NOTE: this is   listing of the minimal revisions that were done to the original Terak FORTRAN code to work as a glass teletype on the Mac in 6/89.

```
C          file: MA.IN.TEXT vrsn: 103 rvsd: 27 Aug 81
C                main routine, block data, reset
C
C**********************************************************************
C
C  PROGRAM   :  EVOLVE
C
C  WRITTEN BY:  FRANK E. PRICE
C               BIOLOGY DEPARTMENT
C               HAMILTON COLLEGE
C               CLINTON, NEW YORK  13323
C               (315) 859-4387, (315) 853-8268
C
C               COPYRIGHT 1981 BY FRANK E. PRICE
C
C     THIS PROGRAM IS DISTRIBUTED ON AN "AS IS" BASIS. CAVEAT EMPTOR,
C  NO WARRANTY IS EXPRESSED OR IMPLIED.
C
C      EVOLVE IS A MONTE CARLO SIMULATION OF EVOLUTION OF A
C  DIALLELIC LOCUS WITH  ALLELES 'W' AND 'M'.  THE USER MAY VARY
C  POPULATION SIZE, AND RATES OF REPRODUCTION, SURVIVAL, EMIGRATION
C  AND IMMIGRATION OF EACH OF THE THREE GENOTYPES.  THUS, EVOLVE WILL
C  SIMULATE THE EFFECTS OF GENETIC DRIFT, GENE FLOW AND NATURAL
C  SELECTION, EITHER ONE AT A TIME OR TOGETHER.
C
C      MUCH INFORMATION HAS BEEN LEFT OUT OF THIS LISTING TO KEEP IT
C  SUCCINCT. SEE USERS' AND INSTRUCTORS' MANUALS FOR ADDITIONAL
C  INFORMATION.
C
C  MISCELLANEOUS NOTES:
C     1) FORTRAN DIALECT: DEC FORTRAN IV V2.1.  NOTE THE FOLLOWING:
C        A) USE OF 'D' IN PLACE OF 'C' IN COL. 1 DENOTES CONDITIONAL
C           COMPILE OF DEBUG STATEMENTS.
C        B) USE OF 'ERR=' EXTENSION IN READ STATEMENTS TO AVOID SYSTEM
C           IO CHECK ERRORS AND AWKWARD ABORTS.  THESE MUST
C           BE REVISED TO IMPROVE TRANSPORTABILITY.
C        C) USE OF "RAN", "SQRT", "ALOG" FORTRAN LIBRARY FUNCTIONS.
C     2) THIS VERSION IS RUNNING AN A TERAK (LSI-11 CPU) MICROCOMPUTER
C        UNDER THE RT-11 OPERATING SYSTEM.  INPUT IS VIA KEYBOARD,
C        COMMUNICATION FROM PROGRAM TO USER IS VIA CRT, PROGRAM OUTPUT
C        IS VIA 132-COLUMN LINE PRINTER.
C     3) CURRENTLY CONFIGURED FOR A 16-BIT CPU. IF EVOLVE IS TO BE
C        IMPLEMENTED ON CPU'S WITH SMALLER WORD SIZES SOME VARIABLES MAY
C        NEED TO BE CHANGED FROM INTEGER TO REAL TO AVOID INTEGER
C        OVERFLOW PROBLEMS.
C     4) EVOLVE USES UNIFORMLY AND NORMALLY DISTRIBUTED PSEUDO-RANDOM
C        NUMBERS (GENERATED BY EXTERNAL FUNCTION "RAN" AND INTERNAL
C        SUBROUTINE "GAUSS"). THE OVERALL RANDOMNESS IN THIS VERSION
C        IS PERHAPS A BIT LOW.  IT HAS BEEN KEPT SO FOR TEACHING AND
C        DEBUGGING PURPOSES.  THIS ASPECT OF THE PROGRAM NEEDS MORE
C        THOROUGH STUDY AND TESTING. THE RANDOMNESS COULD BE
C        SIGNIFICANTLY CHANGED; SEE "RNOTE" IN SUBROUTINES REPROD AND
C        SRVIVL FOR MORE ON MANNER AND PLACES WHERE RANDOMNESS MAY BE
C        CHANGED.
C     5) NOTES ON ARRAYS
C        A) 2-ELEMENT ARRAYS STORING INFORMATION ON ALLELES (E.G.,
C           ALSUR(2), ALFIT(2)): FIRST ELEMENT CORRESPONDS TO THE "W"
```

```
C           ALLELE, THE SECOND ELEMENT CORRESPONDS TO THE "M" ALLELE.
C        B) 3-ELEMENT ARRAYS STORING INFORMATION ON GENOTYPES (E.G.,
C           REPR8(3), SURVR8(3)): FIRST ELEMENT CORRESPONDS TO THE "WW"
C           GENOTYPE, THE SECOND TO THE "WM" GENOTYPE, THE THIRD TO
C           THE "MM" GENOTYPE.
C        C) RSLT(3,125) STORES THE RESULTS, I.E., THE NUMBER OF ADULTS
C           OF EACH GENOTYPE IN EACH GENERATION. THE GENOTYPE
C           IS INDICATED BY THE FIRST INDEX, THE NUMBER OF THE
C           GENERATION IS INDICATED BY THE SECOND INDEX.
C
C  NOTES ON POSSIBLE CHANGES
C     1) STATEMENT NUMBERS SHOULD BE CLEANED UP -- SET IN NUMERICAL
C        ORDER, INCREMENT BY 10'S.
C     2) CLEAN UP LEVELS OF INDENTATION
C     3) IO WILL HAVE TO BE MADE MORE TRANSPORTABLE.
C     4) TOO MUCH OUTPUT IS SENT TO LINE PRINTER.  SHOULD MODIFY TO DO
C        GENERATIONS REQUESTED, DISPLAY SUMMARY TO CRT, ASK USER WHETHER
C        SUMMARIES &/OR GRAPHS ARE NEEDED.  USER SHOULD BE ABLE TO MOVE
C        TO & FROM SUMMARIES AND THE VARIOUS GRAPHS ON THE CRT BEFORE
C        SENDING HARD COPY TO LINE PRINTER IF NEEDED.  IMPLEMENTING
C        THIS FEATURE WILL PROBABLY REQUIRE ANOTHER SUBROUTINE
C        ("DISPLA"?) TO CONTROL MENUES & CALL UP VARIOUS GRAPHS.  WILL
C        NEED TO ADD VARIOUS GLOBAL VARIABLES (PAGESIZE, LINESIZE &
C        #LINESPRINTED, ETC.) TO PAGE SUMRYS OUTPUT, SET GRAPH SIZES TO,
C        FOR E.G., 80 X 24 AND 66 X 132, ETC.
C
C**********************************************************************
C**BEGIN MAIN ROUTINE
C
C   FUNCTION: CONTROL OVERALL FLOW OF THE PROGRAM
C
C   CALLS: INPUT  - INPUT AND CHECK PARAMETERS FOR RUN; RETURNS ALL
C                   VARIABLES NEEDED FOR A RUN.
C          REPROD - DETERMINE NUMBER OF JUVENILES IN CURRENT
C                   GENERATION; RETURNS "JUV" ARRAY.
C          SRVIVL - DETERMINE NUMBER OF ADULTS IN CURRENT GENERATION
C                   FOLLOWING EMIGRATION, MORTALITY & IMMIGRATION;
C                   CRASH EXCESSIVELY LARGE POPULATION; RETURNS
C                   NUMBER OF ADULTS IN "RSLTS" ARRAY.
C          SUMRYS - PRINT SUMMARIES OF GENERATIONS
C          GRAPH  - PRINT VARIOUS GRAPHS OF RESULTS
C          YESNO  - READ & TEST YES/NO RESPONSES; RETURNS EITHER "Y" OR
C                   "N" IN VARIABLE "ANS".
C          RESET  - REINITIALIZE VARIABLES BEFORE A NEW RUN.
C======================================================================
      INCLUDE 'COMMON.f'
C----------------------------------------------------------------------
C     LOCAL DECLARATIONS
      INTEGER ALMFLG, ALWFLG, ATOT, ICOUNT, POPFLG, SCRCNT
C======================================================================
C
C  INPUT DATA, SET ALLELE & POP.  FLAGS & PRINT SUMMARY OF INITIAL POP.
C----------------------------------------------------------------------
C   Change scrolling of output window to single line scrolling
      CALL OutWindowScroll(9999)
C
      WRITE (OUTSCR,10)
   10    FORMAT ('1',80('*'),/,20X,'EVOLVE version 3.1, 27 Aug. 1981',/,
     $      80('*'),/)
  100 CALL INPUT
```

```
D
D     WRITE(OUTLP,200) IGEN,(RSLT(K,IGEN),K=1,3)
D 200 FORMAT(' On return from INPUT:',/,' IGEN = ',I4,' RSLT = ',3(I7))
D     WRITE(OUTLP,205) MXPOP1,MXPOP2
D 205 FORMAT(' K = ',I5,10X,'POST-K = ',I5)
D     WRITE(OUTLP,210) (SURVR8(K),K=1,3)
D 210 FORMAT(' SURVR8 = ',3(F10.2))
D     WRITE(OUTLP,215) (REPR8(K),K=1,3)
D 215 FORMAT(' REPR8  = ',3(F10.2))
D     WRITE(OUTLP,220) (INFLO(K),K=1,3)
D 220 FORMAT(' INFLO  = ',3(I7))
D     WRITE(OUTLP,225) (OUTFLO(K),K=1,3)
D 225 FORMAT(' OUTFLO = ',3(I7))
D     WRITE(OUTLP,230) NOGENS, SUMRY, NEXT
D 230 FORMAT(' NOGENS = ',I7,'SUMRY = ',I7,'NEXT = ',A4)
D     WRITE(OUTLP,235) IGEN
D 235 FORMAT(' IGEN = ',I4,/)
C
C     INITIALIZE FLAGS FOR ALLELE & POPULATION EXTINCTION,  COUNTER FOR
C        SUMMARIES, HEADER FOR PROGRESS REPORT TO SCREEN.
  120 ALWFLG = 0
      ALMFLG = 0
      POPFLG = 0
      ICOUNT = 0
      SCRCNT = 1
      IF (IGEN .GT. 1) GO TO 300
      SUMGEN = IGEN
      PRINT2 = OUTLP
      CALL SUMRYS
  300 WRITE (OUTSCR,301)
  301    FORMAT ('0',X,'Doing generation:')
C
C-----------------------------------------------------------------------
C  DO NUMBER OF GENERATIONS REQUESTED
C-----------------------------------------------------------------------
      DO 390 I=1,NOGENS
C
C        TEST THAT # GENS DONE < 126; CONTINUE IF TRUE, EXIT IF FALSE.
         IF (IGEN .GT. 125) GO TO 400
C
C        WRITE PROGRESS REPORT TO SCREEN
         IGEN = IGEN + 1
         SCRCNT = SCRCNT + 1
  302    WRITE (OUTSCR,303) IGEN
  303       FORMAT (2X,I3,$)
         IF (SCRCNT .LE. 19) GO TO 305
         SCRCNT = 0
C
C      REPRODUCTION AND SURVIVAL - HEART OF THE PROGRAM
  305    CALL REPROD
         CALL SRVIVL
C
C    SUMMARIES (PRINTING THESE TAKES A LOT OF TIME)
         IF (SUMRY .EQ. 0) GO TO 309
         ICOUNT = ICOUNT + 1
         IF (SUMRY .NE. ICOUNT) GO TO 309
         PRINT2 = OUTLP
         SUMGEN = IGEN
         CALL SUMRYS
         ICOUNT = 0
  309    CONTINUE
C
C        TEST FOR EXTINCTIONS & PRINT MESSAGE IF HAS JUST
C           HAPPENED (THESE COULD GO INTO A SUBROUTINE ["XTNCTN"?]
C           TO CLARIFY LOGIC & CLEAN UP MAIN ROUTINE)
C
C        TEST FOR ALLELE EXTINCTION
         IF ((RSLT(1,IGEN) .GT. 0) .OR. (RSLT(2,IGEN) .GT. 0))
     $   GO TO 320
         ALWFLG = ALWFLG + 1
         IF (ALWFLG .GT. 1) GO TO 325
         WRITE (OUTSCR,310) IGEN
  310    FORMAT ('0',/,' W ALLELE BECAME EXTINCT IN GENERATION ',I3)
         WRITE (OUTLP,310) IGEN
         GO TO 325
  320    ALWFLG = 0
C
  325    IF ((RSLT(3,IGEN) .GT. 0) .OR. (RSLT(2,IGEN) .GT. 0))
     1   GO TO 340
         ALMFLG = ALMFLG + 1
         IF (ALMFLG .GT. 1) GO TO 350
         WRITE (OUTSCR,330) IGEN
  330    FORMAT ('0',/,' M ALLELE BECAME EXTINCT IN GENERATION ',I3)
         WRITE (OUTLP,330) IGEN
         GO TO 350
  340    ALMFLG = 0
C
C
C        TEST FOR POPULATION EXTINCTION
  350    ATOT = RSLT(1,IGEN) + RSLT(2,IGEN) + RSLT(3,IGEN)
         IF (ATOT .GT. 0) GO TO 370
         POPFLG = POPFLG + 1
         IF (POPFLG .GT. 1) GO TO 390
         WRITE (OUTSCR,360) IGEN
  360    FORMAT ('0',/,' POPULATION BECAME EXTINCT IN GENERATION ',I3)
         WRITE (OUTLP,360) IGEN
         GO TO 390
  370    POPFLG = 0
C
C
  390 CONTINUE
      WRITE (OUTSCR,395)
  395    FORMAT ('0',/)
C
C----------------------------------------------------------------------
C  CLEANUP AFTER GENERATIONS HAVE BEEN FINISHED
C     (THIS WOULD BE THE PLACE TO ADD THE "DISPLA" SUBROUTINE; MUCH OF
C      THIS CODE WOULD GO INTO THAT SUBROUTINE)
C----------------------------------------------------------------------
C
C     DETERMINE WHETHER GRAPHS OR PARAMETER CHANGE DESIRED
  400 IF (NEXT .EQ. 'GR') GO TO 500
      IF (NEXT .EQ. 'CH') GO TO 100
C
C     CALL SUMRYS, IF REQUESTED
  500 IF ((SUMRY .NE. 0) .AND. (ICOUNT .EQ. 0)) GO TO 600
      SUMGEN = IGEN
      PRINT2 = OUTLP
      CALL SUMRYS
C
```

```
C     PRINT 1ST & LAST GENERATIONS TO SCREEN, ASK IF GRAPHS DESIRED,
C     CALL GRAPH IF NEEDED
  600 SUMGEN = 1
      PRINT2 = OUTSCR
      CALL SUMRYS
      SUMGEN = IGEN
      CALL SUMRYS
      WRITE (OUTSCR, 610)
  610 FORMAT (' Do you wish graphs (Y/N)? ',$)
      CALL YESNO (ANS)
      IF (ANS .EQ. 'N') GO TO 700
      IF (GRF(1) .EQ. '  ') GO TO 700
      CALL GRAPH
C
C     ASK FOR CONTINUATION.  IF NO, STOP; IF YES, RESET GENERATION
C     NUMBER AND RESULTS ARRAY
  700 WRITE (OUTLP,710)
  710    FORMAT (1H1)
      WRITE (OUTSCR,720)
  720 FORMAT (//,3X,'Run finished.  Do you wish to make another ',
     +'run (Y/N)? ',$)
      CALL YESNO (ANS)
      IF (ANS .EQ. 'N') GO TO 999
         CALL RESET
         GO TO 100
C
  999 STOP
      END
C
C**********************************************************************

C**********************************************************************
C
      BLOCK DATA
C      SUBROUTINE INITIALIZEDATA
C
C   FUNCTION: INITIALIZES VARIABLES IN COMMON BLOCKS
C
C======================================================================
      INCLUDE 'COMMON.f'
C======================================================================
C *** INITIALIZE COMMON VARIABLES
C
      DATA   ALSUR/2*0./,  AM/0./,  ANS/' '/,  ASELCO/2*0./,
     1   FITAL/2*0./,  FITGNT/3*0./,  FRQMP/0./,  GNTSUR/3*0./,
     2   GRF/10*'  '/,  GSELCO/3*0./,  IGEN/1/,  IN/5/,  INFLO/3*0/,
     3   IY/12345/,  IZ/12345/,  JUV/3*0./,  LGEN/0/,  MATES/6*0/,
     4   MXPOP1/0/,  MXPOP2/0/,  NEXT/'  '/,  NOADLT/3*0/,  NOGENS/0/,
     5   NOOUT/3*0./,  NOYONG/6*0./,  OBSFR/3*0./,  OUTFLO/3*0/
      DATA   OUTLP/6/,  OUTSCR/6/,  PRINT2/6/,  REPR8/3*0./,
     1   RSLT/375*0/,  S/0./,  SUMGEN/1/,  SUMRY/0/,  SURVR8/3*0./,
     2   TITLE/40*'  '/,  TOADLT/0/,  V/0./
      END
C**********************************************************************
C**********************************************************************
C
      SUBROUTINE RESET
C
C   FUNCTION: REINITIALIZES VARIABLES FOLLOWING A RUN.
