Canadian Technical Report of

Fisheries and Aquatic Sciences 1083

June 1982

A COMPUTER PROGRAM FOR PREDICTING PRECISION AND TAG-LOSS BIAS IN JOLLY-SEBER MARK-RECAPTURE ESTIMATES

by

A. N. Arnason and C. R. Krasey

Department of Computer Science

University of Manitoba

Winnipeg, Manitoba R3T 2N2

and

K. H. Mills

Department of Fisheries and Oceans

Freshwater Institute

Winnipeg, Manitoba R3T 2N6

This is the 144th Technical Report

from the Western Region, Winnipeg

© Minister of Supply and Services Canada 1982

Cat. No. 97-6/1083 ISSN 0706-6457

Correct citation for this publication:

Arnason, A.N., C.R. Krasey, and K.H. Mills. 1982. A computer program for predicting precision and tag-loss bias in Jolly-Seber mark-recapture estimates. Can. Tech. Rep. Fish. Aquat. Sci. 1083: iv + 42p.

TABLE OF CONTENTS

ABSTRACT

Arnason, A.N., C.R. Krasey, and K.H. Mills. 1982. A computer program for predicting precision and tag-loss bias in Jolly-Seber mark-recapture estimates. Can. Tech. Rep. Fish. Aquat. Sci. 1083: iv + 42p.

This technical report provides details on the use of BEFFJOB: an ANSI-FORTRAN-IV program for computing the expected properties of Jolly-Seber estimates for population size, survival rate and birth rates formed from banding or mark-recapture data. The program is designed as a planning tool for sampling experiments where the biologist has a rough idea of the initial population size and turnover (birth, death and removal) rates. The program can then be used to predict the precision that will be achieved in the estimates given specified sample sizes or sampling effort over a specified number of samples. Precision is measured by the standard error and coefficient of variation of the estimate. The program gives results for both the Jolly-Seber full model (allowing for both births and deaths) and the death-only model. In either case, the user may specify given rates of loss-on-capture (e.g. losses due to handling) or of tag loss. The program can be used to assess whether given rates of tag-loss are likely to produce significant bias in the estimates and to show the loss of precision that will be incurred in correcting for this bias. A detailed discussion of example output is given. The FORTRAN program listing is given; it could easily be adapted for use on micro-computers with BASIC, PASCAL or FORTRAN.

Key words: capture-recapture; mark-recapture; Jolly-Seber estimates; survival; abundance; recruitment; tag-loss; bias of estimates; precision; planning experiments; sampling.

RESUME

Arnason, A.N., C.R. Krasey, and K.H. Mills. 1982. A computer program for predicting precision and tag-loss bias in Jolly-Seber mark-recapture estimates. Can. Tech. Rep. Fish. Aquat. Sci. 1083: iv + 42p.

Le présent rapport technique contient des renseignements sur l'utilisation du BEFFJOB, un programme ANSI-FORTRAN-IV destiné à informatiser les propriétés qu'on doit s'attendre à trouver dans les estimations des effectifs de population Jolly-Seber, dans le taux de survie et le taux de naissance découlant des données relatives au marquage ou au marquage-recapture. Le programme a pour objet de servir d'outil de planification lors d'échantillonages au cours desquels le biologiste n'a qu'une idée en gros des effectifs initiaux de population et des taux de roulement (naissance, mort et enlèvement). On pourra alors recourir au programme pour prévoir le degré de précision des estimations que l'on obtiendra lorsqu'on a une grosseur donnée d'échantillons ou lorsque l'échantillonage porte un nombre donné d'échantillons. On obtient ce degré de précision en calculant l'erreur type et le coefficient de variation des estimations. Le programme informatique a donné des résultats pour le modèle de Jolly-Seber "complet" (compte rendu à la fois des naissances et des mortalités) et pour le modèle "mort seulement". Dans l'un ou l'autre cas, l'utilisateur peut préciser quel est le taux donné pour les pertes à la capture (c'est-à-dire, pertes dues à la manipulation) ou les pertes d'étiquette. On peut utiliser le programme pour vérifier si un taux donné de perte d'étiquette concourra à biaiser de façon importante les estimations et à montrer la perte de précision qu'entraînera le fait de rectifier ce biais. Le rapport discute en détail la production d'exemples. Le listage de programme FORTRAN est donné; il porrait être facilement adapté en vue d'être utilisé sur divers microordinateurs (BASIC, PASCAL, FORTRAN).

Mots-clés: capture-recapture; étiquetage; estimations Jolly-Seber; survie; abondance; recrutement; biais des estimations; expérience de planification; échantillonnage.

INTRODUCTION

One of the crucial decisions which must be made in any mark-recapture experiment is the level of resources (time, money, personnel) to be expended in an experiment. The prime question is: can sufficiently precise estimates be obtained to meet our research (or management) needs, given our sampling effort? Mark-recapture estimates of population parameters are generally of little or no value if the standard errors of the estimates are of the same magnitude as the estimates. Robson and Regier (1964) and Jensen (1981) present methods for determining the precision of estimates for Petersen type (2-sample) experiments, given certain levels of effort and true population sizes. These intuitively simple methods are not suitable for more complicated (K-sample) mark-recapture experiments. The Jolly-Seber (Jolly 1965; Seber 1965) multiple mark-recapture models are generally considered the most widely applicable methods for K-sample experiments (Cormack 1968; Arnason and Baniuk 1980; Seber 1973). This report describes the use of a computational program, called BEFFJOB, which we have developed for judging precision in Jolly-Seber estimates. It was originally developed, however, for assessing the bias and precision of Jolly-Seber estimates when tag-loss is present. This feature is retained so that the user can also use the program to judge the bias due to tag loss occurring at a given rate, and hence to determine whether estimates need to be corrected for tag-loss bias (as described by Arnason and Mills 1981). The degree of bias which can be tolerated in estimates depends on the precision of the estimate which, in turn, is related to the true population size, the sampling intensity, and the so-called ´turn-over´ rates. Turn-over rates is a collective term for the rates of mechanisms that tend to decrease the fraction of marked animals in the population: birth (recruitment or immigration) rates, death (physical death, emigration) rates, loss-on-capture (of animals removed permanently by the experimenter) rates, and, if present, tag-loss rates. Our program allows the user to vary all the above parameters at every sample time for a user-set number of sampling times (K > 2), and to see what precision and bias this will lead to in the various estimates. As such, it is useful for planning sampling programs, or for modifying programs as they progress.

PRECISION, BIAS AND NOTATION

In this section, we define some of the more important terms used in this report and describe the notation used both in this report and in the output of BEFFJOB. By precision of an estimate, we mean the magnitude of the estimate relative to some measure of the uncertainty in the estimate (i.e. the ´signal-to-noise´ ratio). In this report, precision will be measured by the inverse of the coefficient of variation of the estimate:

CV(est) = SE(est)/est

where SE(est) stands for the (expected) standard error of the estimate and est stands for (the expected value of) any of the estimated parameters of interest (such as population size, survival rate, or births, at any given sample time). With actual mark-recapture data, both the estimate and the standard error are random variables, but, in this report, they are the expected (or average) values that would arise in an experiment with user-set parameters (for initial population size, turnover and sampling rates). BEFFJOB reports the (expected) CV for each estimate as a percentage (i.e. as CV= 100 * SE(est)/est); note that low CV values correspond to high precision and high CV values to low or poor precision. In general, one should aim to achieve a CV of 20% or less for most of the estimates of interest unless the experimenter needs only very rough estimates of the population parameters. Often, precisions much higher than this (i.e. CV < 20%) are required; for example, when one wishes to test specific hypotheses that, say, survival differs over time or among different sub-groups of animals. In such cases, very high precision may be needed if one is to have any chance of detecting anything but the grossest differences. Some general rules of thumb for the major determinants of precision in the various estimates are given in the next section.

The difference between an estimate of a quantity and the true value of that quantity is influenced by two factors: the variability in the estimate about its average (or expected) value, and the bias in the estimate where bias is the difference between that average and the true value being estimated. Bias in Jolly-Seber estimates results from one of two main causes: small sample bias and failure of model assumptions. Small sample bias arises due to the fact that the (maximum likelihood) estimates are only asymptotically unbiased (as sample sizes and recapture rates become large) even when all model assumptions are satisfied. Small sample bias is generally negligible in experiments that yield reasonably precise estimates. Arnason and Mills (1981) found, using simulations (Arnason and Baniuk 1978), that small sample bias was negligible whenever the expected number of marks (M(I)) and subsequent recaptures (R(I)) at sample time I are not too small (>5). BEFFJOB provides a check that this condition is satisfied by allowing the user to specify cut-off values for these quantities; computations for precision and bias will be suppressed if the M(I) or R(I) fall below these cut-off values. The second source of bias is, in practice, far more serious. The major model assumptions of the Jolly-Seber models are:

(1) Correct closure assumptions (i.e that the analysis used allows for births

and/or deaths if these are in fact occurring.

(2) No tag-loss (or more generally, that the correct capture history of every animal can be identified on capture).

(3) Homogeneity of survival (among all animals within a sub-group of animals to be analysed together, regardless of their age, size, previous history of capture, etc.).

(4) Homogeneity of capture probability (sometimes called the equi-catchability assumption).

These assumptions, their effect on the estimates, and what the experimenter can do to minimize their occurrence (while planning and executing the experiment), and how to test for their presence after the data are in, is discussed elsewhere (Seber 1973, Arnason and Baniuk 1978, Chap. 5 and 6; Arnason and Mills 1981). BEFFJOB can be used to assess the significance of bias arising from causes (1) and (2) above, assuming that assumptions (3) and (4) hold. Investigation of bias from more complicated sources (e.g. where (3) or (4) are also violated) can be investigated using simulation methods (Arnason and Baniuk 1978, Chap. 6). Whether a bias will significantly alter an estimate is a function of the precision of the estimate. For example, if the CV of an estimate is 50% it is hardly worth considering the effect of a tag-loss (relative) bias of 5%. At the other extreme, if the CV is 5-10%, a tag-loss bias of 5% in an estimate is probably worth correcting. To quantify the trade-off between level of precision of an estimate and the magnitude of the bias (due to tag loss or any other assumption violation), BEFFJOB reports a quantity called the <u>effective bias</u> of the estimate (if it is biased), which is the ratio of the relative bias to the CV, or equivalently, of the absolute bias to the SE of the estimate. Absolute, relative and effective bias are defined in Table A.4. For an estimate whose label is ´est´, these quantities will be denoted by AB(est), RB(est) and EB(est), respectively. For reasons given by Arnason and Mills (1981), the bias in an estimate due to a particular cause can usually be ignored provided one is reasonably sure that there are no other major sources of bias, that small sample bias is negligible, and EB(est) is less than 50%.

The notation used above, and in the rest of this report, is based on that of Arnason and Mills (1981), modified to be consistent with the computer output (which only prints upper case, latin letters). The notation and definitions for parameters (user-set rates), expected statistics (expected counts of various class sizes of marked animals), and expected estimates (of population size, rate parameters and their SE and CVs) are given in Tables A.1-A.4 of Appendix A. These definition tables are also printed out by BEFFJOB at the beginning of each job run.

A.1 lists the parameter definitions. Note that these are simply denoted by the spelling of the corresponding Greek letter used in Arnason and Mills (1981), with the exception of the birth parameter B(I) (which is not a rate, but a count of the number of unmarked animals that join the catchable population between time I and I+1; as a parameter, it may be set by the user to any integer value. If this value is negative, the program computes the actual number of births internally to compensate for deaths and losses so that the population size remains constant.

A.2 lists the statistics, all denoted by a single letter followed by parentheses denoting sample time, except for the unobservable counts CN(I) (total population size at time I) and CM(I) (total size of the marked pool at time I). B(I) also appears in this table to denote the true number of births (as set by the user or as internally generated).

A.3 defines the notation for the expected values of the estimates, their standard errors and CVs. Note that each SE is prefixed by the letter S, each CV by the letters CV, and that all estimate labels end in H (for the Hat notation used to denote estimates in Arnason and Mills 1981).

A.4 defines absolute, relative and effective bias for the three main estimates (NH, PHIH, BH denoting population size, survival rate and birth estimates, respectively). This table also defines the notation used for estimates corrected for tag-loss bias. The latter are the Case 3 estimates of Arnason and Mills (1981) and are computed by BEFFJOB if the user specifies that tag loss occurs at some point in the experiment. They are useful for showing the loss in precision that will be incurred if it is necessary to correct the estimates for tag-loss bias.

USING BEFFJOB TO PLAN EXPERIMENTS

Methods for planning Petersen-type experiments (Robson and Regier 1964; Jensen 1981) and our methods for K-sample (multiple) mark-recapture experiments require ´guesstimates´ of initial population size, sampling and turnover rates. Given these rates, BEFFJOB can be used to show the precision that will result from the given sampling rates and, in addition, will show the degree of bias introduced by specified tag-loss rates (if THETA(I), the tag-retention rate for marked animals between times I and I+1, is less than 1.0 for at least one sample time I=1,...K).

Since one can show the results of the Jolly-Seber death-only or full model estimates, regardless of whether the B(I) are all 0, BEFFJOB will also show the bias that results when using the death-only estimates when births are in fact occurring.

Initially, before any experimentation is carried out, the user must use guesstimated parameter values that are hypothetical or reasonable approximations to what he feels might pertain to the population of interest. The user can try several runs using different sampling rates until levels of precision in the estimates are achieved that meet his scientific or managerial needs. Once a rough strategy is obtained, a few more runs, altering the initial population size and population parameters, should be tried to determine that the results will be satisfactory over the range of uncertainty he feels appropriate for the parameters. The user does not specify the true population sizes at all times, but only the initial size, CN(1), and the births, survival rate, and returns-on-capture rate. These parameters then determine what the CN(I) will be for I=2,3...K and the user should check that these are reasonable. In addition, the user also specifies the sampling and tag-retention rates but these have no effect on the CN(I) provided losses-on-capture are 0; their effect is on the precision and bias in the estimates. The user specifies guesstimates for sampling rate (i.e. the fraction of the population that will be captured in sample I), not for sample size at time I. If there are no losses on capture, it is quite easy to determine the sampling rates that result from given sample sizes: do a run of BEFFJOB with the guesstimates for the population parameters and with any arbitrarily chosen sampling rates. Now look at the CN(I) values that result from this run and estimate the desired sampling rates from N(I)/CN(I) where N(I) is the size of sample which you <u>wish</u> to take at time I (not the value printed out by BEFFJOB). If losses-on-capture are substantial, a few iterations of this process will be needed (using the new sampling rate each time) as sampling rate now affects the CN(I).

