



DECUS

PROGRAM LIBRARY

DECUS NO.	8-643
TITLE	LIFE
AUTHOR	Philip Corman
COMPANY	Stewart Radiance Laboratory Bedford, Massachusetts
DATE	June 28, 1973
SOURCE LANGUAGE	8K FORTRAN - OS/8

ATTENTION

This is a USER program. Other than requiring that it conform to submittal and review standards, no quality control has been imposed upon this program by DECUS.

The DECUS Program Library is a clearing house only; it does not generate or test programs. No warranty, express or implied, is made by the contributor, Digital Equipment Computer Users Society or Digital Equipment Corporation as to the accuracy or functioning of the program or related material, and no responsibility is assumed by these parties in connection therewith.

DECUS

Department of Education



Year	Grade	Score
1950	1st	85
1951	1st	88
1952	1st	90
1953	1st	92
1954	1st	95
1955	1st	98
1956	1st	100
1957	1st	100
1958	1st	100
1959	1st	100
1960	1st	100
1961	1st	100
1962	1st	100
1963	1st	100
1964	1st	100
1965	1st	100
1966	1st	100
1967	1st	100
1968	1st	100
1969	1st	100
1970	1st	100
1971	1st	100
1972	1st	100
1973	1st	100
1974	1st	100
1975	1st	100
1976	1st	100
1977	1st	100
1978	1st	100
1979	1st	100
1980	1st	100
1981	1st	100
1982	1st	100
1983	1st	100
1984	1st	100
1985	1st	100
1986	1st	100
1987	1st	100
1988	1st	100
1989	1st	100
1990	1st	100
1991	1st	100
1992	1st	100
1993	1st	100
1994	1st	100
1995	1st	100
1996	1st	100
1997	1st	100
1998	1st	100
1999	1st	100
2000	1st	100
2001	1st	100
2002	1st	100
2003	1st	100
2004	1st	100
2005	1st	100
2006	1st	100
2007	1st	100
2008	1st	100
2009	1st	100
2010	1st	100
2011	1st	100
2012	1st	100
2013	1st	100
2014	1st	100
2015	1st	100
2016	1st	100
2017	1st	100
2018	1st	100
2019	1st	100
2020	1st	100
2021	1st	100
2022	1st	100

Department of Education
1000 ...
...

