A. Sockets

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Vasya has got many devices that work on electricity. He's got n supply-line filters to plug the devices, the i-th supply-line filter has a_i sockets.

Overall Vasya has got m devices and k electrical sockets in his flat, he can plug the devices or supply-line filters directly. Of course, he can plug the supply-line filter to any other supply-line filter. The device (or the supply-line filter) is considered plugged to electricity if it is either plugged to one of k electrical sockets, or if it is plugged to some supply-line filter that is in turn plugged to electricity.

What minimum number of supply-line filters from the given set will Vasya need to plug all the devices he has to electricity? Note that all devices and supply-line filters take one socket for plugging and that he can use one socket to plug either one device or one supply-line filter.

Input

The first line contains three integers n, m, k $(1 \le n, m, k \le 50)$ — the number of supply-line filters, the number of devices and the number of sockets that he can plug to directly, correspondingly. The second line contains n space-separated integers $a_1, a_2, ..., a_n$ $(1 \le a_i \le 50)$ — number a_i stands for the number of sockets on the i-th supply-line filter.

Output

Print a single number — the minimum number of supply-line filters that is needed to plug all the devices to electricity. If it is impossible to plug all the devices even using all the supply-line filters, print -1.

Sample test(s)

input

```
3 5 3
3 1 2
```

output

```
1
```

input

```
4 7 2
3 3 2 4
```

output

```
2
```

input

```
5 5 1
1 3 1 2 1
```

output

```
-1
```

Note

In the first test case he can plug the first supply-line filter directly to electricity. After he plug it, he get 5 (3 on the supply-line filter and 2 remaining sockets for direct plugging) available sockets to plug. Thus, one filter is enough to plug 5 devices.

One of the optimal ways in the second test sample is to plug the second supply-line filter directly and plug the fourth supply-line filter to one of the sockets in the second supply-line filter. Thus, he gets exactly 7 sockets, available to plug: one to plug to the electricity directly, 2 on the second supply-line filter, 4 on the fourth supply-line filter. There's no way he can plug 7 devices if he use one supply-line filter.

B. Playing Cubes

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Petya and Vasya decided to play a little. They found n red cubes and m blue cubes. The game goes like that: the players take turns to choose a cube of some color (red or blue) and put it in a line from left to right (overall the line will have $n + m$ cubes). Petya moves first. Petya's task is to get as many pairs of neighbouring cubes of the same color as possible. Vasya's task is to get as many pairs of neighbouring cubes of different colors as possible.

The number of Petya's points in the game is the number of pairs of neighboring cubes of the same color in the line, the number of Vasya's points in the game is the number of neighbouring cubes of the different color in the line. Your task is to calculate the score at the end of the game (Petya's and Vasya's points, correspondingly), if both boys are playing optimally well. To "play optimally well" first of all means to maximize the number of one's points, and second — to minimize the number of the opponent's points.

Input

The only line contains two space-separated integers n and m $(1 \le n, m \le 10^5)$ — the number of red and blue cubes, correspondingly.

Output

On a single line print two space-separated integers — the number of Petya's and Vasya's points correspondingly provided that both players play optimally well.

Sample test(s)

input

```
3 1
```

output

```
2 1
```

input

```
2 4
```

output

```
3 2
```

Note

In the first test sample the optimal strategy for Petya is to put the blue cube in the line. After that there will be only red cubes left, so by the end of the game the line of cubes from left to right will look as `[blue, red, red, red]`. So, Petya gets 2 points and Vasya gets 1 point.

If Petya would choose the red cube during his first move, then, provided that both boys play optimally well, Petya would get 1 point and Vasya would get 2 points.

C. View Angle

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Flatland has recently introduced a new type of an eye check for the driver's licence. The check goes like that: there is a plane with mannequins standing on it. You should tell the value of the minimum angle with the vertex at the origin of coordinates and with all mannequins standing inside or on the boarder of this angle.

As you spend lots of time "glued to the screen", your vision is impaired. So you have to write a program that will pass the check for you.

Input

The first line contains a single integer n $(1 \le n \le 10^5)$ — the number of mannequins.

Next n lines contain two space-separated integers each: x_i, y_i $(|x_i|, |y_i| \le 1000)$ — the coordinates of the i-th mannequin. It is guaranteed that the origin of the coordinates has no mannequin. It is guaranteed that no two mannequins are located in the same point on the plane.

Output

Print a single real number — the value of the sought angle in degrees. The answer will be considered valid if the relative or absolute error doesn't exceed 10^{-6}.