As the sampling experiment progresses, BEFFJOB should be used between each sample, if time permits (and it often does for long-lived animals where samples can be spaced several weeks or even months apart). Estimates from the real data should be formed as the experiment progresses to see if they are consistent with the previous guesstimates. If they are not, the guesstimates should be reformulated and the plan for the remaining samples should be modified accordingly.

This report is not intended as a complete guide on how one forms guesstimated values nor on how best to plan experiments or allocate sampling effort over space and time. These topics are handled in more detail in Arnason and Mills (1981) and in Arnason and Baniuk (1978, Chap. 6). However, there are some important considerations, arising from the meaning of the Jolly-Seber parameters and their influence on precision, that we wish to point out to the user here.

The death rate (= 1 - survival rate) includes <u>all</u> losses from the catchable population between 2 sample times. It therefore includes true death, permanent emigration beyond the trapping area and any behavioural change in the animal that makes it become uncatchable. Survival operates multiplicatively over time. For example, if survival is thought to be 90% per month (0.9) but samples are taken quarterly, then survival over the 3 month period is 0.9x0.9x0.9 = 0.729. Similarly, the tag-retention rate operates multiplicatively over time. Births include all additions, between time I and I+1, of unmarked animals coming into the catchable population and still present at time I+1. Jolly-Seber models take no account of animals that are not subject to capture for at least one sampling period. Births include true births, recruitment of smaller animals up into the catchable population, new immigrants, and animals which become catchable for some behavioural reason (e.g. emergence from dormancy or a burrow). If new entries persist in the population for a period that is long relative to the sampling interval (i.e. are available for capture at many sample times), then births can be considered to be additive (e.g. 100 births per month leads to B(I)=300 for quarterly samples). However, if the death rate is high, some discounting must be done to reduce the births by the number that would not persist until the next sample. There is no requirement, in Jolly-Seber experiments that sample times be equally spaced. However, if the user plans to use unequal spacing of sample times, it is the user's responsibility to ensure that the guesstimated parameters for births, survival and tag loss are adjusted to account for the intervals between samples.

We have already indicated that sampling rate can be derived from expected sample sizes, but it is not always clear how either rates or sizes are related to the sampling effort that one can afford to expend (net days set, or trap-days used). This is a biological matter, on which we can give little general advice, except to point out that sampling rate is generally not a linear function of effort except at very low sampling rates. At higher sampling rates, one can expect diminishing returns (sample sizes) for a given increase in effort. Moreover, effort is not the only determinant of sampling rate: it may vary with the positioning and baiting of traps, and on behavioural properties of the animal over which the experimenter has no control (such as degree and range of activity). There is no serious objection, however, to using BEFFJOB as a guide to planning sample size objectives which are then used as stopping rules in the field. That is, one finds sampling rates, using BEFFJOB, that lead to satisfactory precision. One then uses the sample sizes N(I), as printed by BEFFJOB, as the size of sample beyond which no further sampling need be done at time I. If this is done, it is particularly important to keep updating the plan as the data come in, as described above, and in particular, to check that the M(I) and MH(I) from the real data are consistent with the projected M(I) and CM(I) given by BEFFJOB so that one has some assurance that the plan is being accurately realized.

Experimentation with BEFFJOB over many different parameter sets will give the biologist an increasing sense of confidence in the precision he can achieve. We strongly recommend that the user undertake this experimentation for himself and confirm that the following rules, as to what determines precision of the Jolly-Seber estimates, hold:

(1) Precision is largely determined by two ratios: the ratio of recaptures out of the marked releases in a sample (R(I)/S(I)), and the marked fraction in the sample (M(I)/N(I)). Anything that tends to increase these ratios tends to improve precision: higher sampling rates, more samples (increased K), decreased turnover (birth, death, loss-on-capture) rates. The general strategy for obtaining good precision is to build up a substantial marked fraction quickly, then maintain that fraction for the period of biological interest or until no further gains in precision are achieved.

(2) Precision decreases as the interval between samples increases (because turnover rates per sample increase), but there is a limit to how soon after one sample the next can be taken. If they are taken too close together, problems of unequal catchability may arise (due to lack of random mixing between marked and unmarked animals, or due to trap happiness, trap shyness, or handling induced inactivity in the animals).

(3) Precision is better in larger populations than in smaller ones. The size of the population is often under some degree of experimental control, through choice of sampling area and the degree of stratification of data (e.g. by age, sex, species etc.).

(4) Precision is better if the data can be analysed using the death-only model rather than the full model and if there is no need to correct estimates for tag loss. When there is recruitment, the death-only model may apply to a subset of older year classes.

(5) Estimates for population size (NH(I)) and survival (PHIH(I)) are generally much more precise than estimates of births (BH(I)) in the same experiment. The precision of estimates also varies with I, generally being best for the middle sample times if the full model is used and being best for the earliest sample times if the death-only model is used.

Finally, we repeat that BEFFJOB can be used to determine whether two specific sources of bias are sufficiently large to require corrective action. The first is bias introduced by the presence of births when one uses the death-only model estimates. If the bias is significant, the user must use the full model estimates and suffer the consequent loss in precision or up his sampling rates to compensate. BEFFJOB can be used to determine how large the B(I) can become before corrective action is required. Second, BEFFJOB will demonstrate the bias introduced by specific tag-loss rates. The corrections required, if these are necessary, are described in Arnason and Mills (1981) as the Case 3 estimates, and these corrections will also lead to a loss in precision. Further discussion of the assessment of bias is given in the chapter following the next, but first we will give detailed instructions for running (a set of) BEFFJOB tasks for calculating the bias and precision in estimates given the population and sampling parameters.

RUNNING BEFFJOB

To run BEFFJOB the user must prepare an input deck, made up of a number of tasks. Each task defines the population parameters (number of samples, initial population size, births, survival, and tag-retention rates) and the sampling rates and returns-on-capture rates. Each of these parameters may change at every sampling time. This set of parameters defines a task, or a single complete experiment, and (if there are no errors) will lead to 1 or 2 sets of output on the resulting precision and bias (depending on whether the user specifies that he wishes to see the results from 1 or 2 Jolly-Seber models). Several tasks can be run together as one job. A detailed description of the output from each task, and its interpretation, is given in the next chapter. At the end of the current chapter, we show how the input deck is combined with the source code deck to run the job. We begin with a description of how the parameters for each task are specified in the input deck.

TASK SPECIFICATION

The task is a set of cards containing the data necessary to perform a BEFFJOB analysis. BEFFJOB will input a task, perform the analysis, and will get the next task. When there are no more tasks, the program will stop executing. Each task is independent of all other tasks. A task consists of three types of cards: a Start-of-Task card, one or more Parameter-Value cards and an End-of-Task card. Once the reader is familiar with the material in this section, he may find the summary of a task given in Table C.2 more convenient for preparing his input deck. Table C.2 gives the form of each of the three types of task cards in symbolic form.

Placement of values on cards

All values in the input deck occupy a space of 10 columns on a card. The first value of a card would occupy columns 1-10, the second value would occupy columns 11-20, and so on. The values (integer/real) are right-justified in the 10 column space. Real values have 5 decimal places, the decimal point is in the 5th column of the 10 column space. Integer values have no decimal point. The placement of values in the first 10 columns would therefore appear as follows (e.g. for integer 500 and real 0.5):

```
PLACEMENT          COLUMN    1
FOR TYPE:          1234567890
=============================

Integer                   500
Real               0.50000
```

Errors and error messages

Any deviation from the above placement scheme may cause a severe execution error that terminates the job. These errors have nothing to do with the errors BEFFJOB may generate during the course of execution of each task. They are generated by the FORTRAN input/output (FIOCS) routines. Most commonly, FIOCS errors occur when the user places a real value in a field where BEFFJOB is attempting to read an integer. There are no traps in BEFFJOB for invalid form or placement of input data, but comprehensive checks for invalid values are done and reported to the user. If BEFFJOB, during the execution of a task, encounters an error, it will generate an error message, suspend the current task, and will perform the next task. For example, if you tried to set the survival rate to 1.20000, BEFFJOB would generate the following message

```
               ***ERROR***
INVALID SURVIVAL RATE (PHI) :    1.20000
```

BEFFJOB then continues to scan through the input deck for the End-of-Task card, at which point it prints the message:

```
>>>>>>>>>>  END OF TASK
```

Thus, any subsequent errors in the same task, after the first error is encountered, will NOT be trapped. Execution of the next task (if any) will then begin.

The Start-of-Task card

The Start-of-Task card is the first card of a task. It contains five values, the first three being integers, the last two being real numbers.

COL 1-10: Initial Population Size

This value specifies the total number of animals in the population just before the first sample is taken. It is an integer value.

Error: if value is less than 1 or greater than 999999.

COL 11-20: Number of Sample Times (K)

This value indicates the number of sample times in the experiment. It is an integer value.

Error: if value less than 3 or greater than 20.

COL 21-30: Model Number

This value indicates which of the two models is to be used for forming estimates. It is an integer value.

number - model

1 - full model
2 - death-only model
3 - both models

Error: if value not 1, 2, or 3.

COL 31-40: Cutoff value for M

This value is used as a limit for calculations using M(I). If M(I) is less than the cutoff value, equations using M(I) will have undefined values for answers. It is a real value.

Valid values are from 1.00000 to 10.00000, if less than 1.00000, value set to 1.00000, if greater than 10.00000, value set to 10.00000.

COL 41-50: Cutoff value for R

Used the same way as for the cutoff value for M, except now checking the R(I) instead of the M(I).

Parameter-Value cards

Each Start-of-Task card is followed by up to K (=number of sample times as specified in columns 11-20 of the Start-of-Task card) Parameter-Value cards, each specifying the population and sampling parameters that hold at the I-th sample time (capture rate, returns-on-capture rate) or between samples I and I+1 (births, survival rate, tag-retention rate). The Parameter-Value card also specifies the sample time (I) and if any value of I is missing for I=2,...K; BEFFJOB assumes that the parameters for the missing times are all identical to the set of parameters at the previous time. The first Parameter-Value card must be for I=1 and is obligatory. The user may supply no further Parameter-Value cards (in which case parameters are constant throughout the experiment), or up to K-1 further cards with the I values on successive cards in increasing order (though possibly having some gaps in the integer sequence). The births, survival rate and tag-retention rate for I=K are not used (because they refer to a period after the end of the experiment) but must be supplied if a card with I=K is used. Each Parameter-Value card has six values; the first two are integers, the last four are reals, as follows:

COL 1-10: Sample Time (I)

This value indicates the sample time for the values on the current card. These values are repeated for succeeding sample times up to just before the sample time specified on the next Parameter-Value card. If there are no more Parameter-Value cards, the values are repeated up to the last sample time. The sample time is an integer value.

Error: an 'INVALID TIME VALUE' error will be generated if one of the following is encountered:

- if sample time value of first Parameter-Value card is not equal to 1.

- if sample time value is greater than the number of sample times (K).

- if sample time value of current Parameter-Value card is not greater than the sample time value of the previous Parameter-Value card.

COL 11-20: Births (B)

This value is the number of untagged additions to the animal population that join the population between the current and next sample time. It is an integer value.

Valid values are -1 and from 0 to 999999. If -1, BEFFJOB calculates a birth value so that the population size at the next sample time will be the same as that for the current sample time.

Error: If value less than -1 or greater than 999999.

COL 21-30: Survival Rate (PHI)

This value is the survival rate of an animal from the specified sample time until the next sample time. It is a real value.

Error: If value is less than or equal to 0.00000, or value is greater than 1.00000.

COL 31-40: Capture Rate (PI)

This value is the probability of capture for every animal alive in the population at the specified time, or equivalently, it is the proportion of the total population that will be taken in the sample (N(I)/CN(I)). It is a real value.

Error: If value is less than or equal to 0.00000, or value is greater than 1.00000.

COL 41-50: Returns-on-Capture Rate (ETA)

This value gives the proportion of animals, out of those caught at the specified sample time, that are returned to the population after the specified sample time (i.e. are not losses-on-capture). It is a real value.

Error: If value is less than 0.00000, or value is greater than 1.00000.

COL 51-60: Tag-Retention Rate (THETA)

This value is the proportion, out of the marked animals alive just after the specified sample is taken, that will retain their tags until the next sample is taken (given that they survive). Equivalently, it is the probability that a marked animal retains its tag until the sample time following the current one. It is a real value.

Error: If value is less than or equal to 0.00000 or greater than 1.00000.

End-of-Task Card

The Parameter-Value card(s) are followed by a single End-of-Task card. It is used as a filler between tasks. It consists of six values, two integers followed by four reals. All values are zero. With some FORTRAN compilers, a blank card is acceptable.

EXAMPLE OF AN INPUT DECK

In Table C.1 we show 17 cards that make up the input deck for running the tasks whose output is discussed in the next section. The reader should compare the actual tasks of Table C.1 with the general form in Table C.2. Table C.2 will be more useful for preparation of input data once the user is familiar with the material in this section. The input cards for the first task (cards 1-3) and last task (cards 13-17) of Table C.1 specify the following experiments:

First Task

(Table C.1, cards 1-3)

Initial population size: 500
Number of sample times: 5
Model: Both models
Cutoff values: m - 3.0
r - 3.0

For all sample times:
Births: 0
Survival Rate: 0.9
Capture Rate: 0.3
Returns-on-Capture Rate: 1.0
Tag-Retention Rate: 1.0

Last Task

(Table C.1, cards 13-17)

Initial population size: 250
Number of sample times: 8
Model: Full Model
Cutoff Values: m - 1.0
r - 2.0

For sample times 1 - 2:
Births: 100
Survival Rate: 0.75
Capture Rate: 0.25
Returns-on-Capture Rate: 1.0
Tag-Retention Rate: 0.95

For sample time 3:
Births: 250
Survival Rate: 0.9
Capture Rate: 0.4
Returns-on-Capture Rate: 0.8
Tag-Retention Rate: 1.0

For sample times 4 - 8:
Births: chosen by BEFFJOB to keep population at current size (CN(4)).
Survival Rate: 0.8
Capture Rate: 0.2
Returns-on-Capture Rate: 0.9
Tag-Retention Rate: 0.9

JOB RUNS

Once an input deck has been punched, the input deck, together with the BEFFJOB source and a few JCL (Job Control Language) cards are put together in the order shown in Table C.3 to form the job deck. This complete job deck is then submitted for execution.

