

Problem A. Problem Statement

Time limit 1000 ms

Mem limit 262144 kB

OS Windows

Slava has almost finished writing the statement for the last problem for the ICPC Kyrgyzstan Qualification Round — just a little more and the problem will be ready!

Moreover, Slava managed to write the statement without spaces, tabs, or line breaks (he himself doesn't understand how he managed to do that).

Overjoyed, Slava went to brew himself some delicious tea.

It was at that moment that Anatoly sneaked up to Slava's laptop and started pressing the Enter key in random places.

When Slava returned, everything was already lost — instead of a coherent statement, he saw only separate pieces, divided by line breaks.

Help Slava assess the damage — calculate how many times Anatoly managed to press the Enter key.

For example, Slava wrote the statement *abracadabra*



But then Anatoly came and pressed the Enter key 4 times:

IdealStatement.md

```

1ab
2rac
3ada
4b
5ra

```

Input

The first line of the input contains an integer n ($1 \leq n \leq 100$) — the number of parts into which the original problem statement has fallen apart.

Each of the following n lines contains a string s_i ($1 \leq |s_i| \leq 10$) — the i -th part of the original statement.

Each part consists only of lowercase Latin letters.

It is guaranteed that the parts are given in the same order they appeared in the original statement.

Output

Output a single integer — the minimum number of times the Enter key must have been pressed by Anatoly to transform the original statement into the presented n parts.

Examples

Input	Output
5 ab rac ada b ra	4

Problem B. Ant Hill

Time limit 2000 ms

Mem limit 262144 kB

OS Windows

You have found a very interesting ant hill. You are so intrigued by it that you decided to find out how many ants were inside at the initial moment (when you discovered it).

To do this, you observed throughout the day and recorded an array a_i — how many ants entered or exited at the i -th moment in time:

- $|a_i|$ represents the number of ants;
- The sign of the number indicates the direction (minus — exited, plus — entered)

.

Using your observations, determine the minimum number of ants that could have been inside the ant hill before the observations began.

Note that at no point in time could there be a negative number of ants inside the ant hill.

Input

The first line contains an integer n ($1 \leq n \leq 10^5$) — the number of observations made.

The second line contains n integers a_1, a_2, \dots, a_n ($-10^6 \leq a_i \leq 10^6$, $i = 1, 2, \dots, n$) — the result of the i -th observation.

Output

In a single line, output an integer — the minimum possible number of ants inside the ant hill before the observations began.

Examples

Input	Output
3 20 -50 30	30

Note

First test example

The number of ants inside the ant hill changed as follows during the observations:

- At the beginning, there were 30;
- 20 entered — a total of 50 became present;
- 50 exited — there were no ants left;
- 30 entered — there were 30 ants inside.

Problem C. Linear Maze

Time limit 2000 ms

Mem limit 262144 kB

OS Windows

There is a maze consisting of n rooms. From the i -th room, you can move to room $i + 1$ (but not the other way).

The entrance to the maze is in room 1, and moving from the n -th room leads to the exit.

But what is the point of a maze if it can be traversed directly?

Therefore, the creators of the maze invented **branch rooms** — if you visit such a room on the 1-st, 3-rd, 5-th, etc. visit (that is, on each odd visit) — then the transition from it will lead back to room 1 instead of the next numbered room (or the exit).

It is known that rooms r_1, r_2, \dots, r_k are branch rooms.

Output the **total number of rooms** that need to be visited to exit the maze.

Input

The first line contains two integers n and k ($1 \leq n \leq 30$, $0 \leq k \leq n$) — the total number of rooms in the maze and the number of branch rooms, respectively.

The next line contains k integers r_1, r_2, \dots, r_k ($1 \leq r_1 < r_2 < \dots < r_k \leq n$) — the numbers of the branch rooms.

Output

In a single line, output an integer — the total number of rooms that need to be visited to exit the maze.

