Codeforces Round #380 (Div. 1, Rated, Based on Technocup 2017 - Elimination Round 2)

A. 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 ($1 \le n \le 2\cdot10^5$, $1 \le k \le 2\cdot10^5$, $2 \le s \le 10^9$, $1 \le t \le 2\cdot10^9$) — 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 $g_1, g_2, ..., g_k$ ($1 \le g_i \le s - 1$) — 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.

B. 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.

C. 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.

D. 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..

E. Tanya is 5!

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

Tanya is now five so all her friends gathered together to celebrate her birthday. There are n children on the celebration, including Tanya.

The celebration is close to its end, and the last planned attraction is gaming machines. There are m machines in the hall, they are numbered 1 through m. Each of the children has a list of machines he wants to play on. Moreover, for each of the machines he knows the exact time he wants to play on it. For every machine, no more than one child can play on this machine at the same time.

It is evening already, so every adult wants to go home. To speed up the process, you can additionally rent second copies of each of the machines. To rent the second copy of the j-th machine, you have to pay p_j burles. After you rent a machine, you can use it for as long as you want.

How long it will take to make every child play according to his plan, if you have a budget of b burles for renting additional machines? There is only one copy of each machine, so it's impossible to rent a third machine of the same type.

The children can interrupt the game in any moment and continue it later. If the i-th child wants to play on the j-th machine, it is allowed after you rent the copy of the j-th machine that this child would play some part of the time on the j-th machine and some part of the time on its copy (each of these parts could be empty). The interruptions and changes take no time and can be performed in any integer moment of time. Of course, a child can't play on more than one machine at the same time.

Remember, that it is not needed to save money (no one saves money at the expense of children happiness!), it is needed to minimize the latest moment of time some child ends his game.

Input

The first line contains three integers n, m and b ($1 \le n \le 40$, $1 \le m \le 10$, $0 \le b \le 10^6$) — the number of children, the number of gaming machines and the budget for renting additional machines.

The second line contains m integers $p_1, p_2, ..., p_m$ ($1 \le p_j \le 10^6$), where p_j is the rent price for the second copy of the j-th machine.

n lines follow, i-th of them describes the wishes of the i-th child. The line starts with an integer k_i ($0 \le k_i \le m$) — the number of machines, the i-th child wants to play on. Then there are k_i pairs in the line, the y-th of them is x_{iy}, t_{iy}. It means that, the i-th child wants to play t_{iy} ($1 \le t_{iy} \le 2500$) minutes on the x_{iy}-th ($1 \le x_{iy} \le m$) machine. In each of these n lines the values x_{iy} are distinct.

Output

In the first line print the minimum time in which all the children can finish their games.

In the second line print a string of length m consisting of zeros and ones. The j-th character is '1', if the copy of j-th machine should be rated, and '0' otherwise.

In the third line print integer g ($0 \le g \le 10^6$) — the total number of time segments of continuous playing for all of the children. Then in g lines print the segments as four integers i, j, s, d, meaning that the i-th child was playing on the j-th machine or its copy from the time moment s ($s \ge 0$) for d minutes ($d \ge 1$). You can print these lines in arbitrary order.

If there are multiple answers, print any of them.

Examples

input

```
2 2 100
3 7
2 1 3 2 1
2 1 3 2 1
```

output

```
4
10
8
1 1 0 1
2 2 0 1
1 1 1 1
2 1 1 1
2 1 2 1
1 1 2 1
1 2 3 1
2 1 3 1
```

input

```
3 2 15
11 7
2 2 10 1 5
1 2 20
2 1 4 2 3
```

output

```
20
```

```
01
17
2 2 0 4
2 2 4 1
1 1 5 2
2 2 5 2
1 2 7 5
2 2 7 5
2 2 12 1
1 2 12 1
3 1 13 4
2 2 13 4
1 2 13 4
1 1 17 2
3 2 17 2
2 2 17 2
1 1 19 1
2 2 19 1
3 2 19 1
```

F. Dirty plates

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

After one of celebrations there is a stack of dirty plates in Nikita's kitchen. Nikita has to wash them and put into a dryer. In dryer, the plates should be also placed in a stack also, and the plates sizes should increase down up. The sizes of all plates are distinct.

Nikita has no so much free space, specifically, he has a place for only one more stack of plates. Therefore, he can perform only such two operations:

- Take any number of plates from 1 to a from the top of the dirty stack, wash them and put them to the *intermediate* stack.
- Take any number of plates from 1 to b from the top of the intermediate stack and put them to the stack in the dryer.

Note that after performing each of the operations, the plates are put in the same order as they were before the operation.

You are given the sizes of the plates $s_1, s_2, ..., s_n$ in the down up order in the dirty stack, and integers a and b. All the sizes are distinct. Write a program that determines whether or not Nikita can put the plates in increasing down up order in the dryer. If he is able to do so, the program should find some sequence of operations (not necessary optimal) to achieve it.

Input

The first line contains three integers n, a and b ($1 \le n \le 2000$, $1 \le a, b \le n$). The second line contains integers $s_1, s_2, ..., s_n$ ($1 \le s_i \le n$) — the sizes of the plates in down up order. All the sizes are distinct.

Output

In the first line print "YES" if there is a solution. In this case, in the second line print integer k — the number of operations. Then in k lines print the operations, one per line. Each operation is described by two integers t_j and c_j, where $t_j = 1$, if the operation is to wash the top c_j places from the dirty stack and put them onto the intermediate stack, and $t_j = 2$, if the operation is to move th top c_j plates from the intermediate stack to the dryer.

In case there is no solution, print single line "NO".

If there are multiple solutions, print any of them. Note that it is not necessary to minimize the number of operations.

Examples

input

```
6 2 3
2 3 6 4 1 5
```

output

```
YES
8
1 2
1 1
2 1
1 2
1 1
2 1
2 1
2 3
```

input

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

output

```
YES
2
1 7
2 7
```

input

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

output

```
YES
14
1 1
1 1
1 1
1 1
1 1
1 1
1 1
2 1
2 1
2 1
2 1
2 1
```

```
2 1
2 1
```

input

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

output

```
NO
```

Note

In the first example the initial order of plates was $2, 3, 6, 4, 1, 5$. Here is how the stacks look like after each of the operations:

- [1 2]: Dirty stack: $6, 4, 1, 5$. Intermediary stack: $2, 3$. The dryer is empty.
- [1 1]: Dirty stack: $4, 1, 5$. Intermediary stack: $6, 2, 3$. The dryer is empty.
- [2 1]: Dirty stack: $4, 1, 5$. Intermediary stack: $2, 3$. Dryer stack: 6.
- [1 2]: Dirty stack: 5. Intermediary stack: $4, 1, 2, 3$. Dryer stack: 6.
- [1 1]: There are no dirty plates. Intermediary stack: $5, 4, 1, 2, 3$. Dryer stack: 6.
- [2 1]: There are no dirty plates. Intermediary stack: $4, 1, 2, 3$. Dryer stack: $5, 6$.
- [2 1]: There are no dirty plates. Intermediary stack: $1, 2, 3$. Dryer stack: $4, 5, 6$.
- [2 3]: All the plates are in the dryer: $1, 2, 3, 4, 5, 6$.

In the second example it is possible to wash all the plates in one operation, and then move them all to the dryer.
This is not possible in the third example, because it is not permitted to move more than one plate at the same time. It is possible to wash plates one by one so that they are placed onto the intermediary stack in the reverse order, and then move plates one by one to the dryer. The final order is correct.