LIFE

DECUS Program Library Write-up

DECUS NO. 8-643

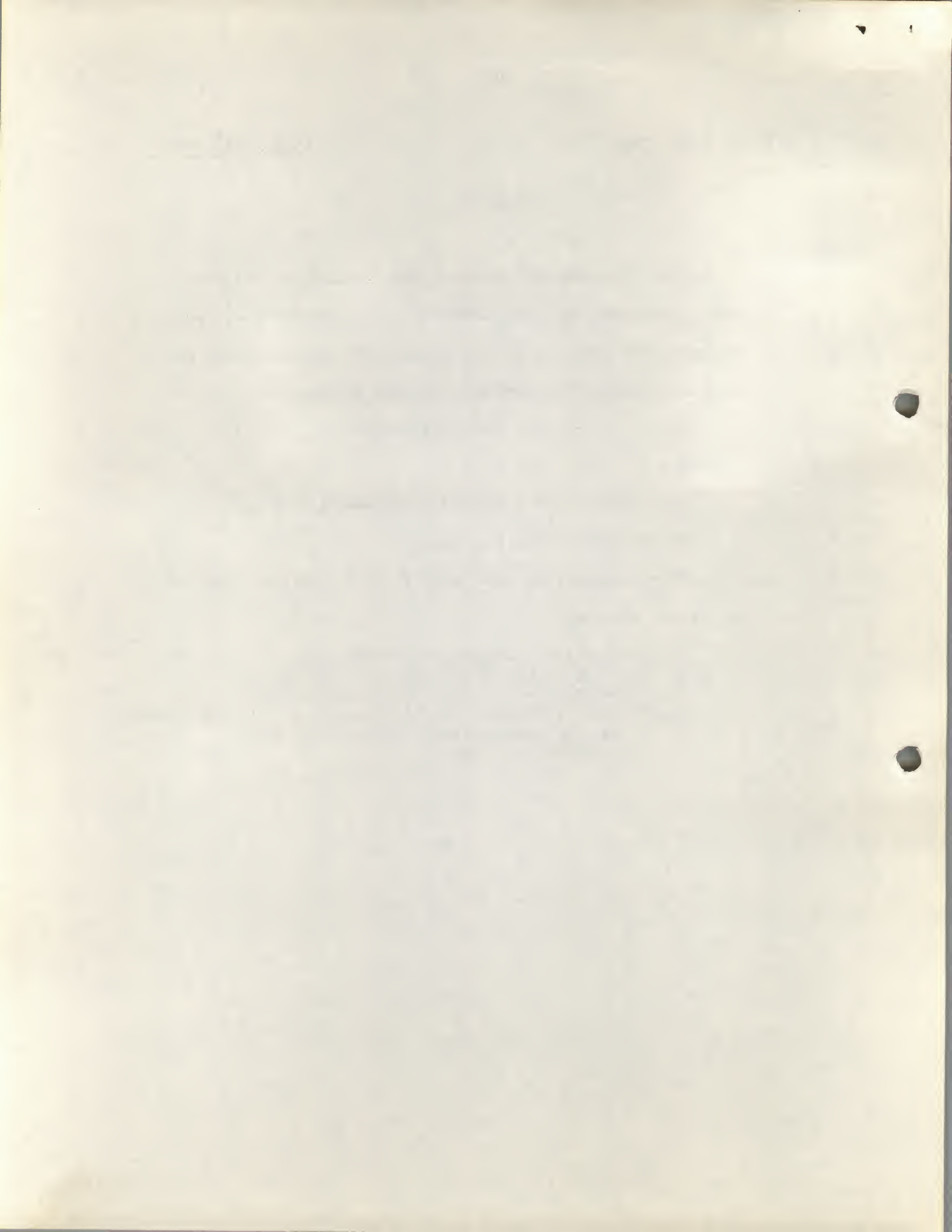
ABSTRACT

A program that executes Conway's game of LIFE as published in several Scientific American articles, under Mathematical Games. The universe consists of a 32 x 32 matrix. Births and deaths are computed according to the number of nearest neighbors:

- 0,1 -- cell dies of loneliness
- 2,3,4 -- cell lives
- 5,6,7,8 -- cell dies of overcrowding
- Exactly 3 -- a cell is born

Printout of each successive generation is to a teletype. Features of the program include:

- 1) Only desired generations are output
- 2) Checking to see if universe has perished
- 3) Checking to see if universe has stabilized
- 4) Warning if boundary may be exceeded in future generation
- 5) Allows the user to start over or alter the universe at any time.



This program is based on articles by Conway in several "Scientific American" articles under the section Mathematical Games. The program took this idea and put it in a computer environment.

DESCRIPTION

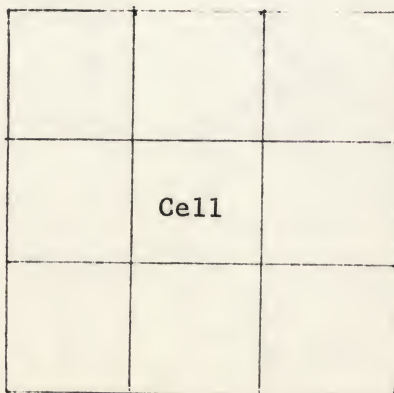
The game consists of a universe, a 32 x 32 array. The size of the universe was limited by core availability (8K - 12 bit words). Each point in the universe is dead or alive, designated by 0 or 1 respectively. An initial pattern of live points, or cells, is input. This is GENERATION number 0. GENERATION 1 and succeeding generations are computed by the following algorithm:

If a live cell has

- 0,1 nearest neighbors - it dies of loneliness in the next generation
- 2,3,4 nearest neighbors - it lives (remains "live" in the next generation)
- 5,6,7,8 nearest neighbors - it dies of loneliness (becomes dead) in the next generation

If a dead cell has

- Exactly 3 nearest neighbors - it is born in the next generation (it becomes a "live" cell)



nearest neighbors of a cell under consideration for birth or death in the next generation as described above

Figure 1

[Faint, illegible text covering the page]



For example:

1	0	1
0	1	0
0	0	0

the circled cell has 2 nearest neighbors,
so it lives into the next generation.

1	0	0
1	1	0
1	1	1

the circled cell in this example dies
of overcrowding.

1	0	0
0	0	1
0	0	1

the circled cell, although dead in this
generation will be born in the next since
it has exactly 3 nearest live neighbors.

This type scanning is performed on all of the 32 x 32 cells in
the array except for the perimeter cells that are always considered
dead since they do not have 8 nearest neighbors.

