Educational Codeforces Round 5

A. Comparing Two Long Integers

2 seconds, 256 megabytes

You are given two very long integers a, b (leading zeroes are allowed). You should check what number a or b is greater or determine that they are equal.

The input size is very large so don't use the reading of symbols one by one. Instead of that use the reading of a whole line or token.

As input/output can reach huge size it is recommended to use fast input/output methods: for example, prefer to use `scanf/printf` instead of `cin/cout` in C++, prefer to use `BufferedReader/PrintWriter` instead of `Scanner/System.out` in `Java`. Don't use the function `input()` in `Python2` instead of it use the function `raw_input()`.

Input

The first line contains a non-negative integer a.

The second line contains a non-negative integer b.

The numbers a, b may contain leading zeroes. Each of them contains no more than 10^6 digits.

Output

Print the symbol "<" if $a < b$ and the symbol ">" if $a > b$. If the numbers are equal print the symbol "=".

input
9 10
output
<

input
11 10
output
>

input
00012345 12345
output
=

input
0123 9
output
>

input
0123 111
output
>

B. Dinner with Emma

1 second, 256 megabytes

Jack decides to invite Emma out for a dinner. Jack is a modest student, he doesn't want to go to an expensive restaurant. Emma is a girl with high taste, she prefers elite places.

Munhattan consists of n streets and m avenues. There is exactly one restaurant on the intersection of each street and avenue. The streets are numbered with integers from 1 to n and the avenues are numbered with integers from 1 to m. The cost of dinner in the restaurant at the intersection of the i-th street and the j-th avenue is c_{ij}.

Jack and Emma decide to choose the restaurant in the following way. Firstly Emma chooses the street to dinner and then Jack chooses the avenue. Emma and Jack makes their choice optimally: Emma wants to maximize the cost of the dinner, Jack wants to minimize it. Emma takes into account that Jack wants to minimize the cost of the dinner. Find the cost of the dinner for the couple in love.

Input
The first line contains two integers n, m $(1 \le n, m \le 100)$ — the number of streets and avenues in Munhattan.

Each of the next n lines contains m integers c_{ij} $(1 \le c_{ij} \le 10^9)$ — the cost of the dinner in the restaurant on the intersection of the i-th street and the j-th avenue.

Output
Print the only integer a — the cost of the dinner for Jack and Emma.

input
3 4 4 1 3 5 2 2 2 2 5 4 5 1
output
2

input
3 3 1 2 3 2 3 1 3 1 2
output
1

In the first example if Emma chooses the first or the third streets Jack can choose an avenue with the cost of the dinner 1. So she chooses the second street and Jack chooses any avenue. The cost of the dinner is 2.

In the second example regardless of Emma's choice Jack can choose a restaurant with the cost of the dinner 1.

C. The Labyrinth

1 second, 256 megabytes

You are given a rectangular field of $n \times m$ cells. Each cell is either empty or impassable (contains an obstacle). Empty cells are marked with '.', impassable cells are marked with '*'. Let's call two empty cells `adjacent` if they share a side.

Let's call a *connected component* any non-extendible set of cells such that any two of them are connected by the path of adjacent cells. It is a typical well-known definition of a connected component.

For each impassable cell (x, y) imagine that it is an empty cell (all other cells remain unchanged) and find the size (the number of cells) of the connected component which contains (x, y). You should do it for each impassable cell independently.

The answer should be printed as a matrix with n rows and m columns. The j-th symbol of the i-th row should be "." if the cell is empty at the start. Otherwise the j-th symbol of the i-th row should contain the only digit —- the answer modulo 10. The matrix should be printed without any spaces.

To make your output faster it is recommended to build the output as an array of n strings having length m and print it as a sequence of lines. It will be much faster than writing character-by-character.

As input/output can reach huge size it is recommended to use fast input/output methods: for example, prefer to use `scanf/printf` instead of `cin/cout` in C++, prefer to use `BufferedReader/PrintWriter` instead of `Scanner/System.out` in `Java`.

Input
The first line contains two integers n, m $(1 \le n, m \le 1000)$ — the number of rows and columns in the field.

Each of the next n lines contains m symbols: "." for empty cells, "*" for impassable cells.

Output
Print the answer as a matrix as described above. See the examples to precise the format of the output.

input
3 3 *.* .*. *.*

output
3.3 .5. 3.3

input
4 5 **..* ..*** .*.*. *.*.*

output
46..3 ..732 .6.4. 5.4.3

In first example, if we imagine that the central cell is empty then it will be included to component of size 5 (cross). If any of the corner cell will be empty then it will be included to component of size 3 (corner).

D. Longest k-Good Segment

1 second, 256 megabytes

The array a with n integers is given. Let's call the sequence of one or more consecutive elements in a `segment`. Also let's call the segment *k*-`good` if it contains no more than k different values.

Find any longest *k*-good segment.

As the input/output can reach huge size it is recommended to use fast input/output methods: for example, prefer to use `scanf/printf` instead of `cin/cout` in C++, prefer to use `BufferedReader/PrintWriter` instead of `Scanner/System.out` in `Java`.

Input

The first line contains two integers n, k ($1 \le k \le n \le 5 \cdot 10^5$) — the number of elements in a and the parameter k.

The second line contains n integers a_i ($0 \le a_i \le 10^6$) — the elements of the array a.

Output

Print two integers l, r ($1 \le l \le r \le n$) — the index of the left and the index of the right ends of some *k*-good longest segment. If there are several longest segments you can print any of them. The elements in a are numbered from 1 to n from left to right.

input
5 5 1 2 3 4 5

output
1 5

input
9 3 6 5 1 2 3 2 1 4 5

output
3 7

input
3 1 1 2 3

output
1 1

E. Sum of Remainders

2 seconds, 256 megabytes

Calculate the value of the sum: $n \bmod 1 + n \bmod 2 + n \bmod 3 + \ldots + n \bmod m$. As the result can be very large, you should print the value modulo $10^9 + 7$ (the remainder when divided by $10^9 + 7$).

The modulo operator $a \bmod b$ stands for the remainder after dividing a by b. For example $10 \bmod 3 = 1$.

Input
The only line contains two integers n, m $(1 \le n, m \le 10^{13})$ — the parameters of the sum.

Output
Print integer s — the value of the required sum modulo $10^9 + 7$.

input
3 4
output
4

input
4 4
output
1

input
1 1
output
0

F. Expensive Strings

6 seconds, 512 megabytes

You are given n strings t_i. Each string has cost c_i.

Let's define the function of string s : $f(s) = \sum_{i=1}^{n} c_i \cdot p_{s,i} \cdot |s|$, where $p_{s,i}$ is the number of occurrences of s in t_i, $|s|$ is the length of the string s. Find the maximal value of function $f(s)$ over all strings.

Note that the string s is not necessarily some string from t.

Input
The first line contains the only integer n $(1 \le n \le 10^5)$ — the number of strings in t.

Each of the next n lines contains contains a non-empty string t_i. t_i contains only lowercase English letters.

It is guaranteed that the sum of lengths of all strings in t is not greater than $5 \cdot 10^5$.

The last line contains n integers c_i (- $10^7 \le c_i \le 10^7$) — the cost of the i-th string.

Output
Print the only integer a — the maximal value of the function $f(s)$ over all strings s. Note one more time that the string s is not necessarily from t.

input
2 aa bb 2 1
output
4

input
2 aa ab 2 1
output
5