Technocup 2017 - Elimination Round 2

A. Interview with Oleg

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

Polycarp has interviewed Oleg and has written the interview down without punctuation marks and spaces to save time. Thus, the interview is now a string s consisting of n lowercase English letters.

There is a filler word `ogo` in Oleg's speech. All words that can be obtained from `ogo` by adding `go` several times to the end of it are also considered to be fillers. For example, the words `ogo`, `ogogo`, `ogogogo` are fillers, but the words `go`, `og`, `ogog`, `ogogog` and `oggo` are not fillers.

The fillers have maximal size, for example, for `ogogoo` speech we can't consider `ogo` a filler and `goo` as a normal phrase. We should consider `ogogo` as a filler here.

To print the interview, Polycarp has to replace each of the fillers with three asterisks. Note that a filler word is replaced with exactly three asterisks regardless of its length.

Polycarp has dealt with this problem in no time. Can you do the same? The clock is ticking!

Input

The first line contains a positive integer n ($1 \le n \le 100$) — the length of the interview.

The second line contains the string s of length n, consisting of lowercase English letters.

Output

Print the interview text after the replacement of each of the fillers with "`***`". It is allowed for the substring "`***`" to have several consecutive occurences.

Examples

input
7 aogogob
output
a***b

input
13 ogogmgogogogo
output
gmg

input
9 ogoogoogo
output
****

Note

The first sample contains one filler word `ogogo`, so the interview for printing is "`a***b`".

The second sample contains two fillers `ogo` and `ogogogo`. Thus, the interview is transformed to "`***gmg***`".

B. Spotlights

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

Theater stage is a rectangular field of size $n \times m$. The director gave you the stage's plan which actors will follow. For each cell it is stated in the plan if there would be an actor in this cell or not.

You are to place a spotlight on the stage in some *good* position. The spotlight will project light in one of the four directions (if you look at the stage from above) — left, right, up or down. Thus, the spotlight's position is a cell it is placed to and a direction it shines.

A position is *good* if two conditions hold:

- there is no actor in the cell the spotlight is placed to;
- there is at least one actor in the direction the spotlight projects.

Count the number of *good* positions for placing the spotlight. Two positions of spotlight are considered to be different if the location cells or projection direction differ.

Input

The first line contains two positive integers n and m $(1 \le n, m \le 1000)$ — the number of rows and the number of columns in the plan.

The next n lines contain m integers, 0 or 1 each — the description of the plan. Integer 1, means there will be an actor in the corresponding cell, while 0 means the cell will remain empty. It is guaranteed that there is at least one actor in the plan.

Output

Print one integer — the number of *good* positions for placing the spotlight.

Examples

input
2 4 0 1 0 0 1 0 1 0
output
9

input
4 4 0 0 0 0 1 0 0 1 0 1 1 0 0 1 0 0
output
20

Note

In the first example the following positions are *good*:

1. the (1, 1) cell and right direction;
2. the (1, 1) cell and down direction;
3. the (1, 3) cell and left direction;
4. the (1, 3) cell and down direction;
5. the (1, 4) cell and left direction;
6. the (2, 2) cell and left direction;
7. the (2, 2) cell and up direction;
8. the (2, 2) and right direction;
9. the (2, 4) cell and left direction.

Therefore, there are 9 *good* positions in this example.

C. Road to Cinema

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

Vasya is currently at a car rental service, and he wants to reach cinema. The film he has bought a ticket for starts in t minutes. There is a straight road of length s from the service to the cinema. Let's introduce a coordinate system so that the car rental service is at the point 0, and the cinema is at the point s.

There are k gas stations along the road, and at each of them you can fill a car with any amount of fuel for free! Consider that this operation doesn't take any time, i.e. is carried out instantly.

There are n cars in the rental service, i-th of them is characterized with two integers c_i and v_i — the price of this car rent and the capacity of its fuel tank in liters. It's not allowed to fuel a car with more fuel than its tank capacity v_i. All cars are completely fueled at the car rental service.

Each of the cars can be driven in one of two speed modes: normal or accelerated. In the normal mode a car covers 1 kilometer in 2 minutes, and consumes 1 liter of fuel. In the accelerated mode a car covers 1 kilometer in 1 minutes, but consumes 2 liters of fuel. The driving mode can be changed at any moment and any number of times.

Your task is to choose a car with minimum price such that Vasya can reach the cinema before the show starts, i.e. not later than in t minutes. Assume that all cars are completely fueled initially.

Input

The first line contains four positive integers n, k, s and t ($1 \le n \le 2 \cdot 10^5$, $1 \le k \le 2 \cdot 10^5$, $2 \le s \le 10^9$, $1 \le t \le 2 \cdot 10^9$) — the number of cars at the car rental service, the number of gas stations along the road, the length of the road and the time in which the film starts.

Each of the next n lines contains two positive integers c_i and v_i ($1 \le c_i, v_i \le 10^9$) — the price of the i-th car and its fuel tank capacity.

The next line contains k **distinct** integers $g_1, g_2, \ldots, g_k$ ($1 \le g_i \le s - 1$) — the positions of the gas stations on the road in arbitrary order.

Output

Print the minimum rent price of an appropriate car, i.e. such car that Vasya will be able to reach the cinema before the film starts (not later than in t minutes). If there is no appropriate car, print `-1`.

