

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

```

60 C      ALLELE, THE SECOND ELEMENT CORRESPONDS TO THE "M" ALLELE.
61 C      B) 3-ELEMENT ARRAYS STORING INFORMATION ON GENOTYPES (E.G.,
62 C      REPR8(3), SURVR8(3)): FIRST ELEMENT CORRESPONDS TO THE "WW"
63 C      GENOTYPE, THE SECOND TO THE "WM" GENOTYPE, THE THIRD TO
64 C      THE "MM" GENOTYPE.
65 C      C) RSLT(3,125) STORES THE RESULTS, I.E., THE NUMBER OF ADULTS
66 C      OF EACH GENOTYPE IN EACH GENERATION. THE GENOTYPE
67 C      IS INDICATED BY THE FIRST INDEX, THE NUMBER OF THE
68 C      GENERATION IS INDICATED BY THE SECOND INDEX.
69 C
70 C  NOTES ON POSSIBLE CHANGES
71 C      1) STATEMENT NUMBERS SHOULD BE CLEANED UP -- SET IN NUMERICAL
72 C      ORDER, INCREMENT BY 10'S.
73 C      2) CLEAN UP LEVELS OF INDENTATION
74 C      3) IO WILL HAVE TO BE MADE MORE TRANSPORTABLE.
75 C      4) TOO MUCH OUTPUT IS SENT TO LINE PRINTER. SHOULD MODIFY TO DO
76 C      GENERATIONS REQUESTED, DISPLAY SUMMARY TO CRT, ASK USER WHETHER
77 C      SUMMARIES &/OR GRAPHS ARE NEEDED. USER SHOULD BE ABLE TO MOVE
78 C      TO & FROM SUMMARIES AND THE VARIOUS GRAPHS ON THE CRT BEFORE
79 C      SENDING HARD COPY TO LINE PRINTER IF NEEDED. IMPLEMENTING
80 C      THIS FEATURE WILL PROBABLY REQUIRE ANOTHER SUBROUTINE
81 C      ("DISPLA"? ) TO CONTROL MENUES & CALL UP VARIOUS GRAPHS. WILL
82 C      NEED TO ADD VARIOUS GLOBAL VARIABLES (PAGESIZE, LINESIZE &
83 C      #LINESPRINTED, ETC.) TO PAGE SUMRYS OUTPUT, SET GRAPH SIZES TO,
84 C      FOR E.G., 80 X 24 AND 66 X 132, ETC.
85 C
86 C*****
87 C**BEGIN MAIN ROUTINE
88 C
89 C  FUNCTION: CONTROL OVERALL FLOW OF THE PROGRAM
90 C
91 C  CALLS: INPUT - INPUT AND CHECK PARAMETERS FOR RUN; RETURNS ALL
92 C              VARIABLES NEEDED FOR A RUN.
93 C      REPROD - DETERMINE NUMBER OF JUVENILES IN CURRENT
94 C              GENERATION; RETURNS "JUV" ARRAY.
95 C      SRVIVL - DETERMINE NUMBER OF ADULTS IN CURRENT GENERATION
96 C              FOLLOWING EMIGRATION, MORTALITY & IMMIGRATION;
97 C              CRASH EXCESSIVELY LARGE POPULATION; RETURNS
98 C              NUMBER OF ADULTS IN "RSLTS" ARRAY.
99 C      SUMRYS - PRINT SUMMARIES OF GENERATIONS
100 C      GRAPH - PRINT VARIOUS GRAPHS OF RESULTS
101 C      YESNO - READ & TEST YES/NO RESPONSES; RETURNS EITHER "Y" OR
102 C              "N" IN VARIABLE "ANS".
103 C      RESET - REINITIALIZE VARIABLES BEFORE A NEW RUN.
104 C=====
105 C      INCLUDE 'COMMON.f'
106 C-----
107 C      LOCAL DECLARATIONS
108 C      INTEGER ALMFLG, ALWFLG, ATOT, ICOUNT, POPFLG, SCRCNT
109 C=====
110 C
111 C  INPUT DATA, SET ALLELE & POP.  FLAGS & PRINT SUMMARY OF INITIAL POP.
112 C-----
113 C  Change scrolling of output window to single line scrolling
114 C      CALL OutWindowScroll(9999)
115 C
116 C      WRITE (OUTSCR,10)
117 C      10  FORMAT ('1',80('*'),/,20X,'EVOLVE version 3.1, 27 Aug. 1981',/,
118 C      $      80('*'),/)
119 C      100 CALL INPUT
    
```

```
120 D
121 D      WRITE(OUTLP,200) IGEN,(RSLT(K,IGEN),K=1,3)
122 D 200 FORMAT(' On return from INPUT:',/, ' IGEN = ',I4,' RSLT = ',3(I7))
123 D      WRITE(OUTLP,205) MXPOP1,MXPOP2
124 D 205 FORMAT(' K = ',I5,10X,'POST-K = ',I5)
125 D      WRITE(OUTLP,210) (SURVR8(K),K=1,3)
126 D 210 FORMAT(' SURVR8 = ',3(F10.2))
127 D      WRITE(OUTLP,215) (REPR8(K),K=1,3)
128 D 215 FORMAT(' REPR8 = ',3(F10.2))
129 D      WRITE(OUTLP,220) (INFLO(K),K=1,3)
130 D 220 FORMAT(' INFLO = ',3(I7))
131 D      WRITE(OUTLP,225) (OUTFLO(K),K=1,3)
132 D 225 FORMAT(' OUTFLO = ',3(I7))
133 D      WRITE(OUTLP,230) NOGENS, SUMRY, NEXT
134 D 230 FORMAT(' NOGENS = ',I7,'SUMRY = ',I7,'NEXT = ',A4)
135 D      WRITE(OUTLP,235) IGEN
136 D 235 FORMAT(' IGEN = ',I4,/)
137 C
138 C      INITIALIZE FLAGS FOR ALLELE & POPULATION EXTINCTION,  COUNTER FOR
139 C      SUMMARIES, HEADER FOR PROGRESS REPORT TO SCREEN.
140 120 ALWFLG = 0
141     ALMFLG = 0
142     POPFLG = 0
143     ICOUNT = 0
144     SCRCNT = 1
145     IF (IGEN .GT. 1) GO TO 300
146     SUMGEN = IGEN
147     PRINT2 = OUTLP
148     CALL SUMRYS
149 300 WRITE (OUTSCR,301)
150 301   FORMAT ('0',X,'Doing generation:')
151 C
152 C-----
153 C DO NUMBER OF GENERATIONS REQUESTED
154 C-----
155     DO 390 I=1,NOGENS
156 C
157 C      TEST THAT # GENS DONE < 126; CONTINUE IF TRUE, EXIT IF FALSE.
158 C      IF (IGEN .GT. 125) GO TO 400
159 C
160 C      WRITE PROGRESS REPORT TO SCREEN
161 C      IGEN = IGEN + 1
162 C      SCRCNT = SCRCNT + 1
163 302   WRITE (OUTSCR,303) IGEN
164 303   FORMAT (2X,I3,$)
165 C      IF (SCRCNT .LE. 19) GO TO 305
166 C      SCRCNT = 0
167 C
168 C      REPRODUCTION AND SURVIVAL - HEART OF THE PROGRAM
169 305   CALL REPROD
170       CALL SRVIVL
171 C
172 C      SUMMARIES (PRINTING THESE TAKES A LOT OF TIME)
173 C      IF (SUMRY .EQ. 0) GO TO 309
174 C      ICOUNT = ICOUNT + 1
175 C      IF (SUMRY .NE. ICOUNT) GO TO 309
176 C      PRINT2 = OUTLP
177 C      SUMGEN = IGEN
178 C      CALL SUMRYS
179 C      ICOUNT = 0
```

```
180 309 CONTINUE
181 C
182 C TEST FOR EXTINCTIONS & PRINT MESSAGE IF HAS JUST
183 C HAPPENED (THESE COULD GO INTO A SUBROUTINE ["XTNCTN"?)
184 C TO CLARIFY LOGIC & CLEAN UP MAIN ROUTINE)
185 C
186 C TEST FOR ALLELE EXTINCTION
187 IF ((RSLT(1,IGEN) .GT. 0) .OR. (RSLT(2,IGEN) .GT. 0))
188 $ GO TO 320
189 ALWFLG = ALWFLG + 1
190 IF (ALWFLG .GT. 1) GO TO 325
191 WRITE (OUTSCR,310) IGEN
192 310 FORMAT ('0',/, ' W ALLELE BECAME EXTINCT IN GENERATION ',I3)
193 WRITE (OUTLP,310) IGEN
194 GO TO 325
195 320 ALWFLG = 0
196 C
197 325 IF ((RSLT(3,IGEN) .GT. 0) .OR. (RSLT(2,IGEN) .GT. 0))
198 1 GO TO 340
199 ALMFLG = ALMFLG + 1
200 IF (ALMFLG .GT. 1) GO TO 350
201 WRITE (OUTSCR,330) IGEN
202 330 FORMAT ('0',/, ' M ALLELE BECAME EXTINCT IN GENERATION ',I3)
203 WRITE (OUTLP,330) IGEN
204 GO TO 350
205 340 ALMFLG = 0
206 C
207 C
208 C TEST FOR POPULATION EXTINCTION
209 350 ATOT = RSLT(1,IGEN) + RSLT(2,IGEN) + RSLT(3,IGEN)
210 IF (ATOT .GT. 0) GO TO 370
211 POPFLG = POPFLG + 1
212 IF (POPFLG .GT. 1) GO TO 390
213 WRITE (OUTSCR,360) IGEN
214 360 FORMAT ('0',/, ' POPULATION BECAME EXTINCT IN GENERATION ',I3)
215 WRITE (OUTLP,360) IGEN
216 GO TO 390
217 370 POPFLG = 0
218 C
219 C
220 390 CONTINUE
221 WRITE (OUTSCR,395)
222 395 FORMAT ('0',/)
223 C
224 C-----
225 C CLEANUP AFTER GENERATIONS HAVE BEEN FINISHED
226 C (THIS WOULD BE THE PLACE TO ADD THE "DISPLA" SUBROUTINE; MUCH OF
227 C THIS CODE WOULD GO INTO THAT SUBROUTINE)
228 C-----
229 C
230 C DETERMINE WHETHER GRAPHS OR PARAMETER CHANGE DESIRED
231 400 IF (NEXT .EQ. 'GR') GO TO 500
232 IF (NEXT .EQ. 'CH') GO TO 100
233 C
234 C CALL SUMRYS, IF REQUESTED
235 500 IF ((SUMRY .NE. 0) .AND. (ICOUNT .EQ. 0)) GO TO 600
236 SUMGEN = IGEN
237 PRINT2 = OUTLP
238 CALL SUMRYS
239 C
```

```

240 C    PRINT 1ST & LAST GENERATIONS TO SCREEN, ASK IF GRAPHS DESIRED,
241 C    CALL GRAPH IF NEEDED
242 600 SUMGEN = 1
243     PRINT2 = OUTSCR
244     CALL SUMRYS
245     SUMGEN = IGEN
246     CALL SUMRYS
247     WRITE (OUTSCR, 610)
248 610 FORMAT (' Do you wish graphs (Y/N)? ', $)
249     CALL YESNO (ANS)
250     IF (ANS .EQ. 'N') GO TO 700
251     IF (GRF(1) .EQ. ' ') GO TO 700
252     CALL GRAPH
253 C
254 C    ASK FOR CONTINUATION.  IF NO, STOP; IF YES, RESET GENERATION
255 C    NUMBER AND RESULTS ARRAY
256 700 WRITE (OUTLP, 710)
257 710     FORMAT (1H1)
258     WRITE (OUTSCR, 720)
259 720 FORMAT (//, 3X, 'Run finished.  Do you wish to make another ',
260 + 'run (Y/N)? ', $)
261     CALL YESNO (ANS)
262     IF (ANS .EQ. 'N') GO TO 999
263     CALL RESET
264     GO TO 100
265 C
266 999 STOP
267     END
268 C
269 C*****
270
271 C*****
272 C
273     BLOCK DATA
274 C     SUBROUTINE INITIALIZEDATA
275 C
276 C     FUNCTION: INITIALIZES VARIABLES IN COMMON BLOCKS
277 C
278 C=====
279     INCLUDE 'COMMON.f'
280 C=====
281 C *** INITIALIZE COMMON VARIABLES
282 C
283     DATA  ALSUR/2*0./, AM/0./, ANS/' '/, ASELCO/2*0./,
284 1  FITAL/2*0./, FITGNT/3*0./, FRQMP/0./, GNTSUR/3*0./,
285 2  GRF/10*' '/, GSELCO/3*0./, IGEN/1/, IN/5/, INFL0/3*0/,
286 3  IY/12345/, IZ/12345/, JUV/3*0./, LGEN/0/, MATES/6*0./,
287 4  MXPOP1/0/, MXPOP2/0/, NEXT/' '/, NOADLT/3*0/, NOGENS/0/,
288 5  NOOUT/3*0./, NOYONG/6*0./, OBSFR/3*0./, OUTFL0/3*0/
289     DATA  OUTLP/6/, OUTSCR/6/, PRINT2/6/, REPR8/3*0./,
290 1  RSLT/375*0/, S/0./, SUMGEN/1/, SUMRY/0/, SURVR8/3*0./,
291 2  TITLE/40*' '/, TOADLT/0/, V/0./
292     END
293 C*****
294 C*****
295 C
296     SUBROUTINE RESET
297 C
298 C     FUNCTION: REINITIALIZES VARIABLES FOLLOWING A RUN.
299 C

```

```
300 C   NOTES:
301 C       1) APPROPRIATE ARRAYS AND VARIABLES ARE RETURNED TO VALUES IN
302 C         BLOCK DATA, EXCEPT THAT FIRST GENERATION (INITIAL POPULATION)
303 C         IN RESULTS ARRAY RETAINS THE VALUES INPUT BY THE USER.
304 C       2) VARIABLES INPUT BY USER RETAIN THEIR VALUES SO USER CAN MAKE
305 C         REPEAT RUNS WITHOUT REENTERING ALL DATA. HOPEFULLY, ANY
306 C         REMAINING VARIABLES NOT RESET ARE NOT IMPORTANT.
307 C       3) ALL MAJOR VARIABLES ARE IN COMMON.
308 C
309 C   CALLED FROM: MAIN
310 C   CALLS: NOTHING
311 C=====
312 C       INCLUDE 'COMMON.f'
313 C=====
314 C
315 C       Type *, 'In Reset'
316 C       DO 100 I = 1,3
317 C         DO 10 J = 2,125
318 C           RSLT(I,J) = 0
319 C       10 CONTINUE
320 C         JUV(I) = 0.
321 C         NOOUT(I) = 0
322 C       C       NOIN(I) = 0
323 C       100 CONTINUE
324 C         DO 200 J = 1,6
325 C           MATES(J) = 0
326 C           NOYONG(J) = 0.
327 C       200 CONTINUE
328 C         IZ = 12345
329 C         IGEN = 1
330 C         RETURN
331 C         END
332 C
333 C*****
334 C       file: in.pu  vrsn: 145  rvsd: 27 Aug 81
335 C       subroutines input, yesno, inpsum
336 C*****
337 C
338 C       SUBROUTINE INPUT
339 C
340 C       FUNCTION: THIS SUBROUTINE READS, CHECKS & PRINTS DATA INPUT.
341 C
342 C       SYNOPSIS:
343 C       1) THE FOLLOWING VARIABLES (ALL IN COMMON) ARE READ IN:
344 C         TITLE - 80 CHARACTER (40A2) TITLE FOR OUTPUT
345 C         IY - RANDOM NUMBER SEED; 0->32760
346 C         RSLT - NUMBER OF EACH GENOTYPE IN FIRST GENERATION; ARRAY
347 C           ELEMENTS (1,1), (2,1), (3,1); EACH 0->4000, TOTAL < 4001
348 C         MXPOP1 - CARRYING CAPACITY, MAXIMUM PERMISSIBLE POPULATION
349 C           SIZE; 10->5000
350 C         MXPOP2 - POST-CRASH POPULATION SIZE; 2->4000
351 C         SURVR8 - SURVIVAL RATE OF EACH GENOTYPE; 0.0->100.0
352 C         REPR8 - REPRODUCTIVE RATE OF EACH GENOTYPE; 0.0->10.0
353 C         INFLO - NUMBER OF EACH GENOTYPE IMMIGRATING PER GENERATION;
354 C           0->2000
355 C         OUTFLO - % OF EACH GENOTYPE EMIGRATING PER GENERATION; 0->100
356 C         NOGENS - NUMBER OF GENERATIONS TO BE RUN WITH THESE DATA;
357 C           1->124
358 C         SUMRY - NUMBER OF GENERATIONS BETWEEN PRINTED SUMMARIES;
359 C           0 = NO SUMMARIES, 1->99
```

```

360 C      NEXT  - WHAT TO DO AFTER GENERATIONS ARE FINISHED;
361 C      "CH" = CHANGE PARAMETERS, "GR" = GRAPH RESULTS
362 C      GRF    - UP TO 10 2-LETTER CODES FOR TYPES OF GRAPHS DESIRED;
363 C      SEE USERS' MANUAL FOR SPECIFICS
364 C      2) USER IS PROMPTED FOR EACH DATUM, WHICH IS CHECKED AGAINST
365 C      LIMITS & AN ERROR MESSAGE PRINTED IF NEEDED. USER CAN THEN
366 C      REENTER DATA.
367 C      3) AFTER ALL DATA ARE ENTERED, SUBROUTINE INPSUM IS CALLED TO
368 C      PRINT A SUMMARY TO THE SCREEN FOR THE USER'S INSPECTION.
369 C      4) IF DATA ARE OK, A SUMMARY IS PRINTED TO THE LINE PRINTER BY
370 C      SUBROUTINE INPSUM AND CONTROL RETURNS TO MAIN.
371 C      5) IF USER DETECTS AN ERROR, VARIABLE "FIX" IS SET TO 'Y' AND
372 C      CONTROL RETURNS TO THE TOP OF SUBROUTINE INPUT. VALUES OF
373 C      DATA ARE DISPLAYED & THE USER PROMPTED TO CORRECT THEM IF
374 C      NECESSARY.
375 C      6) IF VAR. "NEXT" = "CH", SUBROUTINE INPUT HAS BEEN CALLED
376 C      BEFORE, SO "IY" AND INITIAL POPULATION SHOULD NOT BE CHANGED;
377 C      CONTROL JUMPS TO CARRYING CAPACITY.
378 C
379 C      NOTES:
380 C      1) THIS ROUTINE RELIES HEAVILY ON THE "ERR=" DEC FORTRAN
381 C      EXTENSION TO AVOID AWKWARD SYSTEM ABORTS. THIS WILL NEED
382 C      TO BE CHANGED TO IMPROVE TRANSPORTABILITY.
383 C      2) THE RANDOM NUMBER SEED MAY NEED TO BE CHANGED FOR OTHER
384 C      SYSTEMS. THE SEED COULD BE SET IN OTHER WAYS (E.G., FROM A
385 C      SYSTEM CLOCK), BUT I HAVE FOUND IT USEFUL TO HAVE STUDENTS
386 C      BE ABLE TO REPEAT RUNS EXACTLY.
387 C
388 C      CALLED FROM: MAIN
389 C      CALLS: YESNO  - TO READ AND CHECK YES/NO INPUT; RETURNS EITHER "Y"
390 C                  OR "N" IN VARIABLE "ANS".
391 C      INPSUM  - TO PRINT SUMMARIES OF DATA TO SCREEN
392 C              (IF "PRINT2" = "OUTSCR") OR TO LINE PRINTER
393 C              (IF "PRINT2" = "OUTLP")
394 C=====
395 C      INCLUDE 'COMMON.f'
396 C-----
397 C      LOCAL DECLARATIONS
398 C      INTEGER ATOT, GNOT(3), IFLOW, TGENS
399 C      REAL RRATE, SRATE
400 C      STRING*2 FIX, GRFTST(11)
401 C      DATA ATOT/0/, FIX/'N'/,
402 C      1 GNOT/'WW','WM','MM'/,
403 C      2 GRFTST/' ','AF','GF','PO','WW','WM','MM','AW','GW','AS',
404 C      3 'GS'/,
405 C      4 RRATE/0./, SRATE/0./
406 C=====
407 C
408 C      Type *, 'In Input'
409 C
410 C      IF NEXT IS 'CH', SUBROUTINE INPUT HAS ALREADY BEEN
411 C      CALLED, JUMP OVER INITIAL PARAMETERS TO CARRYING CAPACITY
412 C      10 IF (FIX .EQ. 'Y') GO TO 100
413 C      IF (NEXT .EQ. 'CH') GO TO 500
414 C
415 C-----
416 C      INPUT DATA
417 C-----
418 C
419 C      TITLE FOR OUTPUT

