CONWAY’S GAME OF LIFE

This project is basically about the 'survival and death of the cells' , depending upon the several conditions provided.
And this game is all about the formation of dead cells (or) the destruction of live cells which continuously occur depending upon the conditions of a particular live cell or dead cell.

The **Game of Life**, also known simply as **Life**, is a cellular automation devised by the British mathematician John Horton Conway in 1970.

The "game" is actually a zero-player game, meaning that its evolution is determined by its initial state, needing no input from human players . One interacts with the Game of Life by creating an initial configuration and observing how it evolves.

RULES:

A cell **C** is represented by a **1** when alive, or **0** when dead(in this introduction), in An m x m square array of cells.

We calculate **N** - the sum of live cells in C's eight location neighbourhood then cell C is alive or dead in the next generation based on the following table:

| **C** | **N** | **new C** |
|---|---|---|
| 1 | 0,1 | -> 0 # Lonely |
| 1 | 4,5,6,7,8 | -> 0 # Overcrowded |
| 1 | 2,3 | -> 1 # Lives |
| 0 | 3 | -> 1 # It takes three to give birth! |
| 0 | 0,1,2,4,5,6,7,8 | -> 0 # Barren |

Assume cells beyond the boundary are always dead

1.If a live cell is lonely (see the above chart) it dies.

2.If  a live cell has two or three live neighbours ,it lives on to the next generation.

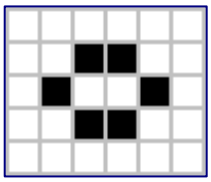
3.If a live cell is overcrowded (see the above chart) it dies.

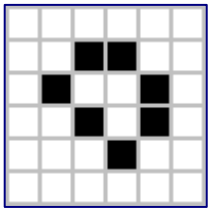
4. If a  dead cell has exactly three live neighbours  it becomes a living cell else it will be a dead cell.

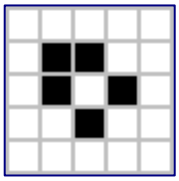
PATTERNS

Patterns of living cells...

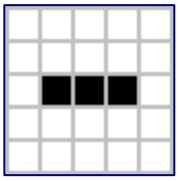
Block 

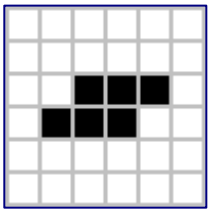
Beehive 

Loaf 

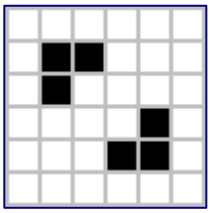
Boat 

Examples of some oscillating patterns that never stops.

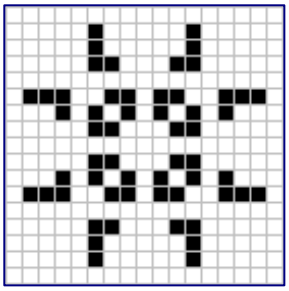
Blinker (period 2) 

Toad (period 2) 

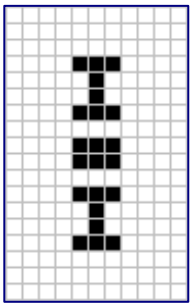
Beacon (period 2)



Pulsar (period 3)



Pentadecathlon
(period 15)



BASIC IDEA

-->In our project initially the pattern is given through coordinates(by using matrices) and that pattern goes through a series of changes.

-->Every generation(i.e the new pattern) is shown by the program till it comes to its end point.

--> It folows the above game rules as mentioned above and changes its pattern.

-->Initialy we will create a two dimensional array to deal with changes in new generations.

-->Now coming to the code ....

>>>We will start with a menu function .... which contains generally start and exit.

-->Next we will sequentially create some functions which implement the following:

>>> Takes input pattern from the user.

>>> That counts the no of living cells around a particular live cell or a dead cell.

>>> That checks neighbour value(i.e a live cell or a dead cell).

>>> That calculates above information based on the conditions(i.e provided in the form of a table.)

>>> Process the immediate pattern(i.e next pattern).

>>> That displays the generations of cells(i.e a new formation).

>>> Finally we will create some function that quits the program.

Note:

> We will try to implement a function that displays the new generations in a regular time interval which delays the occurances of the new generations.

> We will also try to generate patterns where the input is given by the machine itself.

GROUP MEMBERS:

Y D YASASVI IMT2016060
S SHASHANK IMT2016058
Y SRI SUMANTH IMT2016081
T TUSHAR IMT2016059
P RISHI VARMA IMT2016061