C
```

```
C   NOTES:
C      1) APPROPRIATE ARRAYS AND VARIABLES ARE RETURNED TO VALUES IN
C         BLOCK DATA, EXCEPT THAT FIRST GENERATION (INITIAL POPULATION)
C         IN RESULTS ARRAY RETAINS THE VALUES INPUT BY THE USER.
C      2) VARIABLES INPUT BY USER RETAIN THEIR VALUES SO USER CAN MAKE
C         REPEAT RUNS WITHOUT REENTERING ALL DATA. HOPEFULLY, ANY
C         REMAINING VARIABLES NOT RESET ARE NOT IMPORTANT.
C      3) ALL MAJOR VARIABLES ARE IN COMMON.
C
C   CALLED FROM: MAIN
C   CALLS: NOTHING
C======================================================================
      INCLUDE 'COMMON.f'
C======================================================================
C
C     Type *,'In Reset'
      DO 100 I = 1,3
         DO 10 J = 2,125
            RSLT(I,J) = 0
   10    CONTINUE
         JUV(I) = 0.
         NOOUT(I) = 0
C         NOIN(I) = 0
  100 CONTINUE
      DO 200 J = 1,6
         MATES(J) = 0
         NOYONG(J) = 0.
  200    CONTINUE
      IZ = 12345
      IGEN = 1
      RETURN
      END

C***********************************************************************
C          file: in.pu   vrsn: 145   rvsd: 27 Aug 81
C                subroutines input, yesno, inpsum
C**********************************************************************
C
      SUBROUTINE INPUT
C
C  FUNCTION: THIS SUBROUTINE READS, CHECKS & PRINTS DATA INPUT.
C
C   SYNOPSIS:
C      1) THE FOLLOWING VARIABLES (ALL IN COMMON) ARE READ IN:
C         TITLE  - 80 CHARACTER (40A2) TITLE FOR OUTPUT
C         IY     - RANDOM NUMBER SEED; 0->32760
C         RSLT - NUMBER OF EACH GENOTYPE IN FIRST GENERATION; ARRAY
C            ELEMENTS (1,1), (2,1), (3,1); EACH 0->4000, TOTAL < 4001
C         MXPOP1 - CARRYING CAPACITY, MAXIMUM PERMISSIBLE POPULATION
C            SIZE; 10->5000
C         MXPOP2 - POST-CRASH POPULATION SIZE; 2->4000
C         SURVR8 - SURVIVAL RATE OF EACH GENOTYPE; 0.0->100.0
C         REPR8  - REPRODUCTIVE RATE OF EACH GENOTYPE; 0.0->10.0
C         INFLO  - NUMBER OF EACH GENOTYPE IMMIGRATING PER GENERATION;
C            0->2000
C         OUTFLO - % OF EACH GENOTYPE EMIGRATING PER GENERATION; 0->100
C         NOGENS - NUMBER OF GENERATIONS TO BE RUN WITH THESE DATA;
C            1->124
C         SUMRY  - NUMBER OF GENERATIONS BETWEEN PRINTED SUMMARIES;
C            0 = NO SUMMARIES, 1->99
```

```
C         NEXT   - WHAT TO DO AFTER GENERATIONS ARE FINISHED;
C            "CH" = CHANGE PARAMETERS, "GR" = GRAPH RESULTS
C         GRF    - UP TO 10 2-LETTER CODES FOR TYPES OF GRAPHS DESIRED;
C            SEE USERS' MANUAL FOR SPECIFICS
C      2) USER IS PROMPTED FOR EACH DATUM, WHICH IS CHECKED AGAINST
C         LIMITS & AN ERROR MESSAGE PRINTED IF NEEDED. USER CAN THEN
C         REENTER DATA.
C      3) AFTER ALL DATA ARE ENTERED, SUBROUTINE INPSUM IS CALLED TO
C         PRINT A SUMMARY TO THE SCREEN FOR THE USER'S INSPECTION.
C      4) IF DATA ARE OK, A SUMMARY IS PRINTED TO THE LINE PRINTER BY
C         SUBROUTINE INPSUM AND CONTROL RETURNS TO MAIN.
C      5) IF USER DETECTS AN ERROR, VARIABLE "FIX" IS SET TO 'Y' AND
C         CONTROL RETURNS TO THE TOP OF SUBROUTINE INPUT. VALUES OF
C         DATA ARE DISPLAYED & THE USER PROMPED TO CORRECT THEM IF
C         NECESSARY.
C      6) IF VAR. "NEXT" = "CH", SUBROUTINE INPUT HAS BEEN CALLED
C         BEFORE, SO "IY" AND INITIAL POPULATION SHOULD NOT BE CHANGED;
C         CONTROL JUMPS TO CARRYING CAPACITY.
C
C   NOTES:
C      1) THIS ROUTINE RELIES HEAVILY ON THE "ERR=" DEC FORTRAN
C         EXTENSION TO AVOID AWKWARD SYSTEM ABORTS. THIS WILL NEED
C         TO BE CHANGED TO IMPROVE TRANSPORTABILITY.
C      2) THE RANDOM NUMBER SEED MAY NEED TO BE CHANGED FOR OTHER
C         SYSTEMS. THE SEED COULD BE SET IN OTHER WAYS (E.G., FROM A
C         SYSTEM CLOCK), BUT I HAVE FOUND IT USEFUL TO HAVE STUDENTS
C         BE ABLE TO REPEAT RUNS EXACTLY.
C
C   CALLED FROM: MAIN
C   CALLS: YESNO  - TO READ AND CHECK YES/NO INPUT; RETURNS EITHER "Y"
C                   OR "N" IN VARIABLE "ANS".
C          INPSUM - TO PRINT SUMMARIES OF DATA TO SCREEN
C                   (IF "PRINT2" = "OUTSCR") OR TO LINE PRINTER
C                   (IF "PRINT2" = "OUTLP")
C======================================================================
      INCLUDE 'COMMON.f'
C----------------------------------------------------------------------
C  LOCAL DECLARATIONS
      INTEGER  ATOT, GNOT(3), IFLOW, TGENS
      REAL  RRATE, SRATE
      STRING*2 FIX,  GRFTST(11)
      DATA     ATOT/0/,  FIX/'N'/,
     1   GNOT/'WW','WM','MM'/,
     2   GRFTST/'  ','AF','GF','PO','WW','WM','MM','AW','GW','AS',
     3      'GS'/,
     4   RRATE/0./, SRATE/0./
C======================================================================
C
C     Type *,'In Input'
C
C     IF NEXT IS 'CH', SUBROUTINE INPUT HAS ALREADY BEEN
C     CALLED, JUMP OVER INITIAL PARAMETERS TO CARRYING CAPACITY
   10 IF (FIX .EQ. 'Y') GO TO 100
      IF (NEXT .EQ. 'CH') GO TO 500
C
C --------------------------------------------------------------------
C  INPUT DATA
C --------------------------------------------------------------------
C
C     TITLE FOR OUTPUT
```

```
C
C        (THERE MUST BE A MORE ELEGANT WAY TO TEST FOR BLANK TITLE)
  100 DO 102 I=1,40
         IF (TITLE(I) .EQ. '  ') GO TO 102
         GO TO 108
  102    CONTINUE
      GO TO 120
C
  108 WRITE (OUTSCR,110) TITLE
  110 FORMAT (/,5X,'Title is:'/,X,40A2)
      WRITE (OUTSCR,115)
  115 FORMAT (5X,'Is this o.k. (Y/N)? ',$)
      CALL YESNO (ANS)
      IF (ANS .EQ. 'Y') GO TO 200
C
  120 WRITE (OUTSCR,130)
  130 FORMAT ('0',5X,'Enter a title (79 characters max) for output:')
      READ (IN,140) TITLE
  140 FORMAT(40A2)
D     WRITE (OUTSCR,150)TITLE
D 150 FORMAT ('0 1. ', 40A2)
C---------------------------------------------------------------------
C
C  RANDK SEED
C
C     THE SEED SIZE MAY NEED TO BE CHANGED FOR DIFFERENT COMPUTERS
  200 IF (IY .EQ. 12345) GO TO 220
         WRITE (OUTSCR,210) IY
  210       FORMAT (/,5X,'Seed is: ',I5)
         WRITE (OUTSCR,115)
         CALL YESNO (ANS)
         IF (ANS .EQ. 'Y') GO TO 300
  220 WRITE (OUTSCR,230,ERR=250)
  230    FORMAT (/,5X,'Enter random number seed (0-32760): ',$)
      READ (IN,240,ERR=250) IY
  240 FORMAT (I5)
C
C     TEST: RANDK SEED MUST BE BETWEEN 00000 AND 32760
      IF ((IY .GE. 00000) .AND. (IY .LE. 32760)) GO TO 270
  250    WRITE (OUTSCR,260)
  260       FORMAT (' *****ERROR: Seed is incorrect.',/,13X,
     $       'it must be an integer between 0 and 32760)')
         WRITE (OUTSCR,265)
  265       FORMAT (13X,'Please try again.')
         GO TO 220
  270 CONTINUE
C
D     WRITE (OUTSCR,280) IY
D 280 FORMAT ('0 2. RANDOM NUMBER SEED   : ',I5)
C----------------------------------------------------------------------
C
C  INITIAL POPULATION
C
  300 ATOT = RSLT(1,1) + RSLT(2,1) + RSLT(3,1)
      IF (ATOT  .EQ. 0)  GO TO 320
         WRITE (OUTSCR,310) RSLT(1,1), RSLT(2,1), RSLT(3,1), ATOT
  310       FORMAT (/,5X,'The initial population is: WW = ',I4,
     $      ' WM = ',I4,' MM = ',I4,' Total = ',I4)
         WRITE (OUTSCR,115)
         CALL YESNO (ANS)
         IF (ANS .EQ. 'Y') GO TO 500
C
  320 DO 380 I=1,3
  330    WRITE (OUTSCR,340) GNOT(I)
  340       FORMAT (/,5X,'Enter initial number of ',A2,' genotype ',
     $      '(0-4000): ',$)
         READ (IN,350,ERR=360) RSLT(I,1)
  350    FORMAT (I4)
         IF ((RSLT(I,1) .GE. 0) .AND. (RSLT(I,1) .LE. 4000)) GO TO 380
  360       WRITE (OUTSCR,370) GNOT(I)
  370          FORMAT (' *****ERROR: Initial number of ',A2,
     $         ' genotype is incorrect. ',/,13X,'It must be an ',
     $         'integer between 0 and 4000.')
            WRITE (OUTSCR,265)
            GO TO 330
  380    CONTINUE
C
C     TEST: INITIAL POPULATION MUST TOTAL BETWEEN 2 AND 4000 INDIVIDUALS
      ATOT = RSLT(1,1) + RSLT(2,1) + RSLT(3,1)
      IF ((ATOT .GE. 2) .AND. (ATOT .LE. 4000)) GO TO 400
         WRITE (OUTSCR,390)
  390       FORMAT (' *****ERROR: Initial population is out of ',
     $       'range.',/,13X,'The total initial population must be ',
     $       'between 2 and 4000')
         WRITE (OUTSCR,265)
         GO TO 320
  400 CONTINUE
C
D     WRITE (OUTSCR,410) (RSLT(I,1),I=1,3), ATOT
D 410    FORMAT ('0 3. INITIAL POPULATION   : WW = ',I4,2X,'WM = ',I4,
D    $   2X,'MM = ',I4,2X,'TOTAL = ',I4)
C----------------------------------------------------------------------
C
C  CARRYING CAPACITY & POST-CRASH POP SIZE
C
  500 IF ((MXPOP1 .EQ. 0) .AND. (MXPOP2 .EQ. 0)) GO TO 509
         WRITE (OUTSCR,501) MXPOP1, MXPOP2
  501    FORMAT (/,5X,'Carrying capacity is: ',I4,' Post-crash level ',
     $   ' is:',I4)
         WRITE (OUTSCR,115)
         CALL YESNO (ANS)
         IF (ANS .EQ. 'Y') GO TO 600
C......................................................................
C
  509    WRITE (OUTSCR,510)
  510    FORMAT (/,5X,'Enter carrying capacity (10-5000): ',$)
      READ (IN,520,ERR=525) MXPOP1
  520    FORMAT (I4)
C
C     TEST:  CARRYING CAPACITY MUST BE BETWEEN 10 AND 5000 INDIVIDUALS
      IF ((MXPOP1 .GE. 10) .AND. (MXPOP1 .LE. 5000)) GO TO 540
  525    WRITE (OUTSCR,530)
  530       FORMAT (' *****ERROR: Carrying capacity is incorrect.',/,13X,
     $       'it must be an integer between 10 and 5000.')
         WRITE (OUTSCR,265)
         GO TO 509
C......................................................................
C
  540 WRITE (OUTSCR,550)
  550    FORMAT (/,5X,'Enter post-crash population size (2-4000): ',$)
```

```
      READ (IN,560,ERR=575) MXPOP2
  560    FORMAT (I4)
C
C     TEST:  POST-CRASH SIZE MUST BE BETWEEN 2 AND 4000 INDIVIDUALS
  570 IF ((MXPOP2 .GE. 2) .AND. (MXPOP2 .LE. 4000)) GO TO 590
  575    WRITE (OUTSCR,580)
  580    FORMAT (' *****ERROR: Post-crash pop. size is incorrect.',/,
     $   13X,'It must be an integer between 2 and 4000.')
         WRITE (OUTSCR,265)
         GO TO 540
C......................................................................
C
C     TEST:  CARRYING CAPACITY MUST BE GREATER THAN POST-CRASH SIZE
  590 IF (MXPOP1 .GT. MXPOP2) GO TO 594
         WRITE (OUTSCR,592)
  592    FORMAT (' *****ERROR: Post-crash pop. size is greater than',
     1   ' carrying capacity.',/13X,'Please reenter both parameters.')
         GO TO 509
  594 CONTINUE
C
D        WRITE (OUTSCR,596) MXPOP1, MXPOP2
D 596    FORMAT ('  4. CARRYING CAPACITY    : ',I4,/,4X,
D    $   ' POST-CRASH POP. SIZE : ', I4)
C----------------------------------------------------------------------
C
C  SURVIVAL & REPRODUCTIVE RATES OF EACH GENOTYPE
C
  600 RRATE = REPR8(1) + REPR8(2) + REPR8(3)
      SRATE = SURVR8(1) + SURVR8(2) + SURVR8(3)
      IF ((RRATE .EQ. 0.) .AND. (SRATE .EQ. 0.)) GO TO 700
         WRITE (OUTSCR,620) SURVR8(1), SURVR8(2), SURVR8(3), REPR8(1),
     $      REPR8(2), REPR8(3)
  620    FORMAT (/,5X,'Survival rates are:     WW = ',F5.1,' WM = ',
     $      F5.1,' MM = ',F5.1,/,5X,'Reproductive rates are: WW =  ',
     $      F4.1,' WM =  ',F4.1,' MM = ',F4.1,/,5X,'Are survival ',
     $      'rates o.k. (Y/N)?',$)
         CALL YESNO (ANS)
         IF (ANS .EQ. 'Y') GO TO 760
C......................................................................
C
C     SURVIVAL RATES
  700 DO 750 I=1,3
  710    WRITE (OUTSCR,715) GNOT(I)
  715    FORMAT (/,5X,'Enter survival rate of genotype ',A2,
     $   ' (000.0-100.0%): ',$)
         READ (IN,720,ERR=730) SURVR8(I)
  720    FORMAT (F5.1)
         IF ((SURVR8(I) .GE. 0.) .AND. (SURVR8(I) .LE. 100.)) GO TO 750
  730       WRITE (OUTSCR,735) GNOT(I)
  735       FORMAT (' *****ERROR: Survival rate of ',A2,' genotype ',
     $      'is incorrect.',/,13X,'It must be a decimal number ',
     $      'between 000.0 and 100.0%.')
            WRITE (OUTSCR,265)
            GO TO 710
  750    CONTINUE
C......................................................................