```

```
420 C
421 C      (THERE MUST BE A MORE ELEGANT WAY TO TEST FOR BLANK TITLE)
422 100 DO 102 I=1,40
423     IF (TITLE(I) .EQ. ' ') GO TO 102
424     GO TO 108
425 102   CONTINUE
426     GO TO 120
427 C
428 108 WRITE (OUTSCR,110) TITLE
429 110 FORMAT (/ ,5X, 'Title is: ',X,40A2)
430     WRITE (OUTSCR,115)
431 115 FORMAT (5X, 'Is this o.k. (Y/N)? ', $)
432     CALL YESNO (ANS)
433     IF (ANS .EQ. 'Y') GO TO 200
434 C
435 120 WRITE (OUTSCR,130)
436 130 FORMAT ('0',5X, 'Enter a title (79 characters max) for output:')
437     READ (IN,140) TITLE
438 140 FORMAT(40A2)
439 D    WRITE (OUTSCR,150)TITLE
440 D 150 FORMAT ('0 1. ', 40A2)
441 C-----
442 C
443 C  RANDK SEED
444 C
445 C      THE SEED SIZE MAY NEED TO BE CHANGED FOR DIFFERENT COMPUTERS
446 200 IF (IY .EQ. 12345) GO TO 220
447     WRITE (OUTSCR,210) IY
448 210   FORMAT (/ ,5X, 'Seed is: ',I5)
449     WRITE (OUTSCR,115)
450     CALL YESNO (ANS)
451     IF (ANS .EQ. 'Y') GO TO 300
452 220 WRITE (OUTSCR,230,ERR=250)
453 230   FORMAT (/ ,5X, 'Enter random number seed (0-32760): ', $)
454     READ (IN,240,ERR=250) IY
455 240 FORMAT (I5)
456 C
457 C      TEST: RANDK SEED MUST BE BETWEEN 00000 AND 32760
458     IF ((IY .GE. 00000) .AND. (IY .LE. 32760)) GO TO 270
459 250   WRITE (OUTSCR,260)
460 260   FORMAT (' *****ERROR: Seed is incorrect.',/,13X,
461 $         'it must be an integer between 0 and 32760'))
462     WRITE (OUTSCR,265)
463 265   FORMAT (13X, 'Please try again.')
464     GO TO 220
465 270 CONTINUE
466 C
467 D    WRITE (OUTSCR,280) IY
468 D 280 FORMAT ('0 2. RANDOM NUMBER SEED   : ',I5)
469 C-----
470 C
471 C  INITIAL POPULATION
472 C
473 300 ATOT = RSLT(1,1) + RSLT(2,1) + RSLT(3,1)
474     IF (ATOT .EQ. 0) GO TO 320
475     WRITE (OUTSCR,310) RSLT(1,1), RSLT(2,1), RSLT(3,1), ATOT
476 310   FORMAT (/ ,5X, 'The initial population is: WW = ',I4,
477 $         ' WM = ',I4, ' MM = ',I4, ' Total = ',I4)
478     WRITE (OUTSCR,115)
479     CALL YESNO (ANS)
```



```

480         IF (ANS .EQ. 'Y') GO TO 500
481 C
482     320 DO 380 I=1,3
483     330     WRITE (OUTSCR,340) GNOT(I)
484     340     FORMAT (/ ,5X,'Enter initial number of ',A2,' genotype ',
485     $         '(0-4000): ', $)
486     READ (IN,350,ERR=360) RSLT(I,1)
487     350     FORMAT (I4)
488     IF ((RSLT(I,1) .GE. 0) .AND. (RSLT(I,1) .LE. 4000)) GO TO 380
489     360     WRITE (OUTSCR,370) GNOT(I)
490     370     FORMAT (' *****ERROR: Initial number of ',A2,
491     $         ' genotype is incorrect. ',/,13X,'It must be an ',
492     $         'integer between 0 and 4000.')
493     WRITE (OUTSCR,265)
494     GO TO 330
495     380     CONTINUE
496 C
497 C     TEST: INITIAL POPULATION MUST TOTAL BETWEEN 2 AND 4000 INDIVIDUALS
498     ATOT = RSLT(1,1) + RSLT(2,1) + RSLT(3,1)
499     IF ((ATOT .GE. 2) .AND. (ATOT .LE. 4000)) GO TO 400
500     WRITE (OUTSCR,390)
501     390     FORMAT (' *****ERROR: Initial population is out of ',
502     $         'range.',/,13X,'The total initial population must be ',
503     $         'between 2 and 4000')
504     WRITE (OUTSCR,265)
505     GO TO 320
506     400     CONTINUE
507 C
508 D     WRITE (OUTSCR,410) (RSLT(I,1),I=1,3), ATOT
509 D 410     FORMAT ('0 3. INITIAL POPULATION : WW = ',I4,2X,'WM = ',I4,
510 D     $     2X,'MM = ',I4,2X,'TOTAL = ',I4)
511 C-----
512 C
513 C     CARRYING CAPACITY & POST-CRASH POP SIZE
514 C
515     500 IF ((MXPOP1 .EQ. 0) .AND. (MXPOP2 .EQ. 0)) GO TO 509
516     WRITE (OUTSCR,501) MXPOP1, MXPOP2
517     501     FORMAT (/ ,5X,'Carrying capacity is: ',I4,' Post-crash level ',
518     $         ' is:',I4)
519     WRITE (OUTSCR,115)
520     CALL YESNO (ANS)
521     IF (ANS .EQ. 'Y') GO TO 600
522 C.....
523 C
524     509     WRITE (OUTSCR,510)
525     510     FORMAT (/ ,5X,'Enter carrying capacity (10-5000): ', $)
526     READ (IN,520,ERR=525) MXPOP1
527     520     FORMAT (I4)
528 C
529 C     TEST: CARRYING CAPACITY MUST BE BETWEEN 10 AND 5000 INDIVIDUALS
530     IF ((MXPOP1 .GE. 10) .AND. (MXPOP1 .LE. 5000)) GO TO 540
531     525     WRITE (OUTSCR,530)
532     530     FORMAT (' *****ERROR: Carrying capacity is incorrect.',/,13X,
533     $         'it must be an integer between 10 and 5000.')
534     WRITE (OUTSCR,265)
535     GO TO 509
536 C.....
537 C
538     540     WRITE (OUTSCR,550)
539     550     FORMAT (/ ,5X,'Enter post-crash population size (2-4000): ', $)
    
```

```

540     READ (IN,560,ERR=575) MXPOP2
541     560     FORMAT (I4)
542     C
543     C     TEST: POST-CRASH SIZE MUST BE BETWEEN 2 AND 4000 INDIVIDUALS
544     570 IF ((MXPOP2 .GE. 2) .AND. (MXPOP2 .LE. 4000)) GO TO 590
545     575     WRITE (OUTSCR,580)
546     580     FORMAT (' *****ERROR: Post-crash pop. size is incorrect.',/,
547     $     13X,'It must be an integer between 2 and 4000.')
548     WRITE (OUTSCR,265)
549     GO TO 540
550     C.....
551     C
552     C     TEST: CARRYING CAPACITY MUST BE GREATER THAN POST-CRASH SIZE
553     590 IF (MXPOP1 .GT. MXPOP2) GO TO 594
554     WRITE (OUTSCR,592)
555     592     FORMAT (' *****ERROR: Post-crash pop. size is greater than',
556     1     ' carrying capacity.',/13X,'Please reenter both parameters.')
557     GO TO 509
558     594 CONTINUE
559     C
560     D     WRITE (OUTSCR,596) MXPOP1, MXPOP2
561     D 596     FORMAT (' 4. CARRYING CAPACITY      : ',I4,/4X,
562     D $     ' POST-CRASH POP. SIZE : ', I4)
563     C-----
564     C
565     C SURVIVAL & REPRODUCTIVE RATES OF EACH GENOTYPE
566     C
567     600 RRATE = REPR8(1) + REPR8(2) + REPR8(3)
568     SRATE = SURVR8(1) + SURVR8(2) + SURVR8(3)
569     IF ((RRATE .EQ. 0.) .AND. (SRATE .EQ. 0.)) GO TO 700
570     WRITE (OUTSCR,620) SURVR8(1), SURVR8(2), SURVR8(3), REPR8(1),
571     $     REPR8(2), REPR8(3)
572     620     FORMAT (/5X,'Survival rates are:      WW = ',F5.1,' WM = ',
573     $     F5.1,' MM = ',F5.1,/5X,'Reproductive rates are: WW = ',
574     $     F4.1,' WM = ',F4.1,' MM = ',F4.1,/5X,'Are survival ',
575     $     'rates o.k. (Y/N)?',,$)
576     CALL YESNO (ANS)
577     IF (ANS .EQ. 'Y') GO TO 760
578     C.....
579     C
580     C SURVIVAL RATES
581     700 DO 750 I=1,3
582     710     WRITE (OUTSCR,715) GNOT(I)
583     715     FORMAT (/5X,'Enter survival rate of genotype ',A2,
584     $     ' (000.0-100.0%): ',,$)
585     READ (IN,720,ERR=730) SURVR8(I)
586     720     FORMAT (F5.1)
587     IF ((SURVR8(I) .GE. 0.) .AND. (SURVR8(I) .LE. 100.)) GO TO 750
588     730     WRITE (OUTSCR,735) GNOT(I)
589     735     FORMAT (' *****ERROR: Survival rate of ',A2,' genotype ',
590     $     'is incorrect.',/13X,'It must be a decimal number ',
591     $     'between 000.0 and 100.0%.')
592     WRITE (OUTSCR,265)
593     GO TO 710
594     750     CONTINUE
595     C.....
596     C
597     C REPRODUCTIVE RATES OF EACH GENOTYPE
598     760 IF (SRATE .EQ. 0.) GO TO 770
599     WRITE (OUTSCR,765)

```

```

600 765 FORMAT (/5X,'Are reproductive rates o.k. (Y/N)? ', $)
601 CALL YESNO (ANS)
602 IF (ANS .EQ. 'Y') GO TO 800
603 C
604 770 DO 790 I=1,3
605 772 WRITE (OUTSCR,775) GNOT(I)
606 775 FORMAT (/5X,'Enter reproductive rate of genotype ',A2,
607 $ ' (0.0-10.0): ', $)
608 READ (IN,780,ERR=785) REPR8(I)
609 780 FORMAT (F4.1)
610 IF ((REPR8(I) .GE. 0.) .AND. (REPR8(I) .LE. 10.)) GO TO 790
611 785 WRITE (OUTSCR,786) GNOT(I)
612 786 FORMAT (' *****ERROR: Reproductive rate of genotype ',A2,
613 $ ' is incorrect.',/13X,'It must be a decimal number ',
614 $ 'between 00.0 and 10.0.')
615 WRITE (OUTSCR,265)
616 GO TO 772
617 790 CONTINUE
618 C
619 D WRITE (OUTSCR,795) (SURVR8(I), I=1,3)
620 D 795 FORMAT (' 5. GNOT. SURVIVAL RATES : WW = ',F5.1,'% WM = ',
621 D $ F5.1,'% MM = ',F5.1,'%')
622 D WRITE (OUTSCR,796) (REPR8(I), I=1,3)
623 D 796 FORMAT (' GNOT. REPROD. RATES : WW = ',F4.1,2X,'WM = ',
624 D $ F4.1,2X,'MM = ',F4.1)
625 C-----
626 C
627 C GENE FLOW
628 C
629 800 IFLOW = 0
630 DO 810 I=1,3
631 IFLOW = INFLO(I) + OUTFLO(I)
632 810 CONTINUE
633 IF (IFLOW .EQ. 0) GO TO 830
634 WRITE (OUTSCR,820) INFLO(1), INFLO(2), INFLO(3), OUTFLO(1),
635 $ OUTFLO(2), OUTFLO(3)
636 820 FORMAT (/5X,'Gene flow: # in : WW = ',I4,' WM = ',I4,
637 $ ' MM = ',I4,/16X,'% out: WW = ',I3,' WM = ',I3,
638 $ ' MM = ',I3)
639 WRITE (OUTSCR,115)
640 CALL YESNO (ANS)
641 IF (ANS .EQ. 'Y') GO TO 1000
642 C
643 830 WRITE (OUTSCR,840)
644 840 FORMAT (/5X,'Do you wish to have gene flow (Y/N)? ', $)
645 CALL YESNO (ANS)
646 IF (ANS .EQ. 'Y') GO TO 900
647 DO 850 I=1,3
648 INFLO(I) = 0
649 OUTFLO(I) = 0
650 850 CONTINUE
651 GO TO 1000
652 C.....
653 C
654 C IMMIGRATION RATES
655 900 DO 950 I=1,3
656 910 WRITE (OUTSCR,915) GNOT(I)
657 915 FORMAT (/5X,'Enter number of ',A2,' individuals entering ',
658 $ 'per generation (0-4000): ', $)
659 READ (IN,930,ERR=940) INFLO(I)

```

```

660 930 FORMAT (I4)
661      IF ((INFLO(I) .GE. 0) .AND. (INFLO(I) .LE. 4000)) GO TO 950
662 940 WRITE (OUTSCR,945) GNOT(I)
663 945 FORMAT (' *****ERROR: Number of ',A2,' individuals entering ',
664      $ 'is incorrect.',/,13X,'It must be an integer between 0 and ',
665      $ '4000.')
```

---

```

666      WRITE (OUTSCR,265)
667      GO TO 910
668 950 CONTINUE
669 C.....
670 C
671 C      EMIGRATION RATES
672      DO 990 I=1,3
673 960 WRITE (OUTSCR,965) GNOT(I)
674 965 FORMAT (/,5X,'Enter % of ',A2,' individuals leaving per ',
675      $ 'generation (0-100): ', $)
676      READ (IN,970,ERR=980) OUTFLO(I)
677 970 FORMAT (I3)
678      IF ((OUTFLO(I) .GE. 0) .AND. (OUTFLO(I) .LE. 100)) GO TO 990
679 980 WRITE (OUTSCR, 985) GNOT(I)
680 985 FORMAT (' *****ERROR: % of ',A2,' individuals leaving is ',
681      $ 'incorrect.',/,13X,'It must be an integer between 0 and 100.')
```