Examples

Input	Output
5 2 2 4	13

Note

First test example

The sequence of visited rooms will be as follows:

(1, 2, 1, 2, 3, 4, 1, 2, 1, 2, 3, 4, 5), after which the exit follows.

A total of 13 rooms were visited.

Problem D. Grouping

Time limit 2000 ms

Mem limit 262144 kB

OS Windows

A group of numbers is called **good** if

- the size of the set (the amount of numbers) does not exceed k ;
- the difference between the minimum and maximum number does not exceed M ;

The group may contain duplicate values — for example, the size of the group $(2, 7, 3, 7)$ is 4, and the difference between the minimum and maximum number is $7 - 2 = 5$.

Given an array of integers a . Split its elements into **the minimum number** of good groups.

Input

The first line contains integers n, M, k ($1 \leq n, k \leq 10^5, 0 \leq M \leq 10^9$) — the number of elements in the array, the maximum difference between the numbers in the set, and the maximum size of the set, respectively.

The second line contains n integers a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$) — the elements of the array.

Output

In a single line, output the minimum possible number of good groups into which the given array can be split.

Examples

Input	Output
4 3 3 5 8 13 21	3

Input	Output
10 5 3 1 6 4 15 4 10 8 2 12 5	4

Note

The second test example

The array can be split, **for example**, into the following groups:

- (1, 4, 6) — maximum difference 5;
- (2, 5) — maximum difference 3;
- (4, 8) — maximum difference 4;
- (10, 12, 15) — maximum difference 5.

Problem E. Mountain Ranges

Time limit 2000 ms

Mem limit 262144 kB

OS Windows

You are looking out the window and see n mountains, where the height of the i -th mountain is h_i .

You consider that two neighboring mountains i and $(i + 1)$ belong to the same **mountain range** if $|h_i - h_{i+1}| \leq k$, where k is the similarity parameter.

Determine the **minimum possible** similarity parameter k such that all mountains are divided into **exactly** m mountain ranges.

Input

The first line contains two integers n and m ($1 \leq m \leq n \leq 2 \cdot 10^5$) — the number of mountains and the required number of mountain ranges.

The second line contains n integers h_1, h_2, \dots, h_n ($1 \leq h_j \leq 10^6$, $j = 1, 2, \dots, n$) — the height of the j -th mountain.

Output

In a single line, output a single positive integer k — the similarity parameter at which all mountains are divided into exactly m mountain chains.

If there are multiple possible values of k , output the **minimum value**.

If no solution exists, output - 1 as the answer.

Examples

Input	Output
11 3 2 3 1 5 2 1 4 8 7 9 6	3

Input	Output
5 3 3 1 5 7 6	-1

Note

First test example

With $k = 3$, the following $m = 3$ mountain ranges are formed:

- 1... 3: $|2 - 3| \leq 3$, $|3 - 1| \leq 3$, $|1 - 5| > 3$;
- 4... 7: $|5 - 2| \leq 3$, $|2 - 1| \leq 3$, $|1 - 4| \leq 3$, $|4 - 8| > 3$;
- 8... 11: $|8 - 7| \leq 3$, $|7 - 9| \leq 3$, $|9 - 6| \leq 3$.

With $k = 2$, there will be 6 mountain ranges.

Second test example

With $k = 2$, the following 2 mountain ranges are formed:

- 1... 2: $|3 - 1| \leq 2$, $|1 - 5| > 2$;
- 3... 5: $|5 - 7| \leq 2$, $|7 - 6| \leq 2$.

With $k = 1$, there will be 4 mountain ranges.

It is clear that there is no k such that exactly $m = 3$ mountain ranges are formed.

Problem F. Traffic Lights

Time limit 2000 ms
Mem limit 262144 kB
OS Windows

A new school year has begun, which means you'll have to ride the bus through traffic jams to the university again.

You know that there are n intersections with traffic lights between the starting and ending stops.

The starting stop is located immediately after intersection 0, and the ending stop is just after intersection n .

You know that the bus takes a_i seconds to travel from the $(i - 1)$ -th intersection to the i -th intersection. The time spent at stops can be neglected.