C
C  REPRODUCTIVE RATES OF EACH GENOTYPE
  760 IF (SRATE .EQ. 0.) GO TO 770
         WRITE (OUTSCR,765)
  765    FORMAT (/,5X,'Are reproductive rates o.k. (Y/N)? ',$)
         CALL YESNO (ANS)
         IF (ANS .EQ. 'Y') GO TO 800
C
  770 DO 790 I=1,3
  772    WRITE (OUTSCR,775) GNOT(I)
  775    FORMAT (/,5X,'Enter reproductive rate of genotype ',A2,
     $   ' (0.0-10.0): ',$)
         READ (IN,780,ERR=785) REPR8(I)
  780    FORMAT (F4.1)
         IF ((REPR8(I) .GE. 0.) .AND. (REPR8(I) .LE. 10.)) GO TO 790
  785       WRITE (OUTSCR,786) GNOT(I)
  786       FORMAT (' *****ERROR: Reproductive rate of genotype ',A2,
     $      ' is incorrect.',/,13X,'It must be a decimal number ',
     $      'between 00.0 and 10.0.')
            WRITE (OUTSCR,265)
            GO TO 772
  790    CONTINUE
C
D     WRITE (OUTSCR,795) (SURVR8(I), I=1,3)
D 795 FORMAT ('  5. GNOT. SURVIVAL RATES : WW = ',F5.1,'% WM = ',
D    $   F5.1,'% MM = ',F5.1,'%')
D     WRITE (OUTSCR,796) (REPR8(I), I=1,3)
D 796 FORMAT ('     GNOT. REPROD. RATES  : WW = ',F4.1,2X,'WM = ',
D    $   F4.1,2X,'MM = ',F4.1)
C----------------------------------------------------------------------
C
C  GENE FLOW
C
  800 IFLOW = 0
      DO 810 I=1,3
         IFLOW = INFLO(I) + OUTFLO(I)
  810    CONTINUE
      IF (IFLOW .EQ. 0) GO TO 830
      WRITE (OUTSCR,820) INFLO(1), INFLO(2), INFLO(3), OUTFLO(1),
     $   OUTFLO(2), OUTFLO(3)
  820    FORMAT (/,5X,'Gene flow: # in : WW = ',I4,' WM = ',I4,
     $      ' MM = ',I4,/16X,'% out: WW = ',I3,'  WM = ',I3,
     $      '  MM = ',I3)
      WRITE (OUTSCR,115)
      CALL YESNO (ANS)
      IF (ANS .EQ. 'Y') GO TO 1000
C
  830 WRITE (OUTSCR,840)
  840    FORMAT (/,5X,'Do you wish to have gene flow (Y/N)? ',$)
         CALL YESNO (ANS)
         IF (ANS .EQ. 'Y') GO TO 900
            DO 850 I=1,3
               INFLO(I) = 0
               OUTFLO(I) = 0
  850          CONTINUE
            GO TO 1000
C......................................................................
C
C     IMMIGRATION RATES
  900 DO 950 I=1,3
  910    WRITE (OUTSCR,915) GNOT(I)
  915    FORMAT (/,5X,'Enter number of ',A2,' individuals entering ',
     $   'per generation (0-4000): ',$)
         READ (IN,930,ERR=940) INFLO(I)
  930    FORMAT (I4)
         IF ((INFLO(I) .GE. 0) .AND. (INFLO(I) .LE. 4000)) GO TO 950
  940    WRITE (OUTSCR,945) GNOT(I)
  945    FORMAT (' *****ERROR: Number of ',A2,' individuals entering ',
     $   'is incorrect.',/,13X,'It must be an integer between 0 and ',
     $   '4000.')
         WRITE (OUTSCR,265)
         GO TO 910
  950    CONTINUE
C......................................................................
C
C     EMIGRATION RATES
      DO 990 I=1,3
  960    WRITE (OUTSCR,965) GNOT(I)
  965    FORMAT (/,5X,'Enter % of ',A2,' individuals leaving per ',
     $   'generation (0-100): ',$)
         READ (IN,970,ERR=980) OUTFLO(I)
  970    FORMAT (I3)
         IF ((OUTFLO(I) .GE. 0) .AND. (OUTFLO(I) .LE. 100)) GO TO 990
  980    WRITE (OUTSCR, 985) GNOT(I)
  985    FORMAT (' *****ERROR: % of ',A2,' individuals leaving is ',
     $   'incorrect.',/,13X,'It must be an integer getween 0 and 100.')
         WRITE (OUTSCR,265)
         GO TO 960
  990    CONTINUE
C.....................................................................
C
D     WRITE (OUTSCR,995) (INFLO(I), I=1,3)
D 995    FORMAT ('  6. FLOW: # IN PER GNOT. : WW =',I5,3X,'WM =',I5,
D    $   3X,'MM =',I5)
D     WRITE (OUTSCR,996) (OUTFLO(I), I=1,3)
D 996    FORMAT (11X,'% OUT PER GNOT.: WW = ',I3,3X,'WM = ',I3,3X,
D    $   'MM = ',I3)
C---------------------------------------------------------------------
C
C  NUMBER OF GENERATIONS
C
 1000 IF (NOGENS .EQ. 0) GO TO 1002
         WRITE (OUTSCR,1001) NOGENS
 1001    FORMAT (/,5X,'Model set to run for ',I3,' generations.')
      WRITE (OUTSCR,115)
      CALL YESNO (ANS)
      IF (ANS .EQ. 'Y') GO TO 1014
C
 1002 WRITE (OUTSCR,1003)
 1003    FORMAT (/,5X,'Enter number of generations you wish to run ',
     $   'with these data (1-124): ',$)
      READ (IN,1004,ERR=1006) NOGENS
 1004    FORMAT (I3)
      IF ((NOGENS .GE. 1) .AND. (NOGENS .LE. 124)) GO TO 1010
C
 1006    WRITE (OUTSCR,1008)
 1008    FORMAT (' *****ERROR: No. of generations is incorrect.',/,13X,
     $   'It must be an integer between 1 and 124.')
         WRITE (OUTSCR,265)
         GO TO 1002
C
 1010 TGENS = IGEN + NOGENS
      IF (TGENS .LE. 125) GO TO 1012
      WRITE (OUTSCR,1011)
 1011    FORMAT (' +++++WARNING: Total no. of generations requested ',
     $   'exceeds 125.',/,15X,'Run will stop at generation 125.')
      NOGENS = 125 - IGEN
C---------------------------------------------------------------------
C
C     INTERVAL BETWEEN SUMMARIES
C
 1012 IF (SUMRY .NE. 0) GO TO 1014
         WRITE (OUTSCR,1013)
 1013       FORMAT (/,5X,'Summaries will be printed for first and ',
     $         'last generations only.')
         GO TO 1016
C
 1014 WRITE (OUTSCR,1015) SUMRY
 1015    FORMAT (/,5X,'Summaries will be printed after every ',I2,
     $   ' generation(s).')
C
 1016 WRITE(OUTSCR,115)
         CALL YESNO (ANS)
         IF (ANS .EQ. 'Y') GO TO 1030
C
      WRITE (OUTSCR,1017)
 1017    FORMAT (/,5X,'How many generations do you wish between ',
     $   'summaries?',/,13X,'(Enter 0 for no summaries, or 1-99): ',$)
       READ (IN,1018,ERR=1020) SUMRY
 1018    FORMAT (I2)
       IF ((SUMRY .GE. 0) .AND. (SUMRY .LE. 99)) GO TO 1030
 1020    WRITE (OUTSCR,1022)
 1022    FORMAT (' *****ERROR: Summary request is incorrect.',/,13X,
     $   'It must be an integer between 0 and 99.')
         WRITE (OUTSCR,265)
         GO TO 1016
C---------------------------------------------------------------------
C
C     WHAT TO DO AFTER GENERATIONS ARE FINISHED
C
 1030 IF ((NEXT .NE. 'CH') .AND. (NEXT .NE. 'GR')) GO TO 1040
         IF (NEXT .EQ. 'GR') GO TO 1032
            WRITE (OUTSCR,1031)
 1031       FORMAT (/,5X,'The model is set to change parameters.')
            GO TO 1034
C
 1032 WRITE (OUTSCR,1033)
 1033    FORMAT (/,5X,'The model is set to print graphs.')
C
 1034 WRITE (OUTSCR,115)
      CALL YESNO (ANS)
      IF (ANS .EQ. 'Y') GO TO 1060
C
 1040 WRITE (OUTSCR,1042)
 1042     FORMAT (/,5X,'What do you wish to do after the generations ',
     $   'are finished?',/,13X,'Enter "CH" to change parameters or ',
     $   '"GR" to graph results: ',$)
         READ (IN,1044,ERR=1046) NEXT
 1044    FORMAT (A2)
         IF ((NEXT .EQ. 'CH') .OR. (NEXT .EQ. 'GR')) GO TO 1060
 1046       WRITE (OUTSCR,1048)
 1048       FORMAT (' *****ERROR: Request is incorrect.',/,13X,
     $      'It must be either "CH" or "GR".')
            WRITE (OUTSCR,265)
            GO TO 1040
 1060 CONTINUE
C
D     WRITE (OUTSCR,1070) NOGENS, SUMRY, NEXT
D1070    FORMAT (5X,I4,' GENERATIONS WILL BE RUN, ',/,5X,
D    $   'SUMMARIES WILL BE PRINTED EVERY ',I2,' GENERATIONS',/,5X,
D    $   'THEN: ',A2)
D     PAUSE 'HIT <RET> TO CONTINUE.'
C-----------------------------------------------------------------------
C
C  GRAPH REQUESTS
C
      IF (NEXT .NE. 'GR') GO TO 1400
C
C     PRINT MENU
 1200 IF (GRF(1) .EQ. '  ') GO TO 1210
         WRITE (OUTSCR,1201)
 1201    FORMAT (/,5X,'Graphs requested: ',$)
            WRITE (OUTSCR,1202) (GRF(I), I=1,10)
 1202       FORMAT (10X,10(A2,2X))
         WRITE (OUTSCR,115)
         CALL YESNO (ANS)
         IF (ANS .EQ. 'Y') GO TO 1400
C
 1210 WRITE (OUTSCR,1212)
 1212    FORMAT (/,5X,'You may request up to 10 of the following ',
     $   'graphs:',/,10X,'AF (allele frequency)       GF (genotype ',
     $   'frequency)')
      WRITE(OUTSCR,1215)
 1215    FORMAT(10X,'PO (population size)        WW   (number of ',
     $   'WW''S)',/,10X,'WM   (number of WM''S)       MM   (number of ',
     $   'MM''S)')
      WRITE (OUTSCR,1220)
 1220    FORMAT (10X,'AW (allele fitness)         GW (genotype ',
     $   'fitness)',/10X,'AS (allele selection coef.) GS (genotype ',
     $   'selection coef.)')
C
C     ENTER GRAPH REQUESTS
      WRITE (OUTSCR,1230)
 1230    FORMAT (/,5X,'Now enter the codes for the graphs you wish ',
     $   'printed',/,13X,'(Hit <RET> twice after your last request): ')
      DO 1280 I=1,10
 1235    WRITE (OUTSCR,1240) I
 1240    FORMAT (/,5X,'What is the code for graph number ',I2,'? ',$)
         READ (IN,1250,ERR=1260) GRF(I)
 1250    FORMAT (A4)
C
C        IF CURRENT REQUEST IS NOT BLANK, CHECK FOR ERRORS, IF IT IS
C        BLANK, ENSURE THAT REMAINING GRAPH REQUEST SLOTS ARE ALSO
C        BLANK.
         IF (GRF(I) .NE. GRFTST(1)) GO TO 1254
            II = I
            DO 1252 J=II,10
               GRF(I) = GRFTST(1)
 1252          CONTINUE
            GO TO 1290
C
 1254    DO 1255 J=2,11
            IF (GRF(I) .EQ. GRFTST(J)) GO TO 1280
 1255       CONTINUE
```

```
C
 1260    WRITE (OUTSCR,1270) I
 1270       FORMAT (' *****ERROR: Graph request # ',I2,' is ',
     $      'incorrect.',/13X,'It must be one of the following:',/,13X,
     $      '"AF", "GF", "PO", "WW", "WM", "MM",',/,13X,
     $      '"AW", "GW", "AS" OR "GS".')
            WRITE (OUTSCR, 265)
            GRF(I) = '  '
            GO TO 1235
C
 1280    CONTINUE
 1290 CONTINUE
C
D     WRITE (OUTSCR,1300) (GRF(I), I=1,10)
D1300 FORMAT (5X,'THE FOLLOWING GRAPHS WILL BE PRINTED:',10(2X,A2))
D     PAUSE 'TYPE <RET> TO CONTINUE.'
C======================================================================
C
C  PRINT SUMMARY TO SCREEN, ASK IF OK. IF YES, CONTINUE,
C     IF NO, REDO INPUT
C
 1400 PRINT2 = OUTSCR
      CALL INPSUM
C......................................................................
C
 1520 WRITE (OUTSCR,1530)
 1530   FORMAT ('0   Are these data o.k. (Y/N)?',/,5X,
     $  '(If answer is "N", input will restart)',/,5X,
     $  '(If answer is "Y", and nothing happens, be sure printer is',
     $  ' on line.): ',$)
      CALL YESNO (ANS)
      IF (ANS .EQ. 'Y') GO TO 1600
         FIX = 'Y'
         GO TO 10
 1600    FIX = 'N'
C----------------------------------------------------------------------
C
C   PRINT SUMMARY TO LINE PRINTER
      PRINT2 = OUTLP
      CALL INPSUM
C----------------------------------------------------------------------
C
 9999 RETURN
      END
C************************************************************************
C************************************************************************
C
      SUBROUTINE YESNO (ANS)
C
C   FUNCTION: READS AND CHECKS YES/NO INPUT, RETURNS EITHER "Y" OR "N" IN
C        VARIABLE "ANS".
C
C   NOTES:
C      1) THIS ROUTINE RELIES HEAVILY ON THE "ERR=" DEC FORTRAN
C         EXTENSION TO AVOID AWKWARD SYSTEM ABORTS. THIS WILL NEED
C         TO BE CHANGED TO IMPROVE TRANSPORTABILITY.
C
C   CALLED FROM:  INPUT, MAIN
C   CALLS: NOTHING
C========================================================================
```

```fortran
      INCLUDE 'COMMON.f'
C========================================================================
C     Type *,'In YesNo'
C
  100 READ (IN,110,ERR=120) ANS
  110    FORMAT (A1)
      IF ((ANS .EQ. 'Y') .OR. (ANS .EQ. 'N')) GO TO 140
  120     WRITE (OUTSCR,130)
  130        FORMAT (5X,' *****ERROR: Response is incorrect.',/,13X,
     $       'It must be either a "Y" or an "N".',/,13X,
     $       'Please try again:  ',$)
      GO TO 100
C
  140 RETURN
      END
C************************************************************************
C************************************************************************
C
      SUBROUTINE INPSUM
C
C   FUNCTION: PRINTS SUMMARY OF DATA INPUT BY USER.
