B. Ordering Pizza

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

It's another Start[c]up finals, and that means there is pizza to order for the onsite contestants. There are only 2 types of pizza (obviously not, but let's just pretend for the sake of the problem), and all pizzas contain exactly S slices.

It is known that the i-th contestant will eat s_i slices of pizza, and gain a_i happiness for each slice of type 1 pizza they eat, and b_i happiness for each slice of type 2 pizza they eat. We can order any number of type 1 and type 2 pizzas, but we want to buy the minimum possible number of pizzas for all of the contestants to be able to eat their required number of slices. Given that restriction, what is the maximum possible total happiness that can be achieved?

Input

The first line of input will contain integers N and S $(1 \le N \le 10^5, 1 \le S \le 10^5)$, the number of contestants and the number of slices per pizza, respectively. N lines follow.

The i-th such line contains integers s_i, a_i, and b_i $(1 \le s_i \le 10^5, 1 \le a_i \le 10^5, 1 \le b_i \le 10^5)$, the number of slices the i-th contestant will eat, the happiness they will gain from each type 1 slice they eat, and the happiness they will gain from each type 2 slice they eat, respectively.

Output

Print the maximum total happiness that can be achieved.

Examples

input

```
3 12
3 5 7
4 6 7
5 9 5
```

output

```
84
```

input

```
6 10
7 4 7
5 8 8
12 5 8
6 11 6
3 3 7
5 9 6
```

output

```
314
```

Note

In the first example, you only need to buy one pizza. If you buy a type 1 pizza, the total happiness will be $3 \cdot 5 + 4 \cdot 6 + 5 \cdot 9 = 84$, and if you buy a type 2 pizza, the total happiness will be $3 \cdot 7 + 4 \cdot 7 + 5 \cdot 5 = 74$.