18-24, but in view of the fact that the IBM 360/44 provides a much larger core, several modifications have been introduced.

1. Data capacity

   The size of NCOMB, the area of core used for given selections of data for analysis, has been set at 6000 words and may be further enlarged if desired (cf. 1584 words in the IBM 1130 listing, which maximised NCOMB when 24 samples were permitted in the analyses). It should be noted that the increased size of NCOMB makes it less necessary to optimise the format of input data (as outlined on P. 13) because the size of NCOMB is less likely to be limiting. The listed data card format of three animal records per card permits a maximum of 4500 animal records to be included in a given analysis when the size of NCOMB is 6000.

2. Number of samples in analyses

   Up to 36 consecutive samples can be included in a given analysis (cf. 24 in the IBM 1130 listing), but this number requires that the output format of the trellis tables be restricted to three digits per column (the left three columns of the first trellis table are excepted and retain four digits per column). If values larger than three digits are expected, the format must be expanded at the expense of the number of samples permitted in the analysis.

3. Data storage

The program NDATA has been modified to increase the rate of storing data but the file holding the stored data has been reduced from 720 to 250 records.

4. Zero values

It has been necessary in programming for the IBM 360 computer to expand the subroutine JOLLY to bypass calculations including zero values in denominator expressions. The expansions ensure the printing of asterisks where necessary in the table of estimates. Such asterisks are additional to those already printed for any intermediate recapture samples as described on P.8.

B. <u>Flow Diagrams</u>

The flow diagrams presented on Pp. 25-32 are modified by the present listing only to the extent of the changes in sections 3 and 4 above.

NDATA

```
      DIMENSION IA(4),N(4),INC(250),KNC(250)
      COMMON JPOP(312),NA(4),IQQQ(510),NEXP,NX,RQQ(432),IMIG(36)
      COMMON NOW,NBGIN,NBNOW,KOMPL,KOMSP,KOMST,LS(18)
      DEFINE FILE 1(250,317,U,NEXP)
    1 FORMAT (I2,3I1,I3,3(2X,I5))
    2 FORMAT (6I4)
    3 FORMAT (' DISK FULL',I6,' RECORDS')
      READ (5,2) NOW,NBGIN,KOMPL,KOMSP,KOMST,NSW1
C-----IF PRIOR RECORDS LEFT ON DISK WHEN NBGIN IS INCREMENTED, EFFECTIVE
C-----CAPACITY OF NCOMB IS REDUCED
C-----MAKE EACH DISK RECORD IDENTIFICATION ZERO (NSW1=1) FOR FIRST
C-----DATA CARDS ONLY.   AT OTHER TIMES, PUT NSW1=2.
      GO TO (9,12),NSW1
    9 DO 10 I=1,4
   10 NA(I)=0
      DO 11 I=1,250
   11 WRITE (1'I) NA
   12 CONTINUE
      NBNOW=NOW-NBGIN
      DO 1012 NP=1,250
      INC(NP)=0
 1012 KNC(NP)=0
   13 READ (5,1) IA,N
      IB=(IA(1)*1000)+(IA(2)*100)+(IA(3)*10)+IA(4)
      IF (IA(1)-99) 16,28,28
   16 DO 18 NP=1,250
      IF (INC(NP)-IB) 17,1016,18
 1016 IF (KNC(NP)-312) 21,18,18
   17 IF (INC(NP)-1) 1024,18,18
   18 CONTINUE
      WRITE (6,3) NP
      CALL EXIT
   21 READ (1'NP) NA,KOUNT,JPOP
      DO 22 K=1,4
      LOT=KOUNT+K
   22 JPOP(LOT)=N(K)
      KOUNT=KOUNT+4
      WRITE (1'NP) NA,KOUNT,JPOP
      KNC(NP)=KOUNT
      GO TO 13
 1024 DO 25 K=1,4
   25 JPOP(K)=N(K)
      KOUNT=4
      DO 26 J=1,4
   26 NA(J)=IA(J)
      WRITE (1'NP) NA,KOUNT,JPOP
      INC(NP)=IB
      KNC(NP)=KOUNT
      GO TO 13
   28 CONTINUE
      CALL NOGHS
      END
```