C
C   NOTES:
C      1) OUTPUT GOES TO SCREEN IF "PRINT2" = "OUTSCR" AND TO LINE
C         PRINTER IF "PRINT2" = "OUTLP"
C
C   CALLED FROM:  INPUT
C   CALLS: NOTHING
C========================================================================
      INCLUDE 'COMMON.f'
C------------------------------------------------------------------------
C  LOCAL DECLARATIONS
      INTEGER ATOT
C========================================================================
C     Type *,'In Inpsum'
C
      IF (IGEN .GT. 1) GO TO 5
         WRITE (PRINT2,2)
    2       FORMAT (1H1)
    5 IF (PRINT2 .EQ. OUTLP) GO TO 15
          WRITE (PRINT2,10)
   10    FORMAT('0 The following is a summary of the data you ',
     $   'have just input:')
         GO TO 18
C
   15 IF (IGEN .GT. 1) GO TO 45
         WRITE (PRINT2,16)
   16    FORMAT(80(1H-),/,' The following is a summary of ',
     $   'the data you have input: ')
C
   18 WRITE (PRINT2,20)TITLE
   20    FORMAT (5X,'Title: ',/,X,40A2)
C
      WRITE (PRINT2,30) IY
   30    FORMAT (5X,'Random number seed:      ',I5)
C
      ATOT = RSLT(1,1) + RSLT(2,1) + RSLT(3,1)
      WRITE (PRINT2,40) ((RSLT(I,1), I=1,3), ATOT)
   40    FORMAT (5X,'Initial population:     WW = ',I4,3X,'WM = ',I4,
     $   3X,'MM = ',I4,3X,'TOTAL = ',I4)
```

```
C
   45 WRITE (PRINT2,46) IGEN
   46    FORMAT('0After generation ',I3,' parameters are as follows:')
C
      WRITE (PRINT2,50) MXPOP1, MXPOP2
   50    FORMAT (5X,'Carrying capacity:       ',I4,/,4X,
     $   ' Post-crash pop. size:    ', I4)
C
      WRITE (PRINT2,60) (SURVR8(I), I=1,3)
   60    FORMAT (5X,'Survival rates:           WW = ',F5.1,'% WM = ',
     $   F5.1,'% MM = ',F5.1,'%')
C
      WRITE (PRINT2,70) (REPR8(I), I=1,3)
   70    FORMAT (5X,'Reproductive rates:       WW =  ',
     $   F4.1,2X,'WM =  ',F4.1,2X,'MM =  ',F4.1)
C
      WRITE (PRINT2,80) INFLO(1), INFLO(2), INFLO(3), OUTFLO(1),
     $   OUTFLO(2), OUTFLO(3)
   80    FORMAT (5X,'Flow: # in per gen.:      WW =',I5,3X,'WM =',I5,3X,
     $   'MM =',I5,/,11X,'% out per gen.:     WW = ',I4,3X,'WM = ',
     $   I4,3X,'MM = ',I4)
C
      WRITE (PRINT2,90)
   90    FORMAT (/,5X,'Summaries:                 ',$)
      IF (SUMRY .GT. 0) GO TO 110
         WRITE (PRINT2,100)
  100        FORMAT ('+ First and last generations only')
         GO TO 130
C
  110 IF (SUMRY .GT. 1) GO TO 120
         WRITE (PRINT2,112)
  112       FORMAT ('+ Every generation')
         GO TO 130
C
  120 WRITE (PRINT2,122) SUMRY
  122    FORMAT ('+ Every ',I2,' generations')
C
  130 WRITE (PRINT2,140) NOGENS
  140    FORMAT (5X,'Generations to be run:   'I3,/,5X,
     $   'Then:                   ',$)
C
      IF (NEXT .NE. 'CH') GO TO 160
      WRITE (PRINT2,150)
  150    FORMAT ('+ Parameters will be changed ',/,80(1H-))
      GO TO 180
C
  160 WRITE (PRINT2,170) (GRF(I), I=1,10)
  170    FORMAT ('+ The following graphs will be printed:',/,28X,
     $   10(2X,A2),/,80(1H-))
  180 CONTINUE
      RETURN
      END
C
C************************************************************************

C
C**********************************************************************
C          file: MA.RE   vrsn: 29   rvsd: 29 May 81
C                subroutines reprod, m8few, m8many
C**********************************************************************
```

```
C
      SUBROUTINE REPROD
C
C   FUNCTION: CONTROLS MATING AND REPRODUCTION OF POPULATION, RETURNS
C      NUMBER OF JUVENILES OF EACH GENOTYPE IN "JUV" ARRAY.
C
C   SYNOPSIS:
C      1) REPROD ASSIGNS THE ADULTS OF THE PREVIOUS GENERATION TO THE
C         "NOADLT" ARRAY.
C      2) AFTER INITIALIZING VARIABLES, "REPROD" CALLS 1 OR 2 OTHER
C         ROUTINES ("M8FEW" OR "M8MANY") WHICH RANDOMLY MATE INDIVIDUALS
C         FROM "NOADLT" AND RETURN THE NUMBER OF EACH OF THE 6 POSSIBLE
C         COMBINATIONS OF MATES IN THE "MATES" ARRAY.
C      2) REPROD THEN CALCULATES THE NUMBER OF YOUNG FROM EACH TYPE OF
C         MATING FROM THE REPRODUCTIVE RATES ("REPR8") OF THE GENOTYPES
C         AND PLACES THEM IN THE "NOYONG" ARRAY).
C      3) FINALLY, THE NUMBER OF YOUNG OF EACH GENOTYPE ARE DETERMINED
C         FROM "NOYONG" ACCORDING TO MENDELIAN RATIOS AND PLACED INTO
C         THE "JUV" ARRAY WHICH IS RETURNED TO MAIN.
C
C   NOTES:
C      1) TWO METHODS ARE USED TO CALCULATE THE NUMBER OF EACH
C         COMBINATION OF MATES BECAUSE NEITHER WORKS WELL OVER THE FULL
C         RANGE OF POPULATION SIZES:
C         A) SUBROUTINE "M8FEW" USES THE PROPORTION OF A GENOTYPE AMONG
C           THE ADULTS TO DETERMINE ITS PROBABILITY OF BEING PICKED AS
C           A PARENT.  PAIRS OF PARENTS ARE RANDOMLY SELECTED WITHOUT
C           REPLACEMENT FROM THE "NOADLT" ARRAY AND PLACED INTO THE
C           APPROPRIATE CELL OF THE "MATES" ARRAY UNTIL "NOADLT" IS
C           EMPTY OR N = "CUTOFF" INDIVIDUALS HAVE BEEN PAIRED.
C         B) SUBROUTINE "M8MANY" USES "GNTFR", THE PROPORTION OF EACH
C           GENOTYPE REMAINING IN "NOADLT", TO DIRECTLY CALCULATE
C           THE PROBABLE NUMBER OF EACH MATING COMBINATION.
C      2) METHOD A) IS A MONTE CARLO ALGORITHM TO SIMULATE STOCHASTIC
C         COMBINATIONS IN SMALL POPULATIONS. HOWEVER, ITRESULTS IN
C         PROHIBITIVELY LONG RUNNING TIMES WHEN USED WITH LARGE
C         POPULATIONS.  THE RUNNING TIME OF METHOD B) IS ESSENTIALLY
C         CONSTANT REGARDLESS OF POPULATION SIZE, BUT CANNOT PROPERLY
C         SIMULATE MATING IN SMALL POPULATIONS.
C      3) IN SMALL POPULATIONS ONLY "M8FEW" IS CALLED.  FOR POPULATIONS
C         LARGER THAN "CUTOFF" INDIVIDUALS, "M8FEW" IS CALLED TO MATE
C         CUTOFF INDIVIDUALS AND "M8MANY" IS CALLED TO MATE THE REST.
C         THE RUN TIME OF EVOLVE COULD BE REDUCED SOMEWHAT BY REDUCING
C         THE SIZE OF "CUTOFF" IN BLOCK DATA (CURRENTLY SET = 200).
C         SUBROUTINE "M8FEW" COULD BE ELIMINATED IF ONLY DETERMINISTIC
C         MATING IS DESIRED; SUBROUTINE M8MANY COULD BE ELIMINATED IF
C         POPULATION LIMITS WERE REDUCED, RUN TIME WERE NOT IMPORTANT
C         &/OR MUCH DRIFT IS DESIRED.
C      4) THE "MATES" ARRAY HOLDS THE NUMBER OF INDIVIDUALS INVOLVED IN
C         EACH OF THE 6 COMBINATIONS OF PARENTS:
C         MATES(1) = # OF WW-WW MATINGS, MATES(2) = # OF WW-WM MATINGS,
C         MATES(3) = # OF WW-WW MATINGS, MATES(6) = # OF WW-WM MATINGS,
C         MATES(5) = # OF WW-WW MATINGS, MATES(6) = # OF WW-WM MATINGS,
C      5) THE "NOYONG" ARRAY HOLDS THE NUMBER OF YOUNG FROM EACH OF THE
C         6 COMBINATIONS OF PARENTS.  ARRAY ELEMENTS CORRESPOND TO THOSE
C         IN THE MATES ARRAY.
C      6) ALTHOUGH THE DATA ARE INTEGER, VARIABLES "JUV" AND "NOYONG"
C         ARE DECLARED TO BE REAL TO AVOID POSSIBLE INTEGER OVERFLOW IN
C         16-BIT MACHINES. THE VARIABLES POTENTIALLY COULD REACH VALUES
C         OF (REPR8 * MXPOP1) = (10.0 * 5000) = 50,000.
```

```
C
C   CALLED FROM: MAIN
C   CALLS: M8FEW  - RANDOMLY MATES N = "CUTOFF" INDIVIDUALS, RETURNS
C                   "MATES" ARRAY
C          M8MANY - RANDOMLY MATES REMAINDER OF POPULATION, ADDS THEM
C                   TO THE "MATES" ARRAY.
C          GAUSS  - RETURNS "V", A PSEUDORANDOM NUMBER WITH GIVEN MEAN
C                   AND STANDARD DEVIATION TO DETERMINE REPRODUCTIVE
C                   SUCCESS OF EACH COMBINATION OF MATES.
C======================================================================
      INCLUDE 'COMMON.f'
C----------------------------------------------------------------------
C  LOCAL DECLARATIONS
      INTEGER  PGEN
      REAL     X
C======================================================================
C     Type *,'In Reprod'
C
C  INITIALIZE
      PGEN = IGEN - 1
      DO 10 I=1,6
         MATES(I) = 0
         NOYONG(I) = 0.
   10 CONTINUE
      TOADLT = RSLT(1,PGEN) + RSLT(2,PGEN) + RSLT(3,PGEN)
      DO 20 I=1,3
         JUV(I) = 0.
         NOADLT(I) = RSLT(I,PGEN)
   20 CONTINUE
C
C     TEST FOR ZERO ADULTS. RETURN IF TRUE, CONTINUE IF FALSE
      IF (TOADLT .LE. 0) GO TO 999
C----------------------------------------------------------------------
C
C     DETERMINE NUMBER OF EACH COMBINATION OF PARENTAL GENOTYPES
C
      CALL M8FEW
      IF (TOADLT .LT. 1) GO TO 30
      CALL M8MANY
C----------------------------------------------------------------------
C
C     DETERMINE NUMBER OF YOUNG FROM EACH TYPE OF MATING: SUBROUTINE
C  GAUSS IS CALLED AND RETURNS A PSEUDORANDOM NUMBER ("V") WITH A MEAN
C  ("AM") EQUAL TO THE AVERAGE NUMBER OF YOUNG PER PARENT (REPR8(3)).
C  THE MEAN NUMBER OF YOUNG IS MULTIPLIED BY THE NUMBER OF PARENTS
C  (MATES(6)) TO DETERMINE THE TOTAL NUMBER OF YOUNG (NOYONG(6))
C  PRODUCED BY THOSE PARENTS.
C
C  RNOTE: A BETTER METHOD OF SETTING STANDARD DEVIATION OF GAUSS ("S")
C  MAY BE NEEDED.  THE CURRENT METHOD (S = AM / SQRT(X)) IS ESSENTIALLY
C  ANALOGOUS TO THE STANDARD ERROR OF THE MEAN AND ATTEMPTS TO REDUCE
C  THE VARIATION OF GAUSS' OUTPUT AS THE NUMBER OF INDIVIDUALS
C  INCREASES. THIS NEEDS THOUGHT.
C
C  THIS SECTION OF CODE (DOWN TO STATEMENT 90) COULD BE PUT INTO A
C  SUBROUTINE ("NUMYNG") TO IMPROVE PROGRAM CLARITY.
C......................................................................
C
C     NUMBER OF YOUNG FROM WW-WW MATINGS
   30 IF (MATES(1) .LT. 1) GO TO 40
```

```
      AM = REPR8(1)
D     WRITE (OUTSCR,170)
D 170 FORMAT(100('*'))
D 100 FORMAT ('0ENTER REPRODUCTION BLOCK WITH: MATES(1-6) = ',6(I6,1X),
D    1'REPR8(1-3) = ',3(F5.2,1X))
D     WRITE (OUTSCR,100) MATES(1), MATES(2), MATES(3), MATES(4),
D    1MATES(5), MATES(6), REPR8(1), REPR8(2), REPR8(3)
      X = MATES(1)
      IF (X .LT. 1.) X = 1.
      S = AM / SQRT(X)
      CALL GAUSS
      NOYONG(1) = MATES(1) * V + 0.5
D     WRITE (OUTSCR,110) NOYONG(1), V, S, X, AM
D 110 FORMAT('0NOYONG(1) = ',F6.0,2X,'V = ',F5.2,2X,'S = ',F5.2,2X,
D    1'X = ',F5.2,2X,'AM = ',F5.2,/,5X,'NOYONG(1) = MATES(1) * V + 0.5',
D    25X,'S = AM / SQRT(X)',5X,'X = MATES(1) OR 1.0',5X,
D    3' AM = REPR8(1)')
C......................................................................
C
C     NUMBER OF YOUNG FROM WW-WM MATINGS
   40 IF (MATES(2) .LT. 1) GO TO 50
      AM = (REPR8(1) + REPR8(2)) / 2
      X = MATES(2)
      IF (X .LT. 1.) X = 1.
      S = AM / SQRT(X)
      CALL GAUSS
      NOYONG(2) = MATES(2) * V + 0.5
D     WRITE (OUTSCR,120) NOYONG(2), V, S, X, AM
D 120 FORMAT('0NOYONG(2) = ',F6.0,2X,'V = ',F5.2,2X,'S = ',F5.2,2X,
D    1'X = ',F5.2,2X,'AM = ',F5.2,/,5X,'NOYONG(2) = MATES(2) * V + 0.5',
D    25X,'S = AM / SQRT(X)',5X,'X = MATES(2) OR 1.0',5X,
D    3' AM = (REPR8(1) + REPR8(2)) / 2')
C......................................................................
C
C     NUMBER OF YOUNG FROM WW-MM MATINGS
   50 IF (MATES(3) .LT. 1) GO TO 60
      AM = (REPR8(1) + REPR8(3)) / 2
      X = MATES(3)
      IF (X .LT. 1.) X = 1.
      S = AM / SQRT(X)
      CALL GAUSS
      NOYONG(3) = MATES(3) * V + 0.5
D     WRITE (OUTSCR,130) NOYONG(3), V, S, X, AM
D 130 FORMAT('0NOYONG(3) = ',F6.0,2X,'V = ',F5.2,2X,'S = ',F5.2,2X,
D    1'X = ',F5.2,2X,'AM = ',F5.2,/,5X,'NOYONG(3) = MATES(3) * V + 0.5',
D    25X,'S = AM / SQRT(X)',5X,'X = MATES(3) OR 1.0',5X,
D    3' AM = (REPR8(1) + REPR8(3)) / 2')
C......................................................................
C
C     NUMBER OF YOUNG FROM WM-WM MATINGS
   60 IF (MATES(4) .LT. 1) GO TO 70
      AM = REPR8(2)
      X = MATES(4)
      IF (X .LT. 1.) X = 1.
      S = AM / SQRT(X)
      CALL GAUSS
      NOYONG(4) = MATES(4) * V + 0.5
D     WRITE (OUTSCR,140) NOYONG(4), V, S, X, AM
D 140 FORMAT('0NOYONG(4) = ',F6.0,2X,'V = ',F5.2,2X,'S = ',F5.2,2X,
D    1'X = ',F5.2,2X,'AM = ',F5.2,/,5X,'NOYONG(4) = MATES(4) * V + 0.5',
D    25X,'S = AM / SQRT(X)',5X,'X = MATES(4) OR 1.0',5X,
D    3' AM = REPR8(2)
C.......................................................................
C
C     NUMBER OF YOUNG FROM WM-MM MATINGS
   70 IF (MATES(5) .LT. 1) GO TO 80
      AM = (REPR8(2) + REPR8(3)) / 2
      X = MATES(5)
      IF (X .LT. 1.) X = 1.
      S = AM / SQRT(X)
      CALL GAUSS
      NOYONG(5) = MATES(5) * V + 0.5
D     WRITE (OUTSCR,150) NOYONG(5), V, S, X, AM
D 150 FORMAT('0NOYONG(5) = ',F6.0,2X,'V = ',F5.2,2X,'S = ',F5.2,2X,
D    1'X = ',F5.2,2X,'AM = ',F5.2,/,5X,'NOYONG(5) = MATES(5) * V + 0.5',
D    25X,'S = AM / SQRT(X)',5X,'X = MATES(5) OR 1.0',5X,
D    3' AM = (REPR8(2) + REPR8(3)) / 2')
C.......................................................................
C
C     NUMBER OF YOUNG FROM MM-MM MATINGS
   80 IF (MATES(6) .LT. 1) GO TO 90
      AM = REPR8(3)
      X = MATES(6)
      IF (X .LT. 1.) X = 1.