Each successive generation is computed in this manner. The process
is actually a Markov chain since each successive generation depends
only on the generation immediately preceding it. Each generation
may be output to the teletype along with a count of live cells.

PROGRAM REQUIREMENTS

PDP8/E with minimum 8K

8K Fortran, Sabr, LIB8(Fortran library), and Linking Loader
- all included in OS/8

COMPILING AND LOADING PROCEDURE (with OS/8)

.R FORT

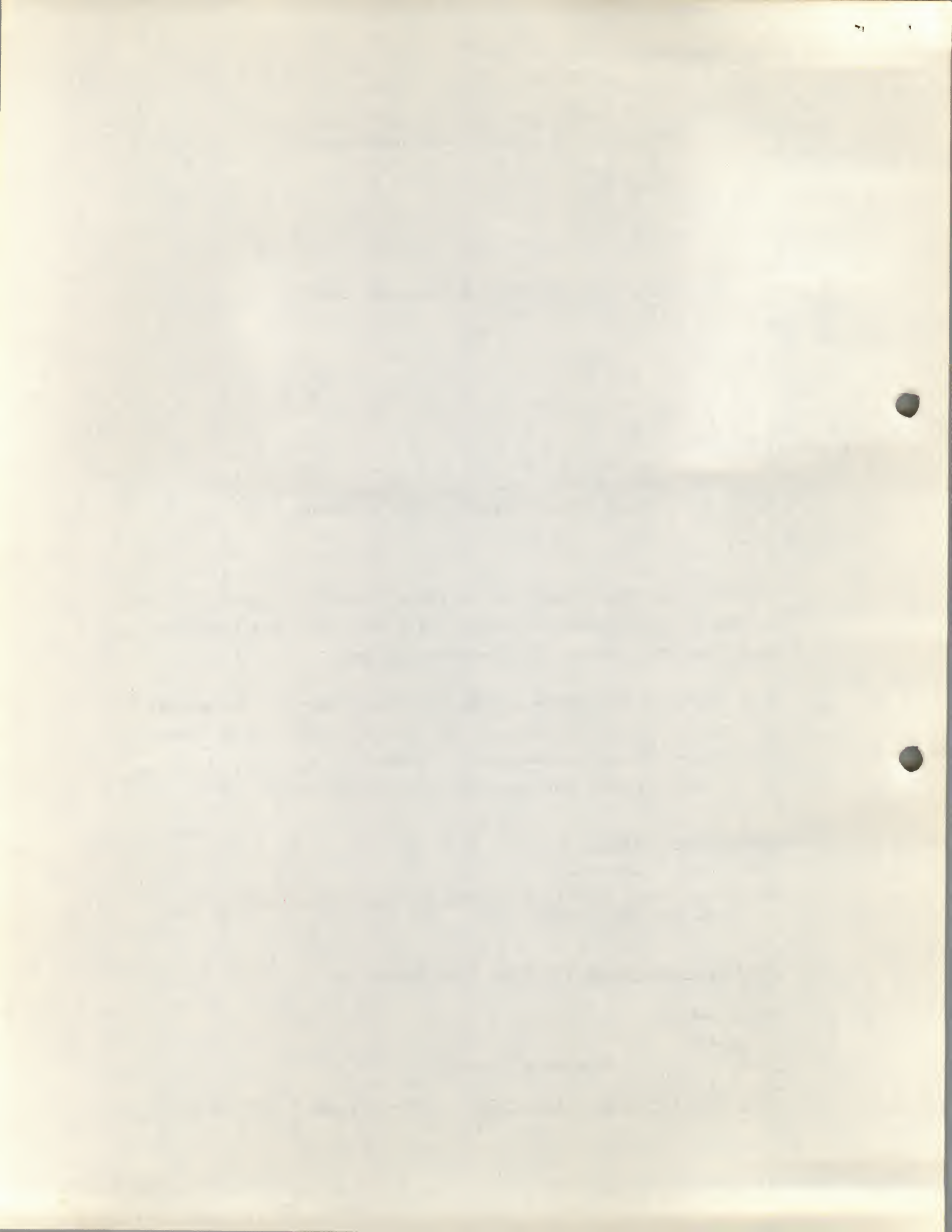
*LIFE.BN←LIFE

.R LOADER

*LIFE\$ (\$ denotes altmode)

.ST

INPUT DEAD/LIVE DESIGNATION CHARS (the program is now running)



BASIC OPERATION

The first request by the program is

"INPUT DEAD/LIVE DESIGNATION CHARS"

This is asking for 2 characters to be typed by the user, not separated by a blank. The first character is that which will be typed to represent a dead cell and the second character is that which will be typed to represent a live cell. A good choice is "space" for dead cells and an "*" for live cells. Hence, one would type " *".