Sample test(s)

input

```
2
2 0
0 2
```

output

```
90.0000000000
```

input

```
3
2 0
0 2
-2 2
```

output

```
135.0000000000
```

input

```
4
2 0
0 2
-2 0
0 -2
```

output

```
270.0000000000
```

input

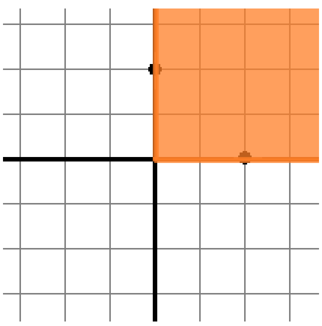
```
2
2 1
1 2
```

output

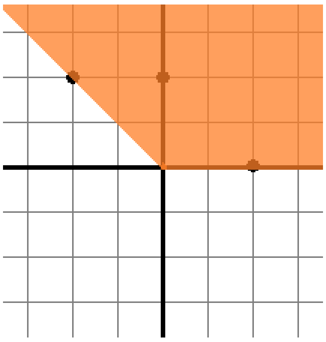
```
36.8698976458
```

Note

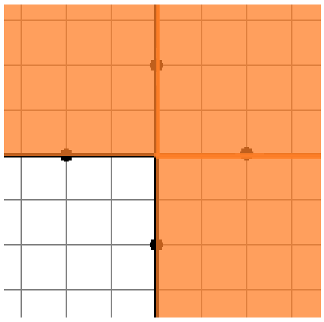
Solution for the first sample test is shown below:



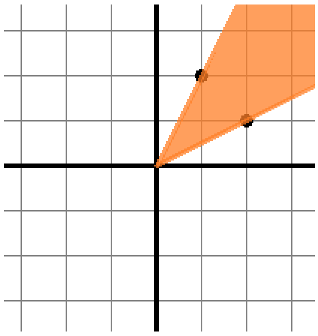
Solution for the second sample test is shown below:



Solution for the third sample test is shown below:



Solution for the fourth sample test is shown below:



D. Sum

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Vasya has found a piece of paper with an array written on it. The array consists of n integers $a_1, a_2, ..., a_n$. Vasya noticed that the following condition holds for the array $a_i \le a_{i+1} \le 2 \cdot a_i$ for any positive integer i ($i < n$).

Vasya wants to add either a "+" or a "-" before each number of array. Thus, Vasya will get an expression consisting of n summands. The value of the resulting expression is the sum of all its elements. The task is to add signs "+" and "-" before each number so that the value of expression s meets the limits $0 \le s \le a_1$. Print a sequence of signs "+" and "-", satisfying the given limits. It is guaranteed that the solution for the problem exists.

Input

The first line contains integer n ($1 \le n \le 10^5$) — the size of the array. The second line contains space-separated integers $a_1, a_2, ..., a_n$ ($0 \le a_i \le 10^9$) — the original array.

It is guaranteed that the condition $a_i \le a_{i+1} \le 2 \cdot a_i$ fulfills for any positive integer i ($i < n$).

Output

In a single line print the sequence of n characters "+" and "-", where the i-th character is the sign that is placed in front of number a_i. The value of the resulting expression s must fit into the limits $0 \le s \le a_1$. If there are multiple solutions, you are allowed to print any of them.

Sample test(s)

input

```
4
1 2 3 5
```

output

```
+++-
```

input

```
3
3 3 5
```

output

```
++-
```

E. Greedy Elevator

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

The m-floor $(m > 1)$ office of international corporation CodeForces has the advanced elevator control system established. It works as follows.

All office floors are sequentially numbered with integers from 1 to m. At time $t = 0$, the elevator is on the first floor, the elevator is empty and nobody is waiting for the elevator on other floors. Next, at times t_i $(t_i > 0)$ people come to the elevator. For simplicity, we assume that one person uses the elevator only once during the reported interval. For every person we know three parameters: the time at which the person comes to the elevator, the floor on which the person is initially, and the floor to which he wants to go.

The movement of the elevator between the floors is as follows. At time t $(t \geq 0$, t is an integer) the elevator is always at some floor. First the elevator releases all people who are in the elevator and want to get to the current floor. Then it lets in all the people waiting for the elevator on this floor. If a person comes to the elevator exactly at time t, then he has enough time to get into it. We can assume that all of these actions (going in or out from the elevator) are made instantly. After that the elevator decides, which way to move and at time $(t + 1)$ the elevator gets to the selected floor.

The elevator selects the direction of moving by the following algorithm.

- If the elevator is empty and at the current time no one is waiting for the elevator on any floor, then the elevator remains at the current floor.
- Otherwise, let's assume that the elevator is on the floor number x $(1 \leq x \leq m)$. Then elevator calculates the directions' "priorities" p_{up} and p_{down}: p_{up} is the sum of the number of people waiting for the elevator on the floors with numbers greater than x, and the number of people in the elevator, who want to get to the floors with the numbers greater than x; p_{down} is the sum of the number of people waiting for the elevator on the floors with numbers less than x, and the number of people in the elevator, who want to get to the floors with the numbers less than x. If $p_{up} \geq p_{down}$, then the elevator goes one floor above the current one (that is, from floor x to floor $x + 1$), otherwise the elevator goes one floor below the current one (that is, from floor x to floor x - 1).

