week2- Arrays Manipulation

A. Anton and Letters

2 seconds, 256 megabytes

Recently, Anton has found a set. The set consists of small English letters. Anton carefully wrote out all the letters from the set in one line, separated by a comma. He also added an opening curved bracket at the beginning of the line and a closing curved bracket at the end of the line.

Unfortunately, from time to time Anton would forget writing some letter and write it again. He asks you to count the total number of distinct letters in his set.

Input

The first and the single line contains the set of letters. The length of the line doesn't exceed 1000. It is guaranteed that the line starts from an opening curved bracket and ends with a closing curved bracket. Between them, small English letters are listed, separated by a comma. Each comma is followed by a space.

Output

Print a single number — the number of distinct letters in Anton's set.

input

```
{a, b, c}
```

output

```
3
```

input

```
{b, a, b, a}
```

output

```
2
```

input

```
{}
```

output

```
0
```

B. Xenia and Ringroad

2 seconds, 256 megabytes

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.

input

```
4 3
3 2 3
```

output

```
6
```

input

```
4 3
2 3 3
```

output

```
2
```

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. Sale

2 seconds, 256 megabytes

Once Bob got to a sale of old TV sets. There were n TV sets at that sale. TV set with index i costs a_i bellars. Some TV sets have a negative price — their owners are ready to pay Bob if he buys their useless apparatus. Bob can «buy» any TV sets he wants. Though he's very strong, Bob can carry at most m TV sets, and he has no desire to go to the sale for the second time. Please, help Bob find out the maximum sum of money that he can earn.

Input

The first line contains two space-separated integers n and m $(1 \le m \le n \le 100)$ — amount of TV sets at the sale, and amount of TV sets that Bob can carry. The following line contains n space-separated integers a_i $(-1000 \le a_i \le 1000)$ — prices of the TV sets.

Output

Output the only number — the maximum sum of money that Bob can earn, given that he can carry at most m TV sets.

input

```
5 3
-6 0 35 -2 4
```

output

```
8
```

input

```
4 2
7 0 0 -7
```

output

```
7
```

D. Points in Segments

1 second, 256 megabytes

You are given a set of n segments on the axis Ox, each segment has integer endpoints between 1 and m inclusive. Segments may intersect, overlap or even coincide with each other. Each segment is characterized by two integers l_i and r_i $(1 \le l_i \le r_i \le m)$ — coordinates of the left and of the right endpoints.

Consider all integer points between 1 and m inclusive. Your task is to print all such points that don't belong to any segment. The point x belongs to the segment $[l; r]$ if and only if $l \le x \le r$.

Input

The first line of the input contains two integers n and m ($1 \le n, m \le 100$) — the number of segments and the upper bound for coordinates.

The next n lines contain two integers each l_i and r_i $(1 \le l_i \le r_i \le m)$ — the endpoints of the i-th segment. Segments may intersect, overlap or even coincide with each other. Note, it is possible that $l_i = r_i$, i.e. a segment can degenerate to a point.

Output

In the first line print one integer k — the number of points that don't belong to any segment.

In the second line print exactly k integers in *any* order — the points that don't belong to any segment. All points you print should be distinct.

If there are no such points at all, print a single integer 0 in the first line and either leave the second line empty or do not print it at all.

input

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

output

```
2
3 4
```

input

```
1 7
1 7
```

output

```
0
```

In the first example the point 1 belongs to the second segment, the point 2 belongs to the first and the second segments and the point 5 belongs to the third segment. The points 3 and 4 do not belong to any segment.

In the second example all the points from 1 to 7 belong to the first segment.

E. Diverse Team

1 second, 256 megabytes

There are n students in a school class, the rating of the i-th student on Codehorses is a_i. You have to form a team consisting of k students ($1 \le k \le n$) such that the ratings of all team members **are distinct**.

If it is impossible to form a suitable team, print "NO" (without quotes). Otherwise print "YES", and then print k distinct numbers which should be the indices of students in the team you form. If there are multiple answers, print any of them.

Input

The first line contains two integers n and k $(1 \le k \le n \le 100)$ — the number of students and the size of the team you have to form.

The second line contains n integers $a_1, a_2, \dots, a_n$ $(1 \le a_i \le 100)$, where a_i is the rating of i-th student.

Output

If it is impossible to form a suitable team, print "NO" (without quotes). Otherwise print "YES", and then print k distinct integers from 1 to n which should be the indices of students in the team you form. All the ratings of the students in the team should be distinct. You may print the indices in any order. If there are multiple answers, print any of them.

Assume that the students are numbered from 1 to n.

input

```
5 3
15 13 15 15 12
```

output

```
YES
1 2 5
```

input

```
5 4
15 13 15 15 12
```

output

```
NO
```

input

```
4 4
20 10 40 30
```

output

```
YES
1 2 3 4
```

All possible answers for the first example:

- {1 2 5}
- {2 3 5}
- {2 4 5}

Note that the order does not matter.

F. Beautiful Matrix

2 seconds, 256 megabytes

You've got a 5×5 matrix, consisting of 24 zeroes and a single number one. Let's index the matrix rows by numbers from 1 to 5 from top to bottom, let's index the matrix columns by numbers from 1 to 5 from left to right. In one move, you are allowed to apply one of the two following transformations to the matrix:

1. Swap two neighboring matrix rows, that is, rows with indexes i and $i+1$ for some integer i $(1 \le i < 5)$.
2. Swap two neighboring matrix columns, that is, columns with indexes j and $j+1$ for some integer j $(1 \le j < 5)$.

You think that a matrix looks *beautiful*, if the single number one of the matrix is located in its middle (in the cell that is on the intersection of the third row and the third column). Count the minimum number of moves needed to make the matrix beautiful.

Input

The input consists of five lines, each line contains five integers: the j-th integer in the i-th line of the input represents the element of the matrix that is located on the intersection of the i-th row and the j-th column. It is guaranteed that the matrix consists of 24 zeroes and a single number one.

Output

Print a single integer — the minimum number of moves needed to make the matrix beautiful.

input

```
0 0 0 0 0
0 0 0 0 1
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
```

output

```
3
```

input

```
0 0 0 0 0
0 0 0 0 0
0 1 0 0 0
0 0 0 0 0
0 0 0 0 0
```

output

```
1
```