DATA STRUCTURES AND ALGORITHMS

ASSIGNMENT-8
Language allowed: C

April 11, 2020

A. The Good Doctor

The genius autistic surgical resident Dr. Shaun Murphy is preparing for his first lead surgery. All the surgical instruments needed for the surgery have been cleaned and set aside. But Shaun is not happy with this arrangement. He wants that arrange the instruments in such a way such that no two instruments made by the same company are kept successively and is unwilling to begin the surgery till he gets his desired arrangement. It's now up to you, Dr. Park, to re-arrange the instruments in the way Shaun desires. Complete this task, *stat*.

Input

The first line of input contains a single integer N ($1 \leq N \leq 10^6$) denoting the number of instruments on the table. The following line contains a sequence of N space-separated integers A ($1 \leq A_i \leq 10^6$) with the i^{th} integer denoting the company that manufactured the i^{th} instrument.

Output

Print a sequence of N integers, denoting the order of the instruments after you rearrange them in such a way that no two instruments manufactured by the same company appear together. If there are multiple answers, print any. If there exists no rearrangement that satisfies the condition, just print -1.

```
input
5
5 5 5 5 2

output
-1
```

```
input
10
1 1 675 87 87 1 76 675 675 1

output
1 87 675 1 76 675 1 87 675 1
```

B. Okabe and the Graph of Worldlines

After more months of research, Okabe Rintarou had yet another epiphany. This time he realized that the worldlines can be visualized as vertices in a graph and the the transitions between those worldlines can be seen as edges. Each transition (between some pair of worldlines) requires a set amount of energy. Okabe can transit between those worldlines only if he overcomes that *transition potential*. To complete his experiment, Okabe needs to know the minimum of all *total transition potentials* of all *spanning sub-graphs* possible in this graph of worldlines. As Okabe is has promised to take Suzuha and Daru out for dinner today, it's up to you, Makise Kurisu, to figure it out. *Total transition potential of a sub-graph* is defined as the sum of all transition potentials in that sub-graph. Assume the transitions to be undirected. A spanning sub-graph is a graph connecting all vertices.

Input

The first line contains two space-separated integers N ($2 \leq \text{N} \leq 5.10^3$) and M ($1 \leq \text{M} \leq \frac{N(N-1)}{2}$) denoting the number of worldlines and the number of transitions respectively. The following M lines contain three space-separated integers U_i, V_i ($0 \leq U_i, V_i \leq$ N-1) and W_i ($0 \leq W_i \leq 10^9$), with W_i denoting the transition potential of the transition between worldlines U_i and V_i. Note that, transitions may not exist between every pair of worldlines. It is guarenteed that the given graph of worldlines will be connected and there won't be multiple disconnected components.

Output

Print a single integer X, denoting the minimum of all *total transition potentials* of all sub-graphs possible in this graph of worldlines.

```
input
7 11
0 1 5
1 4 3
4 6 1
6 5 3
5 2 5
2 0 7
2 3 5
3 6 3
3 4 4
0 4 2
1 5 2

output
16
```

C. Encode it!

In this task, you are expected to find the minimum number of bits required to encode a given string. Formally, you need to find the minimum number of bits you will require to represent each character without ambiguity, i.e. in such a way that the bit representation of a character is not the prefix of the bit representation of some other character.

Input

The only line of input contains a single string S ($1 \leq |S| \leq 10^6$) consisting of latin characters (both uppercase and lowercase). Note that, uppercase and lowercase characters are not equivalent.

Output

In the first line, print a single integer X, denoting the minimum number of bits required to represent the given input string. In the following lines, print the minimum bit representation of each character. (see sample case for clarity). The character and corresponding bit representation pair must be printed in ascending order of ASCII value of the character. If there can be multiple representations for a character, print any.

---

input

```
hwowoaahfajhy
```

output

```
35
a:11
f:101
h:00
j:0100
o:011
w:100
y:0101
```

explanation

For the sample case given above, this is one possible way of encoding the characters. But note that, which ever encoding you follow, the minimum number of bits you will need in this case will be 35.

D. Origin

Robert Langdon is on his was to Barcelona Supercomputing Centre to podcast the presentation Edmond Kirsch wanted to show the world. But to his utter dismay, the presentation in Kirsch's PC at the lab is locked with a check question. The question displays two character sets X and Y and one needs to find if Y is a subset of X before the timer runs out and the presentation is completely erased. Help Robert solve this problem in as soon as possible in $O(|X|)$, so that he can present Kirsch's mega discovery to the world.

Input

The first line of input contains two space-separated integers N ($1 \leq N \leq 26$) and M ($1 \leq M \leq 26$) denoting the sizes of the character sets X and Y respectively. The following line contains N space-separated lowercase latin characters denoting the set X. The next line contains M space-separated lowercase latin characters denoting the set Y.