The second request is

"ENTER COORDINATES OF LIVE CELLS"

The coordinates of the 32 x 32 universe are numbered top to bottom and left to right.

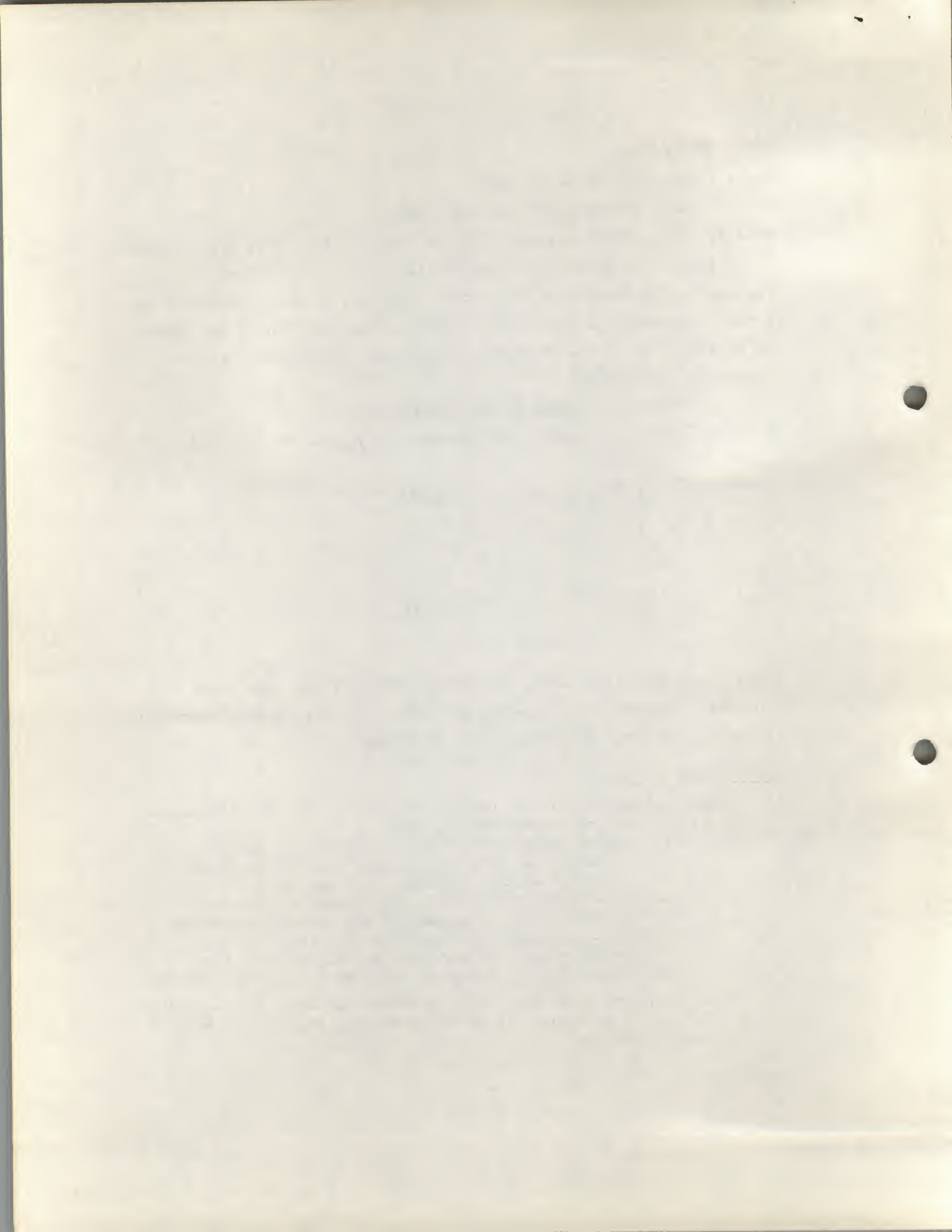
```
1,1 1,2 1,3 ... 1,32
2,1 2,2 . . .
3,1
.
.
.
32,1 . . . 32,32
```

Figure 2

These coordinates are input via fortran FORMAT(2I2). The input of a negative coordinate terminates the input. If a perimeter coordinate is typed, the program responds "OUT OF BOUNDS".

