A. Yellow Cards

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

The final match of the Berland Football Cup has been held recently. The referee has shown n yellow cards throughout the match. At the beginning of the match there were a_1 players in the first team and a_2 players in the second team.

The rules of sending players off the game are a bit different in Berland football. If a player from the first team receives k_1 yellow cards throughout the match, he can no longer participate in the match — he's sent off. And if a player from the second team receives k_2 yellow cards, he's sent off. After a player leaves the match, he can no longer receive any yellow cards. Each of n yellow cards was shown to exactly one player. Even if all players from one team (or even from both teams) leave the match, the game still continues.

The referee has lost his records on who has received each yellow card. Help him to determine the minimum and the maximum number of players that could have been thrown out of the game.

Input

The first line contains one integer a_1 $(1 \le a_1 \le 1\,000)$ — the number of players in the first team.

The second line contains one integer a_2 $(1 \le a_2 \le 1\,000)$ — the number of players in the second team.

The third line contains one integer k_1 $(1 \le k_1 \le 1\,000)$ — the maximum number of yellow cards a player from the first team can receive (after receiving that many yellow cards, he leaves the game).

The fourth line contains one integer k_2 $(1 \le k_2 \le 1\,000)$ — the maximum number of yellow cards a player from the second team can receive (after receiving that many yellow cards, he leaves the game).

The fifth line contains one integer n $(1 \le n \le a_1 \cdot k_1 + a_2 \cdot k_2)$ — the number of yellow cards that have been shown during the match.

Output

Print two integers — the minimum and the maximum number of players that could have been thrown out of the game.

Examples

input

```
2
3
5
1
8
```

output

```
0 4
```

input

```
3
1
6
7
25
```

output

```
4 4
```

input

```
6
4
9
10
89
```

output

```
5 9
```

Note

In the first example it could be possible that no player left the game, so the first number in the output is 0. The maximum possible number of players that could have been forced to leave the game is 4 — one player from the first team, and three players from the second.

In the second example the maximum possible number of yellow cards has been shown $(3 \cdot 6 + 1 \cdot 7 = 25)$, so in any case all players were sent off.

B. The Number of Products

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

You are given a sequence $a_1, a_2, \dots, a_n$ consisting of n non-zero integers (i.e. $a_i \ne 0$).

You have to calculate two following values:

1. the number of pairs of indices (l, r) $(l \le r)$ such that $a_l \cdot a_{l+1} \dots a_{r-1} \cdot a_r$ is negative;
2. the number of pairs of indices (l, r) $(l \le r)$ such that $a_l \cdot a_{l+1} \dots a_{r-1} \cdot a_r$ is positive;

Input

The first line contains one integer n $(1 \le n \le 2 \cdot 10^5)$ — the number of elements in the sequence.

The second line contains n integers $a_1, a_2, \dots, a_n$ $(-10^9 \le a_i \le 10^9; a_i \ne 0)$ — the elements of the sequence.

Output

Print two integers — the number of subsegments with negative product and the number of subsegments with positive product, respectively.

Examples

input

```
5
5 -3 3 -1 1
```

output

```
8 7
```

input

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

output

```
28 27
```

input

```
5
-1 -2 -3 -4 -5
```

output

```
9 6
```

C. Swap Letters

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

Monocarp has got two strings s and t having equal length. Both strings consist of lowercase Latin letters "a" and "b".

Monocarp wants to make these two strings s and t equal to each other. He can do the following operation any number of times: choose an index pos_1 in the string s, choose an index pos_2 in the string t, and swap s_{pos_1} with t_{pos_2}.

You have to determine the minimum number of operations Monocarp has to perform to make s and t equal, and print any optimal sequence of operations — or say that it is impossible to make these strings equal.

Input

The first line contains one integer n $(1 \le n \le 2 \cdot 10^5)$ — the length of s and t.

The second line contains one string s consisting of n characters "a" and "b".

