吃鱼 (fish)
Time Limit:1000ms Memory Limit:128MB

题目描述

LYK 家里有 n 只猫。这一天，LYK 安排了 m 条鱼给这些猫吃。特别地，第 i 只猫吃一条鱼需要花费 a[i]的时间。且一只猫在同一时间最多只会吃一条鱼，且不会有多只猫吃同一条鱼。在第 0 时刻，LYK 给每只猫一条鱼。每当有一只猫吃完鱼时，如果此时还有鱼，它会立刻吃下一条鱼。特别地：如果有 k 只猫在同一时刻一起吃完了鱼，且此时剩下的鱼的个数不足 k，吃的快的猫(即 a[i]较小的猫)会优先吃鱼。

LYK 想知道经过 x 个时间后，有多少条鱼还没被吃过，以及有多少鱼已经被吃了一部分了（也就是说还没吃完，但已经被吃过了）。

输入格式(fish.in)

第一行三个数 m,n,x。表示有 m 条鱼，n 只猫，以及 x。

第二行 n 个数表示 a[i]。

输出格式(fish.out)

两个数，第一个数表示还有多少鱼没被吃过，第二个数表示有多少鱼被吃了一部分但还没被吃完。

输入样例
8 3 5
1 3 4

输出样例
0 1

样例解释：
第一时刻后，第一只猫吃完一只鱼；
第二时刻后，第一只猫又吃完一只鱼；
第三时刻后，第一只猫和第二只猫同时吃完一只；
第四时刻后，第一只和第三只猫也同时吃完鱼，注意此时只剩下一条鱼可以吃，这条鱼会给吃的快的猫，即第一只猫吃；

第五时刻后，第一只猫再吃完一只鱼。此时只有第二只猫吃的那条鱼没被吃完，其它所有鱼都已被吃完。注意此时如果还有鱼，第一只猫也没有时间再吃了的。

数据范围
对于 20%的数据 n,m,x<=100。
对于 50%的数据 n,m,x<=1000。
对于 80%的数据 a[i],x<=100000。
对于 100%的数据 1<=n,m<=100000，1<=a[i],x<=10^9，且 m>=n。

建议使用读入优化，读入优化见文档末位。
01 背包威力加强版 (bag)
Time Limit:1000ms Memory Limit:128MB

题目描述

LYK 有一个容量为 m 的背包，总共有 n 个物品，每个物品有它的体积 pi，价值 vi，从中选若干物品使得它们的体积和不超过 m，且价值之和最大。

但是这个问题过时了！

接下来是这个问题的加强版：

LYK 有 m 块钱，总共有 n 个物品，每个物品有它的价格 pi，参数 qi，价值 vi。LYK 可以用某个顺序来购物。它想买一个物品当且仅当它还剩下的钱不小于该物品的参数 qi。之后为了买这个物品，它的总钱数会减少 pi，相应的，得到的价值总和会增加 vi。

每个物品最多买一次，LYK 可以随便安排一个购物顺序，它想使得满足购物要求的前提下得到的价值之和尽可能高。

输入格式(bag.in)

第一行两个数 n,m。

接下来 n 行，每行三个数 p[i],q[i],v[i]。

输出格式(bag.out)

一个数表示最高多少价值。

输入样例
2 10
1 11 6
5 10 5

输出样例
5

样例解释
LYK 只能买第二个物品。

数据范围
对于 10%的数据 n<=5。
对于 30%的数据 n<=12。
对于 50%的数据 n<=20。
对于另外 10%的数据对于所有 i 有 p[i]=q[i]。
对于再另外 20%的数据对于所有 i 有 v[i]=1。
对于 100%的数据 1<=n<=1000,1<=m<=5000,1<=p[i]<=q[i]<=5000，1<=v[i]<=5000。

崩 (beng)
Time Limit:1000ms Memory Limit:128MB

题目描述

LYK 身处在一个二维平面。但这个平面快坍塌了！

我们可以想象成这个平面是左上角在(-10^12,10^12)右下角在(10^12,-10^12)的一个巨大矩阵。这个矩阵被分割成了若干个 1*1 的格子，一开始这个矩阵的最外层格子都会布满蚂蚁。