FEATURES AND OPTIONS

- 1) Switch Register bit 11 is used by the program to control output:
bit 11 (0) -- each generation is output
bit 11 (1) -- program responds "NEXT GENERATION TO OUTPUT"
The user at this point types the number of the next generation desired to be output.
If the user types "0", then the program starts over.
If the user types the number of the present generation then he may alter the existing universe.
Otherwise, no generations will be output until the generation corresponding to the number typed is reached.
- 2) The program checks to see if the universe has died (ie. population of live cells is zero). If it has, the user is notified and the program starts over.



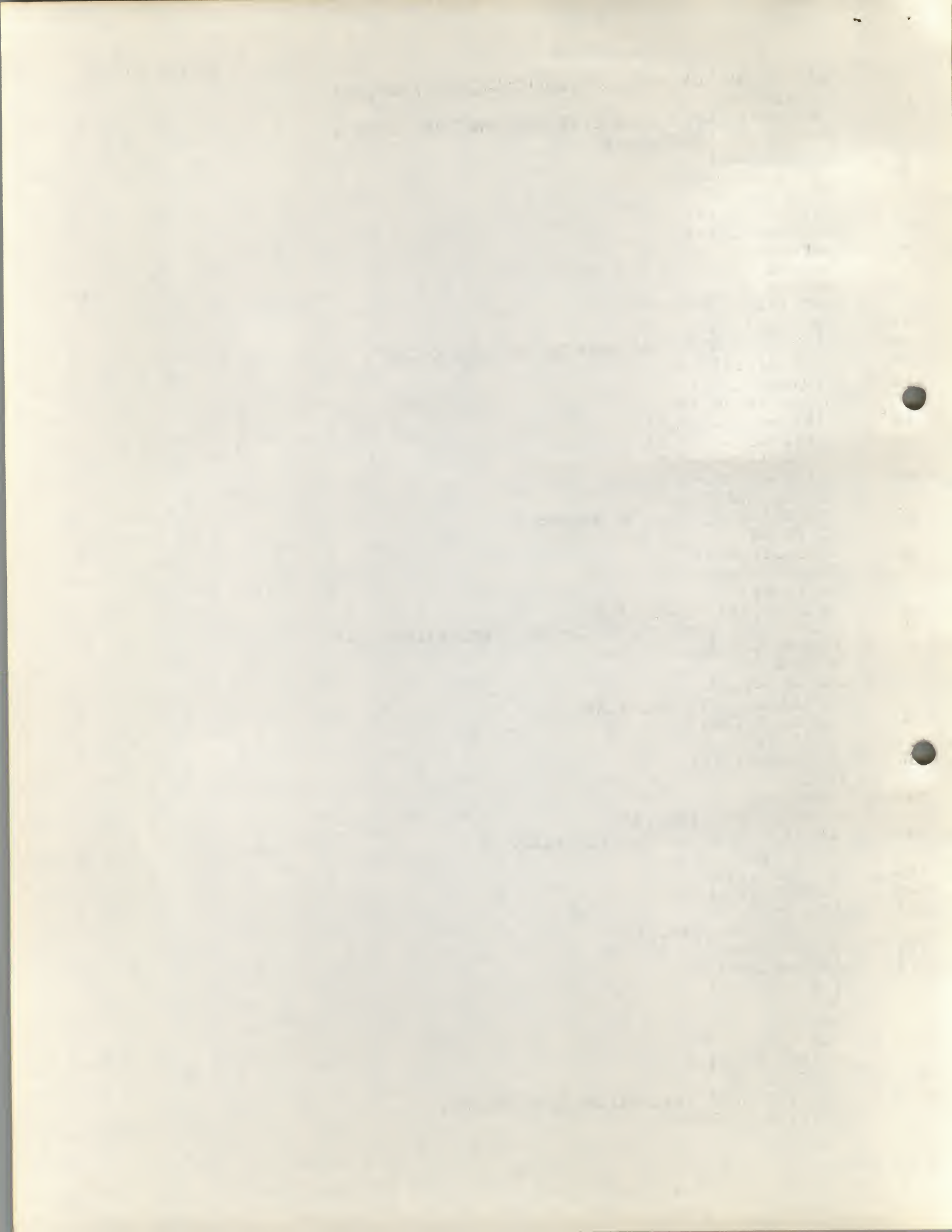
- 3) The program checks to see if the universe has stabilized (ie. no births or deaths will occur in the future generations). If this is true, the user is notified. If bit 11 is set, the message "NEXT GENERATION TO OUTPUT" will be typed and the user may take appropriate **action**.
- 4) The program checks to see if the live cells may grow off the edge of the 32 x 32 universe in the next generation. If it may, the program notifies the user. If bit 11 is set to 1, then the universe is output. Otherwise it continues computing generations.

