



小易经常沉迷于网络游戏.有一次,他在玩一个打怪升级的游戏,他的角色的初始能力值为  $a$ . 在接下来的一段时间内,他将会依次遇见  $n$  个怪物,每个怪物的防御力为  $b_1, b_2, b_3 \dots b_n$ . 如果遇到的怪物防御力  $b_i$  小于等于小易的当前能力值  $c$ ,那么他就能轻松打败怪物,并且使得自己的能力值增加  $b_i$ ;如果  $b_i$  大于  $c$ ,那他也能打败怪物,但他的能力值只能增加  $b_i$  与  $c$  的最大公约数.那么问题来了,在一系列的锻炼后,小易的最终能力值为多少?

```
1  #include <stdio.h>
2  int gcd (int m,int n){
3      if (n == 0)
4          return m;
5      else
6          return gcd(n, m%n);
7  }
8  int main() {
9      int n, a, num;
10     while(~scanf("%d %d", &n, &a)) {
11         for (int i=0; i<n; i++) {
12             scanf("%d", &num);
13             if (num <= a)
14                 a+=num;
15             else
16                 a+=gcd(num, a);
17         }
18         printf("%d\n", a);
19     }
20     return 0;
21 }
```

兰博教训提莫之后,然后和提莫讨论起约德尔人,谈起约德尔人,自然少不了一个人,那就是黑默丁格-----约德尔人历史上最伟大的科学家. 提莫说,黑默丁格最近在思考一个问题:黑默丁格有三个炮台,炮台能攻击到距离它  $R$  的敌人 (两点之间的距离为两点连续的距离,例如  $(3,0),(0,4)$  之间的距离是  $5$ ),如果一个炮台能攻击到敌人,那么就会对敌人造成  $1 \times$  的伤害.黑默丁格将三个炮台放在  $N \times M$  方格中的点上,并且给出敌人 的坐标. 问:那么敌人受到伤害会是多大?

```
1  #include<iostream>
2  #include<cmath>
3  using namespace std;
4  int main(void){
5      int a[8];
6      int R,i;
7      double d1,d;
8      while(cin>>R){
```





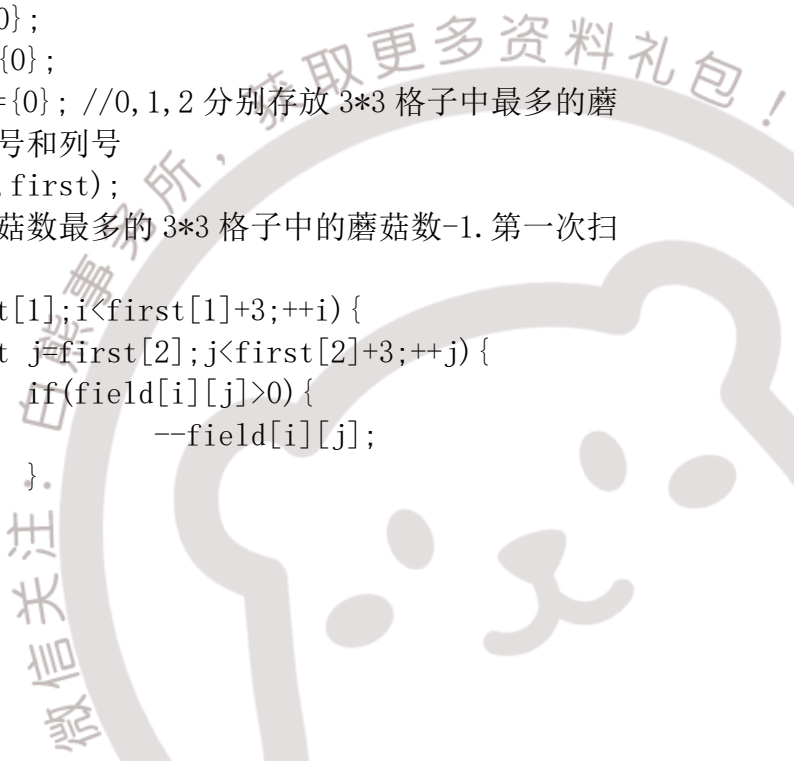
```
9         int count =0;
10        for(i=0;i<8;i++) {
11            int x;
12            cin>>x;
13            a[i]=x;
14        }
15        for(i=0;i<5;i=i+2) {
16            d1=(pow((a[i]-a[6]),2) + pow((a[i+1]-a[7]),2));
17            d=sqrt(d1);
18            if(d <= R)
19                count+=1;
20        }
21        cout<<count<<"x"<<endl;
22    }
23    return 0;
24 }
25 通过所有测试用例
```

