A. Helpful Maths

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

Xenia the beginner mathematician is a third year student at elementary school. She is now learning the addition operation.

The teacher has written down the sum of multiple numbers. Pupils should calculate the sum. To make the calculation easier, the sum only contains numbers 1, 2 and 3. Still, that isn't enough for Xenia. She is only beginning to count, so she can calculate a sum only if the summands follow in non-decreasing order. For example, she can't calculate sum 1+3+2+1 but she can calculate sums 1+1+2 and 3+3.

You've got the sum that was written on the board. Rearrange the summans and print the sum in such a way that Xenia can calculate the sum.

Input

The first line contains a non-empty string s — the sum Xenia needs to count. String s contains no spaces. It only contains digits and characters "+". Besides, string s is a correct sum of numbers 1, 2 and 3. String s is at most 100 characters long.

Output

Print the new sum that Xenia can count.

Sample test(s)

input
```
3+2+1
```
output
```
1+2+3
```

input
```
1+1+3+1+3
```
output
```
1+1+1+3+3
```

input
```
2
```
output
```
2
```

B. Xenia and Ringroad

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

Xenia lives in a city that has n houses built along the main ringroad. The ringroad houses are numbered 1 through n in the clockwise order. The ringroad traffic is one way and also is clockwise.

Xenia has recently moved into the ringroad house number 1. As a result, she's got m things to do. In order to complete the i-th task, she needs to be in the house number a_i and complete all tasks with numbers less than i. Initially, Xenia is in the house number 1, find the minimum time she needs to complete all her tasks if moving from a house to a neighboring one along the ringroad takes one unit of time.

Input

The first line contains two integers n and m $(2 \le n \le 10^5, 1 \le m \le 10^5)$. The second line contains m integers $a_1, a_2, ..., a_m$ $(1 \le a_i \le n)$. Note that Xenia can have multiple consecutive tasks in one house.

Output

Print a single integer — the time Xenia needs to complete all tasks.

Please, do not use the `%lld` specifier to read or write 64-bit integers in C++. It is preferred to use the `cin`, `cout` streams or the `%I64d` specifier.

Sample test(s)

input

```
4 3
3 2 3
```

output

```
6
```

input

```
4 3
2 3 3
```

output

```
2
```

Note

In the first test example the sequence of Xenia's moves along the ringroad looks as follows: $1 \to 2 \to 3 \to 4 \to 1 \to 2 \to 3$. This is optimal sequence. So, she needs 6 time units.

C. Xenia and Weights

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

Xenia has a set of weights and pan scales. Each weight has an integer weight from 1 to 10 kilos. Xenia is going to play with scales and weights a little. For this, she puts weights on the scalepans, one by one. The first weight goes on the left scalepan, the second weight goes on the right scalepan, the third one goes on the left scalepan, the fourth one goes on the right scalepan and so on. Xenia wants to put the total of m weights on the scalepans.

Simply putting weights on the scales is not interesting, so Xenia has set some rules. First, she does not put on the scales two consecutive weights of the same weight. That is, the weight that goes i-th should be different from the $(i+1)$-th weight for any i $(1 \leq i < m)$. Second, every time Xenia puts a weight on some scalepan, she wants this scalepan to outweigh the other one. That is, the sum of the weights on the corresponding scalepan must be strictly greater than the sum on the other pan.

You are given all types of weights available for Xenia. You can assume that the girl has an infinite number of weights of each specified type. Your task is to help Xenia lay m weights on the scales or to say that it can't be done.

Input

The first line contains a string consisting of exactly ten zeroes and ones: the i-th $(i \geq 1)$ character in the line equals "`1`" if Xenia has i kilo weights, otherwise the character equals "`0`". The second line contains integer m $(1 \leq m \leq 1000)$.

Output

In the first line print "`YES`", if there is a way to put m weights on the scales by all rules. Otherwise, print in the first line "`NO`". If you can put m weights on the scales, then print in the next line m integers — the weights' weights in the order you put them on the scales.