For the traffic light at the i -th intersection, three characteristics are known:

- r_i — how many seconds the red light is on at the traffic light;
- g_i — how many seconds the green light is on;
- d_i — how many seconds before your bus boarding the green light turned on at the traffic light.

If the bus arrives exactly when the light switches to red, it will not be able to pass.

Using all this information, determine how many seconds the entire bus trip will take from start to finish.

Input

The first line contains an integer n ($1 \leq n \leq 10^4$) — the number of intersections on your way.

The second line contains integers a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^4$, $i = 1, 2, \dots, n$) — the time it takes for the bus to travel from the $(i - 1)$ -th to the i -th intersection.

Each of the following n lines contains three integers r_i, g_i, d_i ($1 \leq r_i, g_i \leq 1000, 1 \leq d_i \leq 10^4$) — the characteristics of the i -th traffic light.

Output

In a single line, output an integer — the time it will take for you to travel by bus from the starting stop to the ending stop.

Examples

Input	Output
3 100 110 50 80 20 50 60 10 100 30 40 50	370

Note

First test example

- You reached the 1-st traffic light in 100 seconds.
- The 1-st traffic light is green during the time intervals $[-50; -30]$, $[50; 70]$, $[150; 170]$;
- You arrived during the red light, so the bus will wait until 150 seconds.
- The bus will reach the 2-nd traffic light in 110 seconds — it has already been $150 + 110 = 260$ seconds.
- The 2-nd traffic light is green during the time intervals $[-100; -90]$, ... $[180; 190]$, $[250; 260]$, $[320; 330]$.
- The bus arrived exactly at the beginning of the red light, so it will have to wait until 320 seconds.
- The bus will reach the 3-rd traffic light in 50 seconds — it has already been $320 + 50 = 370$ seconds.
- The 3-rd traffic light is green during the time intervals $[-50; -10]$, ... $[300; 340]$, $[370; 410]$.
- The bus arrived exactly at the beginning of the green light, so it passed the intersection immediately without waiting.

- In total, you traveled the entire distance in 370 seconds.

Problem G. Need More Gold

Time limit 2000 ms

Mem limit 262144 kB

OS Windows

A new semester has begun — which means your training in the esports team for Doka 3.

Today you are playing for the first time as a hero named Midas. Experienced teammates have told you that the power of this hero directly depends on the amount of gold he accumulates.

To earn gold, you must defeat monsters.

On the world map, there are n monsters that you can fight in any order. You can fight each monster **only once**.

For the i -th monster, two characteristics are known:

- g_i — how much gold you will receive after defeating the monster;
- b_i — how much gold reserve you need to defeat the monster. The gold is **not spent** in this process.

Initially, you have 0 gold. Your goal in the game is to accumulate at least w gold.

Find the **minimum number** of monsters you need to defeat to achieve this goal, as well as the indices of the monsters in the order you should fight them.

Input

The first line contains an integer w ($1 \leq w \leq 10^9$) — the amount of gold that is your goal in the game.

The second line contains an integer n ($1 \leq n \leq 10^5$) — the number of monsters you can fight.

Each of the following n lines contains two integers g_i and b_i ($1 \leq g_i \leq 10^9$, $0 \leq b_i \leq 10^9$) — the characteristics of the i -th monster as described in the problem.

Output

In the first line, output an integer k — the minimum number of battles with monsters required to achieve the goal of w gold.

If you cannot achieve the goal in any way — output -1 .

If successful, in the second line output k numbers — the indices of the monsters in the order you should fight them.

Assume that the monsters are numbered in the order they are mentioned in the input data.

If there are multiple valid answers that lead to the minimum number of battles, output any of them.

Examples

Input	Output
28 8 5 2 7 25 4 0 6 1 2 20 3 3 8 7 4 15	6 3 4 1 6 7 8

Input	Output
5 1 5 1	-1

Note

First test example

Let's describe the sequence of battles and the gold earned.