The cards shown in Table C.3 that begin with // are the JCL cards. The JCL shown is for IBM or AMDAHL computers running under MVS, MVT or similar operating systems. For other operating systems (e.g. DOS) or for other computers, your local program advisor or computer centre can show you how to set up the program and input decks. Just make sure they are aware that <u>the input deck must be read from UNIT 14</u>. BEFFJOB is written in ANSI standard FORTRAN and should be executable on any machine that has a FORTRAN or WATFIV compiler. It requires a minimum of 64K of memory at run time and requires no ancillary files or library programs (other than the compiler and the standard FORTLIB functions). If you plan to run this job frequently, it may be worthwhile generating the object deck or load module to save reading and compiling the program at every run. Your computer centre or program advisor can set this up for you.

The BEFFJOB source program consists of approximately 900 cards. A full listing of the source is given in Appendix D.

DISCUSSION OF BEFFJOB OUTPUT

The output from BEFFJOB begins with a header (title) page, followed by the four definition tables shown in APPENDIX A. Then for each task (provided no errors were encountered in the input cards for the task), the output consists of 3 or more tables, depending on the parameters specified. The first two of these are always:

THE PARAMETER TABLE

This table gives the parameters, as chosen by the user, that define the experiment. See Table A.1 for definitions of the parameters and Table B.1(a) for a numeric example (corresponding to the first task shown in the input deck of Table C.1). The Parameter Table also gives one or two derived parameters that were not set by the user. These are CHI(I), the recapture probability, and PSI(I), the observable recapture probability for marked animals in the presence of tag loss. The latter is reported only if at least one THETA value is less than 1.0, since otherwise PSI and CHI are identical. The formulae for computing these parameters from the other rate parameters is given in Arnason and Mills (1981, Appendix A). They are useful in that they show what recapture rates can be expected (since R(I) = S(I) * PSI(I)) and a comparison of PSI with CHI shows the reduction in recapture rate due to tag-loss. The user should examine this table carefully to ensure that the parameters read from the input deck are the ones the user intended to use, and that the missing sample time values (if any) have been propagated out as intended. The table will always have exactly K rows, one for each sample time.

THE STATISTICS TABLE

This table gives the expected values of the observable statistics (N(I), M(I), S(I), R(I), Z(I), ZP(I); see Table A.2 for definitions, and Table B.1(a) for a numeric example). This table also includes values for the unobservable statistics CN(I), the population size at time I, CM(I), the total number of marked animals alive at I, and B(I), the births in (I,I+1). These values are determined from the initial population size CN(1) and the rate parameters using formulae given in Appendix A of Arnason and Mills (1981). The user should check that the expected statistics are biologically reasonable (for CN(I) and B(I) in particular), and experimentally feasible (for N(I), the sample size). If the B(I) parameter was set to -1 by the user, the B(I) shown in this table will be derived values (not necessarily whole numbers) calculated to stabilize the population size, (i.e. so that CN(I+1)=CN(I)), given the current loss rates.

In a real experiment, the statistics will be whole numbers (integers) since each is a count over distinct animals. The expected values are reals (not necessarily whole numbers) representing the theoretical average value of the count (over an infinite set of replications of the experiment with the same parameter values). The actual counts, N(I), M(I), R(I), S(I) and Z(I) are random variables, and in a real experiment, will vary about the expected values reported here, even if the parameters are, in reality, identical to those used to generate the Statistics Table. The most important parameters for determining precision are N, M, S and R and these appear to vary according to the Poisson distribution whose mean is the expected value, provided these expected values are not too small (>5). In the Poisson distribution, the theoretical variance equals the mean, so one could expect the count in an actual experiment to fall within MEAN ± 2.0 * SQRT(MEAN) most of the time (with about 95% probability). For example, in Table B.1(a), R(1) = 92.22 whose square root is around 10, so one would expect, in an actual experiment with these parameters, an R(1) value in the range (72,112) most of the time.

The Parameter and Statistics Table output depends only on CN(1), K, and the user-set parameter rates and is not affected by the user's choice of analysis (full model, death-only model, or both). Thus if both models are chosen for analysis, these tables are given only once.

THE ANALYSIS TABLES (EXPECTED ESTIMATES AND BIAS)

Following the first two tables, BEFFJOB prints either one or two sets of model analysis tables, depending on whether the user has specified a single analysis or both analyses. Each model analysis set consists of one or two tables. The first always shows the expected values and expected precision of the estimates, given the parameters and the analysis method. This table is labelled the EXPECTED ESTIMATES TABLE. See Table A.3 for notation and definitions, Table B.1(b) for a numeric example, and Arnason and Mills (1981, section 2) for a description of how they are computed from the parameter and statistics values.

The columns for the <u>parameter</u> estimates (NH(I), PHIH(I), and BH(I)) should be compared to their true or theoretical values as given in the first two tables (i.e. with CN(I) in the Statistics Table, PHI(I) in the Parameter Table, and B(I) in the Statistics Table, respectively). Where there are no sources of bias, these values will coincide exactly with their theoretical values, except at some of the sample times at the beginning or end of the experiment (these unidentifiable estimates will be 0.0; unidentifiability is discussed later in this section). If there is bias, the expected bias is the difference between the expected estimate and its theoretical value. Various bias properties, such as this, will be given in a second table, labelled the BIAS TABLE, which will be printed immediately after the Expected Estimates Table. Beside each column of parameter estimates in the Expected Estimates Table is a column for the expected SE (standard error) of the estimate and for the expected CV (coefficient of variation, expressed as a percent; see Table A.3). These columns give the user the information on the precision he can expect from a given experiment. The precision can be looked at in two ways: either as a measure of the variation in the estimate about its expected value, or as an indication of the expected width of the confidence interval for the parameter. For example, in the death-only analysis in Table B.1(b), at time 1 we have NH(1) = 500, SNH(1) = 27.04 and CVNH(1) = 5.41. This means that if the parameters used are reasonably realistic, the user can expect his estimate to be within ±2 * SE (i.e. within ± 54 in this case) of the true value (with 95% probability). Alternately, the width of the 95% confidence interval for the estimate, relative to the expected value of the estimate, is about 4 * CV (or about 22% of CN(1) in this case).

The user should keep in mind that, in an actual experiment, both the parameter estimate (say, NH) and its SE (say, SNH) are random variables, and that they have strong positive correlation. Thus if, by chance, the estimate happens to be above its true value, the confidence interval will probably be wider than expected; if below, the confidence interval will tend to be narrower than expected. As we have seen above, BEFFJOB provides information on how great the deviation of a parameter estimate like NH might be from its true value, but it provides none on how much an actual SE, like SNH, might deviate from the expected value reported by BEFFJOB. This can, however, be determined from simulations (Arnason and Baniuk 1978, Chap. 6). In situations where precision is marginal (say CV > 20%), a great deal more insight can be obtained from such simulations since POPAN reports the variance as well as the average value of all statistics and estimates. BEFFJOB reports only the (theoretical) average values which are nevertheless adequate for most situations where reasonably good precision in the parameter estimates is achieved.

Certain of the estimates in the Expected Estimates Table will be reported as 0.00 for values near the beginning and end of the experiment. These values are unidentifiable; that is, the mark-recapture data does not contain sufficient information to form unique estimates for these parameters. Where the parameter estimate is unidentifiable, the corresponding SE and CV are also set to zero as are corresponding entries in the Bias Table (if there is one). The range of identifiable estimates in a K-sample experiment, is as follows:

```
IDENTIFIABLE ESTIMATES

     FULL MODEL
     ==== =====

NH(I)    I =   2,...K-2,K-1
PHIH(I)  I = 1,2,...K-2
BH(I)    I =   2,...K-2

   DEATH-ONLY MODEL
   ===== ==== =====

NH(I)    I = 1,2,...K-2,K-1
PHIH(I)  I = 1,2,...K-2
```

If 0´s occur within this range, it is because:

(a) M(I) or R(I) is below its cut-off value (check the Statistics Table for this).

(b) The expected estimate is indeed 0.00 (in which case, the corresponding SE will generally be non-zero and the CV will be undefined; see for example, BH(2) and BH(3) in Table B.1(b)).

When certain statistics or expected estimates are 0.00, it may make other estimates, in this or the Bias Table, undefined (due to zero divides). These undefined values will be indicated by an entry of **********. In rare circumstances, this entry may also result when the estimate is too large to print in a 10 character field. This will only happen if BH(I), NH(I) or their standard errors exceed 9999999.99.

The Expected Estimates Table may be followed by a second table, labelled the BIAS TABLE, if the parameters chosen lead to bias in any of the estimates. The notation and definitions for this table are given in Table A.4 and Table B.2(b) shows a numeric example. As mentioned in the introduction, BEFFJOB accounts for two sources of bias (closure or tag loss) each of which may be present or absent, so there are four output analysis sets to consider. Each of these four cases has been run using both models since the form of the analysis output also depends on which analysis is specified. The four cases of output are shown in Tables B.1-B.4, each table giving the results for each of the two analyses (models). The user´s output will always be identical in form (number and labelling of tables) to one of these four cases. For simplicity, we have generated each case using parameters that are constant throughout an experiment with K=5, but the form would be the same for time-varying parmeters and any other value of K. We will not discuss the Parameter and Statistics Tables (shown in part (a) of each task output in Appendix B) individually. A single task with time varying parameters is shown in Table B.5 and will be discussed at the end of this section.

<u>Tasks with no births and no tag loss (Table B.1)</u>

This experiment is one in which all assumptions are satisfied (no bias) regardless of which analysis is chosen. However, the full model is more general (estimates extra parameters, the BH(I)) than necessary and so will give less precise estimates than the death-only model. Note that, for both models, the expected NH(I) and PHIH(I) coincide exactly with the true values (CN(I) and PHI(I) in Table B.1(a)) except where estimates are not identifiable.

The output from the full model shows the loss in precision incurred by use of too general a model (compare the SNH and CVNH to those for the death-only model, and compare also for SPHIH and CVPHIH). Note that the loss in precision is great for NH but is inconsequential for PHIH. This appears to be generally true. Note that quite large birth estimates are likely to occur at times 2 and 3: e.g. BH(2) might be as large as 110, but in such cases, SBH will also be large (due to the large CV and the correlation between BH and SBH). Inadmissible estimates (BH < 0) are also highly likely.

The output values from the death-only model are the results needed for assessing precision in experiments where one can be reasonably sure that birth, immigration and recruitment can be excluded (e.g. for fish in a closed lake where new recruits can be recognised by their size or age). The precision of the NH is quite good, whereas that for PHIH may not be good enough for some purposes. For example SPHIH(2) is around 0.1 which means that one would probably be unable to exclude the hypothesis that PHI(2) = p for any p-value in the range (0.7-1.0). Values outside this range could probably be rejected but that might not be good enough for the tests and comparisons of interest to the biologist (e.g. for comparison with survival in another age-, sex-, or species-group of fish).

Tasks with no births and some tag loss (Table B.2)

The full model estimates (in B.2(b)) have, as in the previous case, somewhat inflated SE over the death-only estimates (in B.2(c)), due to their allowance for (non-existent) births. It is nevertheless important to print these results out even if one knows there are no births because the precision of the estimates corrected for tag-loss bias are only given under the full model output. Notice in B.2(b) that the full model estimates of NH are not biased by tag loss and so are not given in the Bias Table. PHIH and BH are biased by tag loss (significantly so for these parameters) and the degree of bias is summarized for each of these estimates in the Bias Table. The negative signs for AB, RB and EB of PHIH indicate that PHIH is an under-estimate of the true value. The fact that the magnitude of EB exceeds 50% (ignoring its sign) indicates significant bias. The Case 3 corrected values for PHIH would need to be used. Their properties are summarized in the first triplet of columns of the Bias Table. Comparing these to the corresponding triplet of columns for PHIH in the Expected Estimates Table, we see that the corrections have no effect on the relative precision (CV) but that the confidence interval constructed from PHIH3 will be wider (SPHIH3 > SPHIH). Note that the bias in BH and SBH is sufficient to produce a strong likelihood that BH, in an actual experiment, will differ significantly from 0 (since the CV, at least for BH(3), is approaching 50%). Thus the BH estimates are, in this case, an unreliable guide to whether births are occurring.

The death-only output in B.2(c) shows bias due to tag loss in both NH and PHIH and so a bias table is given with entries for both estimates. Where the EB is well below 50% (in magnitude) one can still use the death-only estimates (Case 1 of Arnason and Mills 1981). In this experiment, however, all estimates are effectively biased and corrections would have to be done. Correction of the NH means using the full model estimates which can be seen (in B.2(b)) to be less precise and to provide no estimate for CN(1). The correction of the PHIH means using the Case 3 estimates (shown in the Bias Table of B.2(b)): in this experiment, this results in very little loss of precision, but an estimate of the tag-retention rate, THETA, would be needed. This means the experimenter would need to use some form of double tagging in his experiment. If no estimate of THETA is available, the Case 2 corrections of Arnason and Mills (1981) can be used, but these will be less precise than the PHIH3. BEFFJOB does not compute precision for the Case 2 corrected PHIH. The (expected) variance of the Case 2 PHIH can be computed by evaluating equation 3.8 of Arnason and Mills (1981) using the NH and SNH values in Table B.2(b), (since the Case 2 NH are the same as for the full model) and the (expected) statistics given in Table B.2(a).

The results of this experiment might also be compared with those in Table B.1 which gives results for the equivalent experiment with no tag loss. This comparison shows that tag loss has produced a loss of precision even in those estimates which are not biased by tag-loss (the full model NH).

Tasks with some births and no tag loss (Table B.3)

The full model estimates have no bias so only an Expected Estimates Table is given. This is the output of interest for judging precision in experiments where births are known to occur but other assumption failures, including tag loss, are negligible. Note for the experiment shown in Table B.3(b) that the NH and PHIH have reasonably good precision while the BH will have unsatisfactory precision. More often than not, the BH estimates will indicate births not differing significantly from 0 and inadmissible estimates (<0) are highly likely.

The death-only estimates will be biased by failure of the closure assumption. The bias is summarized in a Bias Table following the Expected Estimates Table. In this experiment, there is effective bias in the NH but not in the PHIH. However, the PHIH are no more precise than those for the full model, so (at least in this case) if there is any chance of births occurring, it is safest to use the full model estimates.

Tasks with some births and some tag loss (Table B.4)

The full model estimates for PHIH and BH are biased by tag loss and this bias will be summarized in a Bias Table. In the example task (B.4(b)), the bias is significant for all BH and PHIH. These can be corrected only if estimates of THETA are available. The precision of the corrected PHIH is shown by the PHIH3 in the Bias Table. The BH can also be corrected:

```
BH3(I) = BH(I) -
[(NH(I)-N(I)+S(I))*PHIH3(I)*(1-THETA(I))]
```

but the precision, while poorer than that of BH(I), is unknown. The NH(I) are unbiased but can be seen, by comparison with those for the full model in Table B.3(b), to be less precise than in an equivalent experiment with no tag loss.