每次每只蚂蚁都会繁衍出新的蚂蚁爬向上下左右四个方向。

LYK 要圈出一块属于自己的领地。一开始它处于某个格子中，它手里有一桶杀虫剂。LYK 会执行 k 次操作，每次操作都是向四个方向(用 c[i]表示)的某个方向走 a[i]格，在走的过程中它会将杀虫剂倒在所有经过的格子上。并且 LYK 保证在行走的过程中不会走到这个矩阵的最外层格子上（即不会走到二维平面的边缘处）。

蚂蚁不会繁衍到被喷了杀虫剂的格子上。

LYK 想知道经过无限长的时间后有多少格子上是没有蚂蚁的。

输入格式(beng.in)

第一行一个数 k，表示 k 次操作。

接下来 k 行，每行一个字符 ci 和一个数 ai，ci 是 L,R,U,D 中的某一个，表示向左、向右、向上、向下。ai 表示向那个方向走了多少。

输出格式(beng.out)

一个数表示最终有多少格子没有蚂蚁。

输入样例

```
7
R 10
D 2
L 7
U 9
D 2
R 3
D 10
```

输出样例

```
52
```

数据范围

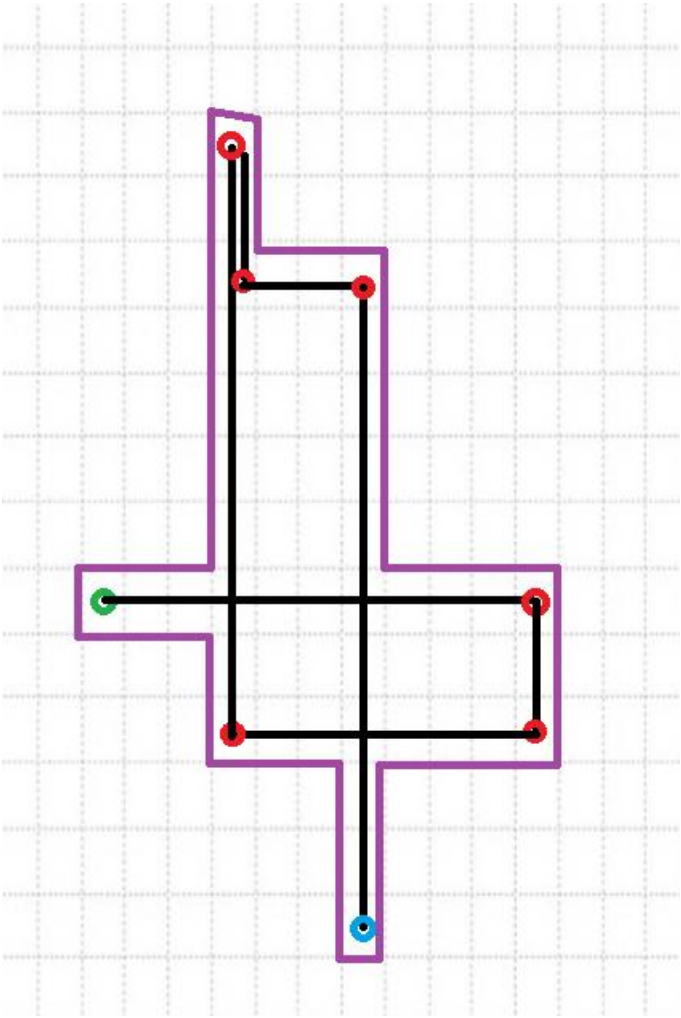
对于 20%的数据 k<=4。

对于 40%的数据 k<=10，且 LYK 保证在走的过程中任意时刻离起点的横纵坐标差均不超过 10。

对于 70%的数据 k<=1000，且 LYK 保证在走的过程中任意时刻离起点的横纵坐标差均不超过 1000。

对于 100%的数据 1<=k<=1000，1<=ai<=10^6。

样例解释：



如图，绿色为起点，会沿着黑边一直走，蓝色为终点。紫色多边形框起来的就是虫子到不了的地方，总共 52 格。

读入优化代码：

```
long long read(){
long long x=0,f=1;char ch=getchar();
while(ch<'0'||ch>'9'){if(ch=='-')f=-1;ch=getchar();}
while(ch>='0'&&ch<='9'){x=x*10+ch-'0';ch=getchar();}
return x*f;
}
```