Output

Print "YES" if Y is a subset of X, else print "NO"

---

input
```
6 4
a l k f t h
a f k h
```

output
```
YES
```

---

input
```
a g d s
g t e
```

output
```
NO
```

E. Count the Occurrences!

In this task, given a string S with many words (words will be space-separated), you need to find the number of times a query word occurs in the given string. Note that pre-processing of the string must be done is $O(|S|)$ time and each query should be answered in expected $O(1)$ time.

Input

The line of input contains the string S ($1 \leq |S| \leq 10^6$). The next line contains a single integer Q ($1 \leq Q \leq 10^3$) denoting the number of queries for the problem. Each of following Q lines contains a single string X_i ($1 \leq |X_i| \leq |S|$) with X_i denoting the i^{th} query word.

Output

Print Q integers line-wise with the i^{th} line containing the answer to the i^{th} query, i.e. the number of occurrences of the i^{th} query word X_i in the given string S.

```
input
blue bear brown bear yellow bear yellow yellow bear
3
yellow
bear
blue

output
3
4
1
```

F. COVID-19

Corona Virus is in the air and researchers are working round the clock to produce a cure. But they are stuck with an interesting problem. They have the weights of the N genes of the virus in an array, and they need your help in finding out if a sub-array exists whose sum is divisible by N. Can you help the researchers out and save humanity? As we are short on time, this problem must be solved in $O(N)$

Input

The first line of input contains a single integer N ($1 \leq N \leq 10^6$) denoting the number of elements in the array A. The following line contains N space-separated integers ($1 \leq A_i \leq 10^9$) denoting the array A.

Output

Print a single integer X, denoting the sum of the sub-array (whose sum is divisible by N) if one exists. Print no such sub-array exists, print -1. If multiple such sub-arrays exist which satisfy the given condition, print the sum of any one of those sub-arrays.

```
input
6
2 3 3 4 1 8

output
6
```

G. Quizzed!

Kailash has quick question for you. He gives you a array of integers (representing the level order traversal) and asks you to check if the given numbers form a min-heap or max-heap or none. So, you decide to write a short program to solve this problem.

Input

The first line contains a single integer N ($1 \leq N \leq 10^6$) denoting the number of elements in the heap. The following line contains N space-separated ($-10^9 \leq A_i \leq 10^9$) integers denoting the array A.

Output

If the elements form a min-heap or max-heap, print "MIN_HEAP" or "MAX_HEAP" accordingly. If they don't form any heap, print "TRICK_QUESTION". In case the elements constitute both a min-heap as well as a max-heap, print "BOTH".

```
input
7
-1 2 3 4 5 6 7

output
MIN_HEAP
```

```
input
4
3 3 3 3

output
BOTH
```

H. Merge those streams!

In this task, you are given K separate streams of integers. You now need to print all the integers after they have been merged and sorted in ascending order. Note that, taking input from all streams into a single array and directly sorting it must not be done. Heaps must be used to solve this problem.

Input

The first line of input contains a single integer K ($1 \leq K \leq 10^2$) denoting the number of streams. Each of the following K lines contain a sequence of integers A ($-10^9 \leq A_{ij} \leq 10^9$; A_{ij} denotes the j^{th} element of the i^{th} stream). It is guaranteed that the size of each stream will be less than 10^3. Note that, the stream size is not given as input.

Output

Print a single sequence of integers B, denoting the final answer after merging the contents of all the streams in ascending order.

```
input
3
1 100 10
2 20 200
3 30 300

output
1 2 3 10 20 30 100 200 300
```

I. Median

You find a weird stream of integers gushing out of a faucet. You are quite intrigued by that and wish to find the *running median* (median at each step) of that stream of integers and decide to write a simple program that will aid you solve this problem. The overall problem must be solved in $o(N^2)$.

Input

The first line of input contains a single integer N ($1 \leq N \leq 10^3$) denoting the number of integers in the stream. The following line contains N space-separated integers ($-10^9 \leq A_i \leq 10^9$) denoting the elements in the stream. Note that, the median at each i^{th} instant should be written as and when you read the i^{th} input.

Output

Print N lines, where the i^{th} line contains a single integer/float M_i denoting the median at the i^{th} instant.

```
input
6
12 4 5 3 8 7

output
12
8
5
4.5
5
6
```

J. Exists?

In this task, you are given an undirected graph. You now need to find if a path exists between any two given vertices. This problem must be solved in $O(M + Q)$ where M is the number of edges and Q is the number of queries. *Hint:* Think of pre-processing the graph into some different data structure other than using DFS or BFS.