NOGHS

```
      SUBROUTINE NOGHS
      COMMON JPOP(312),NA(4),NHOLD(40),NIJ(36),NRI(36),NRIX(36),NZI(36),
     1NZIX(36),NAIJ(36),NMIY(36),NNIY(36),NSIY(36),NREP,NP,NNIX(36),
     2NSIX(36),NMIX(36),NAIJX(36),NEXP,NX,ESTM(36),RECAP(36)
     3,ESTNO(36),ESURV(36),EBORN(36),ESTP(36),VEST(36),SEEST(36),SESUR
     4(36),SEESS(36),SEBOR(36),SENO(36),IMIG(36)
      COMMON NOW,NBGIN,NBNOW,KOMPL,KOMSP,KOMST,LS(18),NCOMB(6000)
C-----NOTE WELL...THE SIZE OF NCOMB MUST BE A MULTIPLE OF THE ARRAY N
C-----IN PROGRAM NDATA
      DEFINE FILE 1(250,317,U,NEXP),2(36,42,U,NREP),3(36,37,U,NP),4(36,3
     17,U,NX)
 1004 FORMAT (//////,' KOMPL',I3,3X,'KOMSP',I2,3X,'KOMST',I2,3X,'KOMSX',
     1I2)
    8 FORMAT (/,I4,2I4,1X,36I3)
 4008 FORMAT (/,7X,'NRI',3X,36I3)
14008 FORMAT (/,13X,36I3)
24008 FORMAT (/,7X,'NZI',3X,36I3)
34008 FORMAT (//////,I16,/)
      NUMBR=6000
      KALC=0
      KOMSX=1
      NREC=36
C-----START SELECTING DATA IN CURRENT CLASS COMBINATION
   29 CALL NSUB1(KOMSX,KOM4,NREC,NUMBR)
      GO TO (276,33,271,288),KOM4
   33 WRITE (6,1004) KOMPL,KOMSP,KOMST,KOMSX
C-----START ANALYSIS
      CALL NSUB2 (NREC,NUMBR,KOMSX,NTWO)
C-----START JOLLY FORULAE
      NAG=0
      CALL JOLLY (NTWO,NAG)
      KALC=KALC+1
  271 IF (KOMSX-2) 272,273,273
  272 KOMSX=2
      GO TO 29
  273 KOMSX=3
      IF (KALC-2) 276,274,276
  274 DO 1274 J=1,36
 1274 IMIG(J)=0
      WRITE (6,1004) KOMPL,KOMSP,KOMST,KOMSX
      NB1=NBNOW+1
      DO 275 NREC=1,NB1
      NFIVE=NREC+3
      READ (2'NREC) NHOLD
      WRITE (6,8) (NHOLD(N),N=1,NFIVE)
  275 IMIG(NREC)=NHOLD(1)
      WRITE (6,4008) (NRIX(J),J=1,NBNOW)
      NAGGX=1
      WRITE (6,34008) NBGIN
      DO 1275 NTIME=1,NBNOW
      NTONE=NTIME+1
      READ (4'NAGGX) NAIJX
      WRITE (6,14008) (NAIJX(J),J=1,NTONE)
 1275 NAGGX=NAGGX+1
      WRITE (6,24008) (NZIX(J),J=1,NBNOW)
      NAG=1
      CALL JOLLY (NTWO,NAG)
  276 KALC=0
      KOMSX=1
      IF (KOMST-3) 278,280,280
  278 KOMST=KOMST+1
      GO TO 29
  280 KOMST=1
      IF (KOMSP-1) 282,284,284
  282 KOMSP=KOMSP+1
      GO TO 29
  284 KOMSP=1
      IF (KOMPL-5) 286,288,288
  286 KOMPL=KOMPL+1
      GO TO 29
  288 CALL EXIT
      END
```