The death-only estimates in this case will be biased both by tag loss and by the failure of the closure assumption. The two sources of bias both produce over-estimates in the NH and so re-inforce one another to produce even greater bias than when either source acts alone. (Compare with the Bias Table entries for NH in Tables B.2(c) and B.3(b)). For the PHIH, the two sources of bias tend to cancel one another out. In the experiment shown in Table B.4(c), this results in estimates which are not effectively biased. In practice however, an experimenter would not want to count on two mechanisms, neither wholly under his control, to balance out the bias. In any case, the PHIH values are no more precise than the Case 3 corrected PHIH3 in Table B.4(b).

Tasks with time-varying parameters (Table B.5)

We show the Parameter and Statistics Tables (only) for a somewhat longer experiment (K=8) in which the parameters were set by the user to values that change with each sample time or interval. It also shows the effect of setting some of the B(I) birth parameters to -1 to obtain a stabilized population. The actual births required to stabilize the population at CN(4)=511.34 can be seen from the B(I) column in the Statistics Table. Since this experiment has both births and tag loss, the form of the analysis output will be the same as in Table B.4(b), but we have not listed the output in this manual.

REFERENCES

ARNASON, A.N., and L. BANIUK. 1978. POPAN-2: a data maintenance and analysis sytem for mark-recapture data. Charles Babbage Research Centre, Box 370, St. Pierre, Man. Canada. 269 p.

-------. 1980. A computer system for mark-recapture analysis of open populations. J. Wildl. Manage. 44:325-332.

ARNASON, A.N., and K.H. MILLS. 1981. Bias and loss of precision due to tag loss in Jolly-Seber estimates for mark-recapture experiments. Can. J. Fish. Aquat. Sci. 38:1077-1095.

CORMACK, R.M. 1968. The statistics of capture-recapture methods. Oceanogr. Mar. Biol. Annu. Rev. 6:455-506.

JENSEN, A.L. 1981. Sample sizes for single mark and single recapture experiments. Trans. Am. Fish. Soc. 110:455-458.

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

ROBSON, D.S., and H. A. REGIER. 1964. Sample size in Petersen mark-recapture experiments. Trans. Am. Fish. Soc. 93: 215-226.

SEBER, G.A.F. 1965. A note on the multiple-recapture census. Biometrika 52: 249-259.

-------. 1973. The estimation of animal abundance and related parameters. Charles Griffin and Company Ltd., London. 506 p.

APPENDIX TABLES

TABLE A.1

PARAMETER TABLE DEFINITIONS

LABEL	DESCRIPTION
B(I)	NUMBER OF ADDITIONS TO POPULATION AFTER TIME (I) THAT ARE STILL ALIVE AT TIME (I+1). IF INPUT VALUE IS -1, B(I) IS CALCULATED SO THAT THE POPULATION SIZE AT TIME (I+1) IS THE SAME AS AT TIME (I). (I.E. BIRTHS COMPENSATE FOR LOSSES)
PHI(I)	PROBABILITY THAT AN ANIMAL ALIVE JUST AFTER TIME (I), IS ALIVE AT TIME (I+1), I.E. THE SURVIVAL RATE AT TIME (I).
PI(I)	PROBABILITY THAT AN ANIMAL IS CAPTURED AT TIME (I).
ETA(I)	PROBABILITY THAT AN ANIMAL CAPTURED AT TIME (I) IS RETURNED TO THE POPULATION; I.E. IS NOT A LOSS-ON-CAPTURE.
THETA(I)	PROBABILITY THAT A LIVING TAGGED ANIMAL AT TIME(I) RETAINS ITS TAG AT TIME (I+1).
PSI(I)	PROBABILITY THAT A TAGGED ANIMAL, ALIVE JUST AFTER TIME (I), IS SEEN AGAIN AT LEAST ONCE WHILE STILL RETAINING ITS TAG.
CHI(I)	PROBABILITY THAT AN ANIMAL, ALIVE JUST AFTER TIME (I), IS SEEN AGAIN AT LEAST ONCE.

TABLE A.2

STATISTICS TABLE DEFINITIONS

LABEL	DESCRIPTION
CN(I)	SIZE OF POPULATION JUST BEFORE TIME (I).
N(I)	SIZE OF SAMPLE TAKEN AT TIME (I).
CM(I)	NUMBER OF MARKED ANIMALS ALIVE JUST BEFORE TIME (I).
M(I)	NUMBER OF N(I) THAT ARE MARKED.
B(I)	NUMBER OF ADDITIONS (BIRTHS) TO POPULATION BETWEEN TIMES (I) AND (I+1).
S(I)	NUMBER OF ANIMALS, CAPTURED AT TIME (I), THAT ARE RETURNED TO THE POPULATION.
R(I)	NUMBER OUT OF S(I) THAT ARE RECAPTURED ONE OR MORE TIMES AFTER TIME (I) (AND RETAIN THEIR TAG, AT LEAST UNTIL THE FIRST RECAPTURE).
Z(I)	NUMBER OF ANIMALS SEEN BEFORE TIME (I), NOT SEEN AT (I), BUT SEEN AGAIN AT LEAST ONCE (WHILE RETAINING ITS TAG) AFTER (I).
ZP(I)	NUMBER OF ANIMALS NOT SEEN AT TIME (I), BUT ARE SEEN AT LEAST ONCE AFTER TIME(I).

TABLE A.3

EXPECTED ESTIMATES TABLE DEFINITIONS

LABEL	DESCRIPTION
NH(I)	EXPECTED ESTIMATE OF CN(I)
SNH(I)	STANDARD ERROR OF NH(I)
CVNH(I)	COEFFICIENT OF VARIATION OF NH(I), EXPRESSED AS A PERCENTAGE I.E. 100 * SNH(I) / NH(I).
PHIH(I)	EXPECTED ESTIMATE OF PHI(I)
SPHIH(I)	STANDARD ERROR OF PHIH(I)
CVPHIH(I)	COEFFICIENT OF VARIATION OF PHIH(I), EXPRESSED AS A PERCENTAGE
BH(I)	EXPECTED ESTIMATE OF B(I)
SBH(I)	STANDARD ERROR OF BH(I)
CVBH(I)	COEFFICIENT OF VARIATION OF BH(I), EXPRESSED AS A PERCENTAGE

NOTE : A TABLE ENTRY CONTAINING '**********' IMPLIES THAT THE VALUE WAS UNDEFINED OR TOO LARGE FOR PRINTING.

A TABLE ENTRY OF 0.0 INDICATES AN ESTIMATE WHICH IS NOT IDENTIFIABLE OR R(I) OF M(I) IS BELOW ITS CUT-OFF VALUE.

TABLE A.4

BIAS TABLE DEFINITIONS

LABEL	DESCRIPTION
PHIH3(I)	CASE 3 CORRECTED PHIH(I): PHIH3(I) = PHIH(I) / THETA(I)
SPHIH3(I)	CASE 3 CORRECTED SPHIH(I): SPHIH3(I) = SPHIH(I) / THETA(I)
CVPHIH3(I)	CASE 3 CORRECTED CVPHIH(I): CVPHIH3(I) = CVPHIH(I)
AB(NH)	ABSOLUTE BIAS OF NH(I) I.E. NH(I) - CN(I).
RB(NH)	RELATIVE BIAS OF NH(I), EXPRESSED AS A PERCENTAGE I.E. 100 * AB(NH) / CN(I).
EB(NH)	EFFECTIVE BIAS OF NH(I), EXPESSED AS A PERCENTAGE I.E. 100 * AB(NH) / SNH(I).
AB(PHIH)	ABSOLUTE BIAS OF PHIH(I)
RB(PHIH)	RELATIVE BIAS OF PHIH(I), EXPRESSED AS A PERCENTAGE
EB(PHIH)	EFFECTIVE BIAS OF PHIH(I), EXPRESSED AS A PERCENTAGE
AB(BH)	ABSOLUTE BIAS OF BH(I)
RB(BH)	RELATIVE BIAS OF BH(I), EXPRESSED AS A PERCENTAGE
EB(BH)	EFFECTIVE BIAS OF BH(I), EXPRESSED AS A PERCENTAGE

TABLE B.1 (a)

Task output with: No births, No tag-loss

>>>>>>>>>> START OF TASK

MODEL(S) SELECTED:

FULL MODEL
DEATH-ONLY MODEL

ORIG. POPULATION: 500
SAMPLE TIMES: 5
CUTOFF VALUES:
M: 3.00000
R: 3.00000

PARAMETER TABLE
========= =====

I	B(I)	PHI(I)	PI(I)	ETA(I)	THETA(I)	CHI(I)
1	0.00	0.90000	0.30000	1.00000	1.00000	0.61478
2	0.00	0.90000	0.30000	1.00000	1.00000	0.54726
3	0.00	0.90000	0.30000	1.00000	1.00000	0.44010
4	0.00	0.90000	0.30000	1.00000	1.00000	0.27000
5	0.00	0.90000	0.30000	1.00000	1.00000	0.00000

STATISTICS TABLE
========== =====

I	CN(I)	N(I)	CM(I)	M(I)	B(I)	S(I)	R(I)	Z(I)	ZP(I)
1	500.00	150.00	0.00	0.00	0.00	150.00	92.22	0.00	215.17
2	450.00	135.00	135.00	40.50	0.00	135.00	73.88	51.72	172.39
3	405.00	121.50	206.55	61.97	0.00	121.50	53.47	63.63	124.77
4	364.50	109.35	239.48	71.84	0.00	109.35	29.52	45.26	68.89
5	328.05	98.41	249.29	74.79	0.00	98.41	0.00	0.00	0.00

TABLE B.1 (b)

FULL MODEL

EXPECTED ESTIMATES TABLE

I	NH(I)	SNH(I)	CVNH(I)	PHIH(I)	SPHIH(I)	CVPHIH(I)	BH(I)	SBH(I)	CVBH(I)
1	0.00	0.00	0.00	0.90000	0.08066	8.96271	0.00	0.00	0.00
2	450.00	62.66	13.93	0.90000	0.10042	11.15782	-0.00	55.98	**********
3	405.00	49.57	12.24	0.90000	0.14547	16.16343	-0.00	35.51	**********
4	364.50	55.71	15.28	0.00000	0.00000	0.00000	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00000	0.00000	0.00000	0.00	0.00	0.00

DEATH-ONLY MODEL

EXPECTED ESTIMATES TABLE

I	NH(I)	SNH(I)	CVNH(I)	PHIH(I)	SPHIH(I)	CVPHIH(I)
1	500.00	27.04	5.41	0.90000	0.07761	8.62321
2	450.00	29.47	6.55	0.90000	0.09804	10.89316
3	405.00	34.67	8.56	0.90000	0.14205	15.78341
4	364.50	47.95	13.16	0.00000	0.00000	0.00000
5	0.00	0.00	0.00	0.00000	0.00000	0.00000

>>>>>>>>>> END OF TASK

TABLE B.2 (a)

Task output with: no births, some tag-loss

```
>>>>>>>>>>  START OF TASK

  MODEL(S) SELECTED:

         FULL MODEL
         DEATH-ONLY MODEL

  ORIG. POPULATION:        500
  SAMPLE TIMES:              5
  CUTOFF VALUES:
            M:    3.00000
            R:    3.00000
```

PARAMETER TABLE
========= =====

I	B(I)	PHI(I)	PI(I)	ETA(I)	THETA(I)	PSI(I)	CHI(I)
1	0.00	0.90000	0.30000	1.00000	0.80000	0.40738	0.61478
2	0.00	0.90000	0.30000	1.00000	0.80000	0.37973	0.54726
3	0.00	0.90000	0.30000	1.00000	0.80000	0.32486	0.44010
4	0.00	0.90000	0.30000	1.00000	0.80000	0.21600	0.27000
5	0.00	0.90000	0.30000	1.00000	0.80000	0.00000	0.00000

STATISTICS TABLE
========== =====

I	CN(I)	N(I)	CM(I)	M(I)	B(I)	S(I)	R(I)	Z(I)	ZP(I)
1	500.00	150.00	0.00	0.00	0.00	150.00	61.11	0.00	288.80
2	450.00	135.00	108.00	32.40	0.00	135.00	51.26	28.71	214.91
3	405.00	121.50	151.63	45.49	0.00	121.50	39.47	34.48	144.67
4	364.50	109.35	163.90	49.17	0.00	109.35	23.62	24.78	74.80
5	328.05	98.41	161.34	48.40	0.00	98.41	0.00	0.00	0.00

TABLE B.2 (b)

FULL MODEL
==== =====

EXPECTED ESTIMATES TABLE
======== ========= =====

I	NH(I)	SNH(I)	CVNH(I)	PHIH(I)	SPHIH(I)	CVPHIH(I)	BH(I)	SBH(I)	CVBH(I)
1	0.00	0.00	0.00	0.72000	0.09953	13.82334	0.00	0.00	0.00
2	450.00	81.67	18.15	0.72000	0.11193	15.54516	81.00	62.73	77.45
3	405.00	67.29	16.61	0.72000	0.14659	20.35919	72.90	47.56	65.25
4	364.50	72.54	19.90	0.00000	0.00000	0.00000	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00000	0.00000	0.00000	0.00	0.00	0.00

BIAS TABLE
==== =====

	CASE 3 CORRECTED PHI GIVEN THETA			EXPECTED BIAS EFFECTS ON PHIH			EXPECTED BIAS EFFECTS ON BH		
I	PHIH3(I)	SPHIH3(I)	CVPHIH3(I)	AB(PHIH)	RB(PHIH)	EB(PHIH)	AB(BH)	RB(BH)	EB(BH)
1	0.90000	0.12441	13.82334	-0.18000	-20.00000	-180.85347	0.00	0.00	0.00
2	0.90000	0.13991	15.54516	-0.18000	-20.00000	-160.82173	81.00	**********	129.12
3	0.90000	0.18323	20.35919	-0.18000	-20.00000	-122.79466	72.90	**********	153.27
4	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00	0.00	0.00
5	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00	0.00	0.00

TABLE B.2 (c)