      S = AM / SQRT(X)
      CALL GAUSS
      NOYONG(6) = MATES(6) * V + 0.5
D     WRITE (OUTSCR,160) NOYONG(6), V, S, X, AM
D 160 FORMAT('0NOYONG(6) = ',F6.0,2X,'V = ',F5.2,2X,'S = ',F5.2,2X,
D    1'X = ',F5.2,2X,'AM = ',F5.2,/,5X,'NOYONG(6) = MATES(6) * V + 0.5',
D    25X,'S = AM / SQRT(X)',5X,'X = MATES(6) OR 1.0',5X,
D    3' AM = REPR8(3)')
D     WRITE (OUTSCR,170)
C-----------------------------------------------------------------------
C
C      CALCULATE NUMBER OF YOUNG OF EACH GENOTYPE FROM THE MENDELIAN
C     RATIOS ASSOCIATED WITH PARTICULAR COMBINATIONS OF PARENTAL
C     GENOTYPES
C
C     NUMBER OF WW YOUNG
   90 JUV(1) = NOYONG(1) + 0.5*FLOAT(NOYONG(2)) + 0.25*FLOAT(NOYONG(4))
     1+ 0.5
C
C     NUMBER OF WM YOUNG
      JUV(2) = 0.5*FLOAT(NOYONG(2)) + NOYONG(3) + 0.5*FLOAT(NOYONG(4))
     1+ 0.5*FLOAT(NOYONG(5)) + 0.5
C
C     NUMBER OF MM YOUNG
      JUV(3) = 0.25*FLOAT(NOYONG(4)) + 0.5*FLOAT(NOYONG(5)) + NOYONG(6)
     1+ 0.5
C
C     type *,'leaving repro'
C     pause
  999 RETURN
      END
C
C***********************************************************************
C***********************************************************************
C
      SUBROUTINE M8FEW
```

```
C
C  FUNCTION: RANDOMLY PICKS UP TO N = "CUTOFF" INDIVIDUALS AS PAIRS OF
C     MATES BY A MONTE CARLO ALGORITH AND ADDS THEM TO THE MATES ARRAY.
C
C  NOTES:
C     1) SEE NOTES IN REPROD FOR A DISCUSSION OF GENERAL METHOD AND
C        RATIONALE.
C
C   CALLED FROM: REPROD
C   CALLS: RAN   - FORTRAN LIBRARY FUNCTION THAT RETURNS "YFL", A
C                  PSEUDORANDOM NUMBER UNIFORMLY DISTRIBUTED 0.0->1.0
C                  USED TO DETERMINE WHETHER A GENOTYPE IS CHOSEN FOR A
C                  PAIR OF MATES.
C=======================================================================
      INCLUDE 'COMMON.f'
C-----------------------------------------------------------------------
C  LOCAL DECLARATIONS
      INTEGER  COUNT, CUTOFF, IPAR(2), MATE, MAXPRS
      REAL     PROB,  YFL
      COUNT = 0
      cutoff = 200
C=======================================================================
C     Type *,'InM8few'
C
C  INITIALIZE
      MAXPRS =(CUTOFF / 2) + 0.5
      DO 10 I=1,2
         IPAR(I) = 0
   10 CONTINUE
C
C-----------------------------------------------------------------------
C   PICK CUTOFF/2 PAIRS OF MATES
C-----------------------------------------------------------------------
C
C     Type *,MAXPRS
C     Pause 'Before loop'
      DO 500 I=1,MAXPRS
D 900    FORMAT ('0ENTERED M8FEW PICK-A-PAIR LOOP WITH:')
D        WRITE (OUTSCR,900)
D        WRITE (OUTSCR,910) NOADLT(1), NOADLT(2), NOADLT(3), TOADLT,
D    1   CUTOFF, MAXPRS, COUNT, IPAR(1), IPAR(2), MATES(1), MATES(2),
D    2   MATES(3), MATES(4), MATES(5), MATES(6)
D 910    FORMAT (' NOADLT(1,2,3) = ',3(I4,1X),2X,'TOADLT = ',I5,5X,
D    1   'CUTOFF = ',I5,2X,'MAXPRS = ',I4,5X,'COUNT = ',I4,/,
D    2   ' IPAR(1,2) = ',2(I1,1X),2X,'MATES(1---6) = ',6(I5,1X))
C
         IF (MAXPRS .LT. 1) GO TO 999
   20    IF (TOADLT .LE. 0) GO TO 999
C
C.......................................................................
C        PICK A PAIR
C.......................................................................
C
         DO 110 J=1,2
D 930       FORMAT ('0IN PICK-A-PAIR LOOP:')
D           WRITE (OUTSCR,930)
            IF (TOADLT .EQ. 0) GO TO 110
            IF ((TOADLT .EQ. 1) .AND. (J .EQ. 1)) GO TO 60
C
            YFL = RAN(IY)
            PROB = FLOAT(NOADLT(1)) / FLOAT(TOADLT)
D           WRITE (OUTSCR,940) YFL, PROB
D 940       FORMAT (' YFL = ',F11.9,5X,'PROB = ',F11.9)
            IF (YFL .GT. PROB) GO TO 30
            IPAR(J) = 1
            NOADLT(1) = NOADLT(1) - 1
            GO TO 50
C
   30       PROB = PROB + FLOAT(NOADLT(2)) / FLOAT(TOADLT)
D           WRITE (OUTSCR,940) YFL, PROB
            IF (YFL .GT. PROB) GO TO 40
            IPAR(J) = 2
            NOADLT(2) = NOADLT(2) - 1
            GO TO 50
C
   40       IPAR(J) = 3
D           WRITE (OUTSCR,940) YFL, PROB
            NOADLT(3) = NOADLT(3) - 1
   50       TOADLT = TOADLT - 1
            GO TO 110
C......................................................................
C
C           MATE SOLITARY INDIVIDUAL WITH ITSELF.  THIS SECTION
C           OF CODE WAS ADDED TO IMPLEMENT "FACULTATIVE INBREEDING" AND
C           PREVENT STEADY DECLINE OF POPULATION SIZE IN SMALL
C           POPULATIONS WITH ODD NUMBERS OF INDIVIDUALS AND LOW ABSOLUTE
C           FITNESSES.  EARLIER VERSIONS OF THE PROGRAM USED "OBLIGATE
C           OUTBREEDING" AND STUDENTS FOUND THE RATHER STEADY DECLINE IN
C           POPULATION SIZE IN SOME RUNS TO BE CONFUSING.
C
   60       IF (NOADLT(1) .NE. 1) GO TO 70
            MATES(1) = MATES(1) + 1
            NOADLT(1) = NOADLT(1) - 1
            GO TO 90
C
   70       IF (NOADLT(2) .NE. 1) GO TO 80
            MATES(4) = MATES(4) + 1
            NOADLT(2) = NOADLT(2) - 1
            GO TO 90
C
   80       IF (NOADLT(3) .NE. 1) GO TO 100
            MATES(6) = MATES(6) + 1
            NOADLT(3) = NOADLT(3) - 1
   90       TOADLT = TOADLT - 1
            GO TO 120
  100       STOP
C......................................................................
C
  110    CONTINUE
  120    CONTINUE
D 920    FORMAT ('0EXIT PICK-A-PAIR LOOP WITH:')
D        WRITE (OUTSCR,920)
D        WRITE (OUTSCR,910) NOADLT(1), NOADLT(2), NOADLT(3), TOADLT,
D    1   CUTOFF, MAXPRS, COUNT, IPAR(1), IPAR(2), MATES(1), MATES(2),
D    2   MATES(3), MATES(4), MATES(5), MATES(6)
C
C----------------------------------------------------------------------
C        ADD PAIR TO MATES ARRAY.
C----------------------------------------------------------------------
C        VALUES OF IPAR ARRAY ARE MULTIPLIED AND THE PRODUCT (MATE)
```

```
C        IS USED TO DETERMINE WHICH ELEMENT OF THE MATES ARRAY TO
C        INCREMENT.
C
         IF ((IPAR(1) .EQ. 0) .OR. (IPAR(2) .EQ. 0)) GO TO 999
         MATE = IPAR(1) * IPAR(2)
         GO TO (300,310,320,330,210,340,210,210,350), MATE
  210    STOP
C
C        WW-WW (1*1=1)
  300    MATES(1) = MATES(1) + 2
         GO TO 360
C
C        WW-WM (1*2=2)
  310    MATES(2) = MATES(2) + 2
         GO TO 360
C
C        WW-MM (1*3=3)
  320    MATES(3) = MATES(3) + 2
         GO TO 360
C
C        WM-WM (2*2=4)
  330    MATES(4) = MATES(4) + 2
         GO TO 360
C
C        WM-MM (2*3=6)
  340    MATES(5) = MATES(5) + 2
         GO TO 360
C
C        MM-MM (3*3=9)
  350    MATES(6) = MATES(6) + 2
C
  360    CONTINUE
D        COUNT = COUNT + 1
         DO 400 K = 1,2
            IPAR(K) = 0
  400    CONTINUE
D 950    FORMAT ('0FINISHED ADD-A-PAIR WITH MATES(1---6) = ',6(I4,1X))
D        WRITE (OUTSCR,950) MATES(1), MATES(2), MATES(3), MATES(4),
D    1   MATES(5), MATES(6)
  500 CONTINUE
C----------------------------------------------------------------------
C
  999 CONTINUE
D     WRITE (OUTSCR,960)
D 960 FORMAT ('0EXIT FROM M8FEW WITH:')
D     WRITE (OUTSCR,910) NOADLT(1), NOADLT(2), NOADLT(3), TOADLT,
D    1   CUTOFF, MAXPRS, COUNT, IPAR(1), IPAR(2), MATES(1), MATES(2),
D    2   MATES(3), MATES(4), MATES(5), MATES(6)
      RETURN
      END
C
C**********************************************************************
C**********************************************************************
C
      SUBROUTINE M8MANY
C
C   FUNCTION:  CALCULATES THE NUMBER OF EACH TYPE OF MATES BY METHOD 2
C      AND THEN ADDS THEM TO THE MATES ARRAY. SEE NOTES IN REPROD.
C
C   NOTES:
```

```
C      1) THE PROBABILITY OF EACH TYPE OF MATING IS DETERMINED BY
C         MULTIPLYING THE FREQUENCIES OF THE APPROPRIATE GENOTYPES
C         "GNTFR" ARRAY, BY EACH OTHER.  THE PROBABILITY OF EACH
C         COMBINATION OF MATES IS THEN MULTIPLIED BY THE NUMBER OF
C         ADULTS LEFT IN "NOADLT", ROUNDED TO THE NEAREST
C         INTEGER AND ADDED TO THE NUMBER OF MATES ESTABLISHED BY
C         SUBROUTINE "M8FEW".
C
C   CALLED FROM: REPROD
C   CALLS: NOTHING
C=======================================================================
      INCLUDE 'COMMON.f'
C-----------------------------------------------------------------------
C  LOCAL DECLARATIONS
      INTEGER  TMATES
      REAL     GNTFR(3)
C=======================================================================
C     Type *,'In M8Many'
C
D 900 FORMAT ('0ENTERED M8MANY WITH:')
D     WRITE (OUTSCR,900)
D     WRITE (OUTSCR,910) NOADLT(1), NOADLT(2), NOADLT(3), TOADLT,
D    1   MATES(1), MATES(2), MATES(3), MATES(4), MATES(5), MATES(6)
D 910 FORMAT (' NOADLT(1-3) = ',3(I5,1X),' TOADLT = ',I5,
D    1   ' MATES(1---6) = ',6(I5,1X))
C
C     CALCULATE FREQUENCY OF EACH GENOTYPE
      DO 10 I=1,3
         GNTFR(I) = FLOAT(NOADLT (I)) / TOADLT
   10 CONTINUE
C
C-----------------------------------------------------------------------
C  CALCULATE NUMBER OF EACH TYPE OF MATING.
C-----------------------------------------------------------------------
      MATES(1) = MATES(1) + (GNTFR(1)**2) * TOADLT + 0.5
      MATES(2) = MATES(2) + (2 * (GNTFR(1)*GNTFR(2))) * TOADLT + 0.5
      MATES(3) = MATES(3) + (2 * (GNTFR(1)*GNTFR(3))) * TOADLT + 0.5
      MATES(4) = MATES(4) + (GNTFR(2)**2) * TOADLT + 0.5
      MATES(5) = MATES(5) + (2 * (GNTFR(2)*GNTFR(3))) * TOADLT + 0.5
      MATES(6) = MATES(6) + (GNTFR(3)**2) * TOADLT + 0.5
C
D     TMATES = MATES(1) + MATES(2) + MATES(3) + MATES(4) + MATES(5) +
D    $   MATES(6)
D     LGEN = IGEN - 1
D     ATOT = RSLT(1,LGEN) + RSLT(2,LGEN) + RSLT(3,LGEN)
D     IF (TMATES .EQ. ATOT) GO TO 930
D     WRITE (OUTSCR,920) TMATES, ATOT
D 920 FORMAT ('0ERROR: TMATES = ',I5,1X,'ATOT = ',I5)
D 930 WRITE (OUTSCR,940)
D 940 FORMAT ('0EXIT FROM M8MANY WITH:')
D     WRITE (OUTSCR,910) NOADLT(1), NOADLT(2), NOADLT(3), TOADLT,
D    $   MATES(1), MATES(2), MATES(3), MATES(4), MATES(5), MATES(6)
C-----------------------------------------------------------------------
C
      RETURN
      END
C***********************************************************************

C
C***********************************************************************
```

```
C file: SR.SU   vrsn:  48   rvsd: 27 Aug 81
C               subroutines srvivl, sumrys
C**********************************************************************
C
      SUBROUTINE SRVIVL
C
C   FUNCTION: TAKES THE NUMBER OF JUVENILES IN THE CURRENT GENERATION
C      AND RETURNS THE NUMBER OF ADULTS IN THE CURRENT GENERATION
C      "RSLTS(1-3,IGEN)"
C
C   SYNOPSIS:
C      1) THIS SUBROUTINE USES THE RATES OF EMIGRATION ("OUTFLOW") TO
C         DETERMINE THE NUMBER OF JUVENILES OF EACH GENOTYPE EMIGRATING
C         ("NOOUT") AND REMOVES THEM FROM THE JUVENILE POPULTION.
C      2) IT THEN DETERMINES THE NUMBER OF JUVENILES DYING FROM THE
C         RATES OF SURVIVAL ("SURVR8") AND REMOVES THEM FROM THE
C         POPULATION TO DETERMINE THE NUMBER SURVIVING TO ADULTHOOD
C         ("ADLT" ARRAY).
C      3) THE NUMBER OF IMMIGRANTS ("INFLO") OF EACH GENOTYPE IS ADDED
C         TO "ADLT".
C      4) IF THE NUMBER OF ADULTS EXCEEDS THE CARRYING CAPACITY
C         ("MXPOP1"), THE TOTAL IS REDUCED TO THE POST-CRASH SIZE
C         ("MXPOP2").
C
C     NOTES:
C        1) VAR. "NOOUT" ADDED & PLACED IN COMMON FOR USE IN DEBUG
C           OUTPUT FROM SUBROUTINE SUMRYS.  IT WILL NOT BE MEANINGFUL
C           IF SUMRYS IS CALLED TO PRINT A GENERATION OTHER THAN THE
C           CURRENT ONE. IT SHOULD PROBABLY BE REMOVED IN THE PRODUCTION
C           VERSION.
C        2) ALTHOUGH VARIABLES "ATOT" & "ADLT" ARE CONCEPTUAL INTEGERS,
C           THEY ARE DECLARED REAL TO AVOID POSSIBLE INTEGER OVERFLOW ON
C           16-BIT MACHINES.  THEORETICALLY, "ADLT" COULD =
C           JUV * SURVR8/100 + INFLO = 50,000 * 100/100 + 2000 = 52,000
C           "ATOT" COULD = 3 * ADLT = 3 * 52,000 = 156,000
C
C   CALLED FROM: MAIN
C   CALLS: NOTHING
C======================================================================
      INCLUDE 'COMMON.f'
C----------------------------------------------------------------------
C  LOCAL DECLARATIONS
      REAL     ADLT(3), ATOT, CRSH
C======================================================================
C     Type *,'In Survival'
C
C  INITIALIZE
C
      DO 10 I=1,3
         ADLT(I) = 0
   10 CONTINUE
      ATOT = 0
C----------------------------------------------------------------------
C
C  EMIGRATION OF JUVENILES
C
C     RNOTE: COULD CALL GAUSS HERE TO GET RANDOM VARIATION IN %
C     EMIGRATING.
C......................................................................
C
```

```
      IF ((OUTFLO(1) .EQ. 0) .AND. (OUTFLO(2) .EQ. 0) .AND.