Your task is to simulate the work of the elevator and for each person to tell the time when the elevator will get to the floor this person needs. Please note that the elevator is large enough to accommodate all the people at once.

Input

The first line contains two space-separated integers: n, m $(1 \leq n \leq 10^5, 2 \leq m \leq 10^5)$ — the number of people and floors in the building, correspondingly.

Next n lines each contain three space-separated integers: t_i, s_i, f_i $(1 \leq t_i \leq 10^9, 1 \leq s_i, f_i \leq m, s_i \neq f_i)$ — the time when the i-th person begins waiting for the elevator, the floor number, where the i-th person was initially located, and the number of the floor, where he wants to go.

Output

Print n lines. In the i-th line print a single number — the moment of time, when the i-th person gets to the floor he needs. The people are numbered in the order, in which they are given in the input.

Please don't use the `%lld` specifier to read or write 64-bit integers in C++. It is preferred to use the `cin`, `cout` streams or the `%I64d` specifier.

Sample test(s)

input

```
3 10
1 2 7
3 6 5
3 4 8
```

output

```
7
11
8
```

input

```
2 10
1 2 5
7 4 5
```

output

```
5
9
```

Note

In the first sample the elevator worked as follows:

- $t = 1$. The elevator is on the floor number 1. The elevator is empty. The floor number 2 has one person waiting. $p_{up} = 1 + 0 = 1$, $p_{down} = 0 + 0 = 0$, $p_{up} \geq p_{down}$. So the elevator goes to the floor number 2.
- $t = 2$. The elevator is on the floor number 2. One person enters the elevator, he wants to go to the floor number 7. $p_{up} = 0 + 1 = 1$, $p_{down} = 0 + 0 = 0$, $p_{up} \geq p_{down}$. So the elevator goes to the floor number 3.
- $t = 3$. The elevator is on the floor number 3. There is one person in the elevator, he wants to go to floor 7. The floors number 4 and 6 have two

people waiting for the elevator. $p_{up} = 2 + 1 = 3$, $p_{down} = 0 + 0 = 0$, $p_{up} \geq p_{down}$. So the elevator goes to the floor number 4.

- $t = 4$. The elevator is on the floor number 4. There is one person in the elevator who wants to go to the floor number 7. One person goes into the elevator, he wants to get to the floor number 8. The floor number 6 has one man waiting. $p_{up} = 1 + 2 = 3$, $p_{down} = 0 + 0 = 0$, $p_{up} \geq p_{down}$. So the elevator goes to the floor number 5.
- $t = 5$. The elevator is on the floor number 5. There are two people in the elevator, they want to get to the floors number 7 and 8, correspondingly. There is one person waiting for the elevator on the floor number 6. $p_{up} = 1 + 2 = 3$, $p_{down} = 0 + 0 = 0$, $p_{up} \geq p_{down}$. So the elevator goes to the floor number 6.
- $t = 6$. The elevator is on the floor number 6. There are two people in the elevator, they want to get to the floors number 7 and 8, correspondingly. One man enters the elevator, he wants to get to the floor number 5. $p_{up} = 0 + 2 = 2$, $p_{down} = 0 + 1 = 1$, $p_{up} \geq p_{down}$. So the elevator goes to the floor number 7.
- $t = 7$. The elevator is on the floor number 7. One person leaves the elevator, this person wanted to get to the floor number 7. There are two people in the elevator, they want to get to the floors with numbers 8 and 5, correspondingly. $p_{up} = 0 + 1 = 1$, $p_{down} = 0 + 1 = 1$, $p_{up} \geq p_{down}$. So the elevator goes to the floor number 8.
- $t = 8$. The elevator is on the floor number 8. One person leaves the elevator, this person wanted to go to the floor number 8. There is one person in the elevator, he wants to go to the floor number 5. $p_{up} = 0 + 0 = 0$, $p_{down} = 0 + 1 = 1$, $p_{up} < p_{down}$. So the elevator goes to the floor number 7.
- $t = 9$. The elevator is on the floor number 7. There is one person in the elevator, this person wants to get to the floor number 5. $p_{up} = 0 + 0 = 0$, $p_{down} = 0 + 1 = 1$, $p_{up} < p_{down}$. So the elevator goes to the floor number 6.
- $t = 10$. The elevator is on the floor number 6. There is one person in the elevator, he wants to get to the floor number 5. $p_{up} = 0 + 0 = 0$, $p_{down} = 0 + 1 = 1$, $p_{up} < p_{down}$. So the elevator goes to the floor number 5.
- $t = 11$. The elevator is on the floor number 5. One person leaves the elevator, this person initially wanted to get to the floor number 5. The elevator is empty and nobody needs it, so the elevator remains at the floor number 5.

Codeforces (c) Copyright 2010-2015 Mike Mirzayanov
The only programming contests Web 2.0 platform