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

Sample test(s)

input

```
0000000101
3
```

output

```
YES
8 10 8
```

input

```
1000000000
2
```

output

```
NO
```

D. Xenia and Bit Operations

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

Xenia the beginner programmer has a sequence a, consisting of 2^n non-negative integers: $a_1, a_2, ..., a_{2^n}$. Xenia is currently studying bit operations. To better understand how they work, Xenia decided to calculate some value v for a.

Namely, it takes several iterations to calculate value v. At the first iteration, Xenia writes a new sequence $a_1\ or\ a_2, a_3\ or\ a_4, ..., a_{2^n - 1}\ or\ a_{2^n}$, consisting of 2^{n-1} elements. In other words, she writes down the bit-wise OR of adjacent elements of sequence a. At the second iteration, Xenia writes the bitwise **exclusive** OR of adjacent elements of the sequence obtained after the first iteration. At the third iteration Xenia writes the bitwise OR of the adjacent elements of the sequence obtained after the second iteration. And so on; the operations of bitwise exclusive OR and bitwise OR alternate. In the end, she obtains a sequence consisting of one element, and that element is v.

Let's consider an example. Suppose that sequence $a = (1, 2, 3, 4)$. Then let's write down all the transformations $(1, 2, 3, 4) \rightarrow (1\ or\ 2 = 3, 3\ or\ 4 = 7) \rightarrow (3\ xor\ 7 = 4)$. The result is $v = 4$.

You are given Xenia's initial sequence. But to calculate value v for a given sequence would be too easy, so you are given additional m queries. Each query is a pair of integers p, b. Query p, b means that you need to perform the assignment $a_p = b$. After each query, you need to print the new value v for the new sequence a.

Input

The first line contains two integers n and m $(1 \le n \le 17, 1 \le m \le 10^5)$. The next line contains 2^n integers $a_1, a_2, ..., a_{2^n}$ $(0 \le a_i < 2^{30})$. Each of the next m lines contains queries. The i-th line contains integers p_i, b_i $(1 \le p_i \le 2^n, 0 \le b_i < 2^{30})$ — the i-th query.

Output

Print m integers — the i-th integer denotes value v for sequence a after the i-th query.

Sample test(s)

input

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

output

```
1
3
3
3
```

Note

For more information on the bit operations, you can follow this link: `http://en.wikipedia.org/wiki/Bitwise_operation`

E. Three Swaps

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

Xenia the horse breeder has n $(n > 1)$ horses that stand in a row. Each horse has its own unique number. Initially, the i-th left horse has number i. That is, the sequence of numbers of horses in a row looks as follows (from left to right): 1, 2, 3, ..., n.

Xenia trains horses before the performance. During the practice sessions, she consistently gives them commands. Each command is a pair of numbers l, r $(1 \le l < r \le n)$. The command l, r means that the horses that are on the l-th, $(l+1)$-th, $(l+2)$-th, ..., r-th places from the left must be rearranged. The horses that initially stand on the l-th and r-th places will swap. The horses on the $(l+1)$-th and $(r-1)$-th places will swap. The horses on the $(l+2)$-th and $(r-2)$-th places will swap and so on. In other words, the horses that were on the segment $[l, r]$ change their order to the reverse one.

For example, if Xenia commanded $l = 2, r = 5$, and the sequence of numbers of horses before the command looked as (2, 1, 3, 4, 5, 6), then after the command the sequence will be (2, 5, 4, 3, 1, 6).

We know that during the practice Xenia gave at most three commands of the described form. You have got the final sequence of numbers of horses by the end of the practice. Find what commands Xenia gave during the practice. Note that you do not need to minimize the number of commands in the solution, find any valid sequence of at most three commands.

Input

The first line contains an integer n $(2 \le n \le 1000)$ — the number of horses in the row. The second line contains n distinct integers $a_1, a_2, ..., a_n$ $(1 \le a_i \le n)$, where a_i is the number of the i-th left horse in the row after the practice.

Output

The first line should contain integer k $(0 \le k \le 3)$ — the number of commads Xenia gave during the practice. In each of the next k lines print two integers. In the i-th line print numbers l_i, r_i $(1 \le l_i < r_i \le n)$ — Xenia's i-th command during the practice.

It is guaranteed that a solution exists. If there are several solutions, you are allowed to print any of them.

Sample test(s)

input

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

output

```
1
2 4
```

input

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

output

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