DEATH-ONLY MODEL
========== =====

EXPECTED ESTIMATES TABLE
======== ========= =====

I	NH(I)	SNH(I)	CVNH(I)	PHIH(I)	SPHIH(I)	CVPHIH(I)
1	858.92	76.84	8.95	0.81609	0.10960	13.42977
2	700.95	69.28	9.88	0.80867	0.12409	15.34441
3	566.84	65.71	11.59	0.80380	0.15893	19.77256
4	455.62	72.37	15.88	0.00000	0.00000	0.00000
5	0.00	0.00	0.00	0.00000	0.00000	0.00000

BIAS TABLE
==== =====

	EXPECTED BIAS EFFECTS ON NH			EXPECTED BIAS EFFECTS ON PHIH		
I	AB(NH)	RB(NH)	EB(NH)	AB(PHIH)	RB(PHIH)	EB(PHIH)
1	358.92	71.78	467.07	-0.08391	-9.32351	-76.56246
2	250.95	55.77	362.22	-0.09133	-10.14815	-73.60538
3	161.84	39.96	246.29	-0.09620	-10.68884	-60.52875
4	91.12	25.00	125.92	0.00000	0.00000	0.00000
5	0.00	0.00	0.00	0.00000	0.00000	0.00000

>>>>>>>>>> END OF TASK

TABLE B.3 (a)

Task output with: Some births, No tag-loss

```
>>>>>>>>>>  START OF TASK

 MODEL(S) SELECTED:

        FULL MODEL
        DEATH-ONLY MODEL

 ORIG. POPULATION:         500
 SAMPLE TIMES:               5
 CUTOFF VALUES:
           M:     3.00000
           R:     3.00000
```

PARAMETER TABLE
========= =====

I	B(I)	PHI(I)	PI(I)	ETA(I)	THETA(I)	CHI(I)
1	50.00	0.90000	0.30000	1.00000	1.00000	0.61478
2	50.00	0.90000	0.30000	1.00000	1.00000	0.54726
3	50.00	0.90000	0.30000	1.00000	1.00000	0.44010
4	50.00	0.90000	0.30000	1.00000	1.00000	0.27000
5	50.00	0.90000	0.30000	1.00000	1.00000	0.00000

STATISTICS TABLE
========== =====

I	CN(I)	N(I)	CM(I)	M(I)	B(I)	S(I)	R(I)	Z(I)	ZP(I)
1	500.00	150.00	0.00	0.00	50.00	150.00	92.22	0.00	319.18
2	500.00	150.00	135.00	40.50	50.00	150.00	82.09	51.72	261.40
3	500.00	150.00	220.05	66.02	50.00	150.00	66.01	67.79	193.48
4	500.00	150.00	273.63	82.09	50.00	150.00	40.50	51.72	109.50
5	500.00	150.00	307.39	92.22	50.00	150.00	0.00	0.00	0.00

TABLE B.3 (b)

FULL MODEL

EXPECTED ESTIMATES TABLE

I	NH(I)	SNH(I)	CVNH(I)	PHIH(I)	SPHIH(I)	CVPHIH(I)	BH(I)	SBH(I)	CVBH(I)
1	0.00	0.00	0.00	0.90000	0.07914	8.79360	0.00	0.00	0.00
2	500.00	70.02	14.00	0.90000	0.09348	10.38700	50.00	66.26	132.53
3	500.00	59.46	11.89	0.90000	0.12858	14.28675	50.00	48.61	97.21
4	500.00	70.02	14.00	0.00000	0.00000	0.00000	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00000	0.00000	0.00000	0.00	0.00	0.00

DEATH-ONLY MODEL

EXPECTED ESTIMATES TABLE

I	NH(I)	SNH(I)	CVNH(I)	PHIH(I)	SPHIH(I)	CVPHIH(I)
1	669.18	38.10	5.69	0.93793	0.08143	8.68160
2	627.64	40.66	6.48	0.93945	0.09683	10.30708
3	589.64	46.89	7.95	0.94220	0.13186	13.99495
4	555.56	63.73	11.47	0.00000	0.00000	0.00000
5	0.00	0.00	0.00	0.00000	0.00000	0.00000

BIAS TABLE

	EXPECTED BIAS EFFECTS ON NH			EXPECTED BIAS EFFECTS ON PHIH		
I	AB(NH)	RB(NH)	EB(NH)	AB(PHIH)	RB(PHIH)	EB(PHIH)
1	169.18	33.84	444.08	0.03793	4.21405	46.57726
2	127.64	25.53	313.91	0.03945	4.38349	40.74294
3	89.64	17.93	191.17	0.04220	4.68850	32.00103
4	55.56	11.11	87.18	0.00000	0.00000	0.00000
5	0.00	0.00	0.00	0.00000	0.00000	0.00000

>>>>>>>>>> END OF TASK

TABLE B.4 (a)

Task output with: Some births, Some tag-loss

```
>>>>>>>>>> START OF TASK

 MODEL(S) SELECTED:

        FULL MODEL
        DEATH-ONLY MODEL

 ORIG. POPULATION:        500
 SAMPLE TIMES:              5
 CUTOFF VALUES:
           M:     3.00000
           R:     3.00000
```

PARAMETER TABLE
========= =====

I	B(I)	PHI(I)	PI(I)	ETA(I)	THETA(I)	PSI(I)	CHI(I)
1	50.00	0.90000	0.30000	1.00000	0.80000	0.40738	0.61478
2	50.00	0.90000	0.30000	1.00000	0.80000	0.37973	0.54726
3	50.00	0.90000	0.30000	1.00000	0.80000	0.32486	0.44010
4	50.00	0.90000	0.30000	1.00000	0.80000	0.21600	0.27000
5	50.00	0.90000	0.30000	1.00000	0.80000	0.00000	0.00000

STATISTICS TABLE
========== =====

I	CN(I)	N(I)	CM(I)	M(I)	B(I)	S(I)	R(I)	Z(I)	ZP(I)
1	500.00	150.00	0.00	0.00	50.00	150.00	61.11	0.00	400.80
2	500.00	150.00	108.00	32.40	50.00	150.00	56.96	28.71	311.91
3	500.00	150.00	162.43	48.73	50.00	150.00	48.73	36.94	218.87
4	500.00	150.00	189.87	56.96	50.00	150.00	32.40	28.71	117.60
5	500.00	150.00	203.69	61.11	50.00	150.00	0.00	0.00	0.00

TABLE B.4 (b)

FULL MODEL
==== =====

EXPECTED ESTIMATES TABLE
======== ========= =====

I	NH(I)	SNH(I)	CVNH(I)	PHIH(I)	SPHIH(I)	CVPHIH(I)	BH(I)	SBH(I)	CVBH(I)
1	0.00	0.00	0.00	0.72000	0.09797	13.60721	0.00	0.00	0.00
2	500.00	90.64	18.13	0.72000	0.10463	14.53198	140.00	74.91	53.51
3	500.00	79.75	15.95	0.72000	0.13017	18.07958	140.00	63.63	45.45
4	500.00	90.64	18.13	0.00000	0.00000	0.00000	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00000	0.00000	0.00000	0.00	0.00	0.00

BIAS TABLE
==== =====

	CASE 3 CORRECTED PHI GIVEN THETA			EXPECTED BIAS EFFECTS ON PHIH			EXPECTED BIAS EFFECTS ON BH		
I	PHIH3(I)	SPHIH3(I)	CVPHIH3(I)	AB(PHIH)	RB(PHIH)	EB(PHIH)	AB(BH)	RB(BH)	EB(BH)
1	0.90000	0.12246	13.60721	-0.18000	-20.00000	-183.72616	0.00	0.00	0.00
2	0.90000	0.13079	14.53198	-0.18000	-20.00000	-172.03438	90.00	180.00	120.14
3	0.90000	0.16272	18.07958	-0.18000	-20.00000	-138.27750	90.00	180.00	141.45
4	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00	0.00	0.00
5	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00	0.00	0.00

TABLE B.4 (c)

DEATH-ONLY MODEL
========== =====

EXPECTED ESTIMATES TABLE
======== ========= =====

I	NH(I)	SNH(I)	CVNH(I)	PHIH(I)	SPHIH(I)	CVPHIH(I)
1	1133.84	104.01	9.17	0.85673	0.11421	13.33059
2	971.40	93.21	9.60	0.84798	0.12208	14.39598
3	823.73	87.69	10.65	0.84305	0.14730	17.47246
4	694.44	95.65	13.77	0.00000	0.00000	0.00000
5	0.00	0.00	0.00	0.00000	0.00000	0.00000

BIAS TABLE
==== =====

	EXPECTED BIAS EFFECTS ON NH			EXPECTED BIAS EFFECTS ON PHIH		
I	AB(NH)	RB(NH)	EB(NH)	AB(PHIH)	RB(PHIH)	EB(PHIH)
1	633.84	126.77	609.40	-0.04327	-4.80779	-37.88742
2	471.40	94.28	505.72	-0.05202	-5.77963	-42.61028
3	323.73	64.75	369.19	-0.05695	-6.32785	-38.66266
4	194.44	38.89	203.29	0.00000	0.00000	0.00000
5	0.00	0.00	0.00	0.00000	0.00000	0.00000

>>>>>>>>>> END OF TASK

TABLE B.5

Task output with: Varying parameters, Stabilized population

>>>>>>>>>> START OF TASK

MODEL(S) SELECTED:

FULL MODEL

ORIG. POPULATION: 250
SAMPLE TIMES: 8
CUTOFF VALUES:
M: 1.00000
R: 2.00000

PARAMETER TABLE
========= =====

I	B(I)	PHI(I)	PI(I)	ETA(I)	THETA(I)	PSI(I)	CHI(I)
1	100.00	0.75000	0.25000	1.00000	0.95000	0.42125	0.46922
2	100.00	0.75000	0.25000	1.00000	0.95000	0.45498	0.50084
3	250.00	0.90000	0.40000	0.80000	1.00000	0.39761	0.44631
4	-1.00	0.80000	0.20000	0.90000	0.90000	0.30224	0.36988
5	-1.00	0.80000	0.20000	0.90000	0.90000	0.27472	0.32794
6	-1.00	0.80000	0.20000	0.90000	0.90000	0.22694	0.26240
7	-1.00	0.80000	0.20000	0.90000	0.90000	0.14400	0.16000
8	-1.00	0.80000	0.20000	0.90000	0.90000	0.00000	0.00000

STATISTICS TABLE
========== =====

I	CN(I)	N(I)	CM(I)	M(I)	B(I)	S(I)	R(I)	Z(I)	ZP(I)
1	250.00	62.50	0.00	0.00	100.00	62.50	26.33	0.00	523.03
2	287.50	71.88	44.53	11.13	100.00	71.88	32.70	15.20	477.48
3	315.63	126.25	75.01	30.00	250.00	101.00	40.16	17.89	383.93
4	511.34	102.27	131.40	26.28	110.45	92.04	27.82	31.77	321.82
5	511.34	102.27	141.96	28.39	110.45	92.04	25.29	31.20	247.38
6	511.34	102.27	148.04	29.61	110.45	92.04	20.89	26.88	170.39
7	511.34	102.27	151.54	30.31	110.45	92.04	13.25	17.46	89.01
8	511.34	102.27	153.56	30.71	110.45	92.04	0.00	0.00	0.00

<< REMAINING OUTPUT NOT SHOWN >>

APPENDIX C

Example Input and Job Decks

TABLE C.1: Input deck for generating output of Appendix B

CARD	COLUMN 1 1234567890	2 1234567890	3 1234567890	4 1234567890	5 1234567890	6 1234567890
1	500	5	3	3.00000	3.00000	
2	1	0	0.90000	0.30000	1.00000	1.00000
3	0	0	0.00000	0.00000	0.00000	0.00000
4	500	5	3	3.00000	3.00000	
5	1	0	0.90000	0.30000	1.00000	0.80000
6	0	0	0.00000	0.00000	0.00000	0.00000
7	500	5	3	3.00000	3.00000	
8	1	50	0.90000	0.30000	1.00000	1.00000
9	0	0	0.00000	0.00000	0.00000	0.00000
10	500	5	3	3.00000	3.00000	
11	1	50	0.90000	0.30000	1.00000	0.80000
12	0	0	0.00000	0.00000	0.00000	0.00000
13	250	8	1	1.00000	2.00000	
14	1	100	0.75000	0.25000	1.00000	0.95000
15	3	250	0.90000	0.40000	0.80000	1.00000
16	4	-1	0.80000	0.20000	0.90000	0.90000
17	0	0	0.00000	0.00000	0.00000	0.00000

TABLE C.2

Symbolic form for each task in input deck

```
COLUMN   1         2         3         4         5         6
123456789012345678901234567890123456789012345678901234567890

NNNNNNNNNN        KK         A    M.CCCCC    R.CCCCC
        I1      BBBB   F.FFFFF    P.PPPPP    R.RRRRR    T.TTTTT
         :
        IK      BBBB   F.FFFFF    P.PPPPP    R.RRRRR    T.TTTTT
         0         0   0.00000    0.00000    0.00000    0.00000

where: NNNNNNN = initial population size (integer)
            KK = number of samples (2 < integer < 21)
             A = analysis number   (1 = full model,
                    2 = death-only model, 3=both models)
       M.CCCCC = cut-off value for M(I) (1.0 < real < 10.0)
       R.CCCCC = cut-off value for R(I) (1.0 < real < 10.0)
       I1...IK = sample time (1 =< integer =< KK)
          BBBB = number of births ( -1 =< integer)
       F.FFFFF = survival rate (0.0 < real =< 1.0)
       P.PPPPP = capture rate  (0.0 < real =< 1.0)
       R.RRRRR = returns-on-capture rate (0.0 =< real =< 1.0)
       T.TTTTT = tag retention rate (0.0 < real =< 1.0)
```

TABLE C.3

Example of complete job deck for running BEFFJOB tasks

```
COLUMN   1         2         3         4         5         6
123456789012345678901234567890123456789012345678901234567890
//jobname JOB 'acct,pswd,T=5,I=5','your name'
//        EXEC FORTHCG,SIZE=256K,CSIZE=512K
//FORT.SYSIN DD *
      :
   BEFFJOB program ... approx. 900 cards.
      :
//GO.FT14F001 DD *
      :
   your input deck
      :
/*
```

