STUDENT'S NAME : Lipakshi UID : 20BCS5082

CLASS & GROUP : 20BCS_WM_607 - B SEMESTER : 5th

EXPERIMENT - 05 (GRAPHS)

PROBLEM STATEMENT :

Journey to the Moon (on HackerRank)

CODE :

```
#include <bits/stdc++.h>
#define MAX 100000
using namespace std;
list<int> *ad;
int *visited;
int vertices;

void DFS(int u){
   visited[u] = 1;
   vertices++;
   list<int>::iterator it;
   for(it=ad[u].begin();it!=ad[u].end();it++){
       if(visited[*it] == 0){
           visited[*it] = 1;
           DFS(*it);
       }
   }
}

int main(){
    int i,m,u,v,numComponents=0,allv=0,temp=2,count=0;
    long long int n;
    int eachC[MAX];
   cin >> n >> m;
   if(n == 1){
     cout <<"0\n";
     return 0;
    }

    ad = new list<int>[n];
   list<int>::iterator it;
```

```
for(i=0;i<m;i++){
        cin >> u >>
        v;
        ad[u].push_back(v);
        ad[v].push_back(u);
   }
   visited = new int[n];
   for(i=0;i<n;i++){
      visited[i] = 0;
   }

   for(i=0;i<n;i++){
      if(visited[i] == 0){
          vertices = 0;
          DFS(i);
          eachC[numComponents] = vertices;
          numComponents++;
      }
    }

     long long int totalWays = n*(n-1) / 2;
     long long int sameWays = 0;

    for(i=0;i<numComponents;i++){
        sameWays = sameWays + (eachC[i]*(eachC[i]-1) / 2);
                                                    }
                                                    c
                                                    o
                                                } u
                                                    t
                                                    <
                                                    <
                                                    (
                                                    t
                                                    o
                                                    t
                                                    a
                                                    l
                                                    W
                                                    a
                                                    y
                                                    s
                                                    -
                                                    s
                                                    a
                                                    m
                                                    e
                                                    W
                                                    a
                                                    y
                                                    s
                                                    )
                                                    <
                                                    <
                                                    e
                                                    n
                                                    d
                                                    l
                                                    ;
                                                    r
                                                    e
                                                    t
                                                    u
                                                    r
                                                    n
                                                    0
```

OUTPUT :

;

Test case 0

Test case 1

Test case 2

Test case 3

Test case 4

Test case 5

Test case 6

Compiler Message

```
Success
```

Input (stdin) Download

```
5 3
0 1
2 3
0 4
```

Expected Output Download

```
6
```

PROBLEM STATEMENT :

Frog in the Maze (on HackerRank)

CODE :

```
#include <bits/stdc++.h>
#define double long double

using namespace std;
const int MAXN = (42);
const double eps = 1e-12;

vector<double> gauss(vector<vector<double>> &a){
   int n = a.size(), m = a[0].size() - 1;
   vector<int> where(m, -1);
   for (int col = 0, row = 0; col < m && row < n; col++){
      int sel = row;
      for (int i = row; i < n; i++){
         if (abs(a[i][col]) > abs(a[sel][col])){
            sel = i;
         }
      }

      if (abs(a[sel][col]) < eps){
         where[col] = -1;
         continue;
      }

      for (int i = col; i <= m; i++){
         swap(a[sel][i], a[row][i]);
      }
      where[col] = row;
      for (int i = 0; i < n; i++){
         if (i != row){
            if (abs(a[i][col]) < eps)
               continue;
            double c = a[i][col] / a[row][col];
            for (int j = 0; j <= m; j++)
               a[i][j] -= c * a[row][j];
         }
      }
      row++;
 }
vector<double> ans(m, 0);
for (int i = 0; i < m; i++){
   if (where[i] != -1){
      ans[i] = a[where[i]][m] / a[where[i]][i];
   }
}
for (int i = 0; i < n; i++){
   double sum = a[i][m];
   for (int j = 0; j < m; j++){
      sum -= ans[j] * a[i][j];
   }
   if (abs(sum) > eps){
      return vector<double>();
   }
}
return ans;
```

```
}

int n, m, k;
string a[MAXN];
int nxt_x[MAXN][MAXN], nxt_y[MAXN][MAXN];

void read(){
   cin >> n >> m >> k; for
   (int i = 0; i < n; i++)
   {
      cin >> a[i];
   }

   for (int i = 0; i < n; i++){
      for (int j = 0; j < m;
      j++){
         nxt_x[i][j] = i, nxt_y[i][j] = j;
      }
   }

   for (int i = 0; i < k; i++){
      int x1, y1, x2, y2;
      cin >> x1 >> y1 >> x2 >>
      y2; x1--;
      y1--;
      x2--;
      y2--;
      nxt_x[x1][y1]      =
      x2;   nxt_y[x1][y1]
      =                  y2;
      nxt_x[x2][y2]      =
      x1;   nxt_y[x2][y2]
      = y1;
   }
}

int N;

int encode(int x, int y){
   return x * m + y;
}

int dirx[4] = {0, 0, 1, -1};
int diry[4] = {1, -1, 0, 0};

bool ok(int x, int y){
   if (x >= n || y >= m || x < 0 || y < 0){
      return false;
   }
   return a[x][y] != '#';
}
void solve(){
N = n * m;
vector<vector<double>> matr;
vector<double> zero(N + 1, 0);

for (int i = 0; i < n; i++){
      for (int j = 0; j < m; j++){
if (a[i][j] == '#'){
   matr.push_back(zero);
   continue;
}
else if (a[i][j] == '*'){
   matr.push_back(zero), matr[matr.size() - 1][encode(i, j)] = 1;
   continue;
}
else if (a[i][j] == '%'){
```

```
            matr.push_back(zero), matr[matr.size() - 1][encode(i, j)] = 1;
            matr[matr.size() - 1][N] = 1;
            continue;
        }

        vector<int> adj;
        for (int d = 0; d < 4; d++){
            if (ok(i + dirx[d], j + diry[d])){
                adj.push_back(encode(nxt_x[i + dirx[d]][j + diry[d]], nxt_y[i + dirx[d]][j + diry[d]]));
            }
        }

        matr.push_back(zero);
        matr[matr.size() - 1][encode(i, j)] = 1;

        for (int v : adj){
            matr[matr.size() - 1][v] = -((double)1 / (double)adj.size());
        }
    }
  }

  vector<double> ans = gauss(matr);

  for (int i = 0; i < n; i++){
    for (int j = 0; j < m;
    j++){
        if (a[i][j] == 'A'){
            cout << setprecision(9) << fixed << ans[encode(i, j)] << endl;
            return;
        }
    }
  }
}

int main(){
  ios_base::sync_with_stdio(false);
  cin.tie(NULL);

  read();
  solve();
  return 0;
}
```

OUTPUT :