NSUB1

```
      SUBROUTINE NSUB1(KOMSX,KOM4,NREC,NUMBR)
      COMMON JPOP(312),NA(4),NHOLD(40),NIJ(36),NRI(36),NRIX(36),NZI(36),
     1NZIX(36),NAIJ(36),NMIY(36),NNIY(36),NSIY(36),NREP,NP,NNIX(36),
     2NSIX(36),NMIX(36),NAIJX(36),NEXP,NX,RQQ(432),IMIG(36)
      COMMON NOW,NBGIN,NBNOW,KOMPL,KOMSP,KOMST,LS(18),NCOMB(6000)
    4 FORMAT (/////,' NCOMB FULL',I5,' EXCEEDS 6000    KOMPL',I3,'  KOMS
     1P',I3,'  KOMST',I3,'  KOMSX',I3)
      IF (KOMSX-1) 10029,10029,30
10029 DO 1029 J=1,40
 1029 NHOLD(J)=0
      DO 2029 J=1,NREC
      NNIX(J)=0
      NSIX(J)=0
      NMIX(J)=0
      NZIX(J)=0
 2029 NRIX(J)=0
   30 DO 1030 I=1,NUMBR
 1030 NCOMB(I)=0
      NUMBR=0
      NEXT=0
      KOM4=1
   31 NEXT=NEXT+1
      READ (1'NEXT) NA,KOUNT
      NPLOT=NA(1)
      IF (NPLOT-1) 137,1031,1031
 1031 NSPEC=NA(2)
      NSTAG=NA(3)
      NSEX=NA(4)
      GO TO (35,135,235,34),KOMPL
   34 KOM4=4
      GO TO 137
   35 IF (NPLOT-2) 36,31,31
  135 IF (NPLOT-2) 31,36,31
  235 IF (NPLOT-2) 36,36,31
   36 GO TO (37,38,120),KOMST
   37 IF (NSTAG-2) 120,31,31
   38 IF (NSTAG-2) 31,120,31
  120 GO TO (121,123),KOMSX
  121 IF (NSEX-2) 130,31,31
  123 IF (NSEX-2) 31,130,31
  130 KOM4=2
      NUMBR=NUMBR+KOUNT
      IF (NUMBR-6000) 133,133,131
  131 WRITE (6,4) KOMPL,KOMSP,KOMST,KOMSX
      KOM4=3
      NUMBR=NUMBR-KOUNT
      GO TO 137
  133 NK=NUMBR-KOUNT
      READ(1'NEXT) NA,KOUNT,JPOP
      DO 134 NTRY=1,KOUNT
      NT=NK+NTRY
  134 NCOMB(NT)=JPOP(NTRY)
      GO TO 31
  137 RETURN
      END
```

