# Game of Life

using System;

namespace GameofLife

{

class Program

{

public static void Main()

{

int R = 10, C = 10;

// Intiliazing the board.

int[,] board = {

{ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 },

{ 0, 0, 0, 1, 1, 0, 0, 0, 0, 0 },

{ 0, 0, 0, 0, 1, 0, 0, 0, 0, 0 },

{ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 },

{ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 },

{ 0, 0, 0, 1, 1, 0, 0, 0, 0, 0 },

{ 0, 0, 1, 1, 0, 0, 0, 0, 0, 0 },

{ 0, 0, 0, 0, 0, 1, 0, 0, 0, 0 },

{ 0, 0, 0, 0, 1, 0, 0, 0, 0, 0 },

{ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 }

};

// Displaying the board

Console.WriteLine("Initial Population");

for (int i = 0; i < R; i++)

{

for (int j = 0; j < C; j++)

{

if (board[i, j] == 0)

Console.Write(".");

else

Console.Write("\*");

}

Console.WriteLine();

}

Console.WriteLine();

nextPopulation(board, R, C);

Console.ReadKey();

}

// Method to display the next Generation Population

static void nextPopulation(int[,] board,

int R, int C)

{

int[,] subsequent = new int[R, C];

// Iterate through every cell

for (int l = 1; l < R - 1; l++)

{

for (int m = 1; m < C - 1; m++)

{

// finding no Of Neighbours

// that are living

int livingNeighbours = 0;

for (int i = -1; i <= 1; i++)

for (int j = -1; j <= 1; j++)

livingNeighbours +=

board[l + i, m + j];

// The cell needs to be subtracted

// from its neighbours as it was

// counted before

livingNeighbours -= board[l, m];

// Implementing the Principles of Life

// Cell is lonely and become extinct

if ((board[l, m] == 1) &&

(livingNeighbours < 2))

subsequent[l, m] = 0;

// Cell is extincted due to over population

else if ((board[l, m] == 1) &&

(livingNeighbours > 3))

subsequent[l, m] = 0;

// Cell multiplication

else if ((board[l, m] == 0) &&

(livingNeighbours == 3))

subsequent[l, m] = 1;

// Population stays fixed

else

subsequent[l, m] = board[l, m];

}

}

Console.WriteLine("Next Population");

for (int i = 0; i < R; i++)

{

for (int j = 0; j < C; j++)

{

if (subsequent[i, j] == 0)

Console.Write(".");

else

Console.Write("\*");

}

Console.WriteLine();

}

}

}

}