
Simulating the Game of life

X13387_en

GRAU-PRO1, FIB (2013-12-16)

The English mathematician John Conway invented in 1970 the so called "Game of life" as follows: Imagine a square matrix with n rows and n columns with free positions and positions occupied by a bacterium. The game considers the evolution of bacteria with local rules.

From a matrix position, there are considered adjacent positions the (at most eight) positions found at distance one either horizontally, vertically or diagonally. At every moment, every position of the matrix is either empty or contains a bacterium. The rules of the game are the following:

- An empty position in an instant t contains a bacterium at time $t+1$ if and only if at time t it has exactly three adjacent bacteria.
- An occupied position in an instant t contains a bacterium at time $t+1$ if and only if at time t it has two or more adjacent bacteria.

We want to make a sequential simulation of the evolution of the game in such a way that, at every moment, only one position is updated. The sequence of positions follows a zig-zag traversal by columns, starting at the upper left position, $(0,0)$, and walking the first column downwards $((0,0)$ to $(n-1,0))$, the second column upwards $((n-1,1)$ to $(0,1))$, and so on.

We call a *round* a complete traversal of all the positions in the matrix, following the zig-zag traversal by columns.

Write a program that, for each given matrix and number of rounds, computes the matrix of the game after the stipulated number of rounds.

Note that if after a certain round the matrix has not changed, it will never change again.

Input

The input is formed by two natural numbers $n > 0$, the size of the matrix, and $r > 0$, the number of rounds. Followed by a list of positions, pairs of integers between 0 and $n-1$, indicating the positions of the bacteria .

Output

Write, for each initial configuration, which is the initial matrix of the game and all the intermediate matrices (after each simulation round), with an empty line after each matrix. Follow the format of the examples .

Observation

Note that, if the matrix after a simulation round has not changed, you should not write this matrix more than twice.

Sample input 1

```
4 5
0 0 0 1 0 2 0 3
1 0 1 1 1 2 1 3
2 0 2 1 2 2 2 3
3 0 3 1 3 2 3 3
```

Sample output 1

```
BBBB
BBBB
BBBB
BBBB

BBBB
BBBB
BBBB
BBBB
```

Sample input 2

```
3 1
0 0 0 1 1 1 1 2 0 2 0 2 1 2 2 2
```

Sample output 2

```
BBB
.BB
..B

BBB
BBB
..B
```

Sample input 3

```
3 2
0 0 0 2 1 1 2 0
```

Sample output 3

```
B.B
.B.
B..

...
...
...

...
...
...
```

Sample input 4

```
3 1
0 2
1 2
2 1 2 2
```

Sample output 4

```
..B
..B
.BB

...
..B
.BB
```

Sample input 5

```
3 3
0 2
1 2
2 1 2 2
```

Sample output 5

```
..B
..B
.BB

...
..B
.BB

...
```

```
.BB
.BB
...
```

Sample input 6

```
10 10
0 0 0 2 0 4 0 6 0 8
1 1 1 3 1 5 1 7 1 9
2 0 2 2 2 4 2 6 2 8
3 1 3 3 3 5 3 7 3 9
4 0 4 2 4 4 4 6 4 8
5 1 5 3 5 5 5 7 5 9
6 0 6 2 6 4 6 6 6 8
7 1 7 3 7 5 7 7 7 9
8 0 8 2 8 4 8 6 8 8
9 1 9 3 9 5 9 7 9 9
```

```
.BB
.BB
```

Sample output 6

```
B.B.B.B.B.
.B.B.B.B.B
B.B.B.B.B.
.B.B.B.B.B
B.B.B.B.B.
.B.B.B.B.B
B.B.B.B.B.
.B.B.B.B.B
B.B.B.B.B.
.B.B.B.B.B
.B.B.B.B.B
.B.B.B.B.B

..BBBBBBB.
.B.B.B.B.B
B.B.B.B.BB
BB.B.B.B.B
B.B.B.B.BB
BB.B.B.B.B
B.B.B.B.BB
BB.B.B.B.B
B.B.B.B.BB
BB.B.B.B.B
B.B.B.B.B.
.BBBBBBBBB

..BBBBBBB.
.B.B.B.B.B
B.B.B.B.BB
BB.B.B.B.B
B.B.B.B.BB
BB.B.B.B.B
B.B.B.B.BB
BB.B.B.B.B
B.B.B.B.B.
.BBBBBBBBB
```

Problem information

Author : Maria J. Blesa i Maria J. Serna
Translator : Maria Serna
Generation : 2013-12-17 16:47:17

© *Jutge.org*, 2006–2013.
<http://www.jutge.org>