NSUB2

```
      SUBROUTINE NSUB2 (NREC,NUMBR,KOMSX,NTWO)
      COMMON JPOP(312),NA(4),NHOLD(40),NIJ(36),NRI(36),NRIX(36),NZI(36),
     1NZIX(36),NAIJ(36),NMIY(36),NNIY(36),NSIY(36),NREP,NP,NNIX(36),
     2NSIX(36),NMIX(36),NAIJX(36),NEXP,NX,RQQ(432),IMIG(36)
      COMMON NOW,NBGIN,NBNOW,KOMPL,KOMSP,KOMST,LS(18),NCOMB(6000)
    5 FORMAT (' ANIMAL NO.',I6,1X,'RECORDED MORE THAN ONCE, KK=',I3,5X,'
     1OR IF KK=0, ERROR IN PROGRAM')
    6 FORMAT (' ERROR IN DISK SEARCH OPERATION, ANIMAL NO.',I6,1X,'NOT
     1RELOCATED, KK=',I3)
 1008 FORMAT (/,I4,2I4,1X,36I3)
 4008 FORMAT (/,7X,'NRI',3X,36I3)
14008 FORMAT (/,13X,36I3)
24008 FORMAT (/,7X,'NZI',3X,36I3)
34008 FORMAT (//////,I16,/)
      NREC=NOW-NBGIN+1
      DO 137 J=1,NREC
      IMIG(J)=0
      NRI(J)=0
      NZI(J)=0
      NNIY(J)=0
      NSIY(J)=0
      NMIY(J)=0
  137 NAIJ(J)=0
      NREL=0
      NREC=1
      NAGGR=1
      NAGGX=1
      DO 220 NTIME=NBGIN,NOW
      NBT=NTIME-NBGIN+1
      NEWM=0
      IMMIG=0
      NMI=0
      MW=0
      NW=0
      MREC=0
      NCHEK=1
      NT=1
      DO 138 J=1,NREC
  138 NAIJX(J)=0
      DO 140 J=1,NBNOW
  140 NIJ(J)=0
  141 NREAL=NCOMB(NT)
      IF (NREAL-500) 1142,1142,142
  142 NREAL=NREAL-500
 1142 IF (NREAL-NTIME) 189,143,189
  143 DO 188 I=1,3
      NTI=NT+I
      N2=NCOMB(NTI)
      IF (N2-1) 189,144,144
  144 IF (N2-1000) 183,183,145
  145 KK=0
      JTIME=NBGIN-1
      LTIME=1
```

```
  146 IF (NCOMB(LTIME)-500) 148,148,147
  147 LREAL=NCOMB(LTIME)-500
      GO TO 149
  148 LREAL=NCOMB(LTIME)
  149 IF (LREAL-NBGIN) 171,150,150
C-----THE FOREGOING TEST ASSUMES NBGIN IS ALWAYS A MARKING TIME.
C-----WHEN NBGIN IS MADE GREATER THAN 1, THIS TEST PROCEEDS
C-----TO OVERLOOK  EARLIER RECORDS AND LISTS AS AN IMMIGRATION THE FIRST
C-----RECAPTURE OF ANY ANIMAL THAT WAS MARKED PRIOR TO NBGIN.
  150 IF (LREAL-NREAL) 154,154,171
  154 DO 168 L=1,3
      LTL=LTIME+L
      IF (NCOMB(LTL)-N2) 168,156,168
  156 IF (NCOMB(LTIME)-500) 158,158,163
  158 JREL=N2/1000
      IF (NREL-JREL) 161,163,1161
C-----THE TEST (NREL-JREL) DETERMINES FROM THE CURRENT (NTIME) RECORDS
C-----THEMSELVES WHETHER OR NOT THE MARKING TIME (JREL) OF A GIVEN
C-----ANIMAL WAS PRIOR TO THE MOST RECENT MARKING TIME (NREL) OF ANIMALS
C-----SO FAR TESTED.          IF ANY MARKED IMMIGRANTS AT NTIME ARE
C-----LISTED BEFORE NREL HAS BEEN UPDATED TO THE MOST RECENT MARKING
C-----TIME, THEY ARE TEMPORARILY CLASSIFIED AS NEWLY MARKED (NEWM) UNTIL
C-----THE UPDATING OF NREL CORRECTLY RECLASSIFIES THEM (SEE 161 + 001).
C-----THE TEST (NREL-JREL) FAILS IN THE SAME UNLIKELY AND UNIMPORTANT
C-----CIRCUMSTANCES LISTED SECOND FOR NCHEK BELOW.
  161 NREL=JREL
      IMMIG=NEWM
      NEWM=0
      GO TO 163
 1161 NCHEK=NCHEK+1
  163 IF (LREAL-NTIME) 169,165,171
