```
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
#include <time.h>


#define N 50


int grid[N][N];
bool visited[N][N];
int dx[] = {-1, 1, 0, 0}; // gore, dole, levo, desno
int dy[] = {0, 0, -1, 1};
bool found = false;


void dfs(int x, int y) {
  if (x < 0 || y < 0 || x >= N || y >= N || visited[x][y] || grid[x][y] == 1)
      return;


  visited[x][y] = true;


  if (grid[x][y] == 3) {
      found = true;
      return;
  }


  for (int i = 0; i < 4; i++) {
      dfs(x + dx[i], y + dy[i]);
  }
```

```
}


int main() {
  srand(time(NULL));


  // Inicijalizacija na nule
  for (int i = 0; i < N; i++)
      for (int j = 0; j < N; j++)
          grid[i][j] = 0;


  // Random pozicije za 1, 2, 3
  for (int i = 0; i < 500; i++) {
      int x = rand() % N;
      int y = rand() % N;
      grid[x][y] = 1;
  }


  int x2 = rand() % N;
  int y2 = rand() % N;
  grid[x2][y2] = 2;


  int x3 = rand() % N;
  int y3 = rand() % N;
  grid[x3][y3] = 3;


  // Prikaz table
  for (int i = 0; i < N; i++) {
```

```
        for (int j = 0; j < N; j++)
            printf("%d ", grid[i][j]);
        printf("\n");
    }


    // DFS pretraga
    dfs(x2, y2);


    if (found)
        printf("Put postoji.\n");
    else
        printf("Put NE postoji.\n");


    return 0;
}
```