APPENDIX D

Program Source Listing

```
C     BEFFJOB - A PROGRAM IN ANSI FORTRAN IV  TO COMPUTE:
C         BIAS (DUE TO TAG-LOSS OR MODEL MIS-SPECIFICATION)  AND
C         PRECISION  (EXPECTED STANDARD ERRORS OF THE ESTIMATES)
C     FOR JOLLY-SEBER MARK-RECAPTURE ESTIMATES
C
      INTEGER EOFLAG,BIFLAG,YES/1/,NO/0/,CN1,K,PRTIME,TIME,BIRTHS
      INTEGER T1,T2,J,MODEL,I
      DOUBLE PRECISION D1/1.0D0/,B(20),PHI(20),PI(20),ETA(20),THETA(20)
      DOUBLE PRECISION PSI(20),CHI(20)
      DOUBLE PRECISION CN(20),N(20),CM(20),M(20),S(20),R(20),Z(20)
      DOUBLE PRECISION ZP(20),ZPSAVE,CUTM,CUTR
      DOUBLE PRECISION MH(20),NH(20),PHIH(20),BH(20)
      DOUBLE PRECISION VNH(20),VPHIH(20),VBH(20),CNN(20),CMM(20)
      DOUBLE PRECISION CMMS(20),RS(20),ALPHA(20),LNUM/1.0D15/,CNNS(20)
      DOUBLE PRECISION SAVE1,SAVE2,SAVE3,SAVE4,SAVE5,SAVE6,SAVE7,BSUM
      DOUBLE PRECISION SBH(20),SPHIH(20),SNH(20)
      DOUBLE PRECISION CVBH(20),CVPHIH(20),CVNH(20)
      DOUBLE PRECISION VALUE(9)
      REAL SURV,CAPT,SOC,TRR,CUTM1,CUTR1
C
      CALL HEADER
      CALL DEFIN
      EOFLAG = NO
    1 CONTINUE
      IF(EOFLAG.EQ.YES) GO TO 7
C
C **********************************************************************
C
C  PART 1)  INPUTTING OF PARAMETER TABLE
C
         BIFLAG = NO
         READ(14,100,END=7) CN1,K,MODEL,CUTM1,CUTR1
         WRITE(6,101)
C
C  ERROR CHECKS FOR FIRST INPUT LINE
C
         IF(CN1.GE.1.AND.CN1.LE.999999) GO TO 103
            WRITE(6,102) CN1
            GO TO 169
  103    CONTINUE
         IF(K.GE.3.AND.K.LE.20) GO TO 105
            WRITE(6,104) K
            GO TO 169
  105    CONTINUE
         IF(MODEL.GE.1.AND.MODEL.LE.3) GO TO 107
            WRITE(6,106) MODEL
            GO TO 169
  107    CONTINUE
C
C  PRINT TYPES OF MODELS TO BE USED
C
         WRITE(6,108)
         IF(MODEL.NE.2) WRITE(6,109)
         IF(MODEL.NE.1) WRITE(6,110)
         PRTIME = 0
C
C  READ SERIES OF PARAMETER INPUT LINES, END IF ERROR FOUND, TIME = 0,
C  OR END OF FILE IS ENCOUNTERED.
C
  111    CONTINUE
            READ(14,112,END=114) TIME,BIRTHS,SURV,CAPT,SOC,TRR
            IF(TIME.EQ.0) GO TO 115
C
C  ERROR CHECKS OF PARAMETER INPUT LINES
C
            IF(PRTIME.LT.TIME.AND.TIME.LE.K) GO TO 116
               WRITE(6,117) TIME
               GO TO 113
  116       CONTINUE
```

```
           IF(PRTIME.NE.0) GO TO 118
              IF(TIME.EQ.1) GO TO 118
                 WRITE(6,117) TIME
                 GO TO 113
  118      CONTINUE
           IF(BIRTHS.GE.-1.AND.BIRTHS.LE.999999) GO TO 180
              WRITE(6,181) BIRTHS
              GO TO 113
  180      CONTINUE
           IF(SURV.GT.0.0.AND.SURV.LE.1.0) GO TO 119
              WRITE(6,120) SURV
              GO TO 113
  119      CONTINUE
           IF(CAPT.GT.0.0.AND.CAPT.LE.1.0) GO TO 121
              WRITE(6,122) CAPT
              GO TO 113
  121      CONTINUE
           IF(SOC.GE.0.0.AND.SOC.LE.1.0) GO TO 123
              WRITE(6,124) SOC
              GO TO 113
  123      CONTINUE
           IF(TRR.GT.0.0.AND.TRR.LE.1.0) GO TO 125
              WRITE(6,126) TRR
              GO TO 113
  125      CONTINUE
C
C  NO ERRORS, FILL UP PARMETER TABLE
C
           IF(TIME.EQ.1) GO TO 127
              T1 = PRTIME + 1
              T2 = TIME - 1
              IF(T1.GT.T2) GO TO 127
                 DO 128 J = T1,T2
                    B(J) = B(PRTIME)
                    PHI(J) = PHI(PRTIME)
                    PI(J) = PI(PRTIME)
                    ETA(J) = ETA(PRTIME)
                    THETA(J) = THETA(PRTIME)
  128            CONTINUE
  127      CONTINUE
           B(TIME) = DBLE(FLOAT(BIRTHS))
           PHI(TIME) = DBLE(SURV)
           PI(TIME) = DBLE(CAPT)
           ETA(TIME) = DBLE(SOC)
           THETA(TIME) = DBLE(TRR)
           IF(TRR.NE.1.0) BIFLAG = YES
           PRTIME = TIME
           GO TO 111
C
C  END OF INPUT, DO CLEANUP AND OUTPUT PARAMETER TABLE IF NO ERROR
C  IS ENCOUNTERED
C
  114   CONTINUE
        EOFLAG = YES
  115   CONTINUE
        IF(PRTIME.NE.0) GO TO 140
          WRITE(6,129)
          GO TO 113
  140   CONTINUE
        T1 = PRTIME + 1
        DO 130 J = T1,K
           B(J) = B(PRTIME)
           PHI(J) = PHI(PRTIME)
           PI(J) = PI(PRTIME)
           ETA(J) = ETA(PRTIME)
           THETA(J) = THETA(PRTIME)
  130   CONTINUE
C
C  SET UNUSED PORTIONS OF PARAMETER TABLE TO 0.
C
        IF(K.EQ.20) GO TO 131
           T1 = K + 1
           DO 132 J = T1,20
              B(J) = 0.0D0
```

```
                PHI(J) = 0.0D0
                PI(J) = 0.0D0
                ETA(J) = 0.0D0
                THETA(J) = 0.0D0
  132        CONTINUE
  131     CONTINUE
C
C  ACCORDING TO VALUE OF BIFLAG (BIAS) CALCULATE PSI AND CHI.
C
          PSI(K) = 0.0D0
          CHI(K) = 0.0D0
          J = K - 1
  133     CONTINUE
          IF(J.EQ.0) GO TO 134
             PSI(J) = D1 - ((D1 - PHI(J)) + PHI(J) * (D1 - THETA(J))
     $                + PHI(J) * THETA(J) * (D1 - PI(J+1))
     $                * (D1 - PSI(J+1)))
             IF(BIFLAG.EQ.YES)
     $          CHI(J) = D1 - ((D1 - PHI(J)) + PHI(J) * (D1 - PI(J+1))
     $                   * (D1 - CHI(J+1)))
             J = J - 1
             GO TO 133
  134     CONTINUE
C
C  ACCORDING TO THE VALUE OF BIFLAG (BIAS), OUTPUT THE PARAMETER
C  TABLE.
C
          WRITE(6,139) CN1,K
          CUTM = DBLE(CUTM1)
          IF(CUTM1.LT.1.0) CUTM = D1
          IF(CUTM1.GT.10.0) CUTM = 10.0D0
          CUTR = DBLE(CUTR1)
          IF(CUTR1.LT.1.0) CUTR = D1
          IF(CUTR1.GT.10.0) CUTR = 10.0D0
          WRITE(6,150) CUTM,CUTR
          IF(BIFLAG.EQ.NO) GO TO 144
             WRITE(6,138)
             DO 135 J = 1,K
                WRITE(6,136) J,B(J),PHI(J),PI(J),ETA(J),
     $                                    THETA(J),PSI(J),CHI(J)
  135        CONTINUE
             GO TO 145
  144     CONTINUE
             WRITE(6,146)
             DO 147 J = 1,K
                WRITE(6,137) J,B(J),PHI(J),PI(J),ETA(J),THETA(J),
     $                       PSI(J)
  147        CONTINUE
  145     CONTINUE
C
C  BRANCH TO CALCULATE STATISTICS
C
          GO TO 200
C
C  END OF JOB, GO BACK TO START TO READ NEW SET OF INPUT DATA.
C
  141     CONTINUE
          WRITE(6,142)
          GO TO 1
C
C  ROUTINE TO HANDLE ERROR IN INPUT
C
  113     CONTINUE
             IF(TIME.EQ.0) GO TO 141
  169     CONTINUE
             IF(EOFLAG.EQ.YES) GO TO 141
             READ(14,112,END=143) TIME,BIRTHS,SURV,CAPT,SOC,TRR
             GO TO 113
  143     CONTINUE
             EOFLAG = YES
             GO TO 141
C
C  ***********************************************************************
C
```

```
C  PART 2)  BUILDING OF STATISTICS TABLE
C
  200 CONTINUE
      J = 1
      BSUM=0.D0
      CN(1) = DBLE(FLOAT(CN1))
      CM(1) = 0.0D0
  204 CONTINUE
      IF(J.GT.K) GO TO 201
         N(J) = CN(J) * PI(J)
         M(J) = CM(J) * PI(J)
         S(J) = N(J) * ETA(J)
         R(J) = S(J) * PSI(J)
         Z(J) = CM(J) * PSI(J) * (D1 - PI(J))
         IF(B(J).GE.0) GO TO 202
            B(J) = (CN(J) - N(J) + S(J)) * (D1 - PHI(J))
     $              + N(J) * (D1 - ETA(J))
  202    CONTINUE
         BSUM = BSUM + B(J)
         IF(J.EQ.K) GO TO 203
            CN(J+1) = (CN(J) - N(J) + S(J)) * PHI(J)
     $                + B(J)
            CM(J+1) = (CM(J) - M(J) + S(J)) * PHI(J) * THETA(J)
  203    CONTINUE
         J = J + 1
         GO TO 204
  201 CONTINUE
      J = K - 1
      ZP(K) = 0.0D0
      ZPSAVE = N(K) - M(K)
  205 CONTINUE
      IF(J.EQ.0) GO TO 206
         ZP(J) = Z(J) + ZPSAVE
         ZPSAVE = ZPSAVE + N(J) - M(J)
         J = J - 1
         GO TO 205
  206 CONTINUE
C
C  OUTPUT STATISTICS TABLE
C
      WRITE(6,207)
      DO 208 J = 1,K
         WRITE(6,209) J,CN(J),N(J),CM(J),M(J),B(J),S(J),R(J),Z(J),
     $                ZP(J)
  208 CONTINUE
C
C  GO TO DO MODELS
C
      GO TO 300
  299 CONTINUE
      GO TO 141
C
C  ***********************************************************************
C
C  PART 3)  BUILDING OF MODEL TABLES
C
C       3A) FULL MODEL
C
C  CALCULATE NH, PHIH, BH
C
  300 CONTINUE
      IF(MODEL.EQ.2) GO TO 350
      MH(1) = 0.0D0
      NH(1) = 0.0D0
      CNN(1) = NH(1) - N(1)
      CMM(1) = MH(1) - M(1)
      CMMS(1) = CMM(1) + S(1)
      RS(1) = LNUM
      IF(R(1).GT.CUTR.AND.S(1).GT.0.0D0)
     $   RS(1) = D1 / R(1) - D1 / S(1)
      ALPHA(1) = 0.0D0
      T1 = K - 1
      DO 301 J = 2,T1
         MH(J) = LNUM
```

```
         IF(R(J).GT.CUTR) MH(J) = S(J) * Z(J) / R(J) + M(J)
         RS(J) = LNUM
         IF(R(J).GT.CUTR.AND.S(J).GT.0.0D0)
     $      RS(J) = D1 / R(J) - D1 / S(J)
         ALPHA(J) = LNUM
         IF(N(J).GT.0.0D0) ALPHA(J) = M(J) / N(J)
         IF(M(J).GT.CUTM) GO TO 306
            NH(J) = LNUM
            PHIH(J-1) = LNUM
            BH(J-1) = LNUM
            GO TO 307
  306    CONTINUE
            NH(J) = MH(J) * N(J) / M(J)
            PHIH(J-1) = LNUM
            SAVE4 = MH(J-1) - M(J-1) + S(J-1)
            IF(SAVE4.GT.0.0D0) PHIH(J-1) = MH(J) / SAVE4
            BH(J-1) = NH(J) - PHIH(J-1) * (CNN(J-1) + S(J-1))
  307    CONTINUE
         CNN(J) = NH(J) - N(J)
         CMM(J) = MH(J) - M(J)
         CMMS(J) = CMM(J) + S(J)
  301 CONTINUE
      BH(1) = 0.0D0
      BH(T1) = 0.0D0
      BH(K) = 0.0D0
      PHIH(T1) = 0.0D0
      PHIH(K) = 0.0D0
      NH(K) = 0.0D0
C
C  CALCULATE SNH, SPHIH, SBH, CVNH, CVPHIH, CVBH
C
      CVNH(1) = 0.0D0
      DO 310 J = 2,T1
         IF(M(J).GT.CUTM.AND.R(J).GT.CUTR.AND.MH(J).GT.0.0D0)
     $      GO TO 311
            VNH(J) = LNUM
            VPHIH(J-1) = LNUM