     1   (OUTFLO(3) .EQ. 0)) GO TO 100
C
      DO 40 I=1,3
         IF (OUTFLO(I) .GT. 0.) GO TO 20
         NOOUT(I) = 0
         GO TO 40
C
   20    IF (JUV(I) .GT. 0.) GO TO 30
         NOOUT(I) = 0
         GO TO 40
C
   30    NOOUT(I) = ((JUV(I))*(OUTFLO(I) / 100)) + 0.5
         JUV(I) = JUV(I) - NOOUT(I)
   40 CONTINUE
C
C----------------------------------------------------------------------
C  CALCULATE TOTAL NUMBER OF ADULTS
C----------------------------------------------------------------------
C     MULTIPLY NUMBER OF YOUNG OF EACH GENOTYPE BY ITS SURVIVAL RATE,
C     ADD IMMIGRANTS
C
C     RNOTE: COULD CALL GAUSS HERE TO GET RANDOM VARIATION IN #
C     IMMIGRATING.
C......................................................................
  100 DO 110 I=1,3
         ADLT(I) = ((SURVR8(I) / 100) * JUV(I)) + .5 + INFLO(I)
         ATOT = ATOT + ADLT(I)
  110 CONTINUE
C
C----------------------------------------------------------------------
C  TEST FOR ADULTS > CARRYING CAPACITY, CRASH IF NECESSARY
C
C     RNOTE: COULD CALL GAUSS HERE TO GET RANDOM VARIATION IN POST-
C     CRASH POPULATION SIZE.
C......................................................................
C
      IF (ATOT .LE. MXPOP1) GO TO 300
D     WRITE (OUTSCR,35) ADLT(1), ADLT(2), ADLT(3), MXPOP2, ATOT
D  35 FORMAT ('0ENTER CRASH BLOCK IN SRVIVL WITH: ',5X,'ADLT(1-3) = ',
D    1   1X,3(I5,1X),5X,'MXPOP2 = ',1X,I5,5X,'ATOT = ',1X,F6.0)
C
      CRSH = FLOAT(MXPOP2) / FLOAT(ATOT)
      DO 200 I=1,3
         ADLT(I) =ADLT(I) * CRSH + .5
  200 CONTINUE
D     WRITE (OUTSCR,36) ADLT(1), ADLT(2), ADLT(3), MXPOP2, ATOT
D  36 FORMAT ('0 EXIT CRASH BLOCK IN SRVIVL WITH: ',5X,'ADLT(1-3) = ',
D    $   1X,3(I5,1X),5X,'MXPOP2 = ',1X,I5,5X,'ATOT = ',1X,F6.0)
C
C----------------------------------------------------------------------
C  PUT ADULTS INTO RESULTS ARRAY
C----------------------------------------------------------------------
C
  300 DO 310 I=1,3
         RSLT(I,IGEN) = ADLT(I)
  310 CONTINUE
C
      RETURN
      END
```

```
C
C**********************************************************************
C**********************************************************************
C
      SUBROUTINE SUMRYS
C
C   FUNCTION: PRINTS A SUMMARY OF FIRST AND LAST GENERATIONS, AND
C      OF OTHER GENERATIONS IF REQUESTED. OUTPUT IS DIRECTED TO CRT OR
C      LINE PRINTER BY VAR. "PRINT2".
C
C   NOTES:
C      1) THIS ROUTINE WAS MUCH USED DURING DEBUGGING AND THERE IS A
C         LOT OF DEBUG CODE.  THERE ARE TWO MAJOR SECTIONS OF "D" CODE;
C         THE FIRST IS FOR GENERATIONS WHEN SELECTION & FITNESS
C         COEFFICIENTS CAN BE CALCULATED, THE SECOND FOR GENERATIONS
C         WHEN THE COEFFICIENTS CANNOT BE CALCULATED.
C      2) DEBUG CODE HAS NOT BEEN USE FOR A LONG TIME & MAY NOT BE
C         RELIABLE. IN THE PRODUCTION VERSION, DEBUG CODE SHOULD
C         PROBABLY BE REMOVED, ALTHOUGH CONDUIT MAY WISH TO CLEAN IT
C         UP FOR PURCHASERS' USE.
C      3) COULD PROBABLY REMOVE JUVENILES AND GENE FLOW FROM PRODUCTION
C         OUTPUT; MIGHT BE SIMPLER FOR STUDENTS.
C      4) PUT IN PAGESIZE & #LINESPRINTED VARS. ETC. TO PAGE OUTPUT
C
C   CALLED FROM: MAIN
C   CALLS: ALFIT  - RETURNS ALLELE FITNESS COEFFICIENTS (DEBUG ONLY)
C          ALFREQ - RETURNS ALLELE FREQUENCY OF M ALLELE
C          ALSEL  - RETURNS ALLELE SELECTION COEFFICIENTS (DEBUG ONLY)
C          GNTFIT - RETURNS GENOTYPE FITNESS COEFFICIENTS (DEBUG ONLY)
C          GNTSEL - RETURNS GENOTYPE SELECTION COEFFICIENTS (DEBUG ONLY)
C======================================================================
      INCLUDE 'COMMON.f'
C----------------------------------------------------------------------
C     LOCAL DECLARATIONS
      INTEGER  ATOT,  ATOTP, INTOT
      REAL     FRQM,  FRQW, OUTTOT, TOJUV
C======================================================================
C
C  PRINT HEADER(S)
C-----------------------------------------------------------------------
C     Type *,'In Sumrys'
C
      WRITE (PRINT2,10)
   10 FORMAT (/,17X,'#      GENE FLOW     #      GNOT.',10X,'ALLELE',/,
     $   1X,'GEN.   GNOT.  YOUNG   # IN  # OUT  ADULTS   FREQ. ALLELE ',
     $   '  FREQ.  ')
C
D     WRITE (OUTLP,15)
D  15    FORMAT (50X,'GNOT.  GNOT.  GNOT.',17X,'ALL.   ALL.   ALL.',/,
D    $   17X,'#     GENE FLOW     #     GNOT.  SURV.  FITN.  SELN.',10X,
D    $   'ALL.   SURV.  FITN.  SELN.',/,1X,'GEN.   GNOT.  YOUNG  # IN ',
D    $   ' # OUT  ADULTS  FREQ.  RATES  COEF.  COEF.  ALLELE  FREQ.   ',
D    $   'RATES  COEF.  COEF.     ***** MEANS A COEF.',/,110X,
D    $   'CANNOT BE CALCULATED')
C
C----------------------------------------------------------------------
C  INITIALIZE VARIABLES, TEST WHETHER COEFFICIENTS CAN BE CALCULATED
C----------------------------------------------------------------------
   20 FRQM = ALFREQ(SUMGEN)
      FRQW = 1 - FRQM
      IF ((RSLT(1,SUMGEN) .EQ. 0) .AND. (RSLT(2,SUMGEN) .EQ. 0))
     $   FRQW = 0.
      TOJUV = JUV(1) + JUV(2) + JUV(3)
      ATOT = RSLT(1,SUMGEN) + RSLT(2,SUMGEN) + RSLT(3,SUMGEN)
      INTOT = INFLO(1) + INFLO(2) + INFLO(3)
      OUTTOT = NOOUT(1) + NOOUT(2) + NOOUT(3)
      DO 25 I=1,3
        IF (ATOT .GT. 0) GO TO 24
        OBSFR(I) = 0.
        GO TO 25
   24   OBSFR(I) = FLOAT(RSLT(I,SUMGEN)) / FLOAT(ATOT)
   25 CONTINUE
C----------------------------------------------------------------------
C
C     PRINT SUMMARY
      WRITE(PRINT2,30) SUMGEN,JUV(1),INFLO(1),NOOUT(1),RSLT(1,SUMGEN),
     $   OBSFR(1),FRQW,JUV(2),INFLO(2),NOOUT(2),RSLT(2,SUMGEN),OBSFR(2),
     $   JUV(3),INFLO(3),NOOUT(3),RSLT(3,SUMGEN),OBSFR(3),FRQM,
     $   TOJUV,INTOT,OUTTOT,ATOT
C
   30 FORMAT(' ',I3,5X,'WW',4X,F6.0,2X,I4,2X,F6.0,3X,I4,3X,F5.3,4X,
     $   'W',5X,F5.3,/,9X,'WM',4X,F6.0,2X,I4,2X,F6.0,3X,I4,3X,F5.3,
     $   /,9X,'MM',4X,F6.0,2X,I4,2X,F6.0,3X,I4,3X,F5.3,4X,'M',5X,
     $   F5.3,/,7X,'TOTAL',3X,F6.0,1X,I5,2X,F6.0,3X,I4,/)
D     GO TO 35
      GO TO 99
C
D----------------------------------------------------------------------
D  CALCULATE ADDITIONAL INITIAL VARS FOR DEBUG OUTPUT;
D     TEST FOR CALCULATABILITY OF COEFFICIENTS
D----------------------------------------------------------------------
D  35 IF (SUMGEN .EQ. 1) GO TO 60
D     LGEN = SUMGEN - 1
D     ATOTP = RSLT(1,LGEN) + RSLT(2,LGEN) + RSLT(3,LGEN)
D     FRQMP = ALFREQ(LGEN)
D     IF ((FRQMP .EQ. 0) .OR. (FRQM .EQ. 0) .OR.(ATOTP .EQ. 0)
D    $   .OR. (FRQMP .EQ. 0.) .OR. (FRQMP .EQ. 1.)) GO TO 60
D     CALL GNTFIT (SUMGEN)
D     CALL ALFIT (SUMGEN)
D     CALL GNTSEL (SUMGEN)
D     CALL ALSEL (SUMGEN)
D
D----------------------------------------------------------------------
D  PRINT DEBUG DATA WHEN FITNESS & SEL. COEFFICIENTS CAN BE CALCULATED
D----------------------------------------------------------------------
D  40 WRITE(OUTLP,50) SUMGEN,JUV(1),INFLO(1),NOOUT(1),RSLT(1,SUMGEN),
D    $   OBSFR(1),GNTSUR(1),FITGNT(1),GSELCO(1),FRQW,ALSUR(1),FITAL(1),
D    $   ASELCO(1),JUV(2),INFLO(2),NOOUT(2),RSLT(2,SUMGEN),OBSFR(2),
D    $   GNTSUR(2),FITGNT(2),GSELCO(2),JUV(3),INFLO(3),NOOUT(3),
D    $   RSLT(3,SUMGEN),OBSFR(3),GNTSUR(3),FITGNT(3),GSELCO(3),FRQM,
D    $   ALSUR(2),FITAL(2),ASELCO(2),TOJUV,INTOT,OUTTOT,ATOT
C
D  50 FORMAT('0',I3,5X,'WW',4X,F5.0,2X,I4,2X,I5,3X,I4,3X,4(F5.3,2X),
D    $  2X,'W',3X,4(2X,F5.3,/,9X,'WM',4X,F5.0,2X,I4,2X,I5,3X,I4,3X,
D    $  4(F5.3,2X),/,9X,'MM',4X,F5.0,2X,I4,2X,I5,3X,I4,3X,4(F5.3,2X),
D    $  2X,'M',3X,4(2X,F5.3),/,7X,'TOTAL',3X,F5.0,2X,I4,2X,I5,3X,
D    $  I4,/))
D     GO TO 99
D
D----------------------------------------------------------------------
```

```
D  PRINT DEBUG DATA WHEN FITNESS & SEL. COEFFICIENTS CAN'T BE CALCULATED
D-----------------------------------------------------------------------
D  60 WRITE(OUTLP,70) SUMGEN,JUV(1),INFLO(1),NOOUT(1),RSLT(1,SUMGEN),
D    $  OBSFR(1),FRQW,JUV(2),INFLO(2),NOOUT(2),RSLT(2,SUMGEN),OBSFR(2),
D    $  JUV(3),INFLO(3),NOOUT(3),RSLT(3,SUMGEN),OBSFR(3),FRQM,TOJUV,
D    $  INTOT,OUTTOT,ATOT
D
D  70 FORMAT('0',I3,5X,'WW',4X,F5.0,2X,I4,2X,I5,3X,I4,3X,F5.3,2X,
D    $ '*****','  *****  *****',4X,'W',5X,F5.3,2X,
D    $ '*****  *****  *****',/,9X,'WM',4X,F5.0,2X,I4,2X,I5,3X,I4,3X,
D    $  F5.3,2X,'*****  *****  ','*****',/,9X,'MM',4X,F5.0,2X,I4,2X,
D    $  I5,3X,I4,3X,F5.3,2X,'*****','*****  *****',4X,'M',5X,F5.3,2X,
D    $  '*****  *****  *****',/,7X,'TOTAL',3X,F5.0,2X,I4,2X,I5,3X,I4)
C-----------------------------------------------------------------------
C
   99 RETURN
      END
C***********************************************************************

C
C***********************************************************************
C          file: GR.MI   vrsn: 43   rvsd: 27 Aug 81
C                Subroutine graph & miscellaneous subroutines
C***********************************************************************
C
      SUBROUTINE GRAPH
C
C   FUNCTION: PRINTS GRAPHS OF RESULTS TO LINE PRINTER
C
C   SYNOPSIS:
C      1) THE MAIN PART OF THE ROUTINE IS DIVIDED INTO TWO MAJOR
C         SECTIONS WHICH DO GRAPHS OF FLOATING POINT AND INTEGER DATA
C         RESPECTIVELY. SUBROUTINE GRAPH FIRST DETERMINES WHETHER THE
C         CURRENT GRAPH IS OF INTEGER OR REAL DATA AND CONTROL JUMPS TO
C         THE APPROPRIATE SECTION
C      2) WITHIN EACH SECTION AN INITIAL PORTION IS DEVOTED TO FILLING A
C         DATA ARRAY WITH THE APPROPRIATE DATA TO BE GRAPHED, AND A
C         SECOND PORTION PLOTS THE DATA FROM THE DATA ARRAY.
C      3) PLOTTING CONSISTS OF COMPOSING A LINE OF PRINT FROM THE DATA
C         ARRAY, THEN PRINTING THE LINE.
C      4) FROM EACH MAJOR SECTION, CONTROL JUMPS TO ONE SECTION THAT PRINTS
C         THE ABSCISSA AND LABEL, THEN RETURNS TO THE TOP OF THE ROUTINE
C         TO REPEAT THE PROCESS.
C
C  NOTES:
C     1) FACTR IS USED TO SCALE VERTICAL AXIS FOR 50 LINES PER PAGE
C     2) "-9" IS USED WHEN COEFFICIENTS CANNOT BE CALCULATED; IT IS
C        USED TO SET PRINTED CHARACTER TO "?".
C
C  NOTES ON POSSIBLE CHANGES:
C     1) CHANGE DIMENSIONS FOR GRAPHS TO VARIABLES THAT CAN BE SET BEFORE
C        GRAPH IS CALLED. E.G., PGLNTH, PGWDTH, HSCALE, VSCALE
C     2) MODIFY TO FLIP FROM ONE GRAPH TO ANOTHER & BACK FOR SCREEN USE
C        (USE "PRINT2").
C        COULD DISPLAY CHARACTER GRAPH TO SCREEN DIMENSIONS, ASK IF USER
C        WANTS HARD COPY TO LINE PRINTER.
C     3) MODIFY TO USE SIGGRAPH CORE-79  ROUTINES ON TERAK.
C     4) MODIFY TO PRINT OUT LENGTH-WISE FOR TELETYPES?
C     5) COULD CLARIFY LOGIC BY BREAKING INTO SUBROUTINES TO, FOR E.G.,
C        FILL DATA ARRAYS, COMPOSE A LINE, ETC.
```

```
C     6) THIS ROUTINE IS SLOW, IT CANNOT DRIVE THE PRINTRONIX AT FULL
C        SPEED.  WILL HAVE TO TRY TO SPEED IT UP.  THIS MAY CONFLICT
C        WITH NOTE 5.
C
C   CALLED FROM: MAIN
C   CALLS: ALFREQ - RETURNS FREQUENCY OF M ALLELE
C          GNTFIT - RETURNS FITNESS COEFFICIENT OF EACH GENOTYPE
C          ALFIT  - RETURNS FITNESS COEFFICIENT OF EACH ALLELE
C          GNTSEL - RETURNS SELECTION COEFFICIENT OF EACH GENOTYPE
C          ALSEL  - RETURNS SELECTION COEFFICIENT OF EACH ALLELE
C=======================================================================
      INCLUDE 'COMMON.f'
C-----------------------------------------------------------------------
C
C  LOCAL DECLARATIONS
      INTEGER  ATOT,  BLNK,  CHAR,  CHARH,  CHARM,  CHARQ,  CHARW,
     $   DASH,  IDATA(125),  IDATUM,  LNBTM,  LINE(125),  LNSIZ,  MAX,
     $   N
      REAL  DATA(3,125),  DATUM1,  DATUM2,  DATUM3,  FACTR,  FLNBTM,
     $   FLNSIZ,  FLNTOP,  FMAX,  IFACTR
      DATA     BLNK/' '/,       CHAR/'@'/,          CHARH/'H'/,
     $         CHARM/'M'/,      CHARQ/'?'/,         CHARW/'W'/,
     $         DASH/'-'/,       IDATA/125*0/
C=======================================================================
C     Type *,'In Graph'
C
C     INITIALIZE, DETERMINE WHETHER ALL GRAPHS HAVE BEEN FINISHED.