---

```

682      WRITE (OUTSCR,265)
683      GO TO 960
684 990 CONTINUE
685 C.....
686 C
687 D      WRITE (OUTSCR,995) (INFLO(I), I=1,3)
688 D 995 FORMAT (' 6. FLOW: # IN PER GNOT. : WW = ',I5,3X,'WM = ',I5,
689 D      $ 3X,'MM = ',I5)
690 D      WRITE (OUTSCR,996) (OUTFLO(I), I=1,3)
691 D 996 FORMAT (11X,'% OUT PER GNOT.: WW = ',I3,3X,'WM = ',I3,3X,
692 D      $ 'MM = ',I3)
693 C-----
694 C
695 C      NUMBER OF GENERATIONS
696 C
697 1000 IF (NOGENS .EQ. 0) GO TO 1002
698      WRITE (OUTSCR,1001) NOGENS
699 1001 FORMAT (/,5X,'Model set to run for ',I3,' generations.')
```

---

```

700      WRITE (OUTSCR,115)
701      CALL YESNO (ANS)
702      IF (ANS .EQ. 'Y') GO TO 1014
703 C
704 1002 WRITE (OUTSCR,1003)
705 1003 FORMAT (/,5X,'Enter number of generations you wish to run ',
706      $ 'with these data (1-124): ', $)
707      READ (IN,1004,ERR=1006) NOGENS
708 1004 FORMAT (I3)
709      IF ((NOGENS .GE. 1) .AND. (NOGENS .LE. 124)) GO TO 1010
710 C
711 1006 WRITE (OUTSCR,1008)
712 1008 FORMAT (' *****ERROR: No. of generations is incorrect.',/,13X,
713      $ 'It must be an integer between 1 and 124.')
```

---

```

714      WRITE (OUTSCR,265)
715      GO TO 1002
716 C
717 1010 TGENS = IGEN + NOGENS
718      IF (TGENS .LE. 125) GO TO 1012
719      WRITE (OUTSCR,1011)
```

```
720 1011 FORMAT (' +++++WARNING: Total no. of generations requested ',
721 $ 'exceeds 125.',/,15X,'Run will stop at generation 125.')
722 NOGENS = 125 - IGEN
723 C-----
724 C
725 C INTERVAL BETWEEN SUMMARIES
726 C
727 1012 IF (SUMRY .NE. 0) GO TO 1014
728 WRITE (OUTSCR,1013)
729 1013 FORMAT (/,5X,'Summaries will be printed for first and ',
730 $ 'last generations only.')
731 GO TO 1016
732 C
733 1014 WRITE (OUTSCR,1015) SUMRY
734 1015 FORMAT (/,5X,'Summaries will be printed after every ',I2,
735 $ ' generation(s).')
736 C
737 1016 WRITE(OUTSCR,115)
738 CALL YESNO (ANS)
739 IF (ANS .EQ. 'Y') GO TO 1030
740 C
741 WRITE (OUTSCR,1017)
742 1017 FORMAT (/,5X,'How many generations do you wish between ',
743 $ 'summaries?',/,13X,'(Enter 0 for no summaries, or 1-99): ',$,)
744 READ (IN,1018,ERR=1020) SUMRY
745 1018 FORMAT (I2)
746 IF ((SUMRY .GE. 0) .AND. (SUMRY .LE. 99)) GO TO 1030
747 1020 WRITE (OUTSCR,1022)
748 1022 FORMAT (' *****ERROR: Summary request is incorrect.',/,13X,
749 $ 'It must be an integer between 0 and 99.')
750 WRITE (OUTSCR,265)
751 GO TO 1016
752 C-----
753 C
754 C WHAT TO DO AFTER GENERATIONS ARE FINISHED
755 C
756 1030 IF ((NEXT .NE. 'CH') .AND. (NEXT .NE. 'GR')) GO TO 1040
757 IF (NEXT .EQ. 'GR') GO TO 1032
758 WRITE (OUTSCR,1031)
759 1031 FORMAT (/,5X,'The model is set to change parameters.')
760 GO TO 1034
761 C
762 1032 WRITE (OUTSCR,1033)
763 1033 FORMAT (/,5X,'The model is set to print graphs.')
764 C
765 1034 WRITE (OUTSCR,115)
766 CALL YESNO (ANS)
767 IF (ANS .EQ. 'Y') GO TO 1060
768 C
769 1040 WRITE (OUTSCR,1042)
770 1042 FORMAT (/,5X,'What do you wish to do after the generations ',
771 $ 'are finished?',/,13X,'Enter "CH" to change parameters or ',
772 $ '"GR" to graph results: ',$,)
773 READ (IN,1044,ERR=1046) NEXT
774 1044 FORMAT (A2)
775 IF ((NEXT .EQ. 'CH') .OR. (NEXT .EQ. 'GR')) GO TO 1060
776 1046 WRITE (OUTSCR,1048)
777 1048 FORMAT (' *****ERROR: Request is incorrect.',/,13X,
778 $ 'It must be either "CH" or "GR".')
779 WRITE (OUTSCR,265)
```

```
780      GO TO 1040
781 1060 CONTINUE
782 C
783 D      WRITE (OUTSCR,1070) NOGENS, SUMRY, NEXT
784 D1070      FORMAT (5X,I4,' GENERATIONS WILL BE RUN, ',/,5X,
785 D      $      'SUMMARIES WILL BE PRINTED EVERY ',I2,' GENERATIONS',/,5X,
786 D      $      'THEN: ',A2)
787 D      PAUSE 'HIT <RET> TO CONTINUE.'
788 C-----
789 C
790 C GRAPH REQUESTS
791 C
792      IF (NEXT .NE. 'GR') GO TO 1400
793 C
794 C PRINT MENU
795 1200 IF (GRF(1) .EQ. ' ') GO TO 1210
796      WRITE (OUTSCR,1201)
797 1201      FORMAT (/,5X,'Graphs requested: ',,$)
798      WRITE (OUTSCR,1202) (GRF(I), I=1,10)
799 1202      FORMAT (10X,10(A2,2X))
800      WRITE (OUTSCR,115)
801      CALL YESNO (ANS)
802      IF (ANS .EQ. 'Y') GO TO 1400
803 C
804 1210 WRITE (OUTSCR,1212)
805 1212      FORMAT (/,5X,'You may request up to 10 of the following ',
806 $      'graphs:',/,10X,'AF (allele frequency)          GF (genotype ',
807 $      'frequency)')
808      WRITE(OUTSCR,1215)
809 1215      FORMAT(10X,'P0 (population size)          WW (number of ',
810 $      'WW'S)',/,10X,'WM (number of WM'S)          MM (number of ',
811 $      'MM'S)')
812      WRITE (OUTSCR,1220)
813 1220      FORMAT (10X,'AW (allele fitness)          GW (genotype ',
814 $      'fitness)',/,10X,'AS (allele selection coef.) GS (genotype ',
815 $      'selection coef.)')
816 C
817 C ENTER GRAPH REQUESTS
818      WRITE (OUTSCR,1230)
819 1230      FORMAT (/,5X,'Now enter the codes for the graphs you wish ',
820 $      'printed',/,13X,'(Hit <RET> twice after your last request): ')
821      DO 1280 I=1,10
822 1235      WRITE (OUTSCR,1240) I
823 1240      FORMAT (/,5X,'What is the code for graph number ',I2,'? ',,$)
824      READ (IN,1250,ERR=1260) GRF(I)
825 1250      FORMAT (A4)
826 C
827 C      IF CURRENT REQUEST IS NOT BLANK, CHECK FOR ERRORS, IF IT IS
828 C      BLANK, ENSURE THAT REMAINING GRAPH REQUEST SLOTS ARE ALSO
829 C      BLANK.
830      IF (GRF(I) .NE. GRFTST(1)) GO TO 1254
831      II = I
832      DO 1252 J=II,10
833          GRF(I) = GRFTST(1)
834 1252      CONTINUE
835      GO TO 1290
836 C
837 1254      DO 1255 J=2,11
838          IF (GRF(I) .EQ. GRFTST(J)) GO TO 1280
839 1255      CONTINUE
```

```

840 C
841 1260 WRITE (OUTSCR,1270) I
842 1270 FORMAT (' *****ERROR: Graph request # ',I2,' is ',
843 $ 'incorrect.',/13X,'It must be one of the following:',/13X,
844 $ '"AF", "GF", "PO", "WW", "WM", "MM",',/13X,
845 $ '"AW", "GW", "AS" OR "GS".')
846 WRITE (OUTSCR, 265)
847 GRF(I) = ' '
848 GO TO 1235
849 C
850 1280 CONTINUE
851 1290 CONTINUE
852 C
853 D WRITE (OUTSCR,1300) (GRF(I), I=1,10)
854 D1300 FORMAT (5X,'THE FOLLOWING GRAPHS WILL BE PRINTED:',10(2X,A2))
855 D PAUSE 'TYPE <RET> TO CONTINUE.'
856 C=====
857 C
858 C PRINT SUMMARY TO SCREEN, ASK IF OK. IF YES, CONTINUE,
859 C IF NO, REDO INPUT
860 C
861 1400 PRINT2 = OUTSCR
862 CALL INPSUM
863 C.....
864 C
865 1520 WRITE (OUTSCR,1530)
866 1530 FORMAT ('0 Are these data o.k. (Y/N)?',/5X,
867 $ '(If answer is "N", input will restart)',/5X,
868 $ '(If answer is "Y", and nothing happens, be sure printer is',
869 $ ' on line.): ',)$
870 CALL YESNO (ANS)
871 IF (ANS .EQ. 'Y') GO TO 1600
872 FIX = 'Y'
873 GO TO 10
874 1600 FIX = 'N'
875 C-----
876 C
877 C PRINT SUMMARY TO LINE PRINTER
878 PRINT2 = OUTLP
879 CALL INPSUM
880 C-----
881 C
882 9999 RETURN
883 END
884 C*****
885 C*****
886 C
887 SUBROUTINE YESNO (ANS)
888 C
889 C FUNCTION: READS AND CHECKS YES/NO INPUT, RETURNS EITHER "Y" OR "N" IN
890 C VARIABLE "ANS".
891 C
892 C NOTES:
893 C 1) THIS ROUTINE RELIES HEAVILY ON THE "ERR=" DEC FORTRAN
894 C EXTENSION TO AVOID AWKWARD SYSTEM ABORTS. THIS WILL NEED
895 C TO BE CHANGED TO IMPROVE TRANSPORTABILITY.
896 C
897 C CALLED FROM: INPUT, MAIN
898 C CALLS: NOTHING
899 C=====
    
```

```

900      INCLUDE 'COMMON.f'
901 C=====
902 C      Type *, 'In YesNo'
903 C
904      100 READ (IN,110,ERR=120) ANS
905      110   FORMAT (A1)
906      IF ((ANS .EQ. 'Y') .OR. (ANS .EQ. 'N')) GO TO 140
907      120   WRITE (OUTSCR,130)
908      130   FORMAT (5X, ' *****ERROR: Response is incorrect.',/,13X,
909      $         'It must be either a "Y" or an "N".',/,13X,
910      $         'Please try again: ', $)
911      GO TO 100
912 C
913      140 RETURN
914      END
915 C*****
916 C*****
917 C
918      SUBROUTINE INPSUM
919 C
920 C      FUNCTION: PRINTS SUMMARY OF DATA INPUT BY USER.
921 C
922 C      NOTES:
923 C          1) OUTPUT GOES TO SCREEN IF "PRINT2" = "OUTSCR" AND TO LINE
924 C             PRINTER IF "PRINT2" = "OUTLP"
925 C
926 C      CALLED FROM:  INPUT
927 C      CALLS:  NOTHING
928 C=====
929      INCLUDE 'COMMON.f'
930 C-----
931 C      LOCAL DECLARATIONS
932 C          INTEGER ATOT
933 C=====
934 C      Type *, 'In Inpsum'
935 C
936      IF (IGEN .GT. 1) GO TO 5
937      WRITE (PRINT2,2)
938      2   FORMAT (1H1)
939      5 IF (PRINT2 .EQ. OUTLP) GO TO 15
940      WRITE (PRINT2,10)
941      10  FORMAT('0 The following is a summary of the data you ',
942      $         'have just input:')
943      GO TO 18
944 C
945      15 IF (IGEN .GT. 1) GO TO 45
946      WRITE (PRINT2,16)
947      16  FORMAT(80(1H-),/, ' The following is a summary of ',
948      $         'the data you have input: ')
949 C
950      18 WRITE (PRINT2,20)TITLE
951      20   FORMAT (5X, 'Title: ',/,X,40A2)
952 C
953      WRITE (PRINT2,30) IY
954      30   FORMAT (5X, 'Random number seed:      ',I5)
955 C
956      ATOT = RSLT(1,1) + RSLT(2,1) + RSLT(3,1)
957      WRITE (PRINT2,40) ((RSLT(I,1), I=1,3), ATOT)
958      40   FORMAT (5X, 'Initial population:      WW = ',I4,3X, 'WM = ',I4,
959      $         3X, 'MM = ',I4,3X, 'TOTAL = ',I4)

```



```

960 C
961 45 WRITE (PRINT2,46) IGEN
962 46 FORMAT('0After generation ',I3,' parameters are as follows:')
963 C
964 WRITE (PRINT2,50) MXPOP1, MXPOP2
965 50 FORMAT (5X,'Carrying capacity: ',I4,/,4X,
966 $ ' Post-crash pop. size: ', I4)
967 C
968 WRITE (PRINT2,60) (SURVR8(I), I=1,3)
969 60 FORMAT (5X,'Survival rates: WW = ',F5.1,'% WM = ',
970 $ F5.1,'% MM = ',F5.1,'%')
971 C
972 WRITE (PRINT2,70) (REPR8(I), I=1,3)
973 70 FORMAT (5X,'Reproductive rates: WW = ',
974 $ F4.1,2X,'WM = ',F4.1,2X,'MM = ',F4.1)
975 C
976 WRITE (PRINT2,80) INFLO(1), INFLO(2), INFLO(3), OUTFLO(1),
977 $ OUTFLO(2), OUTFLO(3)
978 80 FORMAT (5X,'Flow: # in per gen.: WW = ',I5,3X,'WM = ',I5,3X,
979 $ 'MM = ',I5,/,11X,'% out per gen.: WW = ',I4,3X,'WM = ',
980 $ I4,3X,'MM = ',I4)
981 C
982 WRITE (PRINT2,90)
983 90 FORMAT (/,5X,'Summaries: ', $)
984 IF (SUMRY .GT. 0) GO TO 110
985 WRITE (PRINT2,100)
986 100 FORMAT ('+ First and last generations only')
987 GO TO 130
988 C
989 110 IF (SUMRY .GT. 1) GO TO 120
990 WRITE (PRINT2,112)
991 112 FORMAT ('+ Every generation')
992 GO TO 130
993 C
994 120 WRITE (PRINT2,122) SUMRY
995 122 FORMAT ('+ Every ',I2,' generations')
996 C
997 130 WRITE (PRINT2,140) NOGENS
998 140 FORMAT (5X,'Generations to be run: 'I3,/,5X,
999 $ 'Then: ', $)
1000 C
1001 IF (NEXT .NE. 'CH') GO TO 160
1002 WRITE (PRINT2,150)
1003 150 FORMAT ('+ Parameters will be changed ',/,80(1H-))
1004 GO TO 180
1005 C
1006 160 WRITE (PRINT2,170) (GRF(I), I=1,10)
1007 170 FORMAT ('+ The following graphs will be printed:',/,28X,
1008 $ 10(2X,A2),/,80(1H-))
1009 180 CONTINUE
1010 RETURN
1011 END
1012 C
1013 C*****
1014
1015 C
1016 C*****
1017 C file: MA.RE vrsn: 29 rvsd: 29 May 81
1018 C subroutines reprod, m8few, m8many
1019 C*****

