Codeforces Beta Round #72 (Div. 2 Only)

A. Toy Army

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

The hero of our story, Valera, and his best friend Arcady are still in school, and therefore they spend all the free time playing turn-based strategy "GAGA: Go And Go Again". The gameplay is as follows.

There are two armies on the playing field each of which consists of n men (n is always even). The current player specifies for each of her soldiers an enemy's soldier he will shoot (a target) and then all the player's soldiers shot simultaneously. This is a game world, and so each soldier shoots perfectly, that is he absolutely always hits the specified target. If an enemy soldier is hit, he will surely die. It may happen that several soldiers had been indicated the same target. Killed soldiers do not participate in the game anymore.

The game "GAGA" consists of three steps: first Valera makes a move, then Arcady, then Valera again and the game ends.

You are asked to calculate the maximum total number of soldiers that may be killed during the game.

Input

The input data consist of a single integer n ($2 \le n \le 10^8$, n is even). Please note that before the game starts there are $2n$ soldiers on the fields.

Output

Print a single number — a maximum total number of soldiers that could be killed in the course of the game in three turns.

Sample test(s)

input

```
2
```

output

```
3
```

input

```
4
```

output

```
6
```

Note

The first sample test:

1) Valera's soldiers 1 and 2 shoot at Arcady's soldier 1.

2) Arcady's soldier 2 shoots at Valera's soldier 1.

3) Valera's soldier 1 shoots at Arcady's soldier 2.

There are 3 soldiers killed in total: Valera's soldier 1 and Arcady's soldiers 1 and 2.

B. Magical Array

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

Valery is very interested in magic. Magic attracts him so much that he sees it everywhere. He explains any strange and weird phenomenon through intervention of supernatural forces. But who would have thought that even in a regular array of numbers Valera manages to see something beautiful and magical.

Valera absolutely accidentally got a piece of ancient parchment on which an array of numbers was written. He immediately thought that the numbers in this array were not random. As a result of extensive research Valera worked out a wonderful property that a magical array should have: an array is defined as magic if its **minimum and maximum coincide**.

He decided to share this outstanding discovery with you, but he asks you for help in return. Despite the tremendous intelligence and wit, Valera counts very badly and so you will have to complete his work. All you have to do is count the number of magical subarrays of the original array of numbers, written on the parchment. Subarray is defined as **non-empty sequence of consecutive elements**.

Input

The first line of the input data contains an integer n $(1 \le n \le 10^5)$. The second line contains an array of original integers $a_1, a_2, ..., a_n$ $(- 10^9 \le a_i \le 10^9)$.

Output

Print on the single line the answer to the problem: the amount of subarrays, which are magical.

Please do not use the `%lld` specificator to read or write 64-bit numbers in C++. It is recommended to use `cin`, `cout` streams (you can also use the `%I64d` specificator).

Sample test(s)

input

```
4
2 1 1 4
```

output

```
5
```

input

```
5
-2 -2 -2 0 1
```

output

```
8
```

Note

Notes to sample tests:

Magical subarrays are shown with pairs of indices [a;b] of the beginning and the end.

In the first sample: [1;1], [2;2], [3;3], [4;4], [2;3].

In the second sample: [1;1], [2;2], [3;3], [4;4], [5;5], [1;2], [2;3], [1;3].

C. Biathlon

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Perhaps many have heard that the World Biathlon Championship has finished. Although our hero Valera was not present at this spectacular event himself and only watched it on TV, it excited him so much that he decided to enroll in a biathlon section.

Of course, biathlon as any sport, proved very difficult in practice. It takes much time and effort. Workouts, workouts, and workouts, — that's what awaited Valera on his way to great achievements in biathlon.

As for the workouts, you all probably know that every professional biathlete should ski fast and shoot precisely at the shooting range. Only in this case you can hope to be successful, because running and shooting are the two main components of biathlon. Valera has been diligent in his ski trainings, which is why he runs really fast, however, his shooting accuracy is nothing to write home about.

On a biathlon base where Valera is preparing for the competition, there is a huge rifle range with n targets. Each target have shape of a circle, and **the center of each circle is located on the Ox axis**. At the last training session Valera made the total of m shots. To make monitoring of his own results easier for him, one rather well-known programmer (of course it is you) was commissioned to write a program that would reveal how many and which targets Valera hit. More specifically, for each target the program must print the number of **the first** successful shot (in the target), or "-1" if this was not hit. **The target is considered hit if the shot is inside the circle or on its boundary.** Valera is counting on you and perhaps, thanks to you he will one day win international competitions.

Input

The first line of the input file contains the integer n ($1 \le n \le 10^4$), which is the number of targets. The next n lines contain descriptions of the targets. Each target is a circle whose center is located on the Ox axis. Each circle is given by its coordinate of the center x ($-2\cdot 10^4 \le x \le 2\cdot 10^4$) and its radius r ($1 \le r \le 1000$). **It is guaranteed that no two targets coincide, intersect or are nested into each other, but they can touch each other.**

The next line contains integer m ($1 \le m \le 2\cdot 10^5$), which is the number of shots. Next m lines contain descriptions of the shots, which are points on the plane, given by their coordinates x and y ($-2\cdot 10^4 \le x, y \le 2\cdot 10^4$).

All the numbers in the input are integers.

Targets and shots are numbered starting from one in the order of the input.

