

Project Running Example

Game of life

User will enter following data:

2 ← number of generations that you will need to simulate

5 ← number of cells in the object about to be read

2 3 ← (x=2, y=3) coordinates for the various cells of the object

3 1

3 3

4 2

Color scheme followed in this pdf

1. Blue cells are live cells
2. Green & white cells are neighbors' cell (dead cells)

Grid

Size: 10x10

0,0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9
1,0	1,1	1,2	1,3	1,4	1,5	1,6	2,7	1,8	1,9
2,0	2,1	2,2	2,3	2,4	2,5	2,6	3,7	2,8	2,9
3,0	3,1	3,2	3,3	3,4	3,5	3,6	3,7	3,8	3,9
4,0	4,1	4,2	4,3	4,4	4,5	4,6	4,7	4,8	4,9
5,0	5,1	5,2	5,3	5,4	5,5	5,6	5,7	5,8	5,9
6,0	6,1	6,2	6,3	6,4	6,5	6,6	6,7	6,8	6,9
7,0	7,1	7,2	7,3	7,4	7,5	7,6	7,7	7,8	7,9
8,0	8,1	8,2	8,3	8,4	8,5	8,6	8,7	8,8	8,9
9,0	9,1	9,2	9,3	9,4	9,5	9,6	9,7	9,8	9,9

Step 03

Count Neighbors (live cell) for sec - array and nei - array.
Remember live cells can be find using sec - array.

Grid (Generation 0).

Sec - array Last = 4

Count
→

2,3	3,1	3,3	4,2	4,3					
1	1	3	3	2					

nei - array

Last = 16

Count
→

1,2	1,3	1,4	2,2	2,4	3,2	3,4	2,0	2,1	3,0	4,0	4,1	4,4	5,1	5,2	5,3	5,4			
1	1	1	3	2	5	3	1	1	1	1	2	2	1	2	2	1			

Step 04

1. Now Apply Rules

- a. cells (2,3) & (3,1) die because they have less than 2 live neighbors.

Rule: 3

so, these cells will be deleted from sec - array

- b. Cells (3,3), (4,2) & (4,3) remain alive because they have 2 or 3 live neighbors.

Rule: 2

- c. Cells (2,2) & (3,4) will become alive because they have exactly 3 live neighbors.

Rule: 1

This will be removed from nei - array and inserted in sec - array.

2. Remove All element from nei - array.

3. And update the Grid for Generation 1. Display

the output. Grid (Generation 1).

Sec - array

Last = 4

3,3	4,2	4,3	2,2	3,4					
-----	-----	-----	-----	-----	--	--	--	--	--

nei - array

Last = -1

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Step 06

Count Neighbors (live cell) for sec - array and nei - array.
Remember live cells can be find using sec - array.

Grid (Generation 1).

Sec - array

Last = 4

Count
→

3,3	4,2	4,3	2,2	3,4					
4	2	3	1	2					

nei - array

Last = 16

Count
→

2,3	2,4	3,2	4,4	3,1	4,1	5,1	5,2	5,3	5,4	1,1	1,2	1,3	2,1	2,5	3,5	4,5			
3	2	4	3	2	1	1	2	2	1	1	1	1	1	1	1	1			

Step 07

1. Now Apply Rules

- cell (3,3) will die because of overcrowding. **Rule: 4**
so, cell will be deleted from sec - array
- Cells (4,2), (4,3) & (3,4) remain alive because they have 2 or 3 live neighbors.

Rule: 2

- cell (2,2) die because it has less than 2 live neighbors. **Rule: 3**
so, this cell will also be deleted from sec - array
- Cells (2,3) & (4,4) will become alive because they have exactly 3 live neighbors.

Rule: 1

This will be removed from nei - array and inserted in sec - array.

2. Remove All element from nei - array.

3. And update the Grid for Generation 2. Display the output.

Grid (Generation 2).

Sec - array

Last = 4

4,2	4,3	3,4	2,3	4,4					
-----	-----	-----	-----	-----	--	--	--	--	--

nei - array

Last = -1

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Step 8

Display all live cells coordinates on terminal:

2 ← number of generations

5 ← number of live cells

4 2 ← (x, y) coordinates for the various cells of the object

4 3

3 4

2 3

4 4