```

```
1020 C
1021 C      SUBROUTINE REPROD
1022 C
1023 C      FUNCTION: CONTROLS MATING AND REPRODUCTION OF POPULATION, RETURNS
1024 C      NUMBER OF JUVENILES OF EACH GENOTYPE IN "JUV" ARRAY.
1025 C
1026 C      SYNOPSIS:
1027 C      1) REPROD ASSIGNS THE ADULTS OF THE PREVIOUS GENERATION TO THE
1028 C      "NOADLT" ARRAY.
1029 C      2) AFTER INITIALIZING VARIABLES, "REPROD" CALLS 1 OR 2 OTHER
1030 C      ROUTINES ("M8FEW" OR "M8MANY") WHICH RANDOMLY MATE INDIVIDUALS
1031 C      FROM "NOADLT" AND RETURN THE NUMBER OF EACH OF THE 6 POSSIBLE
1032 C      COMBINATIONS OF MATES IN THE "MATES" ARRAY.
1033 C      2) REPROD THEN CALCULATES THE NUMBER OF YOUNG FROM EACH TYPE OF
1034 C      MATING FROM THE REPRODUCTIVE RATES ("REPR8") OF THE GENOTYPES
1035 C      AND PLACES THEM IN THE "NOYONG" ARRAY).
1036 C      3) FINALLY, THE NUMBER OF YOUNG OF EACH GENOTYPE ARE DETERMINED
1037 C      FROM "NOYONG" ACCORDING TO MENDELIAN RATIOS AND PLACED INTO
1038 C      THE "JUV" ARRAY WHICH IS RETURNED TO MAIN.
1039 C
1040 C      NOTES:
1041 C      1) TWO METHODS ARE USED TO CALCULATE THE NUMBER OF EACH
1042 C      COMBINATION OF MATES BECAUSE NEITHER WORKS WELL OVER THE FULL
1043 C      RANGE OF POPULATION SIZES:
1044 C      A) SUBROUTINE "M8FEW" USES THE PROPORTION OF A GENOTYPE AMONG
1045 C      THE ADULTS TO DETERMINE ITS PROBABILITY OF BEING PICKED AS
1046 C      A PARENT. PAIRS OF PARENTS ARE RANDOMLY SELECTED WITHOUT
1047 C      REPLACEMENT FROM THE "NOADLT" ARRAY AND PLACED INTO THE
1048 C      APPROPRIATE CELL OF THE "MATES" ARRAY UNTIL "NOADLT" IS
1049 C      EMPTY OR N = "CUTOFF" INDIVIDUALS HAVE BEEN PAIRED.
1050 C      B) SUBROUTINE "M8MANY" USES "GNTFR", THE PROPORTION OF EACH
1051 C      GENOTYPE REMAINING IN "NOADLT", TO DIRECTLY CALCULATE
1052 C      THE PROBABLE NUMBER OF EACH MATING COMBINATION.
1053 C      2) METHOD A) IS A MONTE CARLO ALGORITHM TO SIMULATE STOCHASTIC
1054 C      COMBINATIONS IN SMALL POPULATIONS. HOWEVER, IT RESULTS IN
1055 C      PROHIBITIVELY LONG RUNNING TIMES WHEN USED WITH LARGE
1056 C      POPULATIONS. THE RUNNING TIME OF METHOD B) IS ESSENTIALLY
1057 C      CONSTANT REGARDLESS OF POPULATION SIZE, BUT CANNOT PROPERLY
1058 C      SIMULATE MATING IN SMALL POPULATIONS.
1059 C      3) IN SMALL POPULATIONS ONLY "M8FEW" IS CALLED. FOR POPULATIONS
1060 C      LARGER THAN "CUTOFF" INDIVIDUALS, "M8FEW" IS CALLED TO MATE
1061 C      CUTOFF INDIVIDUALS AND "M8MANY" IS CALLED TO MATE THE REST.
1062 C      THE RUN TIME OF EVOLVE COULD BE REDUCED SOMEWHAT BY REDUCING
1063 C      THE SIZE OF "CUTOFF" IN BLOCK DATA (CURRENTLY SET = 200).
1064 C      SUBROUTINE "M8FEW" COULD BE ELIMINATED IF ONLY DETERMINISTIC
1065 C      MATING IS DESIRED; SUBROUTINE M8MANY COULD BE ELIMINATED IF
1066 C      POPULATION LIMITS WERE REDUCED, RUN TIME WERE NOT IMPORTANT
1067 C      &/OR MUCH DRIFT IS DESIRED.
1068 C      4) THE "MATES" ARRAY HOLDS THE NUMBER OF INDIVIDUALS INVOLVED IN
1069 C      EACH OF THE 6 COMBINATIONS OF PARENTS:
1070 C      MATES(1) = # OF WW-WW MATINGS, MATES(2) = # OF WW-WM MATINGS,
1071 C      MATES(3) = # OF WW-WW MATINGS, MATES(6) = # OF WW-WM MATINGS,
1072 C      MATES(5) = # OF WW-WW MATINGS, MATES(6) = # OF WW-WM MATINGS,
1073 C      5) THE "NOYONG" ARRAY HOLDS THE NUMBER OF YOUNG FROM EACH OF THE
1074 C      6 COMBINATIONS OF PARENTS. ARRAY ELEMENTS CORRESPOND TO THOSE
1075 C      IN THE MATES ARRAY.
1076 C      6) ALTHOUGH THE DATA ARE INTEGER, VARIABLES "JUV" AND "NOYONG"
1077 C      ARE DECLARED TO BE REAL TO AVOID POSSIBLE INTEGER OVERFLOW IN
1078 C      16-BIT MACHINES. THE VARIABLES POTENTIALLY COULD REACH VALUES
1079 C      OF (REPR8 * MXPOP1) = (10.0 * 5000) = 50,000.
```

```

1080 C
1081 C   CALLED FROM: MAIN
1082 C   CALLS: M8FEW  - RANDOMLY MATES N = "CUTOFF" INDIVIDUALS, RETURNS
1083 C                   "MATES" ARRAY
1084 C       M8MANY  - RANDOMLY MATES REMAINDER OF POPULATION, ADDS THEM
1085 C                   TO THE "MATES" ARRAY.
1086 C       GAUSS   - RETURNS "V", A PSEUDORANDOM NUMBER WITH GIVEN MEAN
1087 C                   AND STANDARD DEVIATION TO DETERMINE REPRODUCTIVE
1088 C                   SUCCESS OF EACH COMBINATION OF MATES.
1089 C=====
1090 C       INCLUDE 'COMMON.f'
1091 C-----
1092 C   LOCAL DECLARATIONS
1093 C       INTEGER  PGEN
1094 C       REAL     X
1095 C=====
1096 C       Type *, 'In Reprod'
1097 C
1098 C   INITIALIZE
1099 C       PGEN = IGEN - 1
1100 C       DO 10 I=1,6
1101 C           MATES(I) = 0
1102 C           NOYONG(I) = 0.
1103 C   10 CONTINUE
1104 C       TOADLT = RSLT(1,PGEN) + RSLT(2,PGEN) + RSLT(3,PGEN)
1105 C       DO 20 I=1,3
1106 C           JUV(I) = 0.
1107 C           NOADLT(I) = RSLT(I,PGEN)
1108 C   20 CONTINUE
1109 C
1110 C       TEST FOR ZERO ADULTS. RETURN IF TRUE, CONTINUE IF FALSE
1111 C       IF (TOADLT .LE. 0) GO TO 999
1112 C-----
1113 C
1114 C       DETERMINE NUMBER OF EACH COMBINATION OF PARENTAL GENOTYPES
1115 C
1116 C       CALL M8FEW
1117 C       IF (TOADLT .LT. 1) GO TO 30
1118 C       CALL M8MANY
1119 C-----
1120 C
1121 C       DETERMINE NUMBER OF YOUNG FROM EACH TYPE OF MATING: SUBROUTINE
1122 C       GAUSS IS CALLED AND RETURNS A PSEUDORANDOM NUMBER ("V") WITH A MEAN
1123 C       ("AM") EQUAL TO THE AVERAGE NUMBER OF YOUNG PER PARENT (REPR8(3)).
1124 C       THE MEAN NUMBER OF YOUNG IS MULTIPLIED BY THE NUMBER OF PARENTS
1125 C       (MATES(6)) TO DETERMINE THE TOTAL NUMBER OF YOUNG (NOYONG(6))
1126 C       PRODUCED BY THOSE PARENTS.
1127 C
1128 C       RNOTE: A BETTER METHOD OF SETTING STANDARD DEVIATION OF GAUSS ("S")
1129 C       MAY BE NEEDED.  THE CURRENT METHOD (S = AM / SQRT(X)) IS ESSENTIALLY
1130 C       ANALOGOUS TO THE STANDARD ERROR OF THE MEAN AND ATTEMPTS TO REDUCE
1131 C       THE VARIATION OF GAUSS' OUTPUT AS THE NUMBER OF INDIVIDUALS
1132 C       INCREASES. THIS NEEDS THOUGHT.
1133 C
1134 C       THIS SECTION OF CODE (DOWN TO STATEMENT 90) COULD BE PUT INTO A
1135 C       SUBROUTINE ("NUMYNG") TO IMPROVE PROGRAM CLARITY.
1136 C.....
1137 C
1138 C       NUMBER OF YOUNG FROM WW-WW MATINGS
1139 C   30 IF (MATES(1) .LT. 1) GO TO 40
    
```

```

1140     AM = REPR8(1)
1141 D    WRITE (OUTSCR,170)
1142 D 170 FORMAT(100('*'))
1143 D 100 FORMAT ('0ENTER REPRODUCTION BLOCK WITH: MATES(1-6) = ',6(I6,1X),
1144 D    1'REPR8(1-3) = ',3(F5.2,1X))
1145 D    WRITE (OUTSCR,100) MATES(1), MATES(2), MATES(3), MATES(4),
1146 D    1MATES(5), MATES(6), REPR8(1), REPR8(2), REPR8(3)
1147     X = MATES(1)
1148     IF (X .LT. 1.) X = 1.
1149     S = AM / SQRT(X)
1150     CALL GAUSS
1151     NOYONG(1) = MATES(1) * V + 0.5
1152 D    WRITE (OUTSCR,110) NOYONG(1), V, S, X, AM
1153 D 110 FORMAT('0NOYONG(1) = ',F6.0,2X,'V = ',F5.2,2X,'S = ',F5.2,2X,
1154 D    1'X = ',F5.2,2X,'AM = ',F5.2,/,5X,'NOYONG(1) = MATES(1) * V + 0.5',
1155 D    25X,'S = AM / SQRT(X)',5X,'X = MATES(1) OR 1.0',5X,
1156 D    3' AM = REPR8(1)')
1157 C.....
1158 C
1159 C    NUMBER OF YOUNG FROM WW-WM MATINGS
1160 40 IF (MATES(2) .LT. 1) GO TO 50
1161     AM = (REPR8(1) + REPR8(2)) / 2
1162     X = MATES(2)
1163     IF (X .LT. 1.) X = 1.
1164     S = AM / SQRT(X)
1165     CALL GAUSS
1166     NOYONG(2) = MATES(2) * V + 0.5
1167 D    WRITE (OUTSCR,120) NOYONG(2), V, S, X, AM
1168 D 120 FORMAT('0NOYONG(2) = ',F6.0,2X,'V = ',F5.2,2X,'S = ',F5.2,2X,
1169 D    1'X = ',F5.2,2X,'AM = ',F5.2,/,5X,'NOYONG(2) = MATES(2) * V + 0.5',
1170 D    25X,'S = AM / SQRT(X)',5X,'X = MATES(2) OR 1.0',5X,
1171 D    3' AM = (REPR8(1) + REPR8(2)) / 2')
1172 C.....
1173 C
1174 C    NUMBER OF YOUNG FROM WW-MM MATINGS
1175 50 IF (MATES(3) .LT. 1) GO TO 60
1176     AM = (REPR8(1) + REPR8(3)) / 2
1177     X = MATES(3)
1178     IF (X .LT. 1.) X = 1.
1179     S = AM / SQRT(X)
1180     CALL GAUSS
1181     NOYONG(3) = MATES(3) * V + 0.5
1182 D    WRITE (OUTSCR,130) NOYONG(3), V, S, X, AM
1183 D 130 FORMAT('0NOYONG(3) = ',F6.0,2X,'V = ',F5.2,2X,'S = ',F5.2,2X,
1184 D    1'X = ',F5.2,2X,'AM = ',F5.2,/,5X,'NOYONG(3) = MATES(3) * V + 0.5',
1185 D    25X,'S = AM / SQRT(X)',5X,'X = MATES(3) OR 1.0',5X,
1186 D    3' AM = (REPR8(1) + REPR8(3)) / 2')
1187 C.....
1188 C
1189 C    NUMBER OF YOUNG FROM WM-WM MATINGS
1190 60 IF (MATES(4) .LT. 1) GO TO 70
1191     AM = REPR8(2)
1192     X = MATES(4)
1193     IF (X .LT. 1.) X = 1.
1194     S = AM / SQRT(X)
1195     CALL GAUSS
1196     NOYONG(4) = MATES(4) * V + 0.5
1197 D    WRITE (OUTSCR,140) NOYONG(4), V, S, X, AM
1198 D 140 FORMAT('0NOYONG(4) = ',F6.0,2X,'V = ',F5.2,2X,'S = ',F5.2,2X,
1199 D    1'X = ',F5.2,2X,'AM = ',F5.2,/,5X,'NOYONG(4) = MATES(4) * V + 0.5',

```

```

1200 D    25X,'S = AM / SQRT(X)',5X,'X = MATES(4) OR 1.0',5X,
1201 D    3' AM = REPR8(2)
1202 C.....
1203 C
1204 C    NUMBER OF YOUNG FROM WM-MM MATINGS
1205 70 IF (MATES(5) .LT. 1) GO TO 80
1206 AM = (REPR8(2) + REPR8(3)) / 2
1207 X = MATES(5)
1208 IF (X .LT. 1.) X = 1.
1209 S = AM / SQRT(X)
1210 CALL GAUSS
1211 NOYONG(5) = MATES(5) * V + 0.5
1212 D    WRITE (OUTSCR,150) NOYONG(5), V, S, X, AM
1213 D 150 FORMAT('0NOYONG(5) = ',F6.0,2X,'V = ',F5.2,2X,'S = ',F5.2,2X,
1214 D    1'X = ',F5.2,2X,'AM = ',F5.2,/,5X,'NOYONG(5) = MATES(5) * V + 0.5',
1215 D    25X,'S = AM / SQRT(X)',5X,'X = MATES(5) OR 1.0',5X,
1216 D    3' AM = (REPR8(2) + REPR8(3)) / 2')
1217 C.....
1218 C
1219 C    NUMBER OF YOUNG FROM MM-MM MATINGS
1220 80 IF (MATES(6) .LT. 1) GO TO 90
1221 AM = REPR8(3)
1222 X = MATES(6)
1223 IF (X .LT. 1.) X = 1.
1224 S = AM / SQRT(X)
1225 CALL GAUSS
1226 NOYONG(6) = MATES(6) * V + 0.5
1227 D    WRITE (OUTSCR,160) NOYONG(6), V, S, X, AM
1228 D 160 FORMAT('0NOYONG(6) = ',F6.0,2X,'V = ',F5.2,2X,'S = ',F5.2,2X,
1229 D    1'X = ',F5.2,2X,'AM = ',F5.2,/,5X,'NOYONG(6) = MATES(6) * V + 0.5',
1230 D    25X,'S = AM / SQRT(X)',5X,'X = MATES(6) OR 1.0',5X,
1231 D    3' AM = REPR8(3)')
1232 D    WRITE (OUTSCR,170)
1233 C-----
1234 C
1235 C    CALCULATE NUMBER OF YOUNG OF EACH GENOTYPE FROM THE MENDELIAN
1236 C    RATIOS ASSOCIATED WITH PARTICULAR COMBINATIONS OF PARENTAL
1237 C    GENOTYPES
1238 C
1239 C    NUMBER OF WW YOUNG
1240 90 JUV(1) = NOYONG(1) + 0.5*FLOAT(NOYONG(2)) + 0.25*FLOAT(NOYONG(4))
1241 1+ 0.5
1242 C
1243 C    NUMBER OF WM YOUNG
1244 JUV(2) = 0.5*FLOAT(NOYONG(2)) + NOYONG(3) + 0.5*FLOAT(NOYONG(4))
1245 1+ 0.5*FLOAT(NOYONG(5)) + 0.5
1246 C
1247 C    NUMBER OF MM YOUNG
1248 JUV(3) = 0.25*FLOAT(NOYONG(4)) + 0.5*FLOAT(NOYONG(5)) + NOYONG(6)
1249 1+ 0.5
1250 C
1251 C    type *, 'leaving repro'
1252 C    pause
1253 999 RETURN
1254 END
1255 C
1256 C*****
1257 C*****
1258 C
1259 SUBROUTINE M8FEW
    
```

```

1260 C
1261 C FUNCTION: RANDOMLY PICKS UP TO N = "CUTOFF" INDIVIDUALS AS PAIRS OF
1262 C MATES BY A MONTE CARLO ALGORITHM AND ADDS THEM TO THE MATES ARRAY.
1263 C
1264 C NOTES:
1265 C 1) SEE NOTES IN REPROD FOR A DISCUSSION OF GENERAL METHOD AND
1266 C RATIONALE.
1267 C
1268 C CALLED FROM: REPROD
1269 C CALLS: RAN - FORTRAN LIBRARY FUNCTION THAT RETURNS "YFL", A
1270 C PSEUDORANDOM NUMBER UNIFORMLY DISTRIBUTED 0.0->1.0
1271 C USED TO DETERMINE WHETHER A GENOTYPE IS CHOSEN FOR A
1272 C PAIR OF MATES.
1273 C=====
1274 C INCLUDE 'COMMON.f'
1275 C-----
1276 C LOCAL DECLARATIONS
1277 C INTEGER COUNT, CUTOFF, IPAR(2), MATE, MAXPRS
1278 C REAL PROB, YFL
1279 C COUNT = 0
1280 C cutoff = 200
1281 C=====
1282 C Type *, 'InM8few'
1283 C
1284 C INITIALIZE
1285 C MAXPRS =(CUTOFF / 2) + 0.5
1286 C DO 10 I=1,2
1287 C IPAR(I) = 0
1288 C 10 CONTINUE
1289 C
1290 C-----
1291 C PICK CUTOFF/2 PAIRS OF MATES
1292 C-----
1293 C
1294 C Type *,MAXPRS
1295 C Pause 'Before loop'
1296 C DO 500 I=1,MAXPRS
1297 D 900 FORMAT ('0ENTERED M8FEW PICK-A-PAIR LOOP WITH:')
1298 D WRITE (OUTSCR,900)
1299 D WRITE (OUTSCR,910) NOADLT(1), NOADLT(2), NOADLT(3), TOADLT,
1300 D 1 CUTOFF, MAXPRS, COUNT, IPAR(1), IPAR(2), MATES(1), MATES(2),
1301 D 2 MATES(3), MATES(4), MATES(5), MATES(6)
1302 D 910 FORMAT (' NOADLT(1,2,3) = ',3(I4,1X),2X,'TOADLT = ',I5,5X,
1303 D 1 'CUTOFF = ',I5,2X,'MAXPRS = ',I4,5X,'COUNT = ',I4,/,
1304 D 2 ' IPAR(1,2) = ',2(I1,1X),2X,'MATES(1---6) = ',6(I5,1X))
1305 C
1306 C IF (MAXPRS .LT. 1) GO TO 999
1307 20 IF (TOADLT .LE. 0) GO TO 999
1308 C
1309 C.....
1310 C PICK A PAIR
1311 C.....
1312 C
1313 C DO 110 J=1,2
1314 D 930 FORMAT ('0IN PICK-A-PAIR LOOP:')
1315 D WRITE (OUTSCR,930)
1316 D IF (TOADLT .EQ. 0) GO TO 110
1317 D IF ((TOADLT .EQ. 1) .AND. (J .EQ. 1)) GO TO 60
1318 C
1319 C YFL = RAN(IY)