            VBH(J-1) = LNUM
            GO TO 313
  311    CONTINUE
         SAVE1 = CMMS(J) / MH(J) * RS(J)
         VNH(J) = NH(J) * (NH(J) - N(J)) * (SAVE1 + (D1 - ALPHA(J))
     $            / M(J))
         IF(CMMS(J-1).GT.0.0D0) GO TO 312
            VPHIH(J-1) = 0.0D0
            VBH(J-1) = 0.0D0
            GO TO 313
  312    CONTINUE
         SAVE2 = SAVE1 * CMM(J) / CM(J)
         SAVE3 = CMM(J-1) / CMMS(J-1) * RS(J-1)
         VPHIH(J-1) = (PHIH(J-1)) ** 2 * (SAVE2 + SAVE3 +
     $                (D1 - PHIH(J-1)) / CM(J))
         IF(ALPHA(J-1).GT.0.0D0) GO TO 314
            VBH(J-1) = LNUM
            GO TO 313
  314    CONTINUE
         SAVE5 = BH(J-1) ** 2 * SAVE2 + SAVE3 * ((PHIH(J-1) * S(J-1) *
     $           (D1 - ALPHA(J-1)) / ALPHA(J-1)) ** 2)
         SAVE6 = CNN(J-1) * (NH(J) - BH(J-1)) * (D1 - ALPHA(J-1)) *
     $           (D1 - PHIH(J-1)) / CMMS(J-1)
         SAVE7 = (NH(J) * CNN(J) * (D1 - ALPHA(J)) / M(J)) +
     $           (PHIH(J-1) ** 2 * NH(J-1) * CNN(J-1) * (D1 -
     $           ALPHA(J-1)) / M(J-1))
         VBH(J-1) = SAVE5 + SAVE6 + SAVE7
  313    CONTINUE
         SNH(J) = DSQRT(VNH(J))
         CVNH(J) = LNUM
         IF(NH(J).GT.0.0D0) CVNH(J) = SNH(J) / NH(J) * 100.0D0
         SPHIH(J-1) = DSQRT(VPHIH(J-1))
         CVPHIH(J-1) = SPHIH(J-1) / PHIH(J-1) * 100.0D0
         SBH(J-1) = DSQRT(VBH(J-1))
         CVBH(J-1) = LNUM
         IF(BH(J-1).GT.0.0D0) CVBH(J-1) = SBH(J-1) / BH(J-1) * 100.0D0
  310 CONTINUE
```

```
      CVBH(1) = 0.0D0
      CVBH(T1) = 0.0D0
      CVBH(K) = 0.0D0
      SBH(1) = 0.0D0
      SBH(T1) = 0.0D0
      SBH(K) = 0.0D0
      CVPHIH(T1) = 0.0D0
      CVPHIH(K) = 0.0D0
      SPHIH(T1) = 0.0D0
      SPHIH(K) = 0.0D0
      SNH(1) = 0.0D0
      SNH(K) = 0.0D0
      CVNH(K) = 0.0D0
C
C  PRINT FIRST TABLE
C
      WRITE(6,322)
      DO 320 J = 1,K
         WRITE(6,321) J,NH(J),SNH(J),CVNH(J),PHIH(J),SPHIH(J),
     $                CVPHIH(J),BH(J),SBH(J),CVBH(J)
  320 CONTINUE
C
C  PRINT SECOND TABLE IF BIASED
C
      IF(BIFLAG.EQ.NO) GO TO 350
         WRITE(6,323)
         WRITE(6,3231)
         DO 324 J = 1,K
            VALUE(1) = PHIH(J) / THETA(J)
            VALUE(2) = SPHIH(J) / THETA(J)
            VALUE(3) = CVPHIH(J)
            IF(J.LT.T1) GO TO 340
               DO 341 I = 4,9
                  VALUE(I) = 0.0D0
  341          CONTINUE
               GO TO 332
  340       CONTINUE
            VALUE(4) = PHIH(J) - PHI(J)
            VALUE(5) = VALUE(4) / PHI(J) * 100.0D0
            VALUE(6) = LNUM
            IF(SPHIH(J).GT.0.0D0) VALUE(6) = VALUE(4) / SPHIH(J)
     $                                       * 100.0D0
            IF(J.GT.1) GO TO 330
               DO 331 I = 7,9
                  VALUE(I) = 0.0D0
  331          CONTINUE
               GO TO 332
  330       CONTINUE
            VALUE(7) = BH(J) - B(J)
            VALUE(8) = LNUM
            IF(B(J).GT.0.0D0) VALUE(8) = VALUE(7) / B(J) * 100.0D0
            VALUE(9) = LNUM
            IF(SBH(J).GT.0.0D0) VALUE(9) = VALUE(7) / SBH(J) * 100.0D0
  332       CONTINUE
            WRITE(6,325) J,(VALUE(I),I = 1,9)
  324    CONTINUE
C
C        3B) DEATH ONLY MODEL
C
C  CALCULATE NH
C
  350 CONTINUE
      IF(MODEL.EQ.1) GO TO 299
      DO 351 J = 1,T1
         NH(J) = LNUM
         IF(R(J).GT.CUTR) NH(J) = S(J) * ZP(J) / R(J) + N(J)
         CNN(J) = NH(J) - N(J)
         CNNS(J) = CNN(J) + S(J)
         RS(J) = LNUM
         IF(R(J).GT.CUTR.AND.S(J).GT.0.0D0)
     $      RS(J) = D1 / R(J) - D1 / S(J)
  351 CONTINUE
      NH(K) = 0.0D0
C
```

```
C  CALCULATE SNH, CVNH, PHIH, SPHIH, CVPHIH
C
      PHIH(T1) = 0.0D0
      VPHIH(T1) = 0.0D0
      DO 352 J = 1,T1
         VNH(J) = CNN(J) * CNNS(J) * RS(J)
         IF(J.EQ.T1) GO TO 353
            PHIH(J) = LNUM
            SAVE1 = NH(J) - N(J) + S(J)
            IF(SAVE1.GT.0.0D0) PHIH(J) = NH(J+1) / SAVE1
            VPHIH(J) = LNUM
            IF(NH(J+1).GT.0.0D0.AND.CNNS(J+1).GT.0.0D0)
     $         VPHIH(J) = PHIH(J) ** 2 * ((RS(J+1) * CNNS(J+1) *
     $                    CNN(J+1) / (NH(J+1) ** 2)) + (CNN(J) /
     $                    CNNS(J) * RS(J)) + ((D1 - PHIH(J)) / NH(J+1)))
  353    CONTINUE
         SNH(J) = DSQRT(VNH(J))
         CVNH(J) = LNUM
         IF(NH(J).GT.0.0D0) CVNH(J) = SNH(J) / NH(J) * 100.0D0
         SPHIH(J) = DSQRT(VPHIH(J))
         CVPHIH(J) = LNUM
         IF(PHIH(J).GT.0.0D0) CVPHIH(J) = SPHIH(J) / PHIH(J) * 100.0D0
  352 CONTINUE
      PHIH(K) = 0.0D0
      SPHIH(K) = 0.0D0
      CVPHIH(K) = 0.0D0
      CVPHIH(T1) = 0.0D0
      SNH(K) = 0.0D0
      CVNH(K) = 0.0D0
C
C  OUTPUT OF DEATH ONLY TABLE
C
      WRITE(6,354)
      DO 355 J = 1,K
         WRITE(6,356) J,NH(J),SNH(J),CVNH(J),PHIH(J),SPHIH(J),CVPHIH(J)
  355 CONTINUE
C
C  BUILD AND OUTPUT SECOND TABLE IF BIASED
C
      IF(BIFLAG.EQ.NO .AND. BSUM.EQ.0.D0) GO TO 299
         WRITE(6,360)
         DO 361 J = 1,K
            IF(J.LT.K) GO TO 370
               DO 371 I = 1,6
                  VALUE(I) = 0.0D0
  371          CONTINUE
               GO TO 372
  370       CONTINUE
            VALUE(1) = NH(J) - CN(J)
            VALUE(2) = LNUM
            IF(CN(J).GT.0.0D0)
     $         VALUE(2) = VALUE(1) / CN(J) * 100.0D0
            VALUE(3) = LNUM
            IF(SNH(J).GT.0.0D0)
     $         VALUE(3) = VALUE(1) / SNH(J) * 100.0D0
            IF(J.LT.T1) GO TO 380
               DO 381 I = 4,6
                  VALUE(I) = 0.0D0
  381          CONTINUE
               GO TO 372
  380       CONTINUE
            VALUE(4) = PHIH(J) - PHI(J)
            VALUE(5) = VALUE(4) / PHI(J) * 100.0D0
            VALUE(6) = LNUM
            IF(SPHIH(J).GT.0.0D0)
     $         VALUE(6) = VALUE(4) / SPHIH(J) * 100.0D0
  372       CONTINUE
            WRITE(6,362) J,(VALUE(I),I=1,6)
  361    CONTINUE
      GO TO 299
C
C  END OF PROGRAM
C
    7 CONTINUE
```

```
      STOP
C
C  FORMAT STATEMENTS
C
  100 FORMAT(3I10,2F10.5)
  101 FORMAT('1','>>>>>>>>>>>  START OF TASK')
  102 FORMAT('0','***ERROR***  INVALID STARTING POPULATION: ',I10)
  104 FORMAT('0','***ERROR***  INVALID NO. OF SAMPLE TIMES: ',I10)
  106 FORMAT('0','***ERROR***  INVALID MODEL NUMBER: ',I10)
  108 FORMAT('0',10X,'MODEL(S) SELECTED:'/)
  109 FORMAT(' ',17X,'FULL MODEL')
  110 FORMAT(' ',17X,'DEATH-ONLY MODEL')
  112 FORMAT(2I10,4F10.5)
  117 FORMAT('0','***ERROR***  INVALID TIME VALUE: ',I10)
  120 FORMAT('0','***ERROR***  INVALID SURVIVAL RATE (PHI) : ',F10.5)
  122 FORMAT('0','***ERROR***  INVALID CAPTURE RATE (PI) : ',F10.5)
  124 FORMAT('0','***ERROR***  INVALID RETURNS-ON-CAPTURE RATE ',
    $        '(ETA) : ',F10.5)
  126 FORMAT('0','***ERROR***  INVALID TAG RETENTION RATE (THETA) ',
    $        ': ',F10.5)
  129 FORMAT('0','***ERROR***  MISSING SET OF DATA')
  136 FORMAT(' ',I10,1X,F10.2,6(1X,F10.5))
  137 FORMAT(' ',I10,1X,F10.2,5(1X,F10.5))
  138 FORMAT(///' ',10X,'PARAMETER TABLE'/' ',10X,'========= ====='/
    $        /' ',9X,'I',5X,'B(I)',
    $        6X,'PHI(I)',6X,'PI(I)',5X,'ETA(I)',4X,'THETA(I)',
    $        4X,'PSI(I)',5X,'CHI(I)'/)
  139 FORMAT('0',10X,'ORIG. POPULATION: ',I10/' ',10X,'SAMPLE TIMES: ',
    $        I14)
  142 FORMAT(////' ','>>>>>>>>>>>  END OF TASK')
  146 FORMAT(///' ',10X,'PARAMETER TABLE'/' ',10X,'========= ====='/
    $        /' ',9X,'I',5X,'B(I)',6X,
    $        'PHI(I)',6X,'PI(I)',5X,'ETA(I)',4X,'THETA(I)',4X,
    $        'CHI(I)'/)
  150 FORMAT(' ',10X,'CUTOFF VALUES:'/' ',20X,'M: ',F10.5/' ',20X,
    $        'R: ',F10.5)
  181 FORMAT('0','***ERROR***  INVALID BIRTH VALUE :',I10)
C
  207 FORMAT('0',10X,'STATISTICS TABLE'/' ',10X,'========== ====='
    $        //' ',9X,'I',5X,'CN(I)',7X,
    $        'N(I)',6X,'CM(I)',7X,'M(I)',7X,'B(I)',7X,'S(I)',7X,'R(I)',
    $        7X,'Z(I)',6X,'ZP(I)'/)
  209 FORMAT(' ',8X,I2,9(1X,F10.2))
C
  321 FORMAT(' ',8X,I2,3X,3(1X,F10.2),7X,3(1X,F10.5),7X,3(1X,F10.2))
  322 FORMAT('1'///' ',10X,'FULL MODEL'/' ',10X,'==== ====='//' ',13X,
    $        'EXPECTED ESTIMATES TABLE'/' ',13X,'======== =========',
    $        ' ====='//' ',9X,'I',6X,'NH(I)',5X,
    $        'SNH(I)',5X,'CVNH(I)',11X,'PHIH(I)',4X,'SPHIH(I)',2X,
    $        'CVPHIH(I)',11X,'BH(I)',5X,'SBH(I)',5X,'CVBH(I)'/)
  323 FORMAT('0',13X,'BIAS TABLE'/' ',13X,'==== ====='
    $        //' ',13X,'CASE 3 CORRECTED PHI ',
    $        'GIVEN THETA',10X,'EXPECTED BIAS EFFECTS ON PHIH',11X,
    $        'EXPECTED BIAS EFFECTS ON BH')
 3231 FORMAT('0',9X,'I',5X,'PHIH3(I)',2X,'SPHIH3(I)',2X,'CVPHIH3(I)',
    $        9X,'AB(PHIH)',3X,'RB(PHIH)',3X,'EB(PHIH)',11X,'AB(BH)',
    $        5X,'RB(BH)',5X,'EB(BH)'/)
  325 FORMAT(' ',8X,I2,3X,3(1X,F10.5),7X,3(1X,F10.5),7X,3(1X,F10.2))
  354 FORMAT('1',10X,'DEATH-ONLY MODEL'/' ',10X,'========== ====='
    $        //' ',13X,'EXPECTED ',
    $        'ESTIMATES TABLE'/' ',13X,'======== ========= ====='
    $        //' ',9X,'I',7X,'NH(I)',5X,'SNH(I)',
    $        5X,'CVNH(I)',10X,'PHIH(I)',4X,'SPHIH(I)',2X,'CVPHIH(I)'/)
  356 FORMAT(' ',8X,I2,3X,3(1X,F10.2),7X,3(1X,F10.5))
  360 FORMAT('0',13X,'BIAS TABLE'/' ',13X,'==== ====='//
    $        ' ',16X,'EXPECTED BIAS EFFECTS ON NH',12X,
    $        'EXPECTED BIAS EFFECTS ON PHIH'
    $        //' ',9X,'I',6X,'AB(NH)',5X,'RB(NH)',5X,'EB(NH)',
    $        11X,'AB(PHIH)',2X,'RB(PHIH)',4X,'EB(PHIH)'/)
  362 FORMAT(' ',8X,I2,3X,3(1X,F10.2),7X,3(1X,F10.5))
C
      END
      SUBROUTINE HEADER
C
```

```
C  SUBROUTINE HEADER
C
C     THIS SUBROUTINE PRINTS THE HEADER 'BEFFJOB'.
C
      WRITE(6,1)
      WRITE(6,2)
      WRITE(6,2)
      WRITE(6,2)
      WRITE(6,3)
      WRITE(6,2)
      WRITE(6,4)
      WRITE(6,4)