The third line contains one string t consisting of n characters "a" and "b".

Output

If it is impossible to make these strings equal, print -1.

Otherwise, in the first line print k — the minimum number of operations required to make the strings equal. In each of the next k

lines print two integers — the index in the string s and the index in the string t that should be used in the corresponding swap operation.

Examples

input

```
4
abab
aabb
```

output

```
2
3 3
3 2
```

input

```
1
a
b
```

output

```
-1
```

input

```
8
babbaabb
abababaa
```

output

```
3
2 6
1 3
7 8
```

Note

In the first example two operations are enough. For example, you can swap the third letter in s with the third letter in t. Then $s =$ "abbb", $t =$ "aaab". Then swap the third letter in s and the second letter in t. Then both s and t are equal to "abab".

In the second example it's impossible to make two strings equal.

D. Ticket Game

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

Monocarp and Bicarp live in Berland, where every bus ticket consists of n digits (n is an even number). During the evening walk Monocarp and Bicarp found a ticket where some of the digits have been erased. **The number of digits that have been erased is even**.

Monocarp and Bicarp have decided to play a game with this ticket. Monocarp hates happy tickets, while Bicarp collects them. A ticket is considered happy if the sum of the first $\frac{n}{2}$ digits of this ticket is equal to the sum of the last $\frac{n}{2}$ digits.

Monocarp and Bicarp take turns (and Monocarp performs the first of them). During each turn, the current player must replace any erased digit with any digit from 0 to 9. The game ends when there are no erased digits in the ticket.

If the ticket is happy after all erased digits are replaced with decimal digits, then Bicarp wins. Otherwise, Monocarp wins. You have to determine who will win if both players play optimally.

Input

The first line contains one **even** integer n $(2 \le n \le 2 \cdot 10^5)$ — the number of digits in the ticket.

The second line contains a string of n digits and "?" characters — the ticket which Monocarp and Bicarp have found. If the i-th character is "?", then the i-th digit is erased. Note that there may be leading zeroes. The number of "?" characters is even.

Output

If Monocarp wins, print "`Monocarp`" (without quotes). Otherwise print "`Bicarp`" (without quotes).

Examples

input

```
4
0523
```

output

```
Bicarp
```

input

```
2
??
```

output

```
Bicarp
```

input

```
8
?054??0?
```

output

```
Bicarp
```

input

```
6
???00?
```

output

```
Monocarp
```

Note

Since there is no question mark in the ticket in the first example, the winner is determined before the game even starts, and it is Bicarp.

In the second example, Bicarp also wins. After Monocarp chooses an erased digit and replaces it with a new one, Bicap can choose another position with an erased digit and replace it with the same digit, so the ticket is happy.

E. Marbles

time limit per test: 4 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

Monocarp has arranged n colored marbles in a row. The color of the i-th marble is a_i. Monocarp likes ordered things, so he wants to rearrange marbles in such a way that all marbles of the same color form a contiguos segment (and there is only one such segment for each color).

In other words, Monocarp wants to rearrange marbles so that, for every color j, if the leftmost marble of color j is l-th in the row, and the rightmost marble of this color has position r in the row, then every marble from l to r has color j.

To achieve his goal, Monocarp can do the following operation any number of times: choose two neighbouring marbles, and swap them.

You have to calculate the minimum number of operations Monocarp has to perform to rearrange the marbles. Note that the order of segments of marbles having equal color does not matter, it is only required that, for every color, all the marbles of this color form exactly one contiguous segment.

Input

The first line contains one integer n $(2 \le n \le 4 \cdot 10^5)$ — the number of marbles.

The second line contains an integer sequence $a_1, a_2, \dots, a_n$ $(1 \le a_i \le 20)$, where a_i is the color of the i-th marble.

Output

Print the minimum number of operations Monocarp has to perform to achieve his goal.