```

```

1320      PROB = FLOAT(NOADLT(1)) / FLOAT(TOADLT)
1321 D      WRITE (OUTSCR,940) YFL, PROB
1322 D 940    FORMAT (' YFL = ',F11.9,5X,'PROB = ',F11.9)
1323      IF (YFL .GT. PROB) GO TO 30
1324      IPAR(J) = 1
1325      NOADLT(1) = NOADLT(1) - 1
1326      GO TO 50
1327 C
1328      30    PROB = PROB + FLOAT(NOADLT(2)) / FLOAT(TOADLT)
1329 D      WRITE (OUTSCR,940) YFL, PROB
1330      IF (YFL .GT. PROB) GO TO 40
1331      IPAR(J) = 2
1332      NOADLT(2) = NOADLT(2) - 1
1333      GO TO 50
1334 C
1335      40    IPAR(J) = 3
1336 D      WRITE (OUTSCR,940) YFL, PROB
1337      NOADLT(3) = NOADLT(3) - 1
1338      50    TOADLT = TOADLT - 1
1339      GO TO 110
1340 C .....
1341 C
1342 C      MATE SOLITARY INDIVIDUAL WITH ITSELF. THIS SECTION
1343 C      OF CODE WAS ADDED TO IMPLEMENT "FACULTATIVE INBREEDING" AND
1344 C      PREVENT STEADY DECLINE OF POPULATION SIZE IN SMALL
1345 C      POPULATIONS WITH ODD NUMBERS OF INDIVIDUALS AND LOW ABSOLUTE
1346 C      FITNESSES. EARLIER VERSIONS OF THE PROGRAM USED "OBLIGATE
1347 C      OUTBREEDING" AND STUDENTS FOUND THE RATHER STEADY DECLINE IN
1348 C      POPULATION SIZE IN SOME RUNS TO BE CONFUSING.
1349 C
1350      60    IF (NOADLT(1) .NE. 1) GO TO 70
1351      MATES(1) = MATES(1) + 1
1352      NOADLT(1) = NOADLT(1) - 1
1353      GO TO 90
1354 C
1355      70    IF (NOADLT(2) .NE. 1) GO TO 80
1356      MATES(4) = MATES(4) + 1
1357      NOADLT(2) = NOADLT(2) - 1
1358      GO TO 90
1359 C
1360      80    IF (NOADLT(3) .NE. 1) GO TO 100
1361      MATES(6) = MATES(6) + 1
1362      NOADLT(3) = NOADLT(3) - 1
1363      90    TOADLT = TOADLT - 1
1364      GO TO 120
1365      100   STOP
1366 C .....
1367 C
1368      110   CONTINUE
1369      120   CONTINUE
1370 D 920    FORMAT ('EXIT PICK-A-PAIR LOOP WITH:')
1371 D      WRITE (OUTSCR,920)
1372 D      WRITE (OUTSCR,910) NOADLT(1), NOADLT(2), NOADLT(3), TOADLT,
1373 D      1    CUTOFF, MAXPRS, COUNT, IPAR(1), IPAR(2), MATES(1), MATES(2),
1374 D      2    MATES(3), MATES(4), MATES(5), MATES(6)
1375 C
1376 C-----
1377 C      ADD PAIR TO MATES ARRAY.
1378 C-----
1379 C      VALUES OF IPAR ARRAY ARE MULTIPLIED AND THE PRODUCT (MATE)
    
```

```

1380 C      IS USED TO DETERMINE WHICH ELEMENT OF THE MATES ARRAY TO
1381 C      INCREMENT.
1382 C
1383 C      IF ((IPAR(1) .EQ. 0) .OR. (IPAR(2) .EQ. 0)) GO TO 999
1384 C      MATE = IPAR(1) * IPAR(2)
1385 C      GO TO (300,310,320,330,210,340,210,210,350), MATE
1386 210 C      STOP
1387 C
1388 C      WW-WW (1*1=1)
1389 300 C      MATES(1) = MATES(1) + 2
1390 C      GO TO 360
1391 C
1392 C      WW-WM (1*2=2)
1393 310 C      MATES(2) = MATES(2) + 2
1394 C      GO TO 360
1395 C
1396 C      WW-MM (1*3=3)
1397 320 C      MATES(3) = MATES(3) + 2
1398 C      GO TO 360
1399 C
1400 C      WM-WM (2*2=4)
1401 330 C      MATES(4) = MATES(4) + 2
1402 C      GO TO 360
1403 C
1404 C      WM-MM (2*3=6)
1405 340 C      MATES(5) = MATES(5) + 2
1406 C      GO TO 360
1407 C
1408 C      MM-MM (3*3=9)
1409 350 C      MATES(6) = MATES(6) + 2
1410 C
1411 360 C      CONTINUE
1412 D      COUNT = COUNT + 1
1413 D      DO 400 K = 1,2
1414 D          IPAR(K) = 0
1415 400 D      CONTINUE
1416 D 950 D      FORMAT ('0FINISHED ADD-A-PAIR WITH MATES(1---6) = ',6(I4,1X))
1417 D      WRITE (OUTSCR,950) MATES(1), MATES(2), MATES(3), MATES(4),
1418 D 1    MATES(5), MATES(6)
1419 500 D      CONTINUE
1420 C-----
1421 C
1422 999 C      CONTINUE
1423 D      WRITE (OUTSCR,960)
1424 D 960 D      FORMAT ('0EXIT FROM M8FEW WITH:')
1425 D      WRITE (OUTSCR,910) NOADLT(1), NOADLT(2), NOADLT(3), TOADLT,
1426 D 1    CUTOFF, MAXPRS, COUNT, IPAR(1), IPAR(2), MATES(1), MATES(2),
1427 D 2    MATES(3), MATES(4), MATES(5), MATES(6)
1428 D      RETURN
1429 D      END
1430 C
1431 C*****
1432 C*****
1433 C
1434 C      SUBROUTINE M8MANY
1435 C
1436 C      FUNCTION: CALCULATES THE NUMBER OF EACH TYPE OF MATES BY METHOD 2
1437 C      AND THEN ADDS THEM TO THE MATES ARRAY. SEE NOTES IN REPROD.
1438 C
1439 C      NOTES:
    
```



```

1440 C      1) THE PROBABILITY OF EACH TYPE OF MATING IS DETERMINED BY
1441 C      MULTIPLYING THE FREQUENCIES OF THE APPROPRIATE GENOTYPES
1442 C      "GNTFR" ARRAY, BY EACH OTHER.  THE PROBABILITY OF EACH
1443 C      COMBINATION OF MATES IS THEN MULTIPLIED BY THE NUMBER OF
1444 C      ADULTS LEFT IN "NOADLT", ROUNDED TO THE NEAREST
1445 C      INTEGER AND ADDED TO THE NUMBER OF MATES ESTABLISHED BY
1446 C      SUBROUTINE "M8FEW".
1447 C
1448 C      CALLED FROM: REPROD
1449 C      CALLS: NOTHING
1450 C=====
1451 C      INCLUDE 'COMMON.f'
1452 C-----
1453 C      LOCAL DECLARATIONS
1454 C      INTEGER  TMATES
1455 C      REAL     GNTFR(3)
1456 C=====
1457 C      Type *, 'In M8Many'
1458 C
1459 D 900 FORMAT ('0ENTERED M8MANY WITH:')
1460 D      WRITE (OUTSCR,900)
1461 D      WRITE (OUTSCR,910) NOADLT(1), NOADLT(2), NOADLT(3), TOADLT,
1462 D      1  MATES(1), MATES(2), MATES(3), MATES(4), MATES(5), MATES(6)
1463 D 910 FORMAT (' NOADLT(1-3) = ',3(I5,1X),' TOADLT = ',I5,
1464 D      1  ' MATES(1---6) = ',6(I5,1X))
1465 C
1466 C      CALCULATE FREQUENCY OF EACH GENOTYPE
1467 C      DO 10 I=1,3
1468 C          GNTFR(I) = FLOAT(NOADLT (I)) / TOADLT
1469 C      10 CONTINUE
1470 C
1471 C-----
1472 C      CALCULATE NUMBER OF EACH TYPE OF MATING.
1473 C-----
1474 C      MATES(1) = MATES(1) + (GNTFR(1)**2) * TOADLT + 0.5
1475 C      MATES(2) = MATES(2) + (2 * (GNTFR(1)*GNTFR(2))) * TOADLT + 0.5
1476 C      MATES(3) = MATES(3) + (2 * (GNTFR(1)*GNTFR(3))) * TOADLT + 0.5
1477 C      MATES(4) = MATES(4) + (GNTFR(2)**2) * TOADLT + 0.5
1478 C      MATES(5) = MATES(5) + (2 * (GNTFR(2)*GNTFR(3))) * TOADLT + 0.5
1479 C      MATES(6) = MATES(6) + (GNTFR(3)**2) * TOADLT + 0.5
1480 C
1481 D      TMATES = MATES(1) + MATES(2) + MATES(3) + MATES(4) + MATES(5) +
1482 D      $      MATES(6)
1483 D      LGEN = IGEN - 1
1484 D      ATOT = RSLT(1,LGEN) + RSLT(2,LGEN) + RSLT(3,LGEN)
1485 D      IF (TMATES .EQ. ATOT) GO TO 930
1486 D      WRITE (OUTSCR,920) TMATES, ATOT
1487 D 920 FORMAT ('0ERROR: TMATES = ',I5,1X,'ATOT = ',I5)
1488 D 930 WRITE (OUTSCR,940)
1489 D 940 FORMAT ('0EXIT FROM M8MANY WITH:')
1490 D      WRITE (OUTSCR,910) NOADLT(1), NOADLT(2), NOADLT(3), TOADLT,
1491 D      $      MATES(1), MATES(2), MATES(3), MATES(4), MATES(5), MATES(6)
1492 C-----
1493 C
1494 C      RETURN
1495 C      END
1496 C*****
1497
1498 C
1499 C*****