  165 KK=KK+1
      NTEST=NCOMB(NT)
      IF (KK-1) 167,168,167
  167 WRITE (6,5) N2,KK
C-----THE ABOVE ERROR MESSAGE IS NOT PRINTED WHEN KOMSX=3
  168 CONTINUE
      GO TO 171
  169 IF (LREAL-JTIME) 171,171,170
  170 JTIME=LREAL
  171 IF (LTIME+3-NUMBR) 172,174,174
  172 LTIME=LTIME+4
      GO TO 146
  174 IF (KK-1) 175,176,188
  175 WRITE (6,6) N2,KK
  176 IF (JTIME-NBGIN+1) 188,178,177
  177 J=JTIME-NBGIN+1
      NIJ(J)=NIJ(J)+1
      NMI=NMI+1
      GO TO 188
  178 IF (NTEST-500) 179,179,180
  179 IF (NCHEK-2) 1180,180,180
C-----NCHEK FOR IMMIGRANTS FAILS IF OTHER ERRORS ARE PRINTED.
C-----NCHEK ALSO FAILS FOR THE IMMIGRANTS WITH THE MOST RECENT MARKING
C-----TIME IF NO NEWLY MARKED ANIMALS ARE PRESENT AT NTIME AND MORE
C-----RECENT MARKINGS ARE ABSENT IN PREVIOUS VALUES OF NTIME.  THE LIKE-
C-----LIHOOD AND IMPORTANCE OF SUCH AN ERROR IS SLIGHT
  180 IMMIG=IMMIG+1
      GO TO 188
 1180 NEWM=NEWM+1
      GO TO 188
  183 IF (N2-300) 185,185,186
  185 NW=NW+1
      GO TO 188
  186 MW=MW+1
  188 NCHEK=1
```

```
 189 NT=NT+4
     IF (NT-NUMBR) 141,191,191
 191 NNI=NEWM+NMI+IMMIG+NW
     NSI=NNI-MW-NW
     NNIY(NBT)=NNI
     NSIY(NBT)=NSI
     NNIX(NBT)=NNIX(NBT)+NNI
     NSIX(NBT)=NSIX(NBT)+NSI
     IF (NTIME-NBGIN-1) 1193,2193,2193
1193 WRITE (6,1008)IMMIG,NNI,NSI,NTIME
     GO TO 201
2193 NMIY(NBT)=NMI
     NMIX(NBT)=NMIX(NBT)+NMI
     NT1=NTIME-NBGIN
     WRITE (6,1008) IMMIG,NNI,NSI,(NIJ(J),J=1,NT1),NTIME
 201 IMIG(NBT)=IMMIG
     IF (KOMSX-2) 205,204,204
 204 READ (2'NREC) NHOLD
 205 NHOLD(1)=NHOLD(1)+IMMIG
     NHOLD(2)=NHOLD(2)+NNI
     NHOLD(3)=NHOLD(3)+NSI
     IF (NTIME-NBGIN-1) 208,1205,1205
1205 NPLUS=NTIME-NBGIN+3
     DO 206 K=4,NPLUS
 206 NHOLD(K)=NHOLD(K)+NIJ(K-3)
     NHOLD(NPLUS+1)=NTIME
     GO TO 209
 208 NHOLD(4)=NTIME
 209 WRITE (2'NREC) NHOLD
     DO 210 J=1,40
 210 NHOLD(J)=0
     IF (NTIME-NBGIN-1) 220,1210,1210
1210 IF (KOMSX-1) 212,212,211
 211 READ (4'NAGGX) NAIJX
 212 NAIJ(1)=NIJ(1)
     NAIJX(1)=NAIJX(1)+NIJ(1)
     IF (NTIME-NBGIN-2) 1214,213,213
 213 NTWO=NTIME-1-NBGIN
     DO 214 I=1,NTWO
     NAIJ(I+1)=NAIJ(I)+NIJ(I+1)
     NAIJX(I+1)=NAIJX(I+1)+NAIJ(I+1)
     NZI(I)=NZI(I)+NAIJ(I)
     NZIX(I)=NZIX(I)+NAIJ(I)
 214 CONTINUE
1214 WRITE (3'NAGGR) (NAIJ(J),J=1,NT1),NTIME
     WRITE (4'NAGGX) (NAIJX(J),J=1,NT1),NTIME
     NAGGR=NAGGR+1
     NAGGX=NAGGX+1
     DO 215 I=1,NBNOW
 215 NRI(I)=NRI(I)+NIJ(I)
 220 NREC=NREC+1
     DO 216 I=1,NBNOW
 216 NRIX(I)=NRIX(I)+NRI(I)
     WRITE (6,4008) (NRI(J),J=1,NBNOW)
     NAGGR=1
     NAGGX=1
     WRITE (6,34008) NBGIN
     DO 1220 NTIME=1,NBNOW
     NTONE=NTIME+1
     READ (3'NAGGR) NAIJ
     WRITE (6,14008) (NAIJ(J),J=1,NTONE)
1220 NAGGR=NAGGR+1
     WRITE (6,24008) (NZI(J),J=1,NBNOW)
     RETURN
     END
```

