

**Minecraft杯模拟赛Ⅰ**

--始作俑者gbakkk5951

|  |  |  |  |
| --- | --- | --- | --- |
| 题目名称 | 红石钟 | 插火把 | 种西瓜 |
| 源程序文件名 | redstoneclock.cpp | torch.cpp | melon.cpp |
| 输入文件名 | redstoneclock.in | torch.in | melon.in |
| 输出文件名 | redstoneclock.out | torch.out | melon.out |
| 每个测试点时限 | 3s | 2s | 1s |
| 内存限制 | 128MB | 128MB | 128MB |
| 测试点数目 | 2 | 10 | 20 |
| 每个测试点分值 | [0]:30,[1]:70 | 10 | 5 |
| 是否打开O2优化 | 否 | 否 | 否 |

--PS:可根据评测环境下std的表现来酌情增加性能。

**128MB,3S,redstoneclock.xxx**

**红石钟**

**问题描述**

[山谷村长]gbakkk5951又在鼓捣一个神秘玩意——红石钟，不过这可不是普通的红石钟，而是可以预定超长时间闹钟的超级红石钟，由于传统的红石钟要么延时不够长，要么耗资巨大，所以他采用2个红石钟共同计时，每个红石钟有一个时钟周期Ti( 单位:tick)，当且仅当2个红石钟同时结束各自的一个时钟周期时，闹钟会被触发，村长想知道如果把时钟周期分别为Ti1 tick和Ti2 tick的红石钟组合起来（0时刻开始运行，0时刻不算触发），闹钟何时会被触发，由于闹钟可能由多个红石钟嵌套而成所以会有多组询问，且时钟周期可能较大。

我们定义一日有24小时，一小时有60分钟，一分钟有20 tick。每次询问的答案即为A日B tick后闹钟第一次被触发。

**输入描述**

第1行一个整数q表示有q次询问

接下来q行，每行两个整数Ti1,Ti2表示两个红石钟的时钟周期。

**输出描述**

3行，每行2个整数A,B ,表示 日、tick的对应和

询问按顺序编号1到q，

第一行表示奇数项询问对应答案的异或和，

第二行表示偶数项询问对应答案的异或和，

第三行表示全部项询问对应答案的数值和。

**样例输入**

5

7 6

1 7

3 8

7 9

11 13

**样例输出**

0 189

0 56

0 279

**数据范围及提示**

50%的数据(30分) 1<=Ti1,Ti2<=100,0000 , 1<=q<=60,0000

100%的数据(100分)1<=Ti1,Ti2<=1000,0000 , 1<=q<=300,0000

**128MB,2S**

**128MB,2S,torch.xxx**

**插火把**

**问题描述**

[山谷村长]gbakkk5951刚刚开辟了一大块农田，这片农田的大小为N行M列，由于天色已晚，黑暗的地方会刷怪，所以，村长想知道，多少块方格有可能刷怪。已知地图上每个点的发光值Li，发光半径Ri以及刷怪亮度Di，设当前点坐标为(xi,yi)，则所有|xj-xi|+|yj-yi|<Ri的方格的亮度会增加Li,若当前方格的亮度<=Di则这个方格有可能刷怪。

由于Minecraft世界中会出现一些蜜汁黑暗区域，所以Li可能小于0。

对于超过农田范围的方格，忽略不计。

**输入描述**

第1行2个整数N，M

接下来N行

每行M组整数，每组2个整数Li,Ri意义见题目描述。

接下来N行，每行M个整数Di，意义见题目描述。

**输出描述**

一行一个整数，表示有可能刷怪的方格个数。

**样例输入**

7 5

0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0

0 0 0 0 1 3 0 0 0 0

0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 1 1

0 0 0 0 0

0 0 0 0 0

0 0 0 0 0

0 0 0 0 0

0 0 0 0 0

0 0 0 0 0

0 0 0 0 0

**样例输出**

21

**数据范围及提示**

30%的数据N,M<=80

50%的数据N,M<=300

80%的数据N,M<=1500

100%的数据N,M<=2200 ,1<=Ri<=105, |Li|<=108,|Di|<=1015

请选手注意空间限制

**128MB,1S,melon.xxx**

**种西瓜**

**问题描述**

[山谷村长]gbakkk5951是个种瓜老手，他认为一列一列地种瓜才能够获得最大的收益，但村里新来的萌新并不这么认为，他认为自己的种法能够获得更大的收益，于是村长便找到了村子里最聪明的你，想让你帮他算算这位萌新的种瓜方式最多能有多少收益，最少能有多少收益。

这位萌新的瓜田可以看做一个平面，由N行M列的方块组成，每个方块初始有3种属性，一种是障碍物，一种是瓜藤，还有一种是空地。（边界可以看做障碍物）若瓜藤的上下左右至少有一个空地，且该瓜藤没有结瓜，那么它会随机选择一个上下左右的一块空地结瓜，已经结瓜的方格可以看做障碍物，瓜只能长在空地上。由于每个瓜藤的结瓜顺序不确定且结瓜方向不确定，所以结瓜的数量也是不确定的。我们假设时间足够即所有能结瓜的瓜藤都已经结瓜。

**输入描述**

第1行3个整数N，M，tp，表示有N行M列，询问类型为tp，tp=0询问最大值,tp=1询问最小值

接下来N行每行M个字母,l表示瓜藤，o表示空地，x表示障碍物。

**输出描述**

一行一个整数，若tp=0 输出最大的结瓜数量，若tp=1输出最小的结瓜数量。

**样例输入**

【样例一】

2 5 0

xlolx

xolox

【样例二】

2 5 1

xlolx

xolox

**样例输出**

【样例一】

3

【样例二】

2

**数据范围及提示**

50%的数据tp=0询问最大值，N×M<=1000,

其中20%的数据N=1

另外50%的数据tp=1询问最小值,N×M<=100,且瓜藤数量<=18

其中20%的数据N=1

**样例解释**

红色箭头表示结瓜方向（请无视图中瓜藤本身的方向）



样例一



样例二