C     IF NOT, DETERMINE WHETHER CURRENT GRAPH IS OF FLOATING POINT OR
C     OF INTEGER DATA.
C-----------------------------------------------------------------------
      DO 6000 I=1,10
         IF (GRF(I) .EQ. '  ') GO TO 9999
         IF ((GRF(I) .EQ. 'PO') .OR. (GRF(I) .EQ. 'WW') .OR.
     $   (GRF(I) .EQ. 'WM') .OR. (GRF(I) .EQ. 'MM')) GO TO 2000
C
C=======================================================================
C FLOATING POINT GRAPHS
C=======================================================================
C
C        INITIALIZE, ZERO DATA MATRIX
         FMAX = 1.0
         FACTR = FMAX/50
         DO 20 J=1,3
            DO 10 K=1,125
               DATA(J,K) = 0.
   10       CONTINUE
   20    CONTINUE
C
C        DETERMINE GRAPH REQUESTED
         IF (GRF(I) .EQ. 'AF') GO TO 100
         IF (GRF(I) .EQ. 'GF') GO TO 200
         IF (GRF(I) .EQ. 'GW') GO TO 300
         IF (GRF(I) .EQ. 'AW') GO TO 400
         IF (GRF(I) .EQ. 'GS') GO TO 500
         IF (GRF(I) .EQ. 'AS') GO TO 600
         WRITE (OUTLP,30) I
   30    FORMAT (1X,'*****ERROR: GRAPH REQUEST #',I2,' IS INCORRECT')
         GO TO 9999
C
C-----------------------------------------------------------------------
```

```
C     SET PARAMETERS FOR FLOATING POINT GRAPHS
C----------------------------------------------------------------------
C
C        SET PARAMETERS FOR ALLELE FREQUENCY GRAPH
  100    DO 110 K=1,IGEN
            DATA(2,K) = ALFREQ(K)
            DATA(1,K) = 1. - DATA(2,K)
            IF((RSLT(1,K) .EQ. 0) .AND. (RSLT(2,K) .EQ. 0))
     $         DATA(1,K) = 0.
            IF((RSLT(3,K) .EQ. 0) .AND. (RSLT(2,K) .EQ. 0))
     $         DATA(2,K) = 0.
  110    CONTINUE
         GO TO 1000
C.......................................................................
C
C        SET PARAMETERS FOR GENOTYPE FREQUENCY GRAPH
  200    DO 230 J=1,IGEN
            ATOT = RSLT(1,J) + RSLT(2,J) + RSLT(3,J)
            IF (ATOT .GT. 0) GO TO 210
            DATA(1,J) = 0.
            DATA(2,J) = 0.
            DATA(3,J) = 0.
            GO TO 230
C
  210       DO 220 K=1,3
               IDATUM = RSLT(K,J)
               IF (IDATUM .GT. 0 ) GO TO 215
                  DATA (K,J) = 0.
                  GO TO 220
  215          DATA(K,J) = FLOAT(IDATUM) / ATOT
  220       CONTINUE
  230    CONTINUE
         GO TO 1000
C.......................................................................
C
C        SET PARAMETERS FOR GENOTYPE FITNESS GRAPH
  300    DATA(1,1) = -9.
         DATA(2,1) = -9.
         DATA(3,1) = -9.
         DO 320 J=2,IGEN
            LGEN = J - 1
            FRQMP = ALFREQ(LGEN)
            ATOT = RSLT(1,J) + RSLT(2,J) + RSLT(3,J)
            IF ((FRQMP .GT. 0.) .AND. (FRQMP .LT. 1.) .AND.
     $      (ATOT .GT. 0)) GO TO 310
C
            DATA(1,J) = -9.
            DATA(2,J) = -9.
            DATA(3,J) = -9.
            GO TO 320
C
  310       CALL GNTFIT(J)
            DATA(1,J) = FITGNT(1)
            DATA(2,J) = FITGNT(2)
            DATA(3,J) = FITGNT(3)
C
  320    CONTINUE
         GO TO 1000
C.......................................................................
C
```

```
C        SET PARAMETERS FOR ALLELE FITNESS GRAPH
  400    DATA(1,1) = -.9
         DATA(2,1) = -.9
C
         DO 420 J=2,IGEN
            LGEN = J - 1
            FRQMP = ALFREQ(LGEN)
            ATOT = RSLT(1,LGEN) + RSLT(2,LGEN) + RSLT(3,LGEN)
            IF ((FRQMP .GT. 0.) .AND. (FRQMP .LT. 1.) .AND.
     1      (ATOT .GT. 0)) GO TO 410
C
            DATA(1,J) = -9.
            DATA(2,J) = -9.
            GO TO 420
C
  410       CALL ALFIT(J)
            DATA(1,J) = FITAL(1)
            DATA(2,J) = FITAL(2)
C
  420    CONTINUE
         GO TO 1000
C......................................................................
C
C        SET PARAMETERS FOR GENOTYPE SELECTION COEF. GRAPH
  500    DATA(1,1) = -9.
         DATA(2,1) = -9.
         DATA(3,1) = -9.
         DO 520 J=2,IGEN
            LGEN = J - 1
            FRQMP = ALFREQ(LGEN)
            ATOT = RSLT(1,J) + RSLT(2,J) + RSLT(3,J)
            IF ((FRQMP .GT. 0.) .AND. (FRQMP .LT. 1.) .AND.
     $      (ATOT .GT. 0)) GO TO 510
C
            DATA(1,J) = -9.
            DATA(2,J) = -9.
            DATA(3,J) = -9.
            GO TO 520
C
  510       CALL GNTSEL (J)
            DATA(1,J) = GSELCO(1)
            DATA(2,J) = GSELCO(2)
            DATA(3,J) = GSELCO(3)
  520    CONTINUE
         GO TO 1000
C......................................................................
C
C        SET PARAMETERS FOR ALLELE SELECTION COEF. GRAPH
  600    DATA(1,1) = -9.
         DATA(2,1) = -9.
         DO 620 J=2,IGEN
            LGEN = J - 1
            FRQMP = ALFREQ(LGEN)
            ATOT = RSLT(1,J) + RSLT(2,J) + RSLT(3,J)
            IF ((FRQMP .GT. 0.) .AND. (FRQMP .LT. 1.) .AND.
     $      (ATOT .GT. 0)) GO TO 610
C
            DATA(1,J) = -9.
            DATA(2,J) = -9.
            GO TO 620
```

```
C
  610       CALL ALSEL (J)
            DATA(1,J) = ASELCO(1)
            DATA(2,J) = ASELCO(2)
  620    CONTINUE
         GO TO 1000
C
C----------------------------------------------------------------------
C        PLOT FLOATING POINT DATA
C----------------------------------------------------------------------
C
 1000    WRITE (OUTLP,5000) TITLE
         WRITE (OUTLP,5002)
C
C        PRINT 50 LINES
         IFACTR=FACTR/2
         DO 1060 J=1,51
C
C           COMPOSE A LINE
            FLNSIZ = FMAX - ((J-1)*FACTR)
            FLNTOP = FLNSIZ + FACTR/2
            FLNBTM = FLNSIZ - FACTR/2
            DO 1010 K=1,125
               LINE(K) = BLNK
 1010       CONTINUE
            IF ((GRF(I) .NE. 'AF') .AND. (GRF(I) .NE. 'AW') .AND.
     $      (GRF(I) .NE. 'AS')) GO TO 1030
C......................................................................
C
C           COMPOSE ALLELE COEFFICIENT LINES
            DO 1020 K = 1,IGEN
               DATUM1 = DATA(1,K)
               DATUM2 = DATA(2,K)
               LINE(K) = BLNK
               IF (DATUM1 .GE. 0.) GO TO 1011
               LINE(K) = CHARQ
               GO TO 1012
 1011          IF ((DATUM1 .GT. FLNBTM) .AND. (DATUM1 .LE.
     $         FLNTOP)) LINE(K) = CHARW
 1012          IF (DATUM2 .GE. 0.) GO TO 1013
               LINE(K) = CHARQ
               GO TO 1020
 1013          IF ((DATUM2 .GT. FLNBTM) .AND. (DATUM2 .LE.
     $         FLNTOP)) LINE(K) = CHARM
 1020          CONTINUE
            IF (J .LT. 51) GO TO 1050
               DO 1025 K=1,125
                  IF ((DATA(2,K) .LE. 0.)  .OR.
     $               (DATA(1,K) .LE. 0.)) LINE(K) = DASH
 1025             CONTINUE
               GO TO 1050
C......................................................................
C
C           COMPOSE GENOTYPE COEFFICIENT LINES
C           NOTE THAT H OVERPRINTS M WHICH OVERPRINTS W
 1030       DO 1040 K = 1,IGEN
               DATUM1 = DATA(1,K)
               DATUM2 = DATA(2,K)
               DATUM3 = DATA(3,K)
               LINE(K) = BLNK
               IF (DATUM1 .GE. 0.) GO TO 1031
               LINE(K) = CHARQ
               GO TO 1032
 1031          IF ((DATUM1 .GT. FLNBTM) .AND. (DATUM1 .LE.
     $         FLNTOP)) LINE(K) = CHARW
 1032          IF (DATUM3 .GE. 0. ) GO TO 1033
               LINE(K) = CHARQ
               GO TO 1034
 1033          IF ((DATUM3 .GT. FLNBTM) .AND. (DATUM3 .LE.
     $         FLNTOP)) LINE(K) = CHARM
 1034          IF (DATUM2 .GE. 0. ) GO TO 1035
               LINE(K) = CHARQ
               GO TO 1040
 1035          IF ((DATUM2 .GT. FLNBTM) .AND. (DATUM2 .LE.
     $         FLNTOP)) LINE(K) = CHARH
 1040          CONTINUE
            IF (J .LT. 51) GO TO 1050
               DO 1045 K=1,125
                   IF ((DATA(2,K) .LE. 0.) .AND. ((DATA(1,K)
     $                .LE. 0.) .OR. (DATA(3,K) .LE. 0.))) LINE(K) = DASH
 1045              CONTINUE
C.......................................................................
C
C           PRINT A LINE
 1050       WRITE (OUTLP,1052) FLNSIZ, LINE
 1052       FORMAT (1X,F5.3,'+',125A1)
 1060    CONTINUE
C
         GO TO 4000
C
C
C======================================================================
C  INTEGER PLOTS (POP, WW, WM, MM)
C======================================================================
C
C        ZERO DATA MATRIX
 2000    DO 2010 K=1,125
            IDATA(K) = 0
 2010    CONTINUE
         IF (GRF(I) .NE. 'PO') GO TO 2200
C
C----------------------------------------------------------------------
C     SET PARAMETERS FOR INTEGER PLOTS
C----------------------------------------------------------------------
C
C        SET PARAMETERS FOR POPULATION SIZE GRAPH
C
C        DETERMINE MAXIMUM POPULATION SIZE DURING RUN, IF < 50, SET = 50
C        (I.E. ONE INDIVIDUAL PER GRAPH LINE)
         MAX = RSLT(1,1) + RSLT(2,1) + RSLT(3,1)
         IDATA(1) = MAX
         DO 2100 J=1,IGEN
            IDATA(J) = RSLT(1,J) + RSLT(2,J) + RSLT(3,J)
            IF (MAX .LT. IDATA(J)) MAX = IDATA(J)
 2100    CONTINUE
         FACTR = FLOAT(MAX)/50
         IF (MAX .GT. 50) GO TO 2110
         FACTR = 1.
         MAX = 50
 2110    GO TO 3000
```

```
C......................................................................
C
C        SET PARAMETERS FOR GENOTYPE NUMBERS GRAPHS
C
C        ZERO DATA MATRIX
 2200    DO 2210 K=1,125
            IDATA(K) = 0
 2210    CONTINUE
C
C        SET GENOTYPE FOR GRAPH
         N = 0
         IF (GRF(I) .NE. 'WW') GO TO 2220
         N = 1
         GO TO 2250
C
 2220    IF (GRF(I) .NE. 'WM') GO TO 2230
         N = 2
         GO TO 2250
C
 2230    IF (GRF(I) .NE. 'MM') GO TO 2240
         N = 3
         GO TO 2250
 2240    WRITE (OUTLP,30) I
         GO TO 9999
C
C        DETERMINE MAXIMUM NUMBER OF GENOTYPE DURING RUN, IF MAX < 50,
C        SET = 50 (I.E., ONE INDIVIDUAL PER GRAPH LINE)
 2250    MAX = RSLT(N,1)
         DO 2260 J=1,IGEN
            IDATA(J) = RSLT(N,J)
            IF (MAX .LT. IDATA(J)) MAX = IDATA(J)
 2260    CONTINUE
         FACTR = FLOAT(MAX)/50
C
         IF (MAX .GT. 50) GO TO 3000
         FACTR = 1.
         MAX = 50
C
C----------------------------------------------------------------------
C        PLOT INTEGER DATA
C----------------------------------------------------------------------
C
 3000    WRITE (OUTLP,5000) TITLE
         WRITE (OUTLP,5002)
C
C        PRINT 50 LINES
         DO 3040 J=1,51
C
C           COMPOSE A LINE
            LNSIZ = MAX - ((J-1)*FACTR) + 0.5
            LNBTM = LNSIZ - FACTR/2
            DO 3010 K=1,125
               LINE(K) = BLNK
 3010       CONTINUE
            DO 3020 K=1,IGEN
               IF (IDATA(K) .GT. LNBTM) LINE(K) = CHAR
 3020       CONTINUE
            IF (J .LT. 51) GO TO 3030
               DO 3025 K=1,125
                  IF (IDATA(K) .EQ. 0) LINE(K) = DASH
 3025             CONTINUE
C
C           PRINT A LINE
 3030       WRITE (OUTLP,3032) LNSIZ, LINE
 3032       FORMAT (1X,I4,' +',125A1)
 3040    CONTINUE
         GO TO 4000
C
C======================================================================
C  PRINT ABSCISSA & LABEL FOR ALL PLOTS
C======================================================================
C
 4000    WRITE(OUTLP,5002)
         WRITE (OUTLP,5003)
         WRITE (OUTLP,5004)
         WRITE (OUTLP,5009)
         IF (GRF(I) .NE. 'AF') GO TO 4010
         WRITE (OUTLP,5010)
         GO TO 4110
 4010    IF (GRF(I) .NE. 'GF') GO TO 4020
         WRITE (OUTLP,5020)
         WRITE (OUTLP,5110)
         GO TO 4110
 4020    IF (GRF(I) .NE. 'GW') GO TO 4030
         WRITE (OUTLP,5030)
         WRITE (OUTLP,5110)
         WRITE (OUTLP,5120)
         GO TO 4110
 4030    IF (GRF(I) .NE. 'AW') GO TO 4040
         WRITE (OUTLP,5040)
         WRITE (OUTLP,5120)
         GO TO 4110
 4040    IF (GRF(I) .NE. 'GS') GO TO 4050
         WRITE (OUTLP,5050)
         WRITE (OUTLP,5110)
         WRITE (OUTLP,5120)
         GO TO 4110
 4050    IF (GRF(I) .NE. 'AS') GO TO 4060
         WRITE (OUTLP,5060)
         WRITE (OUTLP,5120)
         GO TO 4110
 4060    IF (GRF(I) .NE. 'PO') GO TO 4070
         WRITE (OUTLP,5070)
         GO TO 4110
 4070    IF (GRF(I) .NE. 'WW') GO TO 4080
         WRITE (OUTLP,5080)
         GO TO 4110
 4080    IF (GRF(I) .NE. 'WM') GO TO 4090
         WRITE (OUTLP,5090)
         GO TO 4110
 4090    IF (GRF(I) .NE. 'MM') GO TO 4100
 4100    WRITE (OUTLP,5100)
 4110    WRITE (OUTLP,5009)
C
 5000    FORMAT ('1',20X,40A2)
 5002    FORMAT (6X,'+',12('----+----*'),'----+')
 5003    FORMAT (16X,'1',9X,'2',9X,'3',9X,'4',9X,'5',9X,'6',9X,'7',9X,
     $      '8',9X,'9',9X,'10',8X,'11',8X,'12')
 5004    FORMAT (45X,'GENERATION (X10)',/)
 5009    FORMAT (20X,100('*'))
 5010    FORMAT (25X,'FREQUENCY OF W AND M ALLELES')
 5020    FORMAT (25X,'FREQUENCY OF EACH GENOTYPE')
 5030    FORMAT (25X,'RELATIVE FITNESS OF EACH GENOTYPE')
 5040    FORMAT (25X,'RELATIVE FITNESS OF W AND M ALLELES')
 5050    FORMAT (25X,'SELECTION COEFFICIENTS OF GENOTYPES')
 5060    FORMAT (25X,'SELECTION COEFFICIENTS OF W AND M ALLELES')
 5070    FORMAT (25X,'POPULATION SIZE')
 5080    FORMAT (25X,'NUMBER OF WW GENOTYPE')
 5090    FORMAT (25X,'NUMBER OF WM GENOTYPE')
 5100    FORMAT (25X,'NUMBER OF MM GENOTYPE')
 5110    FORMAT ('+',10X,'W = WW, H = WM, M = MM')
 5120    FORMAT ('+',', ? = UNDEFINABLE COEFF')
C
 6000 CONTINUE
      WRITE (OUTLP,6010)
 6010    FORMAT (1H1)
 9999 RETURN
      END
C
C***********************************************************************
C***********************************************************************
C
      SUBROUTINE GNTFIT (DUMY)
C
C   FUNCTION: RETURNS THE RELATIVE FITNESS COEFFICIENT OF EACH GENOTYPE
C      ("FITGNT').