JOLLY

```
      SUBROUTINE JOLLY (NTWO,NAG)
      COMMON JPOP(312),NA(4),NHOLD(40),NIJ(36),NRI(36),NRIX(36),NZI(36),
     1NZIX(36),NAIJ(36),NMIY(36),NNIY(36),NSIY(36),NREP,NP,NNIX(36),
     2NSIX(36),NMIX(36),NAIJX(36),NEXP,NX,ESTM(36),RECAP(36)
     3,ESTNO(36),ESURV(36),EBORN(36),ESTP(36),VEST(36),SEEST(36),SESUR
     4(36),SEESS(36),SEBOR(36),SENO(36),IMIG(36)
      COMMON NOW,NBGIN,NBNOW,KOMPL,KOMSP,KOMST,LS(18)
 5008 FORMAT (////,' TIME  PPTN RECAP  PROB CAP  NO MARKED  TOTAL POP  S
     1E POP   SE ESTPOP   NO BORN  SE BIRTHS  SURV RATE  SE SURV  SE ESTS
     2URV')
 6008 FORMAT (I4,2(5X,F6.4),2X,F8.2,2X,5(2X,F8.2),1X,3(4X,F6.4))
 7008 FORMAT (I4,85X,3(4X,F6.4))
 8008 FORMAT (I4,2(5X,F6.4),2X,F8.2,2X,3(2X,F8.2),/,I4,5X,F6.4,///)
      NB1=NBNOW+1
      IF (NAG-1) 1221,2220,2220
 2220 DO 221 JTIME=1,NB1
      NNIY(JTIME)=NNIX(JTIME)
      NSIY(JTIME)=NSIX(JTIME)
      NMIY(JTIME)=NMIX(JTIME)
      NRI(JTIME)=NRIX(JTIME)
  221 NZI(JTIME)=NZIX(JTIME)
 1221 RNO=0.
      K=0
      L=0
      DO 2221 M=1,18
 2221 LS(M)=1
      L111=1
      ESTM(1)=0.
      DO 226 JTIME=2,NB1
      RNNI=NNIY(JTIME)
      RNSI=NSIY(JTIME)
      RNMI=NMIY(JTIME)
      RIMIG=IMIG(JTIME)
      IF (JTIME-NB1) 12221,22221,226
12221 RNRI=NRI(JTIME)
      RNZI=NZI(JTIME-1)
      IF (RNRI-0.0) 3221,3221,4221
 3221 ESTM(JTIME)=111111.
      GO TO 22221
 4221 ESTM(JTIME)=RNSI*RNZI/RNRI+RNMI
C-----FOLLOWING TEST FOR MARKING TIMES ONLY LIKELY TO FAIL WHEN SAMPLE
C-----SIZE VERY SMALL AND TABLES NOT PRINTED
22221 IF(RNSI-RNMI-RIMIG) 223,223,2222
 2222 RECAP(JTIME)=RNMI/RNNI
      IF (L-1) 3222,4222,4222
 3222 L=JTIME
 4222 IF (L111-1) 223,15222,5222
 5222 K=K+1
      LS(K)=L111
      L111=1
15222 IF (JTIME-NB1) 25222,226,226
25222 IF (RNRI-7.0) 1223,1223,6222
 6222 IF (RNMI-7.0) 1223,1223,224
  223 RECAP(JTIME)=111111.
      ESTP(JTIME)=111111.
      ESTNO(JTIME)=111111.
      SENO(JTIME)=111111.
      SEEST(JTIME)=111111.
      ESURV(JTIME)=111111.
      SESUR(JTIME)=111111.
      SEESS(JTIME)=111111.
      L111=L111+1
      RNO=RNO+1.0
      GO TO 225
 1223 RNO=RNO+1.0
  224 ESTNO(JTIME)=ESTM(JTIME)/RECAP(JTIME)
      ESTP (JTIME)=RNNI/ESTNO(JTIME)
      IF (RNMI-0.0) 30224,30224,10224
10224 IF (RNRI-0.0) 30224,30224,20224
20224 VEST(JTIME)=ESTNO(JTIME)*(ESTNO(JTIME)-RNNI)*(((ESTM(JTIME)-RNMI+
     1RNSI)/ESTM(JTIME))*((1./RNRI)-(1./RNSI))+((1.-RECAP(JTIME))/RNMI))
