Educational Codeforces Round 4

A. The Text Splitting

1 second, 256 megabytes

You are given the string s of length n and the numbers p, q. Split the string s to pieces of length p and q.

For example, the string "Hello" for $p = 2, q = 3$ can be split to the two strings "Hel" and "lo" or to the two strings "He" and "llo".

Note it is allowed to split the string s to the strings only of length p or to the strings only of length q (see the second sample test).

Input

The first line contains three positive integers n, p, q ($1 \le p, q \le n \le 100$).

The second line contains the string s consists of lowercase and uppercase latin letters and digits.

Output

If it's impossible to split the string s to the strings of length p and q print the only number "-1".

Otherwise in the first line print integer k — the number of strings in partition of s.

Each of the next k lines should contain the strings in partition. Each string should be of the length p or q. The string should be in order of their appearing in string s — from left to right.

If there are several solutions print any of them.

input

```
5 2 3
Hello
```

output

```
2
He
llo
```

input

```
10 9 5
Codeforces
```

output

```
2
Codef
orces
```

input

```
6 4 5
Privet
```

output

```
-1
```

input

```
8 1 1
abacabac
```

output

```
8
a
b
a
c
a
b
a
c
```

B. HDD is Outdated Technology

1 second, 256 megabytes

HDD hard drives group data by sectors. All files are split to fragments and each of them are written in some sector of hard drive. Note the fragments can be written in sectors in arbitrary order.

One of the problems of HDD hard drives is the following: the magnetic head should move from one sector to another to read some file.

Find the time need to read file split to n fragments. The i-th sector contains the f_i-th fragment of the file ($1 \le f_i \le n$). Note different sectors contains the different fragments. At the start the magnetic head is in the position that contains the first fragment. The file are reading in the following manner: at first the first fragment is read, then the magnetic head moves to the sector that contains the second fragment, then the second fragment is read and so on until the n-th fragment is read. The fragments are read in the order from the first to the n-th.

It takes $|a - b|$ time units to move the magnetic head from the sector a to the sector b. Reading a fragment takes no time.

Input
The first line contains a positive integer n ($1 \le n \le 2 \cdot 10^5$) — the number of fragments.

The second line contains n different integers f_i ($1 \le f_i \le n$) — the number of the fragment written in the i-th sector.

Output
Print the only integer — the number of time units needed to read the file.

input
3 3 1 2
output
3

input
5 1 3 5 4 2
output
10

In the second example the head moves in the following way:

- 1->2 means movement from the sector 1 to the sector 5, i.e. it takes 4 time units
- 2->3 means movement from the sector 5 to the sector 2, i.e. it takes 3 time units
- 3->4 means movement from the sector 2 to the sector 4, i.e. it takes 2 time units
- 4->5 means movement from the sector 4 to the sector 3, i.e. it takes 1 time units

So the answer to the second example is $4 + 3 + 2 + 1 = 10$.

C. Replace To Make Regular Bracket Sequence

1 second, 256 megabytes

You are given string s consists of opening and closing brackets of four kinds <>, {}, [], (). There are two types of brackets: opening and closing. You can replace any bracket by another of the same type. For example, you can replace < by the bracket {, but you can't replace it by) or >.

The following definition of a regular bracket sequence is well-known, so you can be familiar with it.

Let's define a regular bracket sequence (RBS). Empty string is RBS. Let s_1 and s_2 be a RBS then the strings $<s_1>s_2$, $\{s_1\}s_2$, $[s_1]s_2$, $(s_1)s_2$ are also RBS.

For example the string "[[(){}]<>]" is RBS, but the strings "[)()" and "][()()" are not.

Determine the least number of replaces to make the string s RBS.

Input
The only line contains a non empty string s, consisting of only opening and closing brackets of four kinds. The length of s does not exceed 10^6.

Output
If it's impossible to get RBS from s print Impossible.

Otherwise print the least number of replaces needed to get RBS from s.

input
[<}){}
output
2

input
{()}[]

output
0

input
]]
output
Impossible

D. The Union of k-Segments

4 seconds, 256 megabytes

You are given n segments on the coordinate axis Ox and the number k. The point is *satisfied* if it belongs to at least k segments. Find the smallest (by the number of segments) set of segments on the coordinate axis Ox which contains all *satisfied* points and no others.

