F. Simba on the Circle

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

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.

Examples

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
```

input

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

output

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