

C 市现在要转移一批罪犯到 D 市，C 市有 n 名罪犯，按照入狱时间有顺序，另外每个罪犯有一个罪行值，值越大罪越重。现在为了方便管理，市长决定转移入狱时间连续的 c 名犯人，同时要求转移犯人的罪行值之和不超过 t ，问有多少种选择的方式？

```
1  import java.util.Scanner;
2  public class Main{
3      public static void main(String[] args){
4          Scanner in=new Scanner(System.in);
5          while(in.hasNext()){
6              int n = in.nextInt();//n 个人
7              int t = in.nextInt();//
8              int c = in.nextInt();//连续 c 名
9              int[] a = new int[n];
10             for(int i = 0;i<n;i++){
11                 a[i] = in.nextInt();
12             }
13             int count = 0;
14             int tempt = 0;
15             for(int i = 0;i<c;i++){
16                 tempt += a[i];
17             }
18             if(tempt<=t)
19                 count++;
20             for(int i = c;i<a.length;i++){
21                 tempt = tempt+a[i]-a[i-c];
22                 if(tempt<=t)
23                     count++;
```



```
24         }
25         System.out.println(count);
26     }
27 }
28 }
```

度度熊有一张网格纸，但是纸上有一些点过的点，每个点都在网格点上，若把网格看成一个坐标轴平行于网格线的坐标系的话，每个点可以用一对整数 x , y 来表示。度度熊必须沿着网格线画一个正方形，使所有点在正方形的内部或者边界。然后把这个正方形剪下来。问剪掉正方形的最小面积是多少。

```
1  import java.util.Scanner;
2  public class Main{
3      public static void main(String[] args){
4          Scanner in = new Scanner(System.in);
5          while(in.hasNext()){
6              int n = in.nextInt();
7              int maxX = Integer.MIN_VALUE;
8              int maxY = Integer.MIN_VALUE;
9              int minX = Integer.MAX_VALUE;
10             int minY = Integer.MAX_VALUE;
11             for(int i = 0;i<n;i++){
12                 int x = in.nextInt();
13                 int y = in.nextInt();
14                 maxX = (int) Math.max(maxX, x);
15                 maxY = (int) Math.max(maxY, y);
16                 minX = (int) Math.min(minX, x);
```



```
17             minY = (int) Math.min(minY, y);
18         }
19         int side = Math.max((maxX-minX), (maxY-minY));
20         System.out.println(side*side);
21     }
22 }
23 }
```

ss 请 cc 来家里钓鱼，鱼塘可划分为 $n * m$ 的格子，每个格子每分钟有不同的概率钓上鱼，cc 一直在坐标(x,y)的格子钓鱼，而 ss 每分钟随机钓一个格子。问 t 分钟后他们谁至少钓到一条鱼的概率大？为多少？

```
1  import java.util.Scanner;
2  public class Main{
3      public static void main(String[] args){
4          Scanner in = new Scanner(System.in);
5          while (in.hasNext()) {//注意 while 处理多个 case
6              String[] s1 = in.nextLine().split(" ");
7              int n = Integer.parseInt(s1[0]);
8              int m = Integer.parseInt(s1[1]);
9              int x = Integer.parseInt(s1[2]);
10             int y = Integer.parseInt(s1[3]);
11             int t = Integer.parseInt(s1[4]);
12             //int n = in.nextInt();
13             //int m = in.nextInt();
14             //int x = in.nextInt();
15             //int y = in.nextInt();
```



```
16         //int t = in.nextInt();
17         double ccp = 0.00;
18         double ssp = 0.00;
19         for(int i = 1;i<=n;i++){
20             String[] s = in.nextLine().split(" ");
21             for(int j = 1;j<=m;j++){
22                 double p = 1-Double.parseDouble(s[j-1]);
23                 //double p = 1-in.nextDouble(); //钓不到鱼的概率
24                 if(i==x&&j==y)
25                     ccp = p;
26                 ssp += p;
27             }
28         }
29         ssp /= (n*m); //期望
30         if(ccp<ssp){
31             System.out.println("cc");
32             System.out.printf("%.2f\n", 1-Math.pow(ccp, t));
33         }else if(ccp>ssp){
34             System.out.println("ss");
35             System.out.printf("%.2f\n", 1-Math.pow(ssp, t));
36         }else{
37             System.out.println("equal");
38             System.out.printf("%.2f\n", 1-Math.pow(ccp, t));
39         }
40     }
41 }
```

42 }

现在有两个好友 A 和 B，住在一片长有蘑菇的由 $n \times m$ 个方格组成的草地，A 在(1,1),B 在(n,m)。现在 A 想要拜访 B，由于她只想去 B 的家，所以每次她只会走(i,j+1)或(i+1,j)这样的路线，在草地上有 k 个蘑菇种在格子里(多个蘑菇可能在同一方格),问：A 如果每一步随机选择的话(若她在边界上，则只有一种选择)，那么她不碰到蘑菇走到 B 的家的概率是多少？

```
1 //直接用概率进行 DP，用路径数是不对的
2 import java.util.Scanner;
3
4 public class Main{
5     public static void main(String[] args){
6         Scanner sca = new Scanner(System.in);
7         while(sca.hasNext()){
8             int n = sca.nextInt();
9             int m = sca.nextInt();
10            int k = sca.nextInt();
11            boolean[][] map = new boolean[n][m];
12            for(int i = 0; i < k; i++) {
13                int x = sca.nextInt()-1;
14                int y = sca.nextInt()-1;
15                map[x][y] = true;
16            }
17            double[][] cw = new double[n][m];
18            cw[0][0] = 1;
19            for(int i = 0; i < n; i++){
20                for(int j = 0; j < m; j++){
```

```
21             if(map[i][j]) cw[i][j] = 0;
22             else if(i == 0 && j == 0) {}
23             else cw[i][j] =
24 (j-1<0?0:(i+1<n?cw[i][j-1]*0.5:cw[i][j-1]))+(i-1<0?0:(j+1<m?cw[i-1][j]*0.5:cw[i-1][j]));
25             //System.out.print(String.format("%.5f", cw[i][j])+" ");
26         }
27         //System.out.println();
28     }
29     double res = cw[n-1][m-1];
30     System.out.println(String.format("%.2f", res));
31 }
32 }
```