在  $N \times M$  的草地上,提莫种了  $K$  个蘑菇,蘑菇爆炸的威力极大,兰博不想贸然去闯,而且蘑菇是隐形的.只有一种叫做扫描透镜的物品可以扫描出隐形的蘑菇,于是他回了一趟战争学院,买了 2 个扫描透镜,一个扫描透镜可以扫描出  $(3 \times 3)$  方格中所有的蘑菇,然后兰博就可以清理掉一些隐形的蘑菇.问:兰博最多可以清理多少个蘑菇?  
注意:每个方格被扫描一次只能清除掉一个蘑菇。

```
1  /* 一个扫描透镜可以扫描出 (3*3) 方格中的所有蘑菇,
2   问最多可清理多少个蘑菇就是求二维数组中哪一块 (3*3) 区域中的蘑菇数最多。
3   有两个透镜,那么最多可清理的蘑菇数就是第一个透镜最多清理的加上第二个透
4   镜
5   最多清理的 (将求最多清理蘑菇数写成函数)。
6   需要注意的是对于每个方格如果其中有多蘑菇那么一次扫描只能清理掉一个
7   蘑菇。
8   这要求我们在求出第一个最优解后要对二维数组中的相应方格中的蘑菇数进行
9   减 1 操作。*/
10 #include <iostream>
11 #include <vector>
12 using namespace std;
13
14 void scan(vector<vector<int> > field,int N,int M,int result[])
15 {
16     for(int i=0;i<N-2;++i) {
17         for(int j=0;j<M-2;++j) {
18             //统计 3*3 格子中的蘑菇数,如果一个格子中有多个
19             只算一个
20             int tmp=0; //tmp:每个 3*3 区域中的蘑菇数
```



```
21         for(int p=i;p<i+3;++p) {
22             for(int q=j;q<j+3;++q) {
23                 if(field[p][q] > 0) {
24                     ++tmp;
25                 }
26             }
27         }
28         if(result[0]<tmp) {
29             result[0] = tmp;
30             result[1] = i;
31             result[2] = j;
32         }
33     }
34 }
35 }
36
37 int main()
38 {
39     int N,M,K;
40     int x,y;
41
42     while(cin >> N >> M >> K) {
43         if(N<3)
44             N = 3;
45         if(M<3)
46             M = 3;
47         vector<vector<int> > field(N,vector<int>(M,0));
48         while(K--) {
49             cin >> x >> y;
50             ++field[x-1][y-1];
51         }
52         int first[3]={0};
53         int second[3]={0};
54         //int first[3]={0}; //0,1,2 分别存放 3*3 格子中最多的蘑
55 菇数，该 3*3 区域第一个格子的行号和列号
56         scan(field,N,M,first);
57         //将 field 中蘑菇数最多的 3*3 格子中的蘑菇数-1. 第一次扫
58 描清理掉了
59         for(int i=first[1];i<first[1]+3;++i) {
60             for(int j=first[2];j<first[2]+3;++j) {
61                 if(field[i][j]>0) {
62                     --field[i][j];
63                 }
64             }
```





65

```
    }  
    scan(field, N, M, second);  
    cout << first[0]+second[0] << endl;  
}  
  
return 0;  
}
```