- At the beginning, you have 0 gold, so you can only fight the 3-rd monster and earn 4 gold.
- Next, you fight the 4-th monster and earn 6 gold — now you have 10 gold in total.
- After that, you fight the 1-st monster and earn an additional 5 gold — now you have 15 in total.

- Then you fight the 6-th monster and earn an additional 3 gold — now you have 18 gold in total.
- The last two battles are with monsters 7 and 8 (in any order, as you have enough gold) — in total, you earn 12 gold, bringing your total to 30.
- You need 28 gold, so the goal is achieved.

Note that there are other orders/choices of monsters for battles.

Second test example

You start with 0 gold, and the only available monster requires 1 gold to defeat.

You cannot accumulate $w = 5$ gold, so you must output - 1.

Problem H. Hierarchy

Time limit 2000 ms

Mem limit 262144 kB

OS Windows

You have joined a large corporation.

The corporation is divided into several companies, each managed by a CEO.

All employees of the corporation, except for the CEOs, have exactly one direct supervisor.

You are given an array b — the indexes of the direct supervisors for each employee of the company.

Output the index of the CEO who has the **largest number of employees** in the company, as well as that number itself.

Input

The first line contains an integer n ($1 \leq n \leq 10^5$) — the number of people in the company.

The second line contains n integers b_i ($0 \leq b_i \leq n$), where b_i — the index of the direct supervisor of the i -th employee. If $b_i = 0$, then the i -th employee is the CEO.

It is guaranteed that the described hierarchy is correct — if employee b_i is the direct supervisor of employee i , then employee i is neither directly nor indirectly a supervisor of employee b_i .

Output

In the first line, output two integers — the index of the CEO with the largest number of employees in the company and that number itself.

If there are multiple answers with the largest number of employees, output any.

Examples

Input	Output
15 6 0 0 15 13 2 15 14 13 3 12 0 12 2 11	12 8

Note

First test case

- The CEO with index 2 manages a company with employees 6, 14, 1, and 8 — a total of 5 people.
- The CEO with index 3 manages a company with employee 10 — a total of 2 people.
- The CEO with index 12 manages a company with all the other employees — a total of 8 people.

Problem I. Study Day

Time limit 2000 ms
Mem limit 262144 kB
OS Windows

Today you will attend n lectures, and in the i -th lecture, you can gain a_i knowledge.

To enhance the effect of studying, you can meditate before a lecture — in this case, you will receive **double the amount** of knowledge for that lecture.

Unfortunately, you cannot meditate before two consecutive lectures — the effect will be significantly weakened.

Calculate the **maximum amount of knowledge** you can gain from all the lectures by meditating at optimal times.

Input

The first line contains an integer n ($1 \leq n \leq 10^5$) — the number of lectures.

The second line contains n integers a_1, a_2, \dots, a_n ($0 \leq a_j \leq 10^5, j = 1, 2, \dots, n$) — the amount of knowledge you will gain by default from the j -th lecture.

Output

In the first line, output an integer k — the maximum possible total amount of knowledge you can gain in a day.

In the second line, output n characters $s_1 s_2 \dots s_n$, where the character $s_j = M$ if you meditated before the j -th lecture, and $s_j = O$ if you attended the lecture without meditating.

The characters M and O are uppercase Latin letters.

If there are multiple valid answers with the optimal amount of knowledge, output any.

Examples

Input	Output
6 1 8 1 1 5 2	31 OMOOMO

Input	Output
7 2 4 2 3 5 4 5	39 MOMOMOM

Note

First test example

You decided to meditate before the 2-nd and 5-th lectures — you gained a total of $1 + 8 \cdot 2 + 1 + 1 + 5 \cdot 2 + 2 = 31$.

Second test example

In this case, you meditated before each odd lecture — the total came to $2 \cdot 2 + 4 + 2 \cdot 2 + 3 + 5 \cdot 2 + 4 + 5 \cdot 2 = 39$.

Note that you **could not** meditate before both the 5-th and 6-th lectures.