Refer to Scientific American for maximum enjoyment of this program.

An example run follows.

28/JUN/73

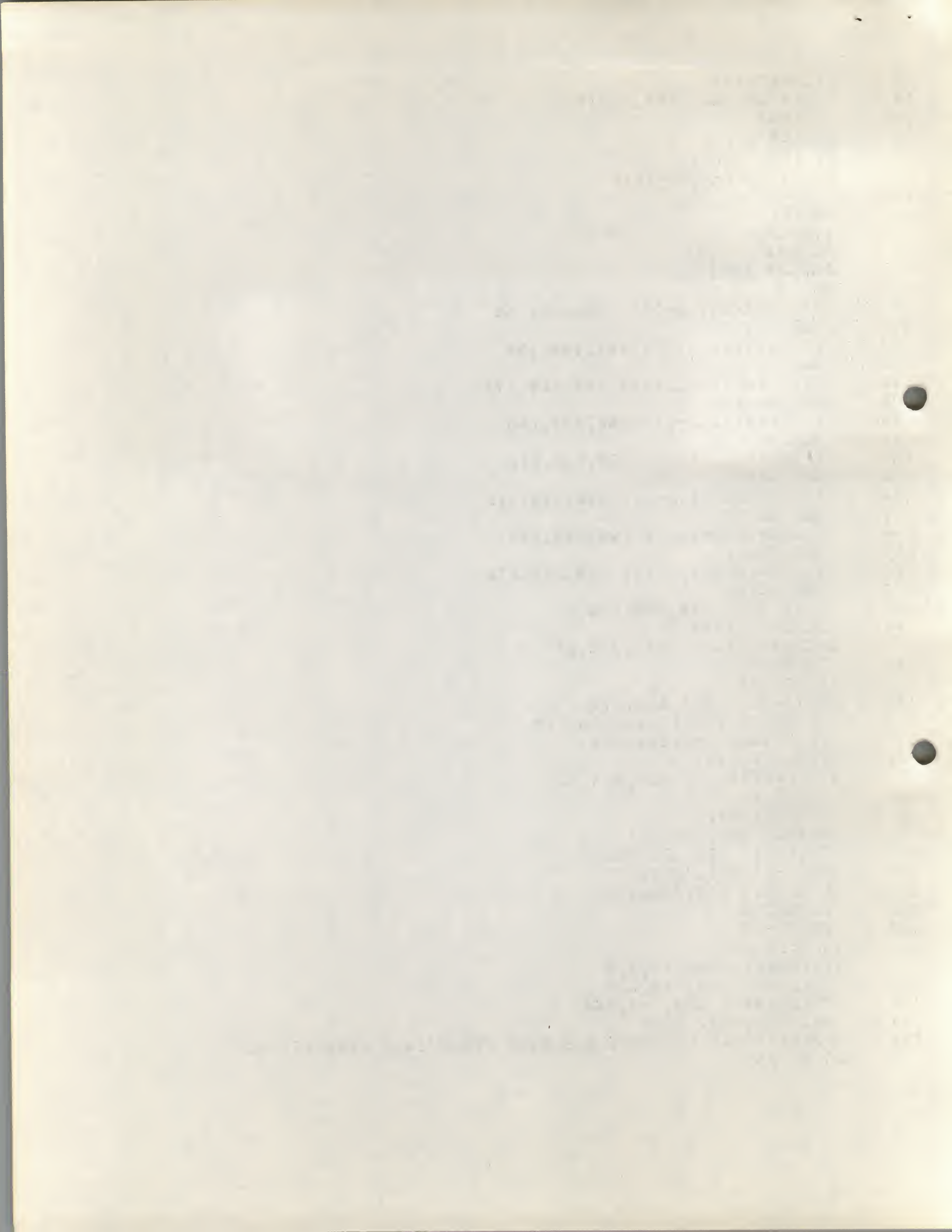
```
DIMENSION ICELL(32,32),ISAVE(32,32),IROW(32)
2 WRITE(1,2)
  FORMAT('INPUT DEAD/LIVE DESIGNATION CWARS')
5 READ(1,5) IDEAD,ILIVE
  FORMAT(2A1)
99 DO 1 J=1,32
  DO 1 I=1,32
  ISAVE(I,J)=0
  ICELL(I,J)=0
  IPOP=0
  IGEN=0
  NGEN=0
  INERT=0
7 WRITE(1,10)
10 FORMAT('ENTER COORDINATES OF LIVE CELLS')
20 READ(1,30)I,J
30 FORMAT(2I2)
  IF(I) 50,40,40
40 IF(I-1) 44,44,41
41 IF(I-32)42,44,44
42 IF(J-1) 44,44,43
43 IF(J-32)49,44,44
44 WRITE(1,45)I,J
45 FORMAT(2I2,' OUT OF BOUNDS')
  GO TO 20
49 ICELL(I,J)=1
  IPOP=IPOP+1
  GO TO 20
50 WRITE(1,60) IGEN,IPOP
60 FORMAT(//'GENERATION 'I4,5X,'POPULATION 'I4)
  DO 100 I=1,32
  NONE=0
  DO 90 J=1,32
  IF(ICELL(I,J)) 80,70,80
70 IROW(J)=IDEAD
  GO TO 90
  IROW(J)=ILIVE
  NONE=1
90 CONTINUE
  IF(NONE) 107,108,107
107 WRITE(1,110) (IROW(J),J=1,32)
  GO TO 100
108 WRITE(1,109)
109 FORMAT(1X)
100 CONTINUE
  IF(INERT) 51,105,51
105 NGEN=IGEN+1
110 FORMAT(32A1)
S CLA CLL
S 7604
S AND (I
S DCA \IND
  IF(IND) 51,59,51
51 WRITE(1,52)
52 FORMAT('NEXT GENERATION TO OUTPUT?')
  READ(1,53) NGEN
```



```

53     FORMAT(I4)
59     IF(NGEN-IGEN) 99,7,115
115    IPOP=0