      IF (VEST(JTIME)-0.0) 1224,1224,2224
C-----THE FOLLOWING DUMMY VALUE FOR VEST(JTIME) ENSURES THAT NO ERRONEOUS VALUE
C-----IS PRINTED FOR SENO(JTIME) WHEN NRI(JTIME)=0
30224 VEST(JTIME)=-ESTNO(JTIME)
 1224 SEEST(JTIME)=111111.
      GO TO 3224
 2224 SEEST(JTIME)=SQRT(VEST(JTIME))
 3224 IF (RNO-0.0) 1225,1225,225
  225 IF (((NBNOW-2)/RNO)-1.04) 260,1225,1225
C-----FINAL VALUE IN THE FOREGOING TEST DETERMINES WHETHER SUFFICIENT
C-----SAMPLING TIMES HAVE ADEQUATE DATA (SEE STATEMENTS 25222 AND 6222)
C-----TO PRINT TABLE.
 1225 EBORN(JTIME)=111111.
      SEBOR(JTIME)=111111.
  226 CONTINUE
```

```
C-----IF POPULATION NOS RATHER UNSTABLE, ESTIMATE N1 FROM TREND, NOT =N2
      ESTNO(1)=ESTNO(L)
      ESURV(1)=ESTM(2)/NSIY(1)
      DO 228 JTIME=1,NTWO
      JTADD=JTIME+1
      RNSI=NSIY(JTIME)
      RNMI=NMIY(JTIME)
      RNRI=NRI(JTIME)
      TNSI=NSIY(JTADD)
      TNMI=NMIY(JTADD)
      TNRI=NRI(JTADD)
      IF (TNRI-0.0) 14227,14227,1226
 1226 VITEM=(((ESTM(JTADD)-TNMI)*(ESTM(JTADD)-TNMI+TNSI))/ESTM(JTADD)**2
     1)*((1./TNRI)-(1./TNSI))
      IF (RNRI-0.0) 14227,14227,2226
 2226 VFORM=((ESTM(JTIME)-RNMI)/(ESTM(JTIME)-RNMI+RNSI))*((1./RNRI)-(1./
     1RNSI))
      IF(JTIME-2)1227,227,227
  227 RNZI=NZI(JTIME-1)
      ESURV(JTIME)=ESTM(JTADD)/((RNSI*RNZI/RNRI)+RNSI)
 1227 REST=ESURV(JTIME)**2*(VITEM+VFORM+((1.-ESURV(JTIME))/ESTM(JTADD)))
      IF (REST-0.0) 2227,2227,3227
 2227 SESUR(JTIME)=111111.
      GO TO 4227
 3227 SESUR(JTIME)=SQRT(REST)
 4227 TEST=SESUR(JTIME)**2-((ESURV(JTIME)**2*(1.-ESURV(JTIME)))/ESTM(JTA
     1DD))
      IF (TEST-0.0) 5227,5227,6227
14227 SESUR(JTIME)=111111.
      ESURV(JTIME)=111111.
 5227 SEESS(JTIME)=111111.
      GO TO 228
 6227 SEESS(JTIME)=SQRT(TEST)
  228 CONTINUE
      K=0
      DO 230 JTIME=2,NTWO
      IF (ESTNO(JTIME)-111111.) 229,5229,5229
  229 IF (ESTNO(JTIME+1)-111111.) 1229,2229,2229
 1229 L111=1
      GO TO 4229
 2229 K=K+1
      L111=LS(K)
      IF (L111-1) 230,230,4229
 4229 JTADD=JTIME+L111
      RNNI=NNIY(JTIME)
      TNNI=NNIY(JTADD)
      TNMI=NMIY(JTADD)
      RNSI=NSIY(JTIME)
      TNSI=NSIY(JTADD)
      RNMI=NMIY(JTIME)
      RNRI=NRI(JTIME)
      TNRI=NRI(JTADD)
      TSURV=1
      J2=JTADD-1
      DO 14229 ITIME=JTIME,J2
14229 TSURV=TSURV*ESURV(ITIME)
      EBORN(JTIME)=ESTNO(JTADD)-TSURV*(ESTNO(JTIME)-RNNI+RNSI)
      IF (NBNOW-JTADD) 230,24229,24229
24229 IF (TNMI-0.0) 74229,74229,34229
