

Monster Trucks

locked

Problem	Submissions	Leaderboard	Discussions
---------	-------------	-------------	-------------

Kyle решил да изкара някой лев за да отиде на рейда на Area 42 и почнал работа на тир паркинг. Всички шофьори карат на 2 ракии и като паркират спират или по-напред или по-назад. Kyle трябва да ги подреди да са успоредни и да са в една линия, за да има повече място за паркиране. Шофьорите не си дават ключовете и се налага Kyle да изпие няколко монстъра, да напегне сили и да избута тирове на място. Kyle, обаче иска да спести повече пари като пие колкото се може по-малко монстъри.

Всички тирове може да си мислите, че са паркирали успоредно, и са или издадени напред или назад. Целта е да се подредят така че да са в една линия, която не е нужно да е средата, тоест 0-лата. Например може всички да се подредят така че да излизат назад с 15 сантиметра или да излизат напред с 2. Важно е да подредбата да е най-евтина за Kyle.

Input Format

На входа ще получите N - брой на камионите

След това на N реда ще получите по 2 числа K M, K е колко сантиметра камионът е напред или назад (в едната посока е положително число, в другата отрицателно), M е колко монстъра са нужни на Kyle, за да помръдне камиона с 1 сантиметър напред или назад.

Constraints

$$0 \leq N \leq 50000$$

$$-10000 \leq K \leq 10000$$

$$0 \leq M \leq 5000$$

Output Format

Минималният брой монстъри необходими на Kyle, за да подреди камионите.

Sample Input 0

```
2
1 5
-2 1
```

Sample Output 0

```
3
```

Explanation 0

За да изравни тирове ще са необходими най-малко 3 монстъра на Kyle да премести тира от -2 на 1 (3 сантиметра по 1 монстръ).

Sample Input 1

```
2
0 0
1 2
```

Sample Output 1

```
0
```

Explanation 1

За да се премести първия тир на Kyle му трябват 0 монстъра, следователно може да го премести 1 сантиметър и да го изравни с втория тир на 1 сантиметър

Sample Input 2

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

Sample Output 2

```
21
```

Current Buffer (saved locally, editable)

C++

```
1 #include <cmath>
2 #include <cstdio>
3 #include <vector>
4 #include <iostream>
5 #include <algorithm>
6 #include<limits.h>
7 using namespace std;
8 unsigned long int monstersNeeded(vector<pair<int,int>> vec,int position,int trucks){
9     unsigned long int counter=0;
10    for(int i=0;i<trucks;i++){
11        counter+=(abs(vec[i].first-position))*vec[i].second;
12    }
13    return counter;
14 }
15
16
17 int main() {
18     int trucksNumber;
19     cin>>trucksNumber;
20     vector<pair<int,int>> trucks;
21     pair<int,int> combination;
22     for(int i=0;i<trucksNumber;i++){
23         cin>>combination.first;
24         cin>>combination.second;
25         trucks.push_back(combination);
26     }
27     int result=INT_MAX;
28     for(int i=-10000;i<=10000;i++){
29         if(monstersNeeded(trucks,i,trucksNumber)<result){
30             result=monstersNeeded(trucks,i,trucksNumber);
31         }
32     }
33     cout<<result;
34     return 0;
35 }
```

Line: 1 Col: 1