Input

The first line contains two integers n and k ($1 \le k \le n \le 10^6$) — the number of segments and the value of k.

The next n lines contain two integers l_i, r_i ($-10^9 \le l_i \le r_i \le 10^9$) each — the endpoints of the i-th segment. The segments can degenerate and intersect each other. The segments are given in arbitrary order.

Output

First line contains integer m — the smallest number of segments.

Next m lines contain two integers a_j, b_j ($a_j \le b_j$) — the ends of j-th segment in the answer. The segments should be listed in the order from left to right.

input
3 2 0 5 -3 2 3 8
output
2 0 2 3 5

input
3 2 0 5 -3 3 3 8
output
1 0 5

E. Square Root of Permutation

2 seconds, 256 megabytes

A *permutation* of length n is an array containing each integer from 1 to n exactly once. For example, $q = [4, 5, 1, 2, 3]$ is a permutation. For the permutation q the square of permutation is the permutation p that $p[i] = q[q[i]]$ for each $i = 1 \dots n$. For example, the square of $q = [4, 5, 1, 2, 3]$ is $p = q^2 = [2, 3, 4, 5, 1]$.

This problem is about the inverse operation: given the permutation p you task is to find such permutation q that $q^2 = p$. If there are several such q find any of them.

Input

The first line contains integer n ($1 \le n \le 10^6$) — the number of elements in permutation p.

The second line contains n distinct integers $p_1, p_2, \dots, p_n$ ($1 \le p_i \le n$) — the elements of permutation p.

Output

If there is no permutation q such that $q^2 = p$ print the number "-1".

If the answer exists print it. The only line should contain n different integers q_i ($1 \le q_i \le n$) — the elements of the permutation q. If there are several solutions print any of them.

input
4 2 1 4 3

output

```
3 4 2 1
```

input

```
4
2 1 3 4
```

output

```
-1
```

input

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

output

```
4 5 1 2 3
```

F. Simba on the Circle

1 second, 256 megabytes

You are given a circular array with n elements. The elements are numbered from some element with values from 1 to n in clockwise order. The i-th cell contains the value a_i. The robot Simba is in cell s.

Each moment of time the robot is in some of the n cells (at the begin he is in s). In one turn the robot can write out the number written in current cell or move to the adjacent cell in clockwise or counterclockwise direction. To write out the number from the cell Simba doesn't spend any time, but to move to adjacent cell Simba spends one unit of time.

Simba wants to write the number from each cell one time, so the numbers will be written in a non decreasing order. Find the least number of time units to write out all numbers.

Input

The first line contains two integers n and s $(1 \le s \le n \le 2000)$ — the number of cells in the circular array and the starting position of Simba.

The second line contains n integers a_i $(-10^9 \le a_i \le 10^9)$ — the number written in the i-th cell. The numbers are given for cells in order from 1 to n. Some of numbers a_i can be equal.

Output

In the first line print the number t — the least number of time units.

Each of the next n lines should contain the direction of robot movement and the number of cells to move in that direction. After that movement the robot writes out the number from the cell in which it turns out. The direction and the number of cells should be printed in the form of $+x$ in case of clockwise movement and $-x$ in case of counterclockwise movement to x cells $(0 \le x \le n-1)$.

Note that the sum of absolute values of x should be equal to t.

input

```
9 1
0 1 2 2 2 1 0 1 1
```

output

```
12
+0
-3
-1
+2
+1
+2
+1
+1
+1
```

input

```
8 1
0 1 0 1 0 1 0 1
```

output

```
13
+0
+2
+2
+2
-1
+2
+2
+2
```

input

```
8 1
1 2 3 4 5 6 7 8
```

output

```
7
+0
+1
+1
+1
+1
+1
+1
+1
```

output

```
7
+0
+1
+1
+1
+1
+1
+1
+1
```

input

```
8 1
0 0 0 0 0 0 0 0
```