```

```
1500 C file: SR.SU   vrsn: 48   rvsd: 27 Aug 81
1501 C           subroutines srvivl, sumrys
1502 C*****
1503 C
1504 C       SUBROUTINE SRVIVL
1505 C
1506 C       FUNCTION: TAKES THE NUMBER OF JUVENILES IN THE CURRENT GENERATION
1507 C               AND RETURNS THE NUMBER OF ADULTS IN THE CURRENT GENERATION
1508 C               "RSLTS(1-3,IGEN)"
1509 C
1510 C       SYNOPSIS:
1511 C           1) THIS SUBROUTINE USES THE RATES OF EMIGRATION ("OUTFLOW") TO
1512 C              DETERMINE THE NUMBER OF JUVENILES OF EACH GENOTYPE EMIGRATING
1513 C              ("NOOUT") AND REMOVES THEM FROM THE JUVENILE POPULTION.
1514 C           2) IT THEN DETERMINES THE NUMBER OF JUVENILES DYING FROM THE
1515 C              RATES OF SURVIVAL ("SURVR8") AND REMOVES THEM FROM THE
1516 C              POPULATION TO DETERMINE THE NUMBER SURVIVING TO ADULTHOOD
1517 C              ("ADLT" ARRAY).
1518 C           3) THE NUMBER OF IMMIGRANTS ("INFLO") OF EACH GENOTYPE IS ADDED
1519 C              TO "ADLT".
1520 C           4) IF THE NUMBER OF ADULTS EXCEEDS THE CARRYING CAPACITY
1521 C              ("MXPOP1"), THE TOTAL IS REDUCED TO THE POST-CRASH SIZE
1522 C              ("MXPOP2").
1523 C
1524 C       NOTES:
1525 C           1) VAR. "NOOUT" ADDED & PLACED IN COMMON FOR USE IN DEBUG
1526 C              OUTPUT FROM SUBROUTINE SUMRYS. IT WILL NOT BE MEANINGFUL
1527 C              IF SUMRYS IS CALLED TO PRINT A GENERATION OTHER THAN THE
1528 C              CURRENT ONE. IT SHOULD PROBABLY BE REMOVED IN THE PRODUCTION
1529 C              VERSION.
1530 C           2) ALTHOUGH VARIABLES "ATOT" & "ADLT" ARE CONCEPTUAL INTEGERS,
1531 C              THEY ARE DECLARED REAL TO AVOID POSSIBLE INTEGER OVERFLOW ON
1532 C              16-BIT MACHINES. THEORETICALLY, "ADLT" COULD =
1533 C              JUV * SURVR8/100 + INFLO = 50,000 * 100/100 + 2000 = 52,000
1534 C              "ATOT" COULD = 3 * ADLT = 3 * 52,000 = 156,000
1535 C
1536 C       CALLED FROM: MAIN
1537 C       CALLS: NOTHING
1538 C=====
1539 C       INCLUDE 'COMMON.f'
1540 C-----
1541 C       LOCAL DECLARATIONS
1542 C           REAL      ADLT(3), ATOT, CRSH
1543 C=====
1544 C           Type *, 'In Survival'
1545 C
1546 C       INITIALIZE
1547 C
1548 C           DO 10 I=1,3
1549 C               ADLT(I) = 0
1550 C       10 CONTINUE
1551 C           ATOT = 0
1552 C-----
1553 C
1554 C       EMIGRATION OF JUVENILES
1555 C
1556 C           RNOTE: COULD CALL GAUSS HERE TO GET RANDOM VARIATION IN %
1557 C           EMIGRATING.
1558 C.....
1559 C
```

```

1560     IF ((OUTFLO(1) .EQ. 0) .AND. (OUTFLO(2) .EQ. 0) .AND.
1561 1    (OUTFLO(3) .EQ. 0)) GO TO 100
1562 C
1563     DO 40 I=1,3
1564     IF (OUTFLO(I) .GT. 0.) GO TO 20
1565     NOOUT(I) = 0
1566     GO TO 40
1567 C
1568 20    IF (JUV(I) .GT. 0.) GO TO 30
1569     NOOUT(I) = 0
1570     GO TO 40
1571 C
1572 30    NOOUT(I) = ((JUV(I))*(OUTFLO(I) / 100)) + 0.5
1573     JUV(I) = JUV(I) - NOOUT(I)
1574 40 CONTINUE
1575 C
1576 C-----
1577 C  CALCULATE TOTAL NUMBER OF ADULTS
1578 C-----
1579 C    MULTIPLY NUMBER OF YOUNG OF EACH GENOTYPE BY ITS SURVIVAL RATE,
1580 C    ADD IMMIGRANTS
1581 C
1582 C    RNOTE: COULD CALL GAUSS HERE TO GET RANDOM VARIATION IN #
1583 C    IMMIGRATING.
1584 C.....
1585 100 DO 110 I=1,3
1586     ADLT(I) = ((SURVR8(I) / 100) * JUV(I)) + .5 + INFLO(I)
1587     ATOT = ATOT + ADLT(I)
1588 110 CONTINUE
1589 C
1590 C-----
1591 C  TEST FOR ADULTS > CARRYING CAPACITY, CRASH IF NECESSARY
1592 C
1593 C    RNOTE: COULD CALL GAUSS HERE TO GET RANDOM VARIATION IN POST-
1594 C    CRASH POPULATION SIZE.
1595 C.....
1596 C
1597     IF (ATOT .LE. MXPOP1) GO TO 300
1598 D    WRITE (OUTSCR,35) ADLT(1), ADLT(2), ADLT(3), MXPOP2, ATOT
1599 D 35  FORMAT ('0ENTER CRASH BLOCK IN SRVIVL WITH: ',5X,'ADLT(1-3) = ',
1600 D 1    1X,3(I5,1X),5X,'MXPOP2 = ',1X,I5,5X,'ATOT = ',1X,F6.0)
1601 C
1602     CRSH = FLOAT(MXPOP2) / FLOAT(ATOT)
1603     DO 200 I=1,3
1604     ADLT(I) =ADLT(I) * CRSH + .5
1605 200 CONTINUE
1606 D    WRITE (OUTSCR,36) ADLT(1), ADLT(2), ADLT(3), MXPOP2, ATOT
1607 D 36  FORMAT ('0 EXIT CRASH BLOCK IN SRVIVL WITH: ',5X,'ADLT(1-3) = ',
1608 D $    1X,3(I5,1X),5X,'MXPOP2 = ',1X,I5,5X,'ATOT = ',1X,F6.0)
1609 C
1610 C-----
1611 C  PUT ADULTS INTO RESULTS ARRAY
1612 C-----
1613 C
1614 300 DO 310 I=1,3
1615     RSLT(I,IGEN) = ADLT(I)
1616 310 CONTINUE
1617 C
1618     RETURN
1619     END
    
```

```

1620 C
1621 C*****
1622 C*****
1623 C
1624     SUBROUTINE SUMRYS
1625 C
1626 C     FUNCTION: PRINTS A SUMMARY OF FIRST AND LAST GENERATIONS, AND
1627 C     OF OTHER GENERATIONS IF REQUESTED. OUTPUT IS DIRECTED TO CRT OR
1628 C     LINE PRINTER BY VAR. "PRINT2".
1629 C
1630 C     NOTES:
1631 C         1) THIS ROUTINE WAS MUCH USED DURING DEBUGGING AND THERE IS A
1632 C            LOT OF DEBUG CODE. THERE ARE TWO MAJOR SECTIONS OF "D" CODE;
1633 C            THE FIRST IS FOR GENERATIONS WHEN SELECTION & FITNESS
1634 C            COEFFICIENTS CAN BE CALCULATED, THE SECOND FOR GENERATIONS
1635 C            WHEN THE COEFFICIENTS CANNOT BE CALCULATED.
1636 C         2) DEBUG CODE HAS NOT BEEN USE FOR A LONG TIME & MAY NOT BE
1637 C            RELIABLE. IN THE PRODUCTION VERSION, DEBUG CODE SHOULD
1638 C            PROBABLY BE REMOVED, ALTHOUGH CONDUIT MAY WISH TO CLEAN IT
1639 C            UP FOR PURCHASERS' USE.
1640 C         3) COULD PROBABLY REMOVE JUVENILES AND GENE FLOW FROM PRODUCTION
1641 C            OUTPUT; MIGHT BE SIMPLER FOR STUDENTS.
1642 C         4) PUT IN PAGESIZE & #LINESPRINTED VARS. ETC. TO PAGE OUTPUT
1643 C
1644 C     CALLED FROM: MAIN
1645 C     CALLS: ALFIT - RETURNS ALLELE FITNESS COEFFICIENTS (DEBUG ONLY)
1646 C            ALFREQ - RETURNS ALLELE FREQUENCY OF M ALLELE
1647 C            ALSEL - RETURNS ALLELE SELECTION COEFFICIENTS (DEBUG ONLY)
1648 C            GNTFIT - RETURNS GENOTYPE FITNESS COEFFICIENTS (DEBUG ONLY)
1649 C            GNTSEL - RETURNS GENOTYPE SELECTION COEFFICIENTS (DEBUG ONLY)
1650 C=====
1651 C     INCLUDE 'COMMON.f'
1652 C-----
1653 C     LOCAL DECLARATIONS
1654 C     INTEGER ATOT, ATOTP, INTOT
1655 C     REAL    FRQM, FRQW, OUTTOT, TOJUV
1656 C=====
1657 C
1658 C     PRINT HEADER(S)
1659 C-----
1660 C         Type *, 'In Sumrys'
1661 C
1662 C         WRITE (PRINT2,10)
1663 C         10 FORMAT (/,17X,'# GENE FLOW # GNOT.',10X,'ALLELE',/,
1664 C            $ 1X,'GEN. GNOT. YOUNG # IN # OUT ADULTS FREQ. ALLELE ',
1665 C            $ ' FREQ. ')
1666 C
1667 C         WRITE (OUTLP,15)
1668 C         15 FORMAT (50X,'GNOT. GNOT. GNOT.',17X,'ALL. ALL. ALL.',/,
1669 C            $ 17X,'# GENE FLOW # GNOT. SURV. FITN. SELN.',10X,
1670 C            $ 'ALL. SURV. FITN. SELN.',/,1X,'GEN. GNOT. YOUNG # IN ',
1671 C            $ '# OUT ADULTS FREQ. RATES COEF. COEF. ALLELE FREQ. ',
1672 C            $ 'RATES COEF. COEF. ***** MEANS A COEF.',/,110X,
1673 C            $ 'CANNOT BE CALCULATED')
1674 C
1675 C-----
1676 C     INITIALIZE VARIABLES, TEST WHETHER COEFFICIENTS CAN BE CALCULATED
1677 C-----
1678 C     20 FRQM = ALFREQ(SUMGEN)
1679 C     FRQW = 1 - FRQM
    
```

```

1680      IF ((RSLT(1,SUMGEN) .EQ. 0) .AND. (RSLT(2,SUMGEN) .EQ. 0))
1681      $   FRQW = 0.
1682      TOJUV = JUV(1) + JUV(2) + JUV(3)
1683      ATOT = RSLT(1,SUMGEN) + RSLT(2,SUMGEN) + RSLT(3,SUMGEN)
1684      INTOT = INFLO(1) + INFLO(2) + INFLO(3)
1685      OUTTOT = NOOUT(1) + NOOUT(2) + NOOUT(3)
1686      DO 25 I=1,3
1687          IF (ATOT .GT. 0) GO TO 24
1688          OBSFR(I) = 0.
1689          GO TO 25
1690      24  OBSFR(I) = FLOAT(RSLT(I,SUMGEN)) / FLOAT(ATOT)
1691      25  CONTINUE
1692  C-----
1693  C
1694  C      PRINT SUMMARY
1695      WRITE(PRINT2,30) SUMGEN,JUV(1),INFLO(1),NOOUT(1),RSLT(1,SUMGEN),
1696      $   OBSFR(1),FRQW,JUV(2),INFLO(2),NOOUT(2),RSLT(2,SUMGEN),OBSFR(2),
1697      $   JUV(3),INFLO(3),NOOUT(3),RSLT(3,SUMGEN),OBSFR(3),FROM,
1698      $   TOJUV,INTOT,OUTTOT,ATOT
1699  C
1700      30  FORMAT(' ',I3,5X,'WW',4X,F6.0,2X,I4,2X,F6.0,3X,I4,3X,F5.3,4X,
1701      $   'W',5X,F5.3,/,9X,'WM',4X,F6.0,2X,I4,2X,F6.0,3X,I4,3X,F5.3,
1702      $   /,9X,'MM',4X,F6.0,2X,I4,2X,F6.0,3X,I4,3X,F5.3,4X,'M',5X,
1703      $   F5.3,/,7X,'TOTAL',3X,F6.0,1X,I5,2X,F6.0,3X,I4,/)
1704  D      GO TO 35
1705      GO TO 99
1706  C
1707  D-----
1708  D  CALCULATE ADDITIONAL INITIAL VARS FOR DEBUG OUTPUT;
1709  D      TEST FOR CALCULATABILITY OF COEFFICIENTS
1710  D-----
1711  D  35  IF (SUMGEN .EQ. 1) GO TO 60
1712  D      LGEN = SUMGEN - 1
1713  D      ATOTP = RSLT(1,LEGEN) + RSLT(2,LEGEN) + RSLT(3,LEGEN)
1714  D      FRQMP = ALFREQ(LEGEN)
1715  D      IF ((FRQMP .EQ. 0) .OR. (FROM .EQ. 0) .OR. (ATOTP .EQ. 0)
1716  D      $   .OR. (FRQMP .EQ. 0.) .OR. (FRQMP .EQ. 1.)) GO TO 60
1717  D      CALL GNTFIT (SUMGEN)
1718  D      CALL ALFIT (SUMGEN)
1719  D      CALL GNTSEL (SUMGEN)
1720  D      CALL ALSEL (SUMGEN)
1721  D
1722  D-----
1723  D  PRINT DEBUG DATA WHEN FITNESS & SEL. COEFFICIENTS CAN BE CALCULATED
1724  D-----
1725  D  40  WRITE(OUTLP,50) SUMGEN,JUV(1),INFLO(1),NOOUT(1),RSLT(1,SUMGEN),
1726  D      $   OBSFR(1),GNTSUR(1),FITGNT(1),GSELCO(1),FRQW,ALSUR(1),FITAL(1),
1727  D      $   ASELCO(1),JUV(2),INFLO(2),NOOUT(2),RSLT(2,SUMGEN),OBSFR(2),
1728  D      $   GNTSUR(2),FITGNT(2),GSELCO(2),JUV(3),INFLO(3),NOOUT(3),
1729  D      $   RSLT(3,SUMGEN),OBSFR(3),GNTSUR(3),FITGNT(3),GSELCO(3),FROM,
1730  D      $   ALSUR(2),FITAL(2),ASELCO(2),TOJUV,INTOT,OUTTOT,ATOT
1731  C
1732  D  50  FORMAT('0',I3,5X,'WW',4X,F5.0,2X,I4,2X,I5,3X,I4,3X,4(F5.3,2X),
1733  D      $   2X,'W',3X,4(2X,F5.3,/,9X,'WM',4X,F5.0,2X,I4,2X,I5,3X,I4,3X,
1734  D      $   4(F5.3,2X),/,9X,'MM',4X,F5.0,2X,I4,2X,I5,3X,I4,3X,4(F5.3,2X),
1735  D      $   2X,'M',3X,4(2X,F5.3),/,7X,'TOTAL',3X,F5.0,2X,I4,2X,I5,3X,
1736  D      $   I4,/)
1737  D      GO TO 99
1738  D
1739  D-----
    
```

```

1740 D PRINT DEBUG DATA WHEN FITNESS & SEL. COEFFICIENTS CAN'T BE CALCULATED
1741 D-----
1742 D 60 WRITE(OUTLP,70) SUMGEN, JUV(1), INFLO(1), NOOUT(1), RSLT(1, SUMGEN),
1743 D $ OBSFR(1), FRQW, JUV(2), INFLO(2), NOOUT(2), RSLT(2, SUMGEN), OBSFR(2),
1744 D $ JUV(3), INFLO(3), NOOUT(3), RSLT(3, SUMGEN), OBSFR(3), FRQM, TOJUV,
1745 D $ INTOT, OUTTOT, ATOT
1746 D
1747 D 70 FORMAT('0', I3, 5X, 'WW', 4X, F5.0, 2X, I4, 2X, I5, 3X, I4, 3X, F5.3, 2X,
1748 D $ '*****', ' ***** *****', 4X, 'W', 5X, F5.3, 2X,
1749 D $ '***** ***** *****', /, 9X, 'WM', 4X, F5.0, 2X, I4, 2X, I5, 3X, I4, 3X,
1750 D $ F5.3, 2X, '***** ***** ', '*****', /, 9X, 'MM', 4X, F5.0, 2X, I4, 2X,
1751 D $ I5, 3X, I4, 3X, F5.3, 2X, '*****', '***** *****', 4X, 'M', 5X, F5.3, 2X,
1752 D $ '***** ***** *****', /, 7X, 'TOTAL', 3X, F5.0, 2X, I4, 2X, I5, 3X, I4)
1753 C-----
1754 C
1755 C 99 RETURN
1756 C END
1757 C*****
1758 C
1759 C
1760 C*****
1761 C file: GR.MI vrsn: 43 rvsd: 27 Aug 81
1762 C Subroutine graph & miscellaneous subroutines
1763 C*****
1764 C
1765 C SUBROUTINE GRAPH
1766 C
1767 C FUNCTION: PRINTS GRAPHS OF RESULTS TO LINE PRINTER
1768 C
1769 C SYNOPSIS:
1770 C 1) THE MAIN PART OF THE ROUTINE IS DIVIDED INTO TWO MAJOR
1771 C SECTIONS WHICH DO GRAPHS OF FLOATING POINT AND INTEGER DATA
1772 C RESPECTIVELY. SUBROUTINE GRAPH FIRST DETERMINES WHETHER THE
1773 C CURRENT GRAPH IS OF INTEGER OR REAL DATA AND CONTROL JUMPS TO
1774 C THE APPROPRIATE SECTION
1775 C 2) WITHIN EACH SECTION AN INITIAL PORTION IS DEVOTED TO FILLING A
1776 C DATA ARRAY WITH THE APPROPRIATE DATA TO BE GRAPHED, AND A
1777 C SECOND PORTION PLOTS THE DATA FROM THE DATA ARRAY.
1778 C 3) PLOTTING CONSISTS OF COMPOSING A LINE OF PRINT FROM THE DATA
1779 C ARRAY, THEN PRINTING THE LINE.
1780 C 4) FROM EACH MAJOR SECTION, CONTROL JUMPS TO ONE SECTION THAT PRINTS
1781 C THE ABSCISSA AND LABEL, THEN RETURNS TO THE TOP OF THE ROUTINE
1782 C TO REPEAT THE PROCESS.
1783 C
1784 C NOTES:
1785 C 1) FACTR IS USED TO SCALE VERTICAL AXIS FOR 50 LINES PER PAGE
1786 C 2) "-9" IS USED WHEN COEFFICIENTS CANNOT BE CALCULATED; IT IS
1787 C USED TO SET PRINTED CHARACTER TO "?".
1788 C
1789 C NOTES ON POSSIBLE CHANGES:
1790 C 1) CHANGE DIMENSIONS FOR GRAPHS TO VARIABLES THAT CAN BE SET BEFORE
1791 C GRAPH IS CALLED. E.G., PGLNTH, PGWDTH, HSCALE, VSCALE
1792 C 2) MODIFY TO FLIP FROM ONE GRAPH TO ANOTHER & BACK FOR SCREEN USE
1793 C (USE "PRINT2").
1794 C COULD DISPLAY CHARACTER GRAPH TO SCREEN DIMENSIONS, ASK IF USER
1795 C WANTS HARD COPY TO LINE PRINTER.
1796 C 3) MODIFY TO USE SIGGRAPH CORE-79 ROUTINES ON TERAK.
1797 C 4) MODIFY TO PRINT OUT LENGTH-WISE FOR TELETYPES?
1798 C 5) COULD CLARIFY LOGIC BY BREAKING INTO SUBROUTINES TO, FOR E.G.,
1799 C FILL DATA ARRAYS, COMPOSE A LINE, ETC.
    
```

```

1800 C      6) THIS ROUTINE IS SLOW, IT CANNOT DRIVE THE PRINTRONIX AT FULL
1801 C      SPEED. WILL HAVE TO TRY TO SPEED IT UP. THIS MAY CONFLICT
1802 C      WITH NOTE 5.
1803 C
1804 C      CALLED FROM: MAIN
1805 C      CALLS: ALFREQ - RETURNS FREQUENCY OF M ALLELE
1806 C      GNTFIT - RETURNS FITNESS COEFFICIENT OF EACH GENOTYPE
1807 C      ALFIT - RETURNS FITNESS COEFFICIENT OF EACH ALLELE
1808 C      GNTSEL - RETURNS SELECTION COEFFICIENT OF EACH GENOTYPE
1809 C      ALSEL - RETURNS SELECTION COEFFICIENT OF EACH ALLELE
1810 C=====
1811 C      INCLUDE 'COMMON.f'
1812 C-----
1813 C
1814 C      LOCAL DECLARATIONS
1815 C      INTEGER ATOT, BLNK, CHAR, CHARH, CHARM, CHARQ, CHARW,
1816 C      $ DASH, IDATA(125), IDATUM, LNBTM, LINE(125), LNSIZ, MAX,
1817 C      $ N
1818 C      REAL DATA(3,125), DATUM1, DATUM2, DATUM3, FACTR, FLNBTM,
1819 C      $ FLNSIZ, FLNTOP, FMAX, IFACTR
1820 C      DATA BLNK/' '/, CHAR/'@'/, CHARH/'H'/,
1821 C      $ CHARM/'M'/, CHARQ/'?'/, CHARW/'W'/,
1822 C      $ DASH/'-'/, IDATA/125*0/
1823 C=====
1824 C      Type *, 'In Graph'
1825 C
1826 C      INITIALIZE, DETERMINE WHETHER ALL GRAPHS HAVE BEEN FINISHED.
1827 C      IF NOT, DETERMINE WHETHER CURRENT GRAPH IS OF FLOATING POINT OR
1828 C      OF INTEGER DATA.
1829 C-----
1830 C      DO 6000 I=1,10
1831 C      IF (GRF(I) .EQ. ' ') GO TO 9999
1832 C      IF ((GRF(I) .EQ. 'PO') .OR. (GRF(I) .EQ. 'WW') .OR.
1833 C      $ (GRF(I) .EQ. 'WM') .OR. (GRF(I) .EQ. 'MM')) GO TO 2000
1834 C
1835 C=====
1836 C      FLOATING POINT GRAPHS
1837 C=====
1838 C
1839 C      INITIALIZE, ZERO DATA MATRIX
1840 C      FMAX = 1.0
1841 C      FACTR = FMAX/50
1842 C      DO 20 J=1,3
1843 C      DO 10 K=1,125
1844 C      DATA(J,K) = 0.
1845 C      10 CONTINUE
1846 C      20 CONTINUE
1847 C
1848 C      DETERMINE GRAPH REQUESTED
1849 C      IF (GRF(I) .EQ. 'AF') GO TO 100
1850 C      IF (GRF(I) .EQ. 'GF') GO TO 200
1851 C      IF (GRF(I) .EQ. 'GW') GO TO 300
1852 C      IF (GRF(I) .EQ. 'AW') GO TO 400
1853 C      IF (GRF(I) .EQ. 'GS') GO TO 500
1854 C      IF (GRF(I) .EQ. 'AS') GO TO 600
1855 C      WRITE (OUTLP,30) I
1856 C      30 FORMAT (1X,'*****ERROR: GRAPH REQUEST #',I2,' IS INCORRECT')
1857 C      GO TO 9999
1858 C
1859 C-----
    
```

```

1860 C      SET PARAMETERS FOR FLOATING POINT GRAPHS
1861 C-----
1862 C
1863 C      SET PARAMETERS FOR ALLELE FREQUENCY GRAPH
1864 100    DO 110 K=1,IGEN
1865         DATA(2,K) = ALFREQ(K)
1866         DATA(1,K) = 1. - DATA(2,K)
1867         IF((RSLT(1,K) .EQ. 0) .AND. (RSLT(2,K) .EQ. 0))
1868             $      DATA(1,K) = 0.
1869         IF((RSLT(3,K) .EQ. 0) .AND. (RSLT(2,K) .EQ. 0))
1870             $      DATA(2,K) = 0.
1871 110    CONTINUE
1872        GO TO 1000
1873 C.....
1874 C
1875 C      SET PARAMETERS FOR GENOTYPE FREQUENCY GRAPH
1876 200    DO 230 J=1,IGEN
1877         ATOT = RSLT(1,J) + RSLT(2,J) + RSLT(3,J)
1878         IF (ATOT .GT. 0) GO TO 210
1879         DATA(1,J) = 0.
1880         DATA(2,J) = 0.
1881         DATA(3,J) = 0.
1882         GO TO 230
1883 C
1884 210    DO 220 K=1,3
1885         IDATUM = RSLT(K,J)
1886         IF (IDATUM .GT. 0 ) GO TO 215
1887         DATA (K,J) = 0.
1888         GO TO 220
1889 215    DATA(K,J) = FLOAT(IDATUM) / ATOT
1890 220    CONTINUE
1891 230    CONTINUE
1892        GO TO 1000
1893 C.....
1894 C
1895 C      SET PARAMETERS FOR GENOTYPE FITNESS GRAPH
1896 300    DATA(1,1) = -9.
1897         DATA(2,1) = -9.
1898         DATA(3,1) = -9.
1899         DO 320 J=2,IGEN
1900             LGEN = J - 1
1901             FRQMP = ALFREQ(LGEN)
1902             ATOT = RSLT(1,J) + RSLT(2,J) + RSLT(3,J)
1903             IF ((FRQMP .GT. 0.) .AND. (FRQMP .LT. 1.) .AND.
1904                 $      (ATOT .GT. 0)) GO TO 310
1905 C
1906         DATA(1,J) = -9.
1907         DATA(2,J) = -9.
1908         DATA(3,J) = -9.
1909         GO TO 320
1910 C
1911 310    CALL GNTFIT(J)
1912         DATA(1,J) = FITGNT(1)
1913         DATA(2,J) = FITGNT(2)
1914         DATA(3,J) = FITGNT(3)
1915 C
1916 320    CONTINUE
1917        GO TO 1000
1918 C.....
1919 C
    
```



```

1920 C      SET PARAMETERS FOR ALLELE FITNESS GRAPH
1921 400    DATA(1,1) = -.9
1922      DATA(2,1) = -.9
1923 C
1924      DO 420 J=2,IGEN
1925          LGEN = J - 1
1926          FRQMP = ALFREQ(LGEN)
1927          ATOT = RSLT(1,LGEN) + RSLT(2,LGEN) + RSLT(3,LGEN)
1928          IF ((FRQMP .GT. 0.) .AND. (FRQMP .LT. 1.) .AND.
1929 1          (ATOT .GT. 0)) GO TO 410
1930 C
1931          DATA(1,J) = -9.
1932          DATA(2,J) = -9.
1933          GO TO 420
1934 C
1935 410      CALL ALFIT(J)
1936          DATA(1,J) = FITAL(1)
1937          DATA(2,J) = FITAL(2)
1938 C
1939 420      CONTINUE
1940          GO TO 1000
1941 C.....
1942 C
1943 C      SET PARAMETERS FOR GENOTYPE SELECTION COEF. GRAPH
1944 500    DATA(1,1) = -9.
1945          DATA(2,1) = -9.
1946          DATA(3,1) = -9.
1947          DO 520 J=2,IGEN
1948              LGEN = J - 1
1949              FRQMP = ALFREQ(LGEN)
1950              ATOT = RSLT(1,J) + RSLT(2,J) + RSLT(3,J)
1951              IF ((FRQMP .GT. 0.) .AND. (FRQMP .LT. 1.) .AND.
1952 $          (ATOT .GT. 0)) GO TO 510
1953 C
1954          DATA(1,J) = -9.
1955          DATA(2,J) = -9.
1956          DATA(3,J) = -9.
1957          GO TO 520
1958 C
1959 510      CALL GNTSEL (J)
1960          DATA(1,J) = GSELCO(1)
1961          DATA(2,J) = GSELCO(2)
1962          DATA(3,J) = GSELCO(3)
1963 520      CONTINUE
1964          GO TO 1000
1965 C.....
1966 C
1967 C      SET PARAMETERS FOR ALLELE SELECTION COEF. GRAPH
1968 600    DATA(1,1) = -9.
1969          DATA(2,1) = -9.
1970          DO 620 J=2,IGEN
1971              LGEN = J - 1
1972              FRQMP = ALFREQ(LGEN)
1973              ATOT = RSLT(1,J) + RSLT(2,J) + RSLT(3,J)
1974              IF ((FRQMP .GT. 0.) .AND. (FRQMP .LT. 1.) .AND.
1975 $          (ATOT .GT. 0)) GO TO 610
1976 C
1977          DATA(1,J) = -9.
1978          DATA(2,J) = -9.
1979          GO TO 620
    
```

```

1980 C
1981   610      CALL ALSEL (J)
1982          DATA(1,J) = ASELCO(1)
1983          DATA(2,J) = ASELCO(2)
1984   620      CONTINUE
1985          GO TO 1000
1986 C
1987 C-----
1988 C          PLOT FLOATING POINT DATA
1989 C-----
1990 C
1991   1000     WRITE (OUTLP,5000) TITLE
1992          WRITE (OUTLP,5002)
1993 C
1994 C          PRINT 50 LINES
1995          IFACR=FACTR/2
1996          DO 1060 J=1,51
1997 C
1998 C          COMPOSE A LINE
1999          FLNSIZ = FMAX - ((J-1)*FACTR)
2000          FLNTOP = FLNSIZ + FACTR/2
2001          FLNBTM = FLNSIZ - FACTR/2
2002          DO 1010 K=1,125
2003              LINE(K) = BLNK
2004   1010     CONTINUE
2005          IF ((GRF(I) .NE. 'AF') .AND. (GRF(I) .NE. 'AW') .AND.
2006              $ (GRF(I) .NE. 'AS')) GO TO 1030
2007 C.....
2008 C
2009 C          COMPOSE ALLELE COEFFICIENT LINES
2010          DO 1020 K = 1,IGEN
2011              DATUM1 = DATA(1,K)
2012              DATUM2 = DATA(2,K)
2013              LINE(K) = BLNK
2014              IF (DATUM1 .GE. 0.) GO TO 1011
2015              LINE(K) = CHARQ
2016              GO TO 1012
2017   1011     IF ((DATUM1 .GT. FLNBTM) .AND. (DATUM1 .LE.
2018              $ FLNTOP)) LINE(K) = CHARW
2019   1012     IF (DATUM2 .GE. 0.) GO TO 1013
2020              LINE(K) = CHARQ
2021              GO TO 1020
2022   1013     IF ((DATUM2 .GT. FLNBTM) .AND. (DATUM2 .LE.
2023              $ FLNTOP)) LINE(K) = CHARM
2024   1020     CONTINUE
2025          IF (J .LT. 51) GO TO 1050
2026          DO 1025 K=1,125
2027              IF ((DATA(2,K) .LE. 0.) .OR.
2028              $ (DATA(1,K) .LE. 0.)) LINE(K) = DASH
2029   1025     CONTINUE
2030          GO TO 1050
2031 C.....
2032 C
2033 C          COMPOSE GENOTYPE COEFFICIENT LINES
2034 C          NOTE THAT H OVERPRINTS M WHICH OVERPRINTS W
2035   1030     DO 1040 K = 1,IGEN
2036              DATUM1 = DATA(1,K)
2037              DATUM2 = DATA(2,K)
2038              DATUM3 = DATA(3,K)
2039              LINE(K) = BLNK