Output

Print on the first line a single number, the number of targets hit by Valera. Print on the second line for each of the targets the number of its first hit or "-1" (without quotes) if this number does not exist. Separate numbers with spaces.

Sample test(s)

input

```
3
2 1
5 2
10 1
5
0 1
1 3
3 0
4 0
4 0
```

output

```
2
3 3 -1
```

input

```
3
3 2
7 1
11 2
4
2 1
6 0
6 4
11 2
```

output

```
3
1 2 4
```

D. Doctor

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

There are n animals in the queue to Dr. Dolittle. When an animal comes into the office, the doctor examines him, gives prescriptions, appoints tests and may appoint extra examination. Doc knows all the forest animals perfectly well and therefore knows exactly that the animal number i in the queue will have to visit his office exactly a_i times. We will assume that an examination takes much more time than making tests and other extra procedures, and therefore we will assume that once an animal leaves the room, it immediately gets to the end of the queue to the doctor. Of course, if the animal has visited the doctor as many times as necessary, then it doesn't have to stand at the end of the queue and it immediately goes home.

Doctor plans to go home after receiving k animals, and therefore what the queue will look like at that moment is important for him. Since the doctor works long hours and she can't get distracted like that after all, she asked you to figure it out.

Input

The first line of input data contains two space-separated integers n and k ($1 \le n \le 10^5$, $0 \le k \le 10^{14}$). In the second line are given space-separated integers $a_1, a_2, ..., a_n$ ($1 \le a_i \le 10^9$).

Please do not use the `%lld` specificator to read or write 64-bit numbers in C++. It is recommended to use `cin`, `cout` streams (you can also use the `%I64d` specificator).

Output

If the doctor will overall carry out less than k examinations, print a single number "-1" (without quotes). Otherwise, print the sequence of numbers — number of animals in the order in which they stand in the queue.

Note that this sequence may be empty. This case is present in pretests. You can just print nothing or print one "End of line"-character. Both will be accepted.

Sample test(s)

input

```
3 3
1 2 1
```

output

```
2
```

input

```
4 10
3 3 2 1
```

output

```
-1
```

input

```
7 10
1 3 3 1 2 3 1
```

output

```
6 2 3
```

Note

In the first sample test:

- Before examination: $\{1, 2, 3\}$
- After the first examination: $\{2, 3\}$
- After the second examination: $\{3, 2\}$
- After the third examination: $\{2\}$

In the second sample test:

- Before examination: $\{1, 2, 3, 4, 5, 6, 7\}$
- After the first examination: $\{2, 3, 4, 5, 6, 7\}$
- After the second examination: $\{3, 4, 5, 6, 7, 2\}$
- After the third examination: $\{4, 5, 6, 7, 2, 3\}$
- After the fourth examination: $\{5, 6, 7, 2, 3\}$
- After the fifth examination: $\{6, 7, 2, 3, 5\}$
- After the sixth examination: $\{7, 2, 3, 5, 6\}$
- After the seventh examination: $\{2, 3, 5, 6\}$
- After the eighth examination: $\{3, 5, 6, 2\}$
- After the ninth examination: $\{5, 6, 2, 3\}$

- After the tenth examination: $\{6, 2, 3\}$

E. Track

time limit per test: 5 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

You already know that Valery's favorite sport is biathlon. Due to your help, he learned to shoot without missing, and his skills are unmatched at the shooting range. But now a smaller task is to be performed, he should learn to complete the path fastest.

The track's map is represented by a rectangle $n \times m$ in size divided into squares. Each square is marked with a lowercase Latin letter (which means the type of the plot), with the exception of the starting square (it is marked with a capital Latin letters S) and the terminating square (it is marked with a capital Latin letter T). The time of movement from one square to another is equal to 1 minute. The time of movement within the cell can be neglected. We can move from the cell only to side-adjacent ones, but it is forbidden to go beyond the map edges. Also the following restriction is imposed on the path: it is not allowed to visit more than k **different types** of squares (squares of one type can be visited an infinite number of times). Squares marked with S and T have no type, so they are not counted. But S must be visited exactly once — at the very beginning, and T must be visited exactly once — at the very end.

Your task is to find the path from the square S to the square T that takes minimum time. Among all shortest paths you should choose the **lexicographically minimal** one. When comparing paths you should lexicographically represent them as a sequence of characters, that is, of plot types.

Input

The first input line contains three integers n, m and k ($1 \le n, m \le 50, n \cdot m \ge 2, 1 \le k \le 4$). Then n lines contain the map. Each line has the length of exactly m characters and consists of lowercase Latin letters and characters S and T. It is guaranteed that the map contains exactly one character S and exactly one character T.

Pretest 12 is one of the maximal tests for this problem.

Output

If there is a path that satisfies the condition, print it as a sequence of letters — the plot types. Otherwise, print "-1" (without quotes). **You shouldn't print the character S in the beginning and T in the end.**

Note that this sequence may be empty. This case is present in pretests. You can just print nothing or print one "End of line"-character. Both will be accepted.

Sample test(s)

input

```
5 3 2
Sba
ccc
aac
ccc
abT
```

output

```
bcccc
```

input

```
3 4 1
Sxyy
yxxx
yyyT
```

output

```
xxxx
```

input

```
1 3 3
TyS
```

output

```
y
```

input

```
1 4 1
SxyT
```

output

```
-1
```

The only programming contests Web 2.0 platform