

TASK	DISPLEJ	PLATFORME	KUHAR	JEDNAKOST
input	standard input			
output	standard output			
time limit (per test case)	1 second			
memory limit (heap+stack)	32 MB			64 MB
points	30	40	60	70
	200			

The game "display" is gaining popularity in casinos around the world. The game is run on a big screen visible to all players. A single integer is displayed on the screen, which changes every minute.

The players attempt to determine the sum of K consecutive numbers, starting from the B -th displayed number. More precisely, if X_n is the n -th number displayed (starting with X_1), then players try to determine the sum $X_B + X_{B+1} + \dots + X_{B+K-1}$.

Mirko is thrilled with this game and often spends five or six hours calculating sums and growing in debt. Recently he realized that the numbers on screen follow a precise pattern; the **same sequence of N integers** repeats over and over.

Help Mirko to find the sought sum, given the sequence of number repeating.

Input

The first line contains three integers N , K and B , $1 \leq N \leq 100$, $1 \leq K \leq 100$, $1 \leq B \leq 10^9$.

The second line contains N non-negative integers less than 100, separated by spaces. This is the sequence that keeps repeating on screen.

Output

Output the sum on a single line.

Example test data

input

6 3 10
29 6 35 0 6 8

output

14

input

5 7 154
1 2 3 4 5

output

24

input

2 100 100000
1 0

output

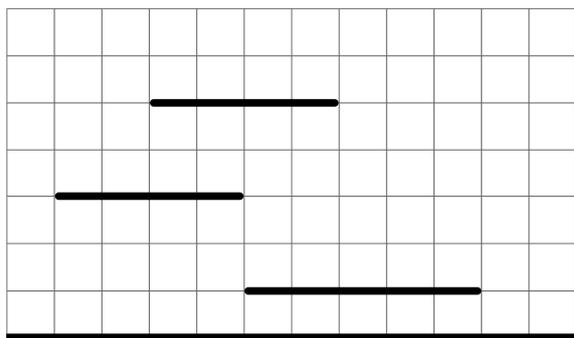
50

In the first example, the tenth number displayed is 0, followed by 6 and 8. The sum is 14.

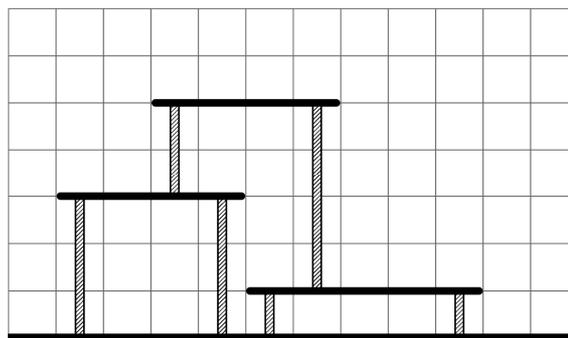
A level is being designed for a new platform game. The locations of the platforms have been chosen. Contrary to popular opinion, platforms can't float in the air, but need pillars for support. More precisely, each of the two ends of the platform needs to be supported by a **pillar standing on the floor or on a different platform**.

You will be given the locations of the platforms in a coordinate system as in the left image below. Each platform's location is determined by its altitude (vertical distance from the ground) and the start and end coordinates in the horizontal direction. Each support pillar is placed **half a unit** from the end of a platform, as in the right image.

Determine the total length of pillars needed to support all the platforms.



Example level with three platforms. The lowest is at altitude 1, the next at altitude 3, and the third at 5.



The total length of support pillars needed is 14.

Input

The first line contains the integer N , $1 \leq N \leq 100$, the number of platforms.

Each of the following N lines contains the position of one platform, three coordinates Y , X_1 and X_2 . The first number is the altitude, the other two the horizontal coordinates. All coordinates will be positive integers less than 10000 satisfying $X_2 > X_1 + 1$ (i.e. the length of each platform will be at least 2).

The input will be such that no two platforms overlap.

Output

Output the total length of pillars needed to support all the platforms.

Example test data

input

```
3
1 5 10
3 1 5
5 3 7
```

output

14

input

```
5
50 50 90
40 40 80
30 30 70
20 20 60
10 10 50
```

output

200

Lisa works as a waitress in a restaurant. Tonight is her birthday so Lisa asked the chef to prepare his special meal for her friends. The chef's meal is made of N ingredients. To prepare one serving of the meal he needs a certain amount of each ingredient.

There are some ingredients already available in the kitchen and Lisa will buy the rest at the grocery store. The store has all the necessary ingredients, each coming in **smaller** and **larger** packages. Lisa has M dollars and wants to spend them so that the chef can make **the most servings** of his meal.

Input

The first line contains two integers N and M , $1 \leq N \leq 100$, $1 \leq M \leq 100000$.

Each of the following N lines contains 6 positive integers, information about one ingredient. These specify, in order:

- X , $10 \leq X \leq 100$, the amount of the ingredient needed in one serving;
- Y , $1 \leq Y \leq 100$, the amount of the ingredient already available in the kitchen;
- S_M , $1 \leq S_M < 100$, the size of the smaller package at the store;
- P_M , $10 \leq P_M < 100$, the price of the smaller package;
- S_V , $S_M < S_V \leq 100$, the size of the larger package; and
- P_V , $P_M < P_V \leq 100$, the price of the larger package.

Output

Output the largest number of servings the chef can make if Lisa spends her money wisely.

Example test data

input

```
2 100
10 8 10 10 13 11
12 20 6 10 17 24
```

output

```
5
```

input

```
3 65
10 5 7 10 13 14
10 5 8 11 14 15
10 5 9 12 15 16
```

output

```
2
```

In the first example, for 99 dollars Lisa will buy three smaller and one larger package of the first ingredient, as well as one smaller and two larger packages of the second ingredient ($3 \cdot 10 + 1 \cdot 11 + 1 \cdot 10 + 2 \cdot 24 = 99$).

The chef will then have 51 units ($8 + 3 \cdot 10 + 1 \cdot 13$) of the first ingredient and 60 units ($20 + 1 \cdot 6 + 2 \cdot 17$) of the second ingredient, enough for 5 servings.

While browsing a math book, Mirko found a strange equation of the form $A=S$. What makes the equation strange is that A and S are not the same, which makes the equation incorrect. Mirko realized that the left side of the equation should have addition operations between some pairs of digits in A .

Write a program that inserts the **smallest number** of addition operations on the left side to make the equation correct. The numbers in the corrected equation may contain arbitrary amounts of leading zeros.

Input

The first line contains the equation in the form $A=S$.

A and S will both be positive integers without leading zeros. They will be different.

A will contain **at most 1000 digits**.

S will be **less than or equal to 5000**.

Note: The input data will guarantee that a solution, although not necessarily unique, will always exist.

Output

Output the corrected equation. If there are multiple solutions, output any of them.

Example test data

input

143175=120

output

14+31+75=120

input

5025=30

output

5+025=30

input

999899=125

output

9+9+9+89+9=125