```

```

2040      IF (DATUM1 .GE. 0.) GO TO 1031
2041      LINE(K) = CHARQ
2042      GO TO 1032
2043 1031      IF ((DATUM1 .GT. FLNBTM) .AND. (DATUM1 .LE.
2044      $      FLNTOP)) LINE(K) = CHARW
2045 1032      IF (DATUM3 .GE. 0. ) GO TO 1033
2046      LINE(K) = CHARQ
2047      GO TO 1034
2048 1033      IF ((DATUM3 .GT. FLNBTM) .AND. (DATUM3 .LE.
2049      $      FLNTOP)) LINE(K) = CHARM
2050 1034      IF (DATUM2 .GE. 0. ) GO TO 1035
2051      LINE(K) = CHARQ
2052      GO TO 1040
2053 1035      IF ((DATUM2 .GT. FLNBTM) .AND. (DATUM2 .LE.
2054      $      FLNTOP)) LINE(K) = CHARH
2055 1040      CONTINUE
2056      IF (J .LT. 51) GO TO 1050
2057      DO 1045 K=1,125
2058          IF ((DATA(2,K) .LE. 0.) .AND. ((DATA(1,K)
2059      $          .LE. 0.) .OR. (DATA(3,K) .LE. 0.))) LINE(K) = DASH
2060 1045      CONTINUE
2061 C.....
2062 C
2063 C      PRINT A LINE
2064 1050      WRITE (OUTLP,1052) FLNSIZ, LINE
2065 1052      FORMAT (1X,F5.3,'+',125A1)
2066 1060      CONTINUE
2067 C
2068      GO TO 4000
2069 C
2070 C
2071 C=====
2072 C  INTEGER PLOTS (POP, WW, WM, MM)
2073 C=====
2074 C
2075 C      ZERO DATA MATRIX
2076 2000      DO 2010 K=1,125
2077          IDATA(K) = 0
2078 2010      CONTINUE
2079      IF (GRF(I) .NE. 'PO') GO TO 2200
2080 C
2081 C-----
2082 C      SET PARAMETERS FOR INTEGER PLOTS
2083 C-----
2084 C
2085 C      SET PARAMETERS FOR POPULATION SIZE GRAPH
2086 C
2087 C      DETERMINE MAXIMUM POPULATION SIZE DURING RUN, IF < 50, SET = 50
2088 C      (I.E. ONE INDIVIDUAL PER GRAPH LINE)
2089      MAX = RSLT(1,1) + RSLT(2,1) + RSLT(3,1)
2090      IDATA(1) = MAX
2091      DO 2100 J=1,IGEN
2092          IDATA(J) = RSLT(1,J) + RSLT(2,J) + RSLT(3,J)
2093          IF (MAX .LT. IDATA(J)) MAX = IDATA(J)
2094 2100      CONTINUE
2095      FACTR = FLOAT(MAX)/50
2096      IF (MAX .GT. 50) GO TO 2110
2097      FACTR = 1.
2098      MAX = 50
2099 2110      GO TO 3000
    
```

```
2100 C.....
2101 C
2102 C      SET PARAMETERS FOR GENOTYPE NUMBERS GRAPHS
2103 C
2104 C      ZERO DATA MATRIX
2105 2200 DO 2210 K=1,125
2106      IDATA(K) = 0
2107 2210 CONTINUE
2108 C
2109 C      SET GENOTYPE FOR GRAPH
2110 N = 0
2111 IF (GRF(I) .NE. 'WW') GO TO 2220
2112 N = 1
2113 GO TO 2250
2114 C
2115 2220 IF (GRF(I) .NE. 'WM') GO TO 2230
2116 N = 2
2117 GO TO 2250
2118 C
2119 2230 IF (GRF(I) .NE. 'MM') GO TO 2240
2120 N = 3
2121 GO TO 2250
2122 2240 WRITE (OUTLP,30) I
2123 GO TO 9999
2124 C
2125 C      DETERMINE MAXIMUM NUMBER OF GENOTYPE DURING RUN, IF MAX < 50,
2126 C      SET = 50 (I.E., ONE INDIVIDUAL PER GRAPH LINE)
2127 2250 MAX = RSLT(N,1)
2128 DO 2260 J=1,IGEN
2129     IDATA(J) = RSLT(N,J)
2130     IF (MAX .LT. IDATA(J)) MAX = IDATA(J)
2131 2260 CONTINUE
2132 FACTR = FLOAT(MAX)/50
2133 C
2134 IF (MAX .GT. 50) GO TO 3000
2135 FACTR = 1.
2136 MAX = 50
2137 C
2138 C-----
2139 C      PLOT INTEGER DATA
2140 C-----
2141 C
2142 3000 WRITE (OUTLP,5000) TITLE
2143 WRITE (OUTLP,5002)
2144 C
2145 C      PRINT 50 LINES
2146 DO 3040 J=1,51
2147 C
2148 C      COMPOSE A LINE
2149 LNSIZ = MAX - ((J-1)*FACTR) + 0.5
2150 LNBTM = LNSIZ - FACTR/2
2151 DO 3010 K=1,125
2152     LINE(K) = BLNK
2153 3010 CONTINUE
2154 DO 3020 K=1,IGEN
2155     IF (IDATA(K) .GT. LNBTM) LINE(K) = CHAR
2156 3020 CONTINUE
2157 IF (J .LT. 51) GO TO 3030
2158 DO 3025 K=1,125
2159     IF (IDATA(K) .EQ. 0) LINE(K) = DASH
```

```

2160 3025          CONTINUE
2161 C
2162 C          PRINT A LINE
2163 3030          WRITE (OUTLP,3032) LNSIZ, LINE
2164 3032          FORMAT (1X,I4,'+',125A1)
2165 3040          CONTINUE
2166          GO TO 4000
2167 C
2168 C=====
2169 C  PRINT ABSCISSA & LABEL FOR ALL PLOTS
2170 C=====
2171 C
2172 4000          WRITE(OUTLP,5002)
2173          WRITE (OUTLP,5003)
2174          WRITE (OUTLP,5004)
2175          WRITE (OUTLP,5009)
2176          IF (GRF(I) .NE. 'AF') GO TO 4010
2177          WRITE (OUTLP,5010)
2178          GO TO 4110
2179 4010          IF (GRF(I) .NE. 'GF') GO TO 4020
2180          WRITE (OUTLP,5020)
2181          WRITE (OUTLP,5110)
2182          GO TO 4110
2183 4020          IF (GRF(I) .NE. 'GW') GO TO 4030
2184          WRITE (OUTLP,5030)
2185          WRITE (OUTLP,5110)
2186          WRITE (OUTLP,5120)
2187          GO TO 4110
2188 4030          IF (GRF(I) .NE. 'AW') GO TO 4040
2189          WRITE (OUTLP,5040)
2190          WRITE (OUTLP,5120)
2191          GO TO 4110
2192 4040          IF (GRF(I) .NE. 'GS') GO TO 4050
2193          WRITE (OUTLP,5050)
2194          WRITE (OUTLP,5110)
2195          WRITE (OUTLP,5120)
2196          GO TO 4110
2197 4050          IF (GRF(I) .NE. 'AS') GO TO 4060
2198          WRITE (OUTLP,5060)
2199          WRITE (OUTLP,5120)
2200          GO TO 4110
2201 4060          IF (GRF(I) .NE. 'PO') GO TO 4070
2202          WRITE (OUTLP,5070)
2203          GO TO 4110
2204 4070          IF (GRF(I) .NE. 'WW') GO TO 4080
2205          WRITE (OUTLP,5080)
2206          GO TO 4110
2207 4080          IF (GRF(I) .NE. 'WM') GO TO 4090
2208          WRITE (OUTLP,5090)
2209          GO TO 4110
2210 4090          IF (GRF(I) .NE. 'MM') GO TO 4100
2211 4100          WRITE (OUTLP,5100)
2212 4110          WRITE (OUTLP,5009)
2213 C
2214 5000          FORMAT ('1',20X,40A2)
2215 5002          FORMAT (6X,'+',12('-----+----*'),'-----+')
2216 5003          FORMAT (16X,'1',9X,'2',9X,'3',9X,'4',9X,'5',9X,'6',9X,'7',9X,
2217 $            '8',9X,'9',9X,'10',8X,'11',8X,'12')
2218 5004          FORMAT (45X,'GENERATION (X10)',/)
2219 5009          FORMAT (20X,100('*'))
    
```

```

2220 5010  FORMAT (25X,'FREQUENCY OF W AND M ALLELES')
2221 5020  FORMAT (25X,'FREQUENCY OF EACH GENOTYPE')
2222 5030  FORMAT (25X,'RELATIVE FITNESS OF EACH GENOTYPE')
2223 5040  FORMAT (25X,'RELATIVE FITNESS OF W AND M ALLELES')
2224 5050  FORMAT (25X,'SELECTION COEFFICIENTS OF GENOTYPES')
2225 5060  FORMAT (25X,'SELECTION COEFFICIENTS OF W AND M ALLELES')
2226 5070  FORMAT (25X,'POPULATION SIZE')
2227 5080  FORMAT (25X,'NUMBER OF WW GENOTYPE')
2228 5090  FORMAT (25X,'NUMBER OF WM GENOTYPE')
2229 5100  FORMAT (25X,'NUMBER OF MM GENOTYPE')
2230 5110  FORMAT ('+',10X,'W = WW, H = WM, M = MM')
2231 5120  FORMAT ('+',',', '? = UNDEFINABLE COEFF')
2232 C
2233 6000  CONTINUE
2234      WRITE (OUTLP,6010)
2235 6010  FORMAT (1H1)
2236 9999  RETURN
2237      END
2238 C
2239 C*****
2240 C*****
2241 C
2242      SUBROUTINE GNTFIT (DUMY)
2243 C
2244 C      FUNCTION: RETURNS THE RELATIVE FITNESS COEFFICIENT OF EACH GENOTYPE
2245 C      ("FITGNT").
2246 C
2247 C      NOTES:
2248 C          1) THE METHODS USED FOR THIS AND OTHER COEFFICIENT SUBROUTINES
2249 C              ARE DETAILED ON PAGE 52 IN: WILSON, E. O., AND W. H. BOSSERT
2250 C              (1971) "A PRIMER OF POPULATION BIOLOGY". STAMFORD, CONN.,
2251 C              SINAUER ASSOC.
2252 C
2253 C      CALLED FROM: SUMRYS, GRAPH, GNTSEL
2254 C      CALLS: ALFREQ - RETURNS FREQUENCY OF M ALLELE
2255 C=====
2256      INCLUDE 'COMMON.f'
2257 C-----
2258 C      LOCAL DECLARATIONS
2259 C      REAL  EXPFRM, EXPFRW, EXPFR(3)
2260 C      INTEGER      ATOT,  DUMY
2261 C=====
2262 C      Type *,'In Gntfit'
2263 C
2264 C      TEST THAT COEFFICIENTS CAN BE CALCULATED
2265 C      IF (IGEN .NE. 1) GO TO 20
2266 C      WRITE (OUTLP,10)
2267 10  FORMAT (' *** WARNING: ENTRY INTO GNTFIT WITH IGEN = 1')
2268 C      GO TO 999
2269 20  ATOT = (RSLT(1,DUMY)+RSLT(2,DUMY)+RSLT(3,DUMY))
2270 C      LGEN = DUMY - 1
2271 C      FRQMP = ALFREQ(LGEN)
2272 C      IF ((FRQMP .GT. 0.) .AND. (FRQMP .LT. 1.) .AND. (ATOT .GT. 0))
2273 1  GO TO 40
2274 C      WRITE (OUTLP,30)
2275 30  FORMAT (' *****WARNING: ENTRY INTO GNTFIT WITH M ALLELE FREQ.',
2276 1  ' IN PREVIOUS GEN = 0 OR 1, OR TOTAL # ADULTS IN PRESENT GEN',
2277 2  ' = 0')
2278 C      GO TO 999
2279 C-----
    
```

```
2280 C
2281 C    INITIALIZE VARIABLES
2282 40 DO 50 J=1,3
2283     FITGNT(J) = 0.
2284     OBSFR(J) = 0.
2285     EXPFR(J) = 0.
2286 50 CONTINUE
2287 C
2288 C    CALCULATE OBSERVED GENOTYPE FREQUENCY
2289 DO 60 J=1,3
2290     OBSFR(J) = FLOAT(RSLT(J,DUMY))/(RSLT(1,DUMY)+
2291 + RSLT(2,DUMY)+RSLT(3,DUMY))
2292 60 CONTINUE
2293 C
2294 C    CALCULATE EXPECTED FREQUENCY OF EACH ALLELE (BY THE HARDY-WEINBERG
2295 C    LAW, THE EXPECTED ALLELE FREQUENCIES EQUAL THE FREQUENCIES IN THE
2296 C    PREVIOUS GENERATION). THE EXPECTED GENOTYPE FREQUENCIES (BY THE
2297 C    HARDY-WEINBERG LAW) = P*P, 2PQ, & Q*Q FOR WW, WM & MM
2298 C    RESPECTIVELY.
2299 EXPFRM = 0.
2300 EXPFRW = 0.
2301 LGEN = DUMY - 1
2302 C
2303 EXPFRM = ALFREQ(LGEN)
2304 EXPFRW = 1 - EXPFRM
2305 C
2306 EXPFR(1) = EXPFRW * EXPFRW
2307 EXPFR(2) = 2 * EXPFRW * EXPFRM
2308 EXPFR(3) = EXPFRM * EXPFRM
2309 C
2310 C    CALCULATE GENOTYPE SURVIVAL RATE, OR ABSOLUTE FITNESS OF EACH
2311 C    GENOTYPE
2312 DO 70 J=1,3
2313     GNTSUR(J) = OBSFR(J) / EXPFR(J)
2314 70 CONTINUE
2315 C
2316 C    DETERMINE WHICH GENOTYPE HAS THE HIGHEST SURVIVAL RATE, CALCULATE
2317 C    RELATIVE FITNESS COEFFICIENTS BY DIVIDING SURVIVAL RATE OF EACH
2318 C    GENOTYPE BY THE HIGHEST SURVIVAL RATE
2319 IF ((GNTSUR(2) .GE. GNTSUR(1)) .AND. (GNTSUR(2) .GE. GNTSUR(3)))
2320 1 GO TO 90
2321 IF ((GNTSUR(3) .GE. GNTSUR(1)) .AND. (GNTSUR(3) .GE. GNTSUR(2)))
2322 1 GO TO 100
2323 C
2324 C    WW HAS HIGHEST SURVIVAL RATE
2325 C    ((GNTSUR(1) .GE. GNTSUR(2)) .AND. (GNTSUR(1) .GE. GNTSUR(3)))
2326 80 FITGNT(1) = 1.0
2327     FITGNT(2) = GNTSUR(2) / GNTSUR(1)
2328     FITGNT(3) = GNTSUR(3) / GNTSUR(1)
2329 GO TO 999
2330 C
2331 C    WM HAS HIGHEST SURVIVAL RATE
2332 90 FITGNT(2) = 1.0
2333     FITGNT(1) = GNTSUR(1) / GNTSUR(2)
2334     FITGNT(3) = GNTSUR(3) / GNTSUR(2)
2335 GO TO 999
2336 C
2337 C    MM HAS HIGHEST SURVIVAL RATE
2338 100 FITGNT(3) = 1.0
2339     FITGNT(1) = GNTSUR(1) / GNTSUR(3)
```

```

2340     FITGNT(2) = GNTSUR(2) / GNTSUR(3)
2341 C
2342     999 RETURN
2343     END
2344 C
2345 C*****
2346 C*****
2347 C
2348     SUBROUTINE ALFIT (DUMY)
2349 C
2350 C     FUNCTION: RETURNS THE RELATIVE FITNESS COEFFICIENT OF EACH ALLELE
2351 C     ('FITAL').
2352 C
2353 C     NOTES:
2354 C     1) FOR METHODOLOGY, SEE NOTE IN SUBROUTINE GNTFIT
2355 C
2356 C     CALLED FROM: SUMRYS, GRAPH, ALSEL
2357 C     CALLS: ALFREQ - RETURNS FREQUENCY OF M ALLELE
2358 C=====
2359     INCLUDE 'COMMON.f'
2360 C-----
2361 C     LOCAL DECLARATIONS
2362 C     INTEGER ATOT, DUMY
2363 C=====
2364 C     Type *,'In Alfit'
2365 C     TEST THAT COEFFICIENT CAN BE CALCULATED
2366 C     IF (IGEN .GT. 1) GO TO 10
2367 C     WRITE (OUTLP,1)
2368 C     1    FORMAT (' ***WARNING: ENTRY INTO ALFIT WITH IGEN = 1')
2369 C     GO TO 99
2370 C     10 LGEN = DUMY - 1
2371 C     ATOT = RSLT(1,LGEN)+RSLT(2,LGEN)+RSLT(3,LGEN)
2372 C     ALFRMP = ALFREQ(LGEN)
2373 C     IF ((ALFRMP .NE. 0.) .AND. (ALFRMP .NE. 1.) .AND. (ATOT .NE. 0))
2374 C     $    GO TO 30
2375 C     WRITE (OUTLP,20)
2376 C     20 FORMAT (' ***WARNING: ENTRY INTO ALFIT WITH FREQ. OF M IN ',
2377 C     $    'PREVIOUS GENERATION = 0. OR 1., OR TOTAL # ADULTS = 0. ')
2378 C     GO TO 99
2379 C
2380 C     INITIALIZE VARIABLES
2381 C     30 ALSUR(1) = 0.
2382 C     ALSUR(2) = 0.
2383 C     FITAL(1) = 0.
2384 C     FITAL(2) = 0.
2385 C
2386 C     CALCULATE ALLELE SURVIVAL RATES
2387 C     ALSUR(1) = (1-ALFREQ(DUMY)) / (1-ALFREQ(LGEN))
2388 C     ALSUR(2) = (ALFREQ(DUMY)) / (ALFREQ(LGEN))
2389 C
2390 C     DETERMINE WHICH ALLELE HAS THE HIGHEST SURVIVAL RATE
2391 C     IF (ALSUR(1) .LT. ALSUR(2)) GO TO 40
2392 C
2393 C     ALLELE W HAS THE HIGHEST SURVIVAL RATE
2394 C     FITAL(1) = 1.0
2395 C     FITAL(2) = ALSUR(2) / ALSUR(1)
2396 C     GO TO 99
2397 C
2398 C     ALLELE M HAS THE HIGHEST SURVIVAL RATE
2399 C     40 FITAL(2) = 1.0

