

**Department of CSE**

**Project Report of**

**Rat In a Maze**

**Course Name:** Data Structure & Algorithm

**Course Code :** CSE 207

**Section No :** **07**

**Group No : 05**

**Submitted To : Dr. Md. Manowarul Islam**

**Associate Professor, Department of CSE**

.

**Submitted By:**

|  |  |
| --- | --- |
| **Student Name** | **ID** |
| **Sheikh Sarafat Hossain** | **2022-3-60-109** |
| **Rijia Parveen Raya** | **2022-3-60-192** |
| **Sayma Sultana** | **2021-3-60-105** |
| **Prottoy Debnath** | **2020-3-60-080** |

**Contribution Table**

|  |  |  |
| --- | --- | --- |
| **Name** | **ID** | **Contribution Details** |
| Sheikh Sarafat Hossain | 2022-3-60-109 | 1. Project Concept making & Research. 2. Preparing Project and doing programming. 3. Report Writing. 4. PowerPoint slide making |
| Rijia Parveen Raya | 2022-3-60-192 | 1. Project Concept idea generation. 2. Preparing Project and doing programming. 3. Report Writing. 4. PowerPoint slide making. |
| Sayma Sultana | 2021-3-60-105 | 1. Project Concept making & Research. 2. Doing basic programing in solving. 3. Preparing Project and group handling 4. Report Writing. |
| Prottoy Debnath | 2020-3-60-080 | 1. Project Concept making & Research. 2. Doing programing in error solving 3. Report Writing. 4. PowerPoint slide making. |

**Project Procedure & Features**

**App Used:**

Code blocks

**Objectives:**

to navigate a maze that is created at random and is constructed using a matrix with elements of 0 and 1. where 0 denotes a closed path and 1 denotes an open path. Once the maze has been solved, print the waypoint.

**Theory:**

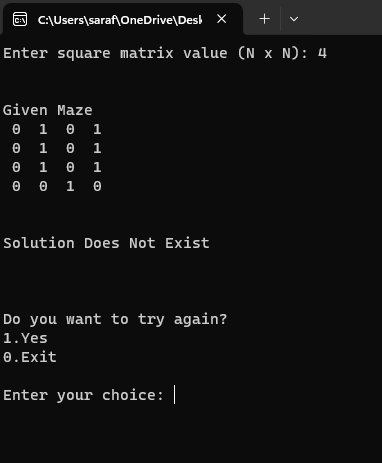
We have used matrix size of N\*N and random function to generate the matrix randomly. Then we have used back-tracking to check each element of the matrix to ensure whether the rat can move in that direction or not.   
  
Uses of recursion was used here to determine the way point. After doing and finding out the way the solution is printed along with the generated matrix.

**Limitation:**

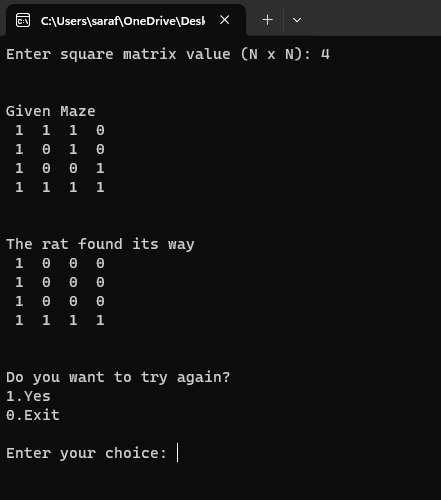
1. The rat can not run on the back or upper direction.
2. Need to generate multiple mazes based on the given size to reach the solution.

**Output:**

1. **No solution**



1. **With Solution**



**Source Code:**

**#include <iostream>**

**#include <random>**

**#include <ctime>**

**using namespace std;**

**bool isSafe(int x, int y, int N, int maze[][20])**

**{**

**return (x >= 0 && x < N && y >= 0 && y < N && maze[x][y] == 1);**

**}**

**bool ratInMaze(int N, int x, int y, int maze[][20], int sol[][20])**

**{**

**if (isSafe(x, y, N, maze))**

**{**

**if (x == N - 1 && y == N - 1)**

**{**

**sol[x][y] = 1;**

**return true;**

**}**

**sol[x][y] = 1;**

**if (ratInMaze(N, x + 1, y, maze, sol))**

**return true;**

**if (ratInMaze(N, x, y + 1, maze, sol))**

**return true;**

**sol[x][y] = 0;**

**return false;**

**}**

**return false;**

**}**

**void printSolution(int N, int sol[][20])**

**{**

**for (int i = 0; i < N; i++)**

**{**

**for (int j = 0; j < N; j++)**

**cout << " " << sol[i][j] << " ";**

**cout << endl;**

**}**

**}**

**int main()**

**{**

**srand(time(NULL));**

**int N, choice;**

**do**

**{**

**cout << "Enter square matrix value (N x N): ";**

**cin >> N;**

**int givenMaze[20][20];**

**int solMaze[20][20];**

**for (int i = 0; i < N; i++)**

**{**

**for (int j = 0; j < N; j++)**

**{**

**solMaze[i][j] = 0;**

**}**

**}**

**for (int i = 0; i < N; i++)**

**{**

**for (int j = 0; j < N; j++)**

**{**

**givenMaze[i][j] = rand() % 2; // 0 or 1**

**}**

**}**

**if (ratInMaze(N, 0, 0, givenMaze, solMaze) == false)**

**{**

**cout << endl << endl << "Given Maze" << endl;**

**for (int i = 0; i < N; i++)**

**{**

**for (int j = 0; j < N; j++)**

**cout << " " << givenMaze[i][j] << " ";**

**cout << endl;**

**}**

**cout << endl << endl << "Solution Does Not Exist" << endl << endl;**

**cout << endl << endl << "Do you want to try again?" << endl;**

**cout << "1.Yes" << endl;**

**cout << "0.Exit" << endl << endl;**

**cout << "Enter your choice: ";**

**cin >> choice;**

**if (choice == 1)**

**{**

**system("cls"); // Clear screen (for Windows)**

**}**

**}**

**else**

**{**

**cout << endl << endl << "Given Maze" << endl;**

**for (int i = 0; i < N; i++)**

**{**

**for (int j = 0; j < N; j++)**

**cout << " " << givenMaze[i][j] << " ";**

**cout << endl;**

**}**

**cout << endl << endl << "The rat found its way" << endl;**

**printSolution(N, solMaze);**

**cout << endl << endl << "Do you want to try again?" << endl;**

**cout << "1.Yes" << endl;**

**cout << "0.Exit" << endl << endl;**

**cout << "Enter your choice: ";**

**cin >> choice;**

**if (choice == 1)**

**{**

**system("cls"); // Clear screen (for Windows)**

**}**

**}**

**}**

**while (choice != 0);**

**return 0;**

**}**