      WRITE(6,5)
      WRITE(6,10)
      WRITE(6,11)
      WRITE(6,12)
      WRITE(6,13)
      WRITE(6,14)
      RETURN
C
    1 FORMAT('1'////////////' ',35X,'BBBBBB ',2X,
     $       'EEEEEEE',2X,'FFFFFFF',2X,'FFFFFFF',2X,'      J',2X,
     $       ' OOOOO ',2X,'BBBBBB ')
    2 FORMAT(' ',35X,'B     B',2X,'E      ',2X,'F      ',2X,
     $       'F      ',2X,'      J',2X,'O     O',2X,'B     B')
    3 FORMAT(' ',35X,'BBBBBB ',2X,'EEEEEE ',2X,'FFFFF  ',2X,
     $       'FFFFF  ',2X,'      J',2X,'O     O',2X,'BBBBBB ')
    4 FORMAT(' ',35X,'B     B',2X,'E      ',2X,'F      ',2X,
     $       'F      ',2X,'J     J',2X,'O     O',2X,'B     B')
    5 FORMAT(' ',35X,'BBBBBB ',2X,'EEEEEEE',2X,'F      ',2X,
     $       'F      ',2X,' JJJJJ ',2X,' OOOOO ',2X,'BBBBBB ')
   10 FORMAT(///' ',40X,'AUTHORS:',5X,'A. N. ARNASON  -  DEPT. OF ',
     $      'COMPUTER SCIENCE'//' ',53X,'C. R. KRASEY   -  ',
     $      'UNIV. OF MANITOBA'//' ',53X,'K. H. MILLS    -  ',
     $      'FRESHWATER INSTITUTE'/' ',71X,'D.F.O., WINNIPEG, MANITOBA')
   11 FORMAT(///' ',30X,'DISCLAIMER:'/' ',30X,'=========='/)
   12 FORMAT(' ',30X,'     NO WARRANTY , EXPRESSED OR IMPLIED , IS ',
     $       'MADE BY THE AUTHORS  OF  THIS'/' ',30X,'PROGRAM  , OR ',
     $       'BY  THE UNIVERSITY OF MANITOBA  , AS TO THE ACCURACY ',
     $       'AND THE'/' ',30X,'FUNCTIONING OF THIS PROGRAM AND ',
     $       'RELATED PROGRAM MATERIAL , NOR  SHALL  THE')
   13 FORMAT(' ',30X,'FACT OF DISTRIBUTION CONSTITUTE ANY SUCH ',
     $       'WARRANTY.   NO RESPONSIBILITY  IS'/' ',30X,'ASSUMED ',
     $       'BY THE AUTHORS OR THE UNIVERSITY OF MANITOBA IN ',
     $       'CONNECTION  THERE-'/' ',30X,'WITH.')
   14 FORMAT(///' ',35X,'VERSION 2',38X,'SEPTEMBER 1981')
C
      END
      SUBROUTINE DEFIN
C
C  SUBROUTINE DEFIN
C
C     THIS SUBROUTINE PRINTS OUT THE DEFINITION TABLES USED IN THIS
C     PROGRAM.  THIS IS DONE BEFORE THE SET OF INPUT JOBS, ARE
C     PROCESSED.
C
C  PARAMETER TABLE
C
      WRITE(6,10)
      WRITE(6,99)
      WRITE(6,11)
      WRITE(6,111)
      WRITE(6,99)
      WRITE(6,12)
      WRITE(6,99)
      WRITE(6,13)
      WRITE(6,99)
      WRITE(6,14)
      WRITE(6,99)
      WRITE(6,15)
      WRITE(6,99)
      WRITE(6,16)
      WRITE(6,99)
      WRITE(6,17)
```

```
C
C   STATISTICS TABLE
C
      WRITE(6,20)
      WRITE(6,99)
      WRITE(6,21)
      WRITE(6,99)
      WRITE(6,22)
      WRITE(6,99)
      WRITE(6,23)
      WRITE(6,99)
      WRITE(6,24)
      WRITE(6,99)
      WRITE(6,25)
      WRITE(6,99)
      WRITE(6,26)
      WRITE(6,99)
      WRITE(6,27)
      WRITE(6,99)
      WRITE(6,28)
      WRITE(6,99)
      WRITE(6,29)
C
C   EXPECTED ESTIMATES TABLE
C
      WRITE(6,30)
      WRITE(6,99)
      WRITE(6,31)
      WRITE(6,99)
      WRITE(6,32)
      WRITE(6,99)
      WRITE(6,33)
      WRITE(6,99)
      WRITE(6,34)
      WRITE(6,99)
      WRITE(6,35)
      WRITE(6,99)
      WRITE(6,36)
      WRITE(6,99)
      WRITE(6,37)
      WRITE(6,99)
      WRITE(6,38)
      WRITE(6,99)
      WRITE(6,39)
      WRITE(6,399)
C
C   BIAS TABLE
C
      WRITE(6,40)
      WRITE(6,99)
      WRITE(6,41)
      WRITE(6,99)
      WRITE(6,42)
      WRITE(6,99)
      WRITE(6,43)
      WRITE(6,99)
      WRITE(6,44)
      WRITE(6,99)
      WRITE(6,45)
      WRITE(6,99)
      WRITE(6,46)
      WRITE(6,99)
      WRITE(6,47)
      WRITE(6,99)
      WRITE(6,48)
      WRITE(6,99)
      WRITE(6,49)
      WRITE(6,99)
      WRITE(6,50)
      WRITE(6,99)
      WRITE(6,51)
      WRITE(6,99)
      WRITE(6,52)
      WRITE(6,399)
```

```
      RETURN
C
   10 FORMAT('1'///' ',10X,'PARAMETER TABLE DEFINITIONS'/' ',
     $        10X,'========= ===== ==========='///' ','LABEL',
     $        10X,'DESCRIPTION'//' ',80('-'))
   11 FORMAT(' ','B(I)',8X,'|',2X,'NUMBER OF ADDITIONS TO POPULATION ',
     $        'AFTER TIME (I) THAT ARE STILL ALIVE AT')
  111 FORMAT(' ',12X,'|',2X,'TIME (I+1).  IF INPUT VALUE IS -1, ',
     $        'B(I) IS CALCULATED SO THAT THE POPULATION'/' ',12X,'|',
     $        2X,'SIZE AT TIME (I+1) IS THE SAME AS TIME (I).  (I.E. ',
     $        'BIRTHS COMPENSATE FOR LOSSES)')
   12 FORMAT(' ','PHI(I)',6X,'|',2X,'PROBABILITY THAT AN ANIMAL ALIVE ',
     $        'JUST AFTER TIME (I), IS ALIVE AT'/' ',12X,'|',2X,
     $        'TIME (I+1), I.E. THE SURVIVAL RATE AT TIME (I).')
   13 FORMAT(' ','PI(I)',7X,'|',2X,'PROBABILITY THAT AN ANIMAL IS',
     $        ' CAPTURED AT TIME (I).')
   14 FORMAT(' ','ETA(I)',6X,'|',2X,'PROBABILITY THAT AN ANIMAL ',
     $        'CAPTURED AT TIME (I) IS RETURNED TO'/' ',12X,'|',2X,
     $        'THE POPULATION; I.E. IS NOT A LOSS-ON-CAPTURE.')
   15 FORMAT(' ','THETA(I)',4X,'|',2X,'PROBABILITY THAT A LIVING ',
     $        'TAGGED ANIMAL AT TIME(I) RETAINS ITS TAG'/' ',12X,
     $        '|',2X,'AT TIME (I+1).')
   16 FORMAT(' ','PSI(I)',6X,'|',2X,'PROBABILITY THAT A TAGGED ANIMAL,',
     $        ' ALIVE JUST AFTER TIME (I), IS SEEN'/' ',12X,'|',2X,
     $        'AGAIN AT LEAST ONCE WHILE STILL RETAINING ITS TAG.')
   17 FORMAT(' ','CHI(I)',6X,'|',2X,'PROBABILITY THAT AN ANIMAL,',
     $        ' ALIVE JUST AFTER TIME (I), IS SEEN AGAIN AT'/' ',12X,
     $        '|',2X,'LEAST ONCE.')
   20 FORMAT('1'///' ',10X,'STATISTICS TABLE DEFINITIONS'/' ',10X,
     $        '========== ===== ==========='///' ','LABEL',
     $         10X,'DESCRIPTION'//' ',80('-'))
   21 FORMAT(' ','CN(I)',7X,'|',2X,'SIZE OF POPULATION JUST BEFORE ',
     $        'TIME (I).')
   22 FORMAT(' ','N(I)',8X,'|',2X,'SIZE OF SAMPLE TAKEN AT TIME (I).')
   23 FORMAT(' ','CM(I)',7X,'|',2X,'NUMBER OF MARKED ANIMALS ALIVE ',
     $        'JUST BEFORE TIME (I).')
   24 FORMAT(' ','M(I)',8X,'|',2X,'NUMBER OF N(I) THAT ARE MARKED.')
   25 FORMAT(' ','B(I)',8X,'|',2X,'NUMBER OF ADDITIONS (BIRTHS) TO ',
     $        'POPULATION BETWEEN TIMES (I) AND (I+1).')
   26 FORMAT(' ','S(I)',8X,'|',2X,'NUMBER OF ANIMALS, CAPTURED AT ',
     $        'TIME (I), THAT ARE RETURNED TO'/' ',12X,
     $        '|',2X,'THE POPULATION.')
   27 FORMAT(' ','R(I)',8X,'|',2X,'NUMBER OUT OF S(I) THAT ARE ',
     $        'RECAPTURED ONE OR MORE TIMES AFTER TIME'/' ',12X,'|',2X,
     $        '(I) (AND RETAIN THEIR TAG, AT LEAST UNTIL THE FIRST ',
     $        'RECAPTURE).')
   28 FORMAT(' ','Z(I)',8X,'|',2X,'NUMBER OF ANIMALS SEEN BEFORE ',
     $        'TIME (I), NOT SEEN AT (I), BUT SEEN AGAIN'/' ',12X,'|',
     $        2X,'AT LEAST ONCE (WHILE RETAINING ITS TAG) AFTER (I).')
   29 FORMAT(' ','ZP(I)',7X,'|',2X,'NUMBER OF ANIMALS NOT SEEN AT ',
     $        'TIME (I), BUT ARE SEEN AT LEAST ONCE'/' ',12X,'|',2X,
     $        'AFTER TIME(I).')
   30 FORMAT('1'///' ',10X,'EXPECTED ESTIMATES TABLE DEFINITIONS'/
     $        ' ',10X,'======== ========= ===== ==========='///' ',
     $        'LABEL',10X,'DESCRIPTION'//' ',80('-'))
   31 FORMAT(' ','NH(I)',7X,'|',2X,'EXPECTED ESTIMATE OF CN(I)')
   32 FORMAT(' ','SNH(I)',6X,'|',2X,'STANDARD ERROR OF NH(I)')
   33 FORMAT(' ','CVNH(I)',5X,'|',2X,'COEFFICIENT OF VARIATION ',
     $        'OF NH(I), EXPRESSED AS A PERCENTAGE'/' ',12X,
     $        '|',2X,'I.E. 100 * SNH(I) / NH(I).')
   34 FORMAT(' ','PHIH(I)',5X,'|',2X,'EXPECTED ESTIMATE OF PHI(I)')
   35 FORMAT(' ','SPHIH(I)',4X,'|',2X,'STANDARD ERROR OF PHIH(I)')
   36 FORMAT(' ','CVPHIH(I)',3X,'|',2X,'COEFFICIENT OF VARIATION OF ',
     $        'PHIH(I), EXPRESSED AS A PERCENTAGE')
   37 FORMAT(' ','BH(I)',7X,'|',2X,'EXPECTED ESTIMATE OF B(I)')
   38 FORMAT(' ','SBH(I)',6X,'|',2X,'STANDARD ERROR OF BH(I)')
   39 FORMAT(' ','CVBH(I)',5X,'|',2X,'COEFFICIENT OF VARIATION OF ',
     $        'BH(I), EXPRESSED AS A PERCENTAGE')
  399 FORMAT('-','NOTE :',9X,'A TABLE ENTRY CONTAINING ''**********'' ',
     $        'IMPLIES THAT THE VALUE WAS UNDEFINED OR TOO LARGE FOR ',
     $        'PRINTING.'/' ',15X,'A TABLE ENTRY OF 0.0 INDICATES AN ',
     $        'ESTIMATE WHICH IS NOT IDENTIFIABLE OR R(I) OR M(I) IS ',
     $        'BELOW ITS CUT-OFF VALUE.')
   40 FORMAT('1'///' ',10X,'BIAS TABLE DEFINITIONS'/
```

```
      $        ' ',10X,'==== ===== ==========='///' ','LABEL',10X,
      $        'DESCRIPTION'//' ',80('-'))
   41 FORMAT(' ','PHIH3(I)',4X,'|',2X,'CASE 3 CORRECTED PHIH(I):  ',
      $        'PHIH3(I) = PHIH(I) / THETA(I)')
   42 FORMAT(' ','SPHIH3(I)',3X,'|',2X,'CASE 3 CORRECTED SPHIH(I):  ',
      $        'SPHIH3(I) = SPHIH(I) / THETA(I)')
   43 FORMAT(' ','CVPHIH3(I)',2X,'|',2X,'CASE 3 CORRECTED CVPHIH(I):  ',
      $        'CVPHIH3(I) = CVPHIH(I)')
   44 FORMAT(' ','AB(NH)',6X,'|',2X,'ABSOLUTE BIAS OF NH(I)'/' ',12X,
      $        '|',2X,'I.E. NH(I) - CN(I).')
   45 FORMAT(' ','RB(NH)',6X,'|',2X,'RELATIVE BIAS OF NH(I), EXPRESSED',
      $        ' AS A PERCENTAGE'/' ',12X,
      $        '|',2X,'I.E. 100 * AB(NH) / CN(I).')
   46 FORMAT(' ','EB(NH)',6X,'|',2X,'EFFECTIVE BIAS OF NH(I), EXPESSED',
      $        ' AS A PERCENTAGE'/' ',12X,
      $        '|',2X,'I.E. 100 * AB(NH) / SNH(I).')
   47 FORMAT(' ','AB(PHIH)',4X,'|',2X,'ABSOLUTE BIAS OF PHIH(I)')
   48 FORMAT(' ','RB(PHIH)',4X,'|',2X,'RELATIVE BIAS OF PHIH(I), ',
      $        'EXPRESSED AS A PERCENTAGE')
   49 FORMAT(' ','EB(PHIH)',4X,'|',2X,'EFFECTIVE BIAS OF PHIH(I), ',
      $        'EXPRESSED AS A PERCENTAGE')
   50 FORMAT(' ','AB(BH)',6X,'|',2X,'ABSOLUTE BIAS OF BH(I)')
   51 FORMAT(' ','RB(BH)',6X,'|',2X,'RELATIVE BIAS OF BH(I), EXPRESSED',
      $        ' AS A PERCENTAGE')
   52 FORMAT(' ','EB(BH)',6X,'|',2X,'EFFECTIVE BIAS OF BH(I), ',
      $        'EXPRESSED AS A PERCENTAGE')
   99 FORMAT(' ',12X,'|'/' ',12X,'|')
      END
C$ENTRY
```