```



```
2400      FITAL(1) = ALSUR(1) / ALSUR(2)
2401 C
2402      99 RETURN
2403      END
2404 C
2405 C*****
2406 C*****
2407 C
2408      SUBROUTINE ELSEL (DUMY)
2409 C
2410 C      FUNCTION: RETURNS THE SELECTION COEFFICIENT OF EACH ALLELE
2411 C      ("ASELCO").
2412 C
2413 C      NOTES:
2414 C          1) FOR METHODOLOGY, SEE NOTE IN SUBROUTINE GNTFIT
2415 C          2) THIS SUBROUTINE COULD BE INCORPORATED INTO ALFIT
2416 C
2417 C      CALLED FROM: SUMRYS, GRAPH
2418 C      CALLS: ALFIT - RETURNS FITNESS COEFFICIENT OF EACH ALLELE
2419 C=====
2420      INCLUDE 'COMMON.f'
2421 C-----
2422 C      LOCAL DECLARATIONS
2423 C      INTEGER DUMY
2424 C=====
2425 C      Type *,'In Alsel'
2426 C
2427 C      ALLELE SELECTION COEFFICIENT = 1 - ALLELE FITNESS COEFFICIENT
2428 C      CALL ALFIT (DUMY)
2429 C      DO 10 J=1,2
2430 C          ASELCO(J) = 1 - FITAL(J)
2431 C      10 CONTINUE
2432 C      RETURN
2433 C      END
2434 C
2435 C*****
2436 C*****
2437 C
2438      SUBROUTINE GNTSEL (DUMY)
2439 C
2440 C      FUNCTION: RETURNS THE SELECTION COEFFICIENT OF EACH GENOTYPE
2441 C      ("GSELCO").
2442 C
2443 C      NOTES:
2444 C          1) FOR METHODOLOGY, SEE NOTE IN SUBROUTINE GNTFIT
2445 C          2) THIS SUBROUTINE COULD BE INCORPORATED INTO GNTFIT
2446 C
2447 C      CALLED FROM: SUMRYS, GRAPH
2448 C      CALLS: GNTFIT - RETURNS FITNESS COEFFICIENT OF EACH GENOTYPE
2449 C=====
2450      INCLUDE 'COMMON.f'
2451 C-----
2452 C      LOCAL DECLARATIONS
2453 C      INTEGER DUMY
2454 C=====
2455 C      Type *,'In Gntsel'
2456 C
2457 C      GENOTYPE SELECTION COEFFICIENT = 1 - GENOTYPE FITNESS COEFFICIENT
2458 C      CALL GNTFIT (DUMY)
2459 C      DO 10 J=1,3
```

```

2460         GSELCO(J) = 1 - FITGNT(J)
2461     10 CONTINUE
2462         RETURN
2463     END
2464 C
2465 C*****
2466 C
2467     FUNCTION ALFREQ (DUMY)
2468 C
2469 C     FUNCTION: FREQUENCY OF MUTANT ALLELE M (ALFREQ) IN SPECIFIED
2470 C     GENERATION
2471 C
2472 C     NOTES:
2473 C     1)
2474 C
2475 C     CALLED FROM: SRVIVL, GRAPH, GNTFIT
2476 C     CALLS: NOTHING
2477 C=====
2478     INCLUDE 'COMMON.f'
2479 C-----
2480 C     LOCAL DECLARATIONS
2481 C     INTEGER DUMY
2482 C=====
2483 C     Type *,'In Alfreq'
2484 C
2485 C     VARIABLE ALFREQ IS THE ALLELE FREQUENCY OF THE M ALLELE
2486 C     IF THERE ARE NO WM OR MM ADULTS, SET ALFREQ=0, OTHERWISE COMPUTE
2487 C     ALFREQ
2488 C     IF ((RSLT(2,DUMY) .EQ. 0) .AND. (RSLT(3,DUMY) .EQ. 0)) GO TO 10
2489 C
2490 C     ALFREQ = (RSLT(2,DUMY) + (2*FLOAT(RSLT(3,DUMY)))) /
2491 C     $      (2*((RSLT(1,DUMY)+RSLT(2,DUMY)+RSLT(3,DUMY))))
2492 C     GO TO 20
2493 C
2494 C     10 ALFREQ = 0.
2495 C     20 RETURN
2496 C     END
2497 C
2498 C*****
2499 C*****
2500 C
2501     SUBROUTINE GAUSS
2502 C
2503 C     FUNCTION: RETURNS FLOATING POINT NUMBER "V" WITH MEAN = AM AND
2504 C     STD. DEV. = S
2505 C
2506 C     NOTES:
2507 C     1) THIS ALGORITHM WAS PROVIDED BY A COMPUTER PERSON WHO HAS
2508 C     SINCE MOVED. HE DID NOT PROVIDE DOCUMENTATION, OTHER THAN
2509 C     A NOTE TO SEE VOL. 2 OF KNUTH'S "ART OF COMPUTER PROGRAMMING"
2510 C     2) BEYOND AN AWARENESS THAT RANDOM NUMBER ALGORITHMS ARE A
2511 C     TARPIT FOR THE UNWARY, I AM NOT KNOWLEDGABLE IN THIS AREA.
2512 C     I CANNOT VOUCH FOR THE VALIDITY OR TRANSPORTABILITY OF THIS
2513 C     ALGORITHM. I HOPE CONDUIT CAN HELP BY VALIDATING THIS
2514 C     ALGORITHM OR DONATING A MORE TRANSPORTABLE ONE.
2515 C
2516 C     CALLED FROM: REPROD
2517 C     CALLS: RAND - FORTRAN LIBRARY FUNCTION, RETURNS RANDOM NUMBERS
2518 C             UNIFORMLY DISTRIBUTED FROM 0.0-->1.0
2519 C     SQRT - FORTRAN LIBRARY FUNCTION, RETURNS SQUARE ROOT
    
```

```

2520 C          ALOG  - FORTRAN LIBRARY FUNCTION, RETURNS NATURAL LOGARITHM
2521 C=====
2522 C          INCLUDE 'COMMON.f'
2523 C-----
2524 C          LOCAL DECLARATIONS
2525 C          REAL      Q,  U1,  U2,   V1,  V2,  X1
2526 C=====
2527 C          Type *, 'In Gauss'
2528 C          Type *, IY
2529 C          Type *, IZ
2530 C          Pause
2531 C
2532 C      100 U1 = RAN(IY)
2533 C          U2 = RAN(IY)
2534 C          V1 = 2 * U1 - 1
2535 C          V2 = 2 * U2 - 1
2536 C          Q = V1 * V1 + V2 * V2
2537 C          IF (Q .GE. 1) GO TO 100
2538 C          X1 = V1 * SQRT(-2 * ALOG(Q) / Q)
2539 C          V = X1 * S + AM
2540 C          RETURN
2541 C          END
2542 C
2543 C*****
2544
2545
2546 C      file: COMN   vrsn: 14   rvsd: 13 Aug 81
2547 C-----
2548 C GLOBAL DECLARATIONS
2549 C      COMMON /A/  ALSUR(2),  AM,  ASELCO(2),  FITAL(2),
2550 C      1  FITGNT(3), FRQMP,  GNTSUR(3),  GRF(10),  GSELCO(3),  IGEN,
2551 C      2  IN,  INFLO(3),  IY,  IZ,  JUV(3),  LGEN,  MATES(6),  MXPOP1,
2552 C      3  MXPOP2,  NEXT,  NOADLT(3),  NOGENS,  NOOUT(3),  NOYONG(6),
2553 C      4  OBSFR(3),  OUTFLO(3),  OUTLP,  OUTSCR,  PRINT2,  REPR8(3),
2554 C      5  RSLT(3,125),  S,  SUMGEN,  SUMRY,  SURVR8(3),  TITLE(40)
2555 C      COMMON /B/  TOADLT,  V
2556 C
2557 C      INTEGER IGEN,  IN,  INFLO,  IY,  IZ,
2558 C      1  LGEN,  MATES,  MXPOP1,  MXPOP2,  NOADLT,  NOGENS,
2559 C      2  OUTFLO,  OUTLP,  OUTSCR,  PRINT2,  RSLT,  SUMGEN,
2560 C      3  SUMRY,  TOADLT
2561 C
2562 C      REAL ALSUR,  AM,  ASELCO,  FITAL,  FITGNT,  FRQMP,  GNTSUR,
2563 C      1  GSELCO,  JUV,  NOOUT,  NOYONG,  OBSFR,  REPR8,  S,  SURVR8,  V
2564 C
2565 C      STRING*2  ANS,  GRF,  NEXT
2566 C
2567 C      STRING*80  TITLE
2568 C-----

```