Examples

input

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

output

```
3
```

input

```
5
20 1 14 10 2
```

output

```
0
```

input

```
13
5 5 4 4 3 5 7 6 5 4 4 6 5
```

output

```
21
```

Note

In the first example three operations are enough. Firstly, Monocarp should swap the third and the fourth marbles, so the sequence of colors is $[3, 4, 3, 2, 4, 2, 2]$. Then Monocarp should swap the second and the third marbles, so the sequence is $[3, 3, 4, 2, 4, 2, 2]$. And finally, Monocarp should swap the fourth and the fifth marbles, so the sequence is $[3, 3, 4, 4, 2, 2, 2]$.

In the second example there's no need to perform any operations.

F. Radio Stations

time limit per test: 7 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

In addition to complaints about lighting, a lot of complaints about insufficient radio signal covering has been received by Bertown city hall recently. n complaints were sent to the mayor, all of which are suspiciosly similar to each other: in the i-th complaint, one of the radio fans has mentioned that the signals of two radio stations x_i and y_i are not covering some parts of the city, and demanded that the signal of **at least one** of these stations can be received in the whole city.

Of cousre, the mayor of Bertown is currently working to satisfy these complaints. A new radio tower has been installed in Bertown, it can transmit a signal with any integer power from 1 to M (let's denote the signal power as f). The mayor has decided that he will choose a set of radio stations and establish a contract with every chosen station. To establish a contract with the i-th station, the following conditions should be met:

- the signal power f should be not less than l_i, otherwise the signal of the i-th station won't cover the whole city;
- the signal power f should be not greater than r_i, otherwise the signal will be received by the residents of other towns which haven't established a contract with the i-th station.

All this information was already enough for the mayor to realise that choosing the stations is hard. But after consulting with specialists, he learned that some stations the signals of some stations may interfere with each other: there are m pairs of stations (u_i, v_i) that use the same signal frequencies, and for each such pair it is impossible to establish contracts with both stations. **If stations x and y use the same frequencies, and y and z use the same frequencies, it does not imply that x and z use the same frequencies**.

The mayor finds it really hard to analyze this situation, so he hired you to help him. You have to choose signal power f and a set of stations to establish contracts with such that:

- all complaints are satisfied (formally, for every $i \in [1, n]$ the city establishes a contract either with station x_i, or with station y_i);
- no two chosen stations interfere with each other (formally, for every $i \in [1, m]$ the city **does not** establish a contract either with station u_i, or with station v_i);
- for each chosen station, the conditions on signal power are met (formally, for each chosen station i the condition $l_i \le f \le r_i$ is met).

Input

The first line contains 4 integers n, p, M and m ($2 \le n, p, M, m \le 4 \cdot 10^5$) — the number of complaints, the number of radio stations, maximum signal power and the number of interfering pairs, respectively.

Then n lines follow, which describe the complains. Each line contains two integers x_i and y_i ($1 \le x_i < y_i \le p$) — the indices of the radio stations mentioned in the i-th complaint). **All complaints are distinct**.

Then p lines follow, which describe the radio stations. Each line contains two integers l_i and r_i ($1 \le l_i \le r_i \le M$) — the constrains on signal power that should be satisfied if the city establishes a contract with the i-th station.

Then m lines follow, which describe the pairs of interfering radio stations. Each line contains two integers u_i and v_i ($1 \le u_i < v_i \le p$) — the indices of interfering radio stations. **All these pairs are distinct**.

Output

If it is impossible to choose signal power and a set of stations to meet all conditions, print -1.

Otherwise print two integers k and f in the first line — the number of stations in the chosen set and the chosen signal power, respectively. In the second line print k **distinct** integers from 1 to p — the indices of stations to establish contracts with (in any order). If there are multiple answers, print any of them; you don't have to minimize/maximize the number of chosen stations, and the same applies to signal power.

Examples

input

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

output

```
2 3
1 3
```

input

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

output

```
-1
```