Examples

input
3 1 8 10 10 8 5 7 11 9 3
output
10

input
2 2 10 18 10 4 20 6 5 3
output
20

Note

In the first sample, Vasya can reach the cinema in time using the first or the third cars, but it would be cheaper to choose the first one. Its price is equal to 10, and the capacity of its fuel tank is 8. Then Vasya can drive to the first gas station in the accelerated mode in 3 minutes, spending 6 liters of fuel. After that he can full the tank and cover 2 kilometers in the normal mode in 4 minutes, spending 2 liters of fuel. Finally, he drives in the accelerated mode covering the remaining 3 kilometers in 3 minutes and spending 6 liters of fuel.

D. Sea Battle

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

Galya is playing one-dimensional Sea Battle on a $1 \times n$ grid. In this game a ships are placed on the grid. Each of the ships consists of b consecutive cells. No cell can be part of two ships, however, the ships **can touch** each other.

Galya doesn't know the ships location. She can shoot to some cells and after each shot she is told if that cell was a part of some ship (this case is called "hit") or not (this case is called "miss").

Galya has already made k shots, all of them were misses.

Your task is to calculate the minimum number of cells such that if Galya shoot at all of them, she would hit at least one ship.

It is guaranteed that there is at least one valid ships placement.

Input

The first line contains four positive integers n, a, b, k ($1 \le n \le 2 \cdot 10^5$, $1 \le a, b \le n$, $0 \le k \le n - 1$) — the length of the grid, the number of ships on the grid, the length of each ship and the number of shots Galya has already made.

The second line contains a string of length n, consisting of zeros and ones. If the i-th character is one, Galya has already made a shot to this cell. Otherwise, she hasn't. It is guaranteed that there are exactly k ones in this string.

Output

In the first line print the minimum number of cells such that if Galya shoot at all of them, she would hit at least one ship.

In the second line print the cells Galya should shoot at.

Each cell should be printed exactly once. You can print the cells in arbitrary order. The cells are numbered from 1 to n, starting from the left.

If there are multiple answers, you can print any of them.

Examples

input
5 1 2 1 00100
output
2 4 2

input
13 3 2 3 1000000010001
output
2 7 11

Note

There is one ship in the first sample. It can be either to the left or to the right from the shot Galya has already made (the "1" character). So, it is necessary to make two shots: one at the left part, and one at the right part.

E. Subordinates

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

There are n workers in a company, each of them has a unique id from 1 to n. **Exaclty** one of them is a chief, his id is s. Each worker except the chief has exactly one immediate superior.

There was a request to each of the workers to tell how how many superiors (not only immediate). Worker's superiors are his immediate superior, the immediate superior of the his immediate superior, and so on. For example, if there are three workers in the company, from which the first is the chief, the second worker's immediate superior is the first, the third worker's immediate superior is the second, then the third worker has two superiors, one of them is immediate and one not immediate. The chief is a superior to all the workers except himself.

Some of the workers were in a hurry and made a mistake. You are to find the minimum number of workers that could make a mistake.

Input

The first line contains two positive integers n and s ($1 \le n \le 2 \cdot 10^5$, $1 \le s \le n$) — the number of workers and the id of the chief.

The second line contains n integers $a_1, a_2, ..., a_n$ ($0 \le a_i \le n$ - 1), where a_i is the number of superiors (not only immediate) the worker with id i reported about.

Output

Print the minimum number of workers that could make a mistake.

Examples

input
3 2 2 0 2
output
1

input
5 3 1 0 0 4 1
output
2

Note

In the first example it is possible that only the first worker made a mistake. Then:

- the immediate superior of the first worker is the second worker,
- the immediate superior of the third worker is the first worker,
- the second worker is the chief.

F. Financiers Game

time limit per test: 2 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

This problem has unusual memory constraint.

At evening, Igor and Zhenya the financiers became boring, so they decided to play a game. They prepared n papers with the income of some company for some time periods. Note that the income can be positive, zero or negative.

Igor and Zhenya placed the papers in a row and decided to take turns making moves. Igor will take the papers from the left side, Zhenya will take the papers from the right side. Igor goes first and takes 1 or 2 (on his choice) papers from the left. Then, on each turn a player can take k or $k + 1$ papers from his side if the opponent took exactly k papers in the previous turn. Players can't skip moves. The game ends when there are no papers left, or when some of the players can't make a move.

Your task is to determine the difference between the sum of incomes on the papers Igor took and the sum of incomes on the papers Zhenya took, assuming both players play optimally. Igor wants to maximize the difference, Zhenya wants to minimize it.

Input

The first line contains single positive integer n ($1 \le n \le 4000$) — the number of papers.

The second line contains n integers $a_1, a_2, ..., a_n$ ($-10^5 \le a_i \le 10^5$), where a_i is the income on the i-th paper from the left.

Output

Print the difference between the sum of incomes on the papers Igor took and the sum of incomes on the papers Zhenya took, assuming both players play optimally. Igor wants to maximize the difference, Zhenya wants to minimize it.

Examples

input
3 1 3 1
output
4

input
5 -1 -2 -1 -2 -1
output
0

input
4 -4 -2 4 5
output
-13

Note

In the first example it's profitable for Igor to take two papers from the left to have the sum of the incomes equal to 4. Then Zhenya wouldn't be able to make a move since there would be only one paper, and he would be able to take only 2 or 3..