C
C   NOTES:
C      1) THE METHODS USED FOR THIS AND OTHER COEFFICIENT SUBROUTINES
C         ARE DETAILED ON PAGE 52 IN: WILSON, E. O., AND W. H. BOSSERT
C         (1971) "A PRIMER OF POPULATION BIOLOGY".  STAMFORD, CONN.,
C         SINAUER ASSOC.
C
C   CALLED FROM: SUMRYS, GRAPH, GNTSEL
C   CALLS: ALFREQ - RETURNS FREQUENCY OF M ALLELE
C=======================================================================
      INCLUDE 'COMMON.f'
C-----------------------------------------------------------------------
C       LOCAL DECLARATIONS
      REAL   EXPFRM,  EXPFRW,  EXPFR(3)
      INTEGER         ATOT,   DUMY
C=======================================================================
C     Type *,'In Gntfit'
C
C     TEST THAT COEFFICIENTS CAN BE CALCULATED
      IF (IGEN .NE. 1) GO TO 20
      WRITE (OUTLP,10)
   10 FORMAT (' *** WARNING: ENTRY INTO GNTFIT WITH IGEN = 1')
      GO TO 999
   20 ATOT = (RSLT(1,DUMY)+RSLT(2,DUMY)+RSLT(3,DUMY))
      LGEN = DUMY - 1
      FRQMP = ALFREQ(LGEN)
      IF ((FRQMP .GT. 0.) .AND. (FRQMP .LT. 1.) .AND. (ATOT .GT. 0))
     1   GO TO 40
      WRITE (OUTLP,30)
   30 FORMAT (' *****WARNING: ENTRY INTO GNTFIT WITH M ALLELE FREQ.',
     1   ' IN PREVIOUS GEN = 0 OR 1, OR TOTAL # ADULTS IN PRESENT GEN',
     2   ' = 0')
      GO TO 999
C-----------------------------------------------------------------------
```

```
C
C     INITIALIZE VARIABLES
   40 DO 50 J=1,3
         FITGNT(J) = 0.
         OBSFR(J) = 0.
         EXPFR(J) = 0.
   50 CONTINUE
C
C     CALCULATE OBSERVED GENOTYPE FREQUENCY
      DO 60 J=1,3
         OBSFR(J) = FLOAT(RSLT(J,DUMY))/(RSLT(1,DUMY)+
     +   RSLT(2,DUMY)+RSLT(3,DUMY))
   60 CONTINUE
C
C     CALCULATE EXPECTED FREQUENCY OF EACH ALLELE (BY THE HARDY-WEINBERG
C     LAW, THE EXPECTED ALLELE FREQUENCIES EQUAL THE FREQUENCIES IN THE
C     PREVIOUS GENERATION). THE EXPECTED GENOTYPE FREQUENCIES (BY THE
C     HARDY-WEINBERG LAW) =  P*P,  2PQ, & Q*Q FOR WW, WM & MM
C     RESPECTIVELY.
      EXPFRM = 0.
      EXPFRW = 0.
      LGEN = DUMY - 1
C
      EXPFRM = ALFREQ(LGEN)
      EXPFRW = 1 - EXPFRM
C
      EXPFR(1) = EXPFRW * EXPFRW
      EXPFR(2) = 2 * EXPFRW * EXPFRM
      EXPFR(3) = EXPFRM * EXPFRM
C
C     CALCULATE GENOTYPE SURVIVAL RATE, OR ABSOLUTE FITNESS OF EACH
C     GENOTYPE
      DO 70 J=1,3
         GNTSUR(J) = OBSFR(J) / EXPFR(J)
   70 CONTINUE
C
C     DETERMINE WHICH GENOTYPE HAS THE HIGHEST SURVIVAL RATE, CALCULATE
C     RELATIVE FITNESS COEFFICIENTS  BY DIVIDING SURVIVAL RATE OF EACH
C     GENOTYPE BY THE HIGHEST SURVIVAL RATE
      IF ((GNTSUR(2) .GE. GNTSUR(1)) .AND. (GNTSUR(2) .GE. GNTSUR(3)))
     1   GO TO 90
      IF ((GNTSUR(3) .GE. GNTSUR(1)) .AND. (GNTSUR(3) .GE. GNTSUR(2)))
     1   GO TO 100
C
C     WW HAS HIGHEST SURVIVAL RATE
C     ((GNTSUR(1) .GE. GNTSUR(2)) .AND. (GNTSUR(1) .GE. GNTSUR(3)))
   80 FITGNT(1) = 1.0
      FITGNT(2) = GNTSUR(2) / GNTSUR(1)
      FITGNT(3) = GNTSUR(3) / GNTSUR(1)
      GO TO 999
C
C     WM HAS HIGHEST SURVIVAL RATE
   90 FITGNT(2) = 1.0
      FITGNT(1) = GNTSUR(1) / GNTSUR(2)
      FITGNT(3) = GNTSUR(3) / GNTSUR(2)
      GO TO 999
C
C     MM HAS HIGHEST SURVIVAL RATE
  100 FITGNT(3) = 1.0
      FITGNT(1) = GNTSUR(1) / GNTSUR(3)
      FITGNT(2) = GNTSUR(2) / GNTSUR(3)
C
  999 RETURN
      END
C
C**********************************************************************
C**********************************************************************
C
      SUBROUTINE ALFIT (DUMY)
C
C   FUNCTION: RETURNS THE RELATIVE FITNESS COEFFICIENT OF EACH ALLELE
C      ('FITAL").
C
C   NOTES:
C      1) FOR METHODOLOGY, SEE NOTE IN SUBROUTINE GNTFIT
C
C   CALLED FROM: SUMRYS, GRAPH, ALSEL
C   CALLS: ALFREQ - RETURNS FREQUENCY OF M ALLELE
C======================================================================
      INCLUDE 'COMMON.f'
C----------------------------------------------------------------------
C  LOCAL DECLARATIONS
      INTEGER ATOT, DUMY
C======================================================================
C     Type *,'In Alfit'
C     TEST THAT COEFFICIENT CAN BE CALCULATED
      IF (IGEN .GT. 1) GO TO 10
      WRITE (OUTLP,1)
    1    FORMAT (' ***WARNING: ENTRY INTO ALFIT WITH IGEN = 1')
      GO TO 99
   10 LGEN = DUMY - 1
      ATOT = RSLT(1,LGEN)+RSLT(2,LGEN)+RSLT(3,LGEN)
      ALFRMP = ALFREQ(LGEN)
      IF ((ALFRMP .NE. 0.) .AND. (ALFRMP .NE. 1.) .AND. (ATOT .NE. 0))
     $   GO TO 30
      WRITE (OUTLP,20)
   20 FORMAT (' ***WARNING: ENTRY INTO ALFIT WITH FREQ. OF M IN ',
     $   'PREVIOUS GENERATION = 0. OR 1., OR TOTAL # ADULTS = 0.')
      GO TO 99
C
C     INITIALIZE VARIABLES
   30 ALSUR(1) = 0.
      ALSUR(2) = 0.
      FITAL(1) = 0.
      FITAL(2) = 0.
C
C     CALCULATE ALLELE SURVIVAL RATES
      ALSUR(1) = (1-ALFREQ(DUMY)) / (1-ALFREQ(LGEN))
      ALSUR(2) = (ALFREQ(DUMY)) / (ALFREQ(LGEN))
C
C     DETERMINE WHICH ALLELE HAS THE HIGHEST SURVIVAL RATE
      IF (ALSUR(1) .LT. ALSUR(2)) GO TO 40
C
C     ALLELE W HAS THE HIGHEST SURVIVAL RATE
      FITAL(1) = 1.0
      FITAL(2) = ALSUR(2) / ALSUR(1)
      GO TO 99
C
C     ALLELE M HAS THE HIGHEST SURVIVAL RATE
   40 FITAL(2) = 1.0
```

```
      FITAL(1) = ALSUR(1) / ALSUR(2)
C
   99 RETURN
      END
C
C***********************************************************************
C**********************************************************************
C
      SUBROUTINE ALSEL (DUMY)
C
C   FUNCTION: RETURNS THE SELECTION COEFFICIENT OF EACH ALLELE
C      ("ASELCO").
C
C   NOTES:
C      1) FOR METHODOLOGY, SEE NOTE IN SUBROUTINE GNTFIT
C      2) THIS SUBROUTINE COULD BE INCORPORATED INTO ALFIT
C
C   CALLED FROM: SUMRYS, GRAPH
C   CALLS: ALFIT - RETURNS FITNESS COEFFICIENT OF EACH ALLELE
C======================================================================
       INCLUDE 'COMMON.f'
C----------------------------------------------------------------------
C  LOCAL DECLARATIONS
       INTEGER DUMY
C======================================================================
C     Type *,'In Alsel'
C
C     ALLELE SELECTION COEFFICIENT = 1 - ALLELE FITNESS COEFFIENT
      CALL ALFIT (DUMY)
      DO 10 J=1,2
         ASELCO(J) = 1 - FITAL(J)
   10 CONTINUE
      RETURN
      END
C
C***********************************************************************
C**********************************************************************
C
      SUBROUTINE GNTSEL (DUMY)
C
C   FUNCTION: RETURNS THE SELECTION COEFFICIENT OF EACH GENOTYPE
C      ("GSELCO").
C
C   NOTES:
C      1) FOR METHODOLOGY, SEE NOTE IN SUBROUTINE GNTFIT
C      2) THIS SUBROUTINE COULD BE INCORPORATED INTO GNTFIT
C
C   CALLED FROM: SUMRYS, GRAPH
C   CALLS: GNTFIT - RETURNS FITNESS COEFFICIENT OF EACH GENOTYPE
C======================================================================
      INCLUDE 'COMMON.f'
C----------------------------------------------------------------------
C  LOCAL DECLARATIONS
      INTEGER DUMY
C======================================================================
C     Type *,'In Gntsel'
C
C     GENOTYPE SELECTION COEFFICIENT = 1 - GENOTYPE FITNESS COEFFICIENT
      CALL GNTFIT (DUMY)
      DO 10 J=1,3
```

```
         GSELCO(J) = 1 - FITGNT(J)
   10 CONTINUE
      RETURN
      END
C
C***********************************************************************
C
      FUNCTION ALFREQ (DUMY)
C
C   FUNCTION: FREQUENCY OF MUTANT ALLELE M (ALFREQ) IN SPECIFIED
C      GENERATION
C
C   NOTES:
C      1)
C
C   CALLS FROM: SRVIVL, GRAPH, GNTFIT
C   CALLS: NOTHING
C=======================================================================
      INCLUDE 'COMMON.f'
C----------------------------------------------------------------------
C  LOCAL DECLARATIONS
      INTEGER DUMY
C=======================================================================
C     Type *,'In Alfreq'
C
C     VARIABLE ALFREQ IS THE ALLELE FREQUENCY OF THE M ALLELE
C     IF THERE ARE NO WM OR MM ADULTS, SET ALFREQ=0, OTHERWISE COMPUTE
C     ALFREQ
      IF ((RSLT(2,DUMY) .EQ. 0) .AND. (RSLT(3,DUMY) .EQ. 0)) GO TO 10
C
      ALFREQ = (RSLT(2,DUMY) + (2*FLOAT(RSLT(3,DUMY)))) /
     $   (2*((RSLT(1,DUMY)+RSLT(2,DUMY)+RSLT(3,DUMY))))
      GO TO 20
C
   10 ALFREQ = 0.
   20 RETURN
      END
C
C**********************************************************************
C**********************************************************************
C
      SUBROUTINE GAUSS
C
C   FUNCTION: RETURNS FLOATING POINT NUMBER "V" WITH MEAN = AM AND
C      STD. DEV. = S
C
C   NOTES:
C      1) THIS ALGORITHM WAS PROVIDED BY A COMPUTER PERSON WHO HAS
C         SINCE MOVED.  HE DID NOT PROVIDE DOCUMENTATION, OTHER THAN
C         A NOTE TO SEE VOL. 2 OF KNUTH'S "ART OF COMPUTER PROGRAMMING"
C      2) BEYOND AN AWARENESS THAT RANDOM NUMBER ALGORITHMS ARE A
C         TARPIT FOR THE UNWARY, I AM NOT KNOWLEDGABLE IN THIS AREA.
C         I CANNOT VOUCH FOR THE VALIDITY OR TRANSPORTABILITY OF THIS
C         ALGORITHM. I HOPE CONDUIT CAN HELP BY VALIDATING THIS
C         ALGORITHM OR DONATING A MORE TRANSPORTABLE ONE.
C
C   CALLED FROM: REPROD
C   CALLS: RAN   - FORTRAN LIBRARY FUNCTION, RETURNS RANDOM NUMBERS
C                  UNIFORMLY DISTRIBUTED FROM 0.0->1.0
C          SQRT  - FORTRAN LIBRARY FUNCTION, RETURNS SQUARE ROOT
```

```
C          ALOG  - FORTRAN LIBRARY FUNCTION, RETURNS NATURAL LOGARITHM
C======================================================================
      INCLUDE 'COMMON.f'
C----------------------------------------------------------------------
C     LOCAL DECLARATIONS
      REAL     Q,  U1,  U2,   V1,  V2,  X1
C======================================================================
C     Type *,'In Gauss'
C     Type *, IY
C     Type *, IZ
C     Pause
C
  100 U1 = RAN(IY)
      U2 = RAN(IY)
      V1 = 2 * U1 - 1
      V2 = 2 * U2 - 1
      Q = V1 * V1 + V2 * V2
      IF (Q .GE. 1) GO TO 100
      X1 = V1 * SQRT(-2 * ALOG(Q) / Q)
      V = X1 * S + AM
      RETURN
      END
C
C**********************************************************************


C     file: COMN   vrsn: 14   rvsd: 13 Aug 81
C-----------------------------------------------------------------------
C GLOBAL DECLARATIONS
      COMMON /A/  ALSUR(2),  AM,  ASELCO(2),  FITAL(2),
     1   FITGNT(3), FRQMP,  GNTSUR(3),  GRF(10),  GSELCO(3),  IGEN,
     2   IN,  INFLO(3),  IY,  IZ,  JUV(3),  LGEN,  MATES(6),  MXPOP1,
     3   MXPOP2,  NEXT,  NOADLT(3),  NOGENS,  NOOUT(3),  NOYONG(6),
     4   OBSFR(3),  OUTFLO(3),  OUTLP,  OUTSCR,  PRINT2,  REPR8(3),
     5   RSLT(3,125),  S,  SUMGEN,  SUMRY,  SURVR8(3),  TITLE(40)
      COMMON /B/  TOADLT,  V
C
      INTEGER  IGEN,  IN,  INFLO,  IY,  IZ,
     1   LGEN,  MATES,  MXPOP1,  MXPOP2,  NOADLT,  NOGENS,
     2   OUTFLO,  OUTLP,  OUTSCR,  PRINT2,  RSLT,  SUMGEN,
     3   SUMRY,  TOADLT
C
      REAL  ALSUR,  AM,  ASELCO,  FITAL,  FITGNT,  FRQMP,  GNTSUR,
     1   GSELCO,  JUV,  NOOUT,  NOYONG,  OBSFR,  REPR8,  S,  SURVR8,  V
C
      STRING*2  ANS,  GRF,  NEXT
C
      STRING*80  TITLE
C-----------------------------------------------------------------------
```