34229 IF (RNMI-0.0) 74229,74229,44229
44229 XTRA=(ESTNO(JTADD)*(ESTNO(JTADD)-TNNI)*((1.-RECAP(JTADD))/TNMI))+
     1TSURV**2*ESTNO(JTIME)*(ESTNO(JTIME)-RNNI)*((1.-RECAP(JTIME))/RNMI)
      IF (TNRI-0.0) 74229,74229,54229
54229 IF (RNRI-0.0) 74229,74229,64229
64229 VITEM=(((ESTM(JTADD)-TNMI)*(ESTM(JTADD)-TNMI+TNSI))/ESTM(JTADD)**2
     1)*((1./TNRI)-(1./TNSI))
      VFORM=((ESTM(JTIME)-RNMI)/(ESTM(JTIME)-RNMI+RNSI))*((1./RNRI)-(1./
     1RNSI))
      WEST=EBORN(JTIME)**2*VITEM+(VFORM*(((TSURV*RNSI*(1.-RECAP(JTIME)))
     1/RECAP(JTIME))**2))+(((ESTNO(JTIME)-RNNI)*(ESTNO(JTADD)-EBORN(JTIM
     2E))*(1.-RECAP(JTIME))*(1.-TSURV))/(ESTM(JTIME)-RNMI+RNSI))+XTRA
      IF (WEST-0.0) 74229,74229,84229
74229 SEBOR(JTIME)=111111.
      GO TO 230
84229 SEBOR(JTIME)=SQRT(WEST)
      GO TO 230
 5229 IF (JTIME-2) 230,6229,230
 6229 K=1
  230 CONTINUE
```

```
      EBORN(1)=ESTNO(L)-ESURV(1)*(ESTNO(1)-NNIY(1)+NSIY(1))
      DO 238 JTIME=L,NBNOW
      TOTAL=ESTNO(L)*((ESTNO(L)-EBORN(1))/ESTNO(1))
      IF (ESTNO(JTIME)-111111.) 231,238,238
  231 IF (L-2) 232,232,233
  232 M=0
      GO TO 1233
  233 M=1
 1233 SUM=0
      J1=JTIME-1
      DO 237 I=1,J1
      K=M
      IF (ESTNO(I)-111111.) 1231,237,237
 1231 TERMA=EBORN(I)
      IF (JTIME-I) 236,236,2231
 2231 I1=I+1
      MM=0
      DO 6235 J=I1,J1
      IF (ESTNO(J)-111111.) 234,6235,6235
  234 MM=MM+1
      IF (ESTNO(J+1)-111111.) 2235,1235,1235
 1235 K=K+1
      L111=LS(K)
      IF (MM-1) 3235,11235,3235
11235 M=K
      GO TO 3235
 2235 L111=1
 3235 JL111=J+L111
      IF (SUM) 5235,4235,5235
 4235 TOTAL=TOTAL*((ESTNO(JL111)-EBORN(J))/ESTNO(J))
 5235 TERMA=TERMA*((ESTNO(JL111)-EBORN(J))/ESTNO(J))
 6235 CONTINUE
  236 SUM=SUM+TERMA*TERMA/EBORN(I)
  237 CONTINUE
      YEST=(VEST(JTIME))+ESTNO(JTIME)-SUM-TOTAL*TOTAL/ESTNO(1)
      IF (YEST-0.0) 1237,1237,2237
 1237 SENO(JTIME)=111111.
      GO TO 238
 2237 SENO(JTIME)=SQRT(YEST)
  238 CONTINUE
      WRITE (6,5008)
      WRITE (6,7008) NBGIN,ESURV(1),SESUR(1),SEESS(1)
      DO 253 JTIME=2,NTWO
      NTIME=JTIME+NBGIN-1
  253 WRITE (6,6008) NTIME,RECAP(JTIME),ESTP(JTIME),ESTM(JTIME),ESTNO(JT
     1IME),SENO(JTIME),SEEST(JTIME),EBORN(JTIME),SEBOR(JTIME),ESURV
     2(JTIME),SESUR(JTIME),SEESS(JTIME)
      JTNOW=NOW-1
      WRITE (6,8008) JTNOW,RECAP(NBNOW),ESTP(NBNOW),ESTM(NBNOW),ESTNO(NB
     1NOW),SENO(NBNOW),SEEST(NBNOW),NOW,RECAP(NB1)
  260 RETURN
      END
```