      DO 120 J=2,31
      DO 120 I=2,31
      ISAVE(I,J)=ICELL(I,J)
120    ICELL(I,J)=0
      IER601=1
      INERT=1
      DO 330 J=2,31
      DO 330 I=2,31
      NBR=0
      IF(ISAVE(I-1,J-1)) 140,140,130
130    NBR=NBR+1
140    IF(ISAVE(I-1,J)) 160,160,150
150    NBR=NBR+1
160    IF(ISAVE(I-1,J+1)) 180,180,170
170    NBR=NBR+1
180    IF(ISAVE(I,J-1)) 200,200,190
190    NBR=NBR+1
200    IF(ISAVE(I,J+1)) 220,220,210
210    NBR=NBR+1
220    IF(ISAVE(I+1,J-1)) 240,240,230
230    NBR=NBR+1
240    IF(ISAVE(I+1,J)) 260,260,250
250    NBR=NBR+1
260    IF(ISAVE(I+1,J+1)) 280,280,270
270    NBR=NBR+1
280    IF(2-NBR) 300,300,290
290    ICELL(I,J)=0
      IF(ISAVE(I,J)) 295,330,295
295    INERT=0
      GO TO 330
300    IF(3-NBR) 310,320,310
310    IF(ISAVE(I,J)) 315,290,315
315    IF(5-NBR) 290,290,320
320    ICELL(I,J)=1
      IF(ISAVE(I,J)) 325,324,325
324    INERT=0
325    IPOP=IPOP+1
      IF(I-2) 321,328,321
321    IF(I-31) 322,328,322
322    IF(J-2) 323,328,323
323    IF(J-31) 330,328,330
328    IER601=0
330    CONTINUE
      IGEN=IGEN+1
      IF(INERT) 390,334,390
334    IF(IPOP) 335,340,335
335    IF(IER601) 380,360,380
340    WRITE(1,350) IGEN
350    FORMAT('THE UNIVERSE HAS DIED AFTER 'I4,' GENERATIONS ')
      GO TO 99

