

6131 Arranging Heaps

A mining company extracts terbium, a rare metal used for constructing lightweight magnets, from river sand. They mine the Long River at N mining points, each of them identified by its distance from the river source. At each mining point, a relatively small but highly valued heap of mineral ore is extracted from the river.

To collect the mineral ore, the company regroups the N produced heaps into a smaller number of K heaps, each located at one of the initial mining points. The newly formed heaps are then collected by trucks.

To regroup the N heaps, they use a barge, which in practice can carry any amount of mineral ore because it is very large. The barge starts at the river source and can only travel downriver, so the heap produced at a mining point X can be taken to a mining point Y only if $Y > X$. Each heap is moved completely to another mining point, or not moved at all. The cost of moving a heap of weight W from a mining point X to a mining point Y is $W \times (Y - X)$. The total cost of the regrouping is the sum of the costs for each heap movement. Notice that a heap which is not moved has no influence on the total cost.

Given the values for N and K , the N mining points, and the weight of the heap each mining point produced, write a program that calculates the minimum total cost to regroup the N initial heaps into K heaps.

Input

Each test case is described using several lines. The first line contains two integers N and K denoting respectively the number of initial heaps and the desired number of heaps after regrouping ($1 \leq K < N \leq 1000$). Each of the next N lines describes one of the initial heaps with two integers X and W indicating that the mining point X produced a heap of weight W ($1 \leq X, W \leq 10^6$). Within each test case the heaps are given in strictly ascending order considering their mining points.

Sample Output

For each test case output a line with an integer representing the minimum total cost to regroup the N initial heaps into K heaps.

Sample Input

```
3 1
20 1
30 1
40 1
3 1
11 3
12 2
13 1
6 2
10 15
12 17
16 18
18 13
```

```
30 10
32 1
6 3
10 15
12 17
16 18
18 13
30 10
32 1
```

Sample Output

```
30
8
278
86
```