```



```
368 WRITE(1,375) IGEN
378 FORMAT('*** WARNING--BOUNDARY MAY BE EXCEEDED - GENERATION 'I4)
S CLA CLL
S 7684
S AND (I
S DCA \IND
IF(IND) 375,388,375
375 NGEN=IGEN
388 IF(NGEN-IGEN) 115,58,115
398 WRITE(1,400) IGEN
400 FORMAT('THE UNIVERSE WAS STABILIZED-GENERATION'I4)
GO TO 58
CALL EXIT
STOP
END
```

THE UNIVERSITY OF CHICAGO

PHYSICS DEPARTMENT



1C
.R FORT
*LIFE.BN<LIFE

.R LOADER
*LIFES
.ST

INPUT DEAD/LIVE DESIGNATION CHARS

*
ENTER COORDINATES OF LIVE CELLS

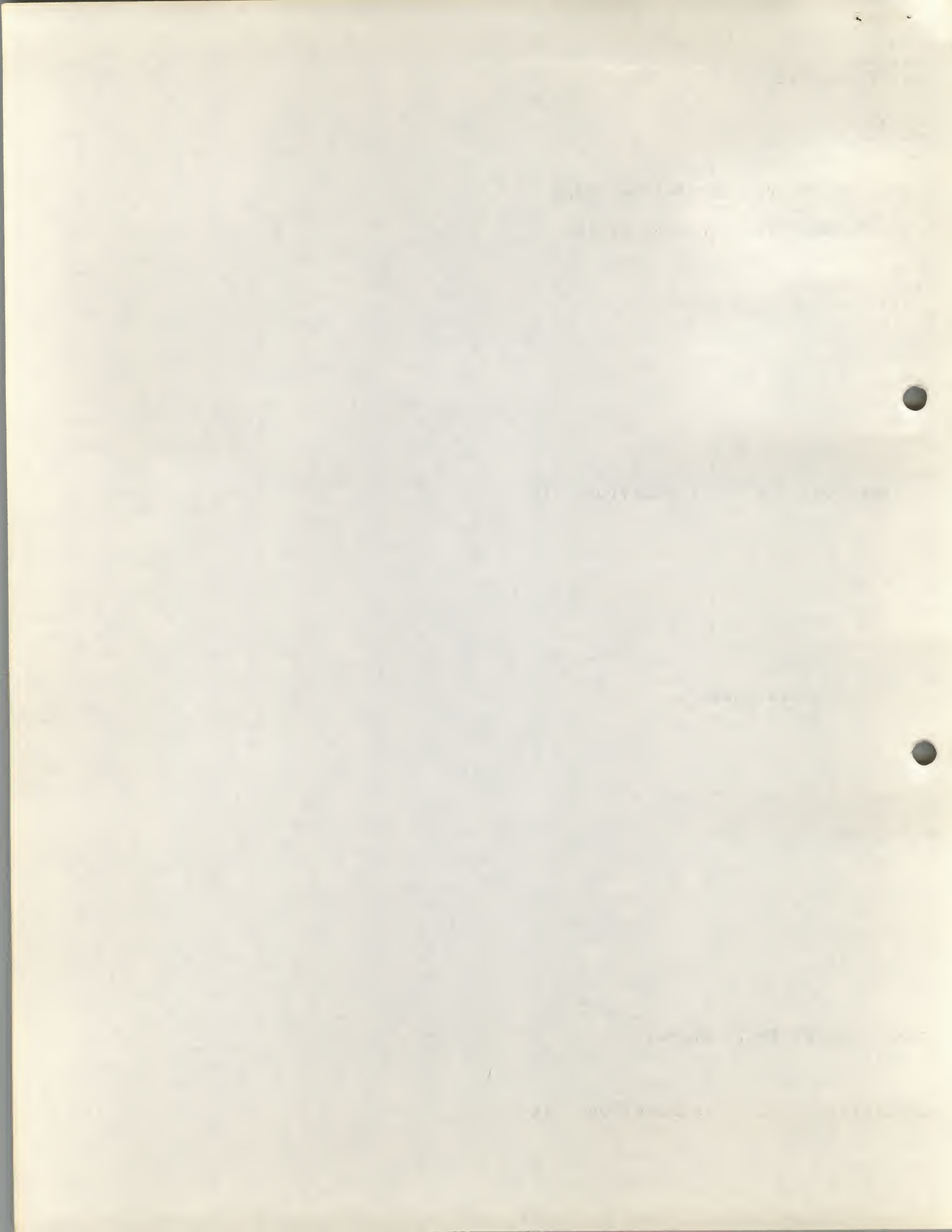
1510
1511
1512
1513
1514
1516
1517
1518
1519
1520
-1

GENERATION 0 POPULATION 10

NEXT GENERATION TO OUTPUT?

2

GENERATION 2 POPULATION 16





NEXT GENERATION TO OUTPUT?

0

ENTER COORDINATES OF LIVE CELLS

1515

1516

-1

GENERATION 0 POPULATION 2

**

THE UNIVERSE HAS DIED AFTER 1 GENERATIONS
ENTER COORDINATES OF LIVE CELLS