Input

The first line of input contains a two space-separated integers N ($2 \leq$ N $\leq 10^5$) and M ($1 \leq$ M $\leq \frac{N(N-1)}{2}$) denoting the number of vertices and edges respectively. The following M lines contain two space-separated integers U_i, V_i ($0 \leq U_i$, $V_i \leq$ N-1) denoting an edge between vertices U_i and V_i. The next line contains a single integer Q ($1 \leq$ Q $\leq 10^3$) denoting the number of queries for the problem. Each of the following Q lines contains two space-separated A_i and B_i denoting the vertices between whom the existence of a path must be checked.

Output

Print "YES" if a path exists between the given vertices else print "NO". Each output must be printed in a new line (i^{th} line should contain the answer the i^{th} query).

```
input
15 14
9 8
9 4
1 9
1 5
5 4
6 10
11 12
10 3
14 11
11 13
3 2
2 0
0 10
2 6
4
3 9
14 0
8 5
2 10

output
NO
NO
YES
YES
```

K. Fire Emblem

The evil orgs have trapped prince Marth in an intricate dungeon (which is in the form of a grid) and his friend Roy Guide is on his way to save him. The orgs, expecting Roy, have set up trap spells in various rooms of the dungeon such that if Roy enters a room with a trap spell, he will *immediately* lose some life points. If Roy loses all his life points, he faints and cannot proceed. Luckily, some of the rooms have crystal stones which can boost Roy's life points. Roy is currently at the entrance of the dungeon and is wondering which path will lead him to Marth in such a way that he will reach (without fainting anywhere) with *minimum* life points. Help Roy find that path, so that he can be cautious in his quest.

Assume that, a room will either contain a crystal stone or a trap spell, but not both. It is guaranteed that there will not be any trap spells at the entrance and in the room where Marth is trapped. Further, Roy always begins with non-zero life points and can only move one cell to the *right* or one cell *down*. Note that, the entrance corresponds to the $(0, 0)^{th}$ cell of the grid and Marth is always trapped in the $(N-1, M-1)^{th}$ cell. You are expected to solve this problem in $O(NM)$.

Input

The first line of input contains three space-separated integers N, M ($1 \leq$ N, M $\leq 10^3$) and L ($1 \leq$ L $\leq 10^3$) denoting the dimensions of the grid and Roy's initial life points. Each of the following N lines contains M space-separated integers ($-10^3 \leq D_{ij} \leq 10^3$) such that if D_{ij} is negative, it refers to a trap spell and Roy losses equal amount of life-points and if D_{ij} is positive, it refers to a crystal stone and Roy gains equal amount of life points.

Output

If Roy cannot reach Marth through any path, print -1. Otherwise, print a single integer X, denoting the *minimum* possible life points with which Roy can reach Marth.

```
input
3 4 10
1 -2 -3 6
7 8 -6 8
6 -7 -1 2

output
8
```

L. Nearest Neighbours

In this task, you are given N unique point in the cartesian plane. Your task is to find the *minimum* distance between any two points in the given set. You are expected to solve this problem in $o(N^2)$.

Input

The first line contains a single integer N denoting the number of points in the given set. Each of the following N lines contains a two space-separated floats X_i and Y_i denoting the coordinates of the points in the set (the i^{th} line contains the coordinates of the i^{th} point).

Output

Print a single integer/float D denoting the *minimum* distance between any two points in the set.

```
input
14
-4.02, 2.2
-1.52, 2.54
-2.36, 1.40
-1.46, 0.58
-3.28, 1.20
0.0, 4.0
0.0, 3.0
-3.2, 0.08
-2.5, 2.42
-2.26, 0.32
-2.0, 0.0
-0.34, 0.28
-1.08, 1.42
4.66, -1.48

output
0.3624
```

M. Stock Trading 101

After some thought, you decide to invest in stocks. After enquiring with your friend who works at the stock market, you get a long list of stock rates (to either to buy/sell) for the next N days starting from today. So, you decide to buy a stock on some particular day and sell on some other day (after the buying day). Once you sell a stock, the profit you get is the difference between the rates at which you sell and buy (which can also be negative, indicating a loss). As usual, you decide to trade such that you get the *maximum possible profit* and decide to write a code that will help you find the days on which you should buy and sell the stock. Note that, *you trade stocks only once*, and a stock once bought must be sold at a later stage. It is not mandatory to buy stock always when it an obvious loss is perceived, but if you have the opportunity to buy a stock and make no loss, you will surely buy it. You are expected to solve this problem in $O(N)$.

Input

The first line of input contains a single integer N ($1 \leq N \leq 10^6$), denoting the number of days for which you find the stock rates. The following line contains a sequence of N space-separated integers R ($1 \leq R_i \leq 10^9$) denoting the rates (the i^{th} integer denotes the stock rate on the i^{th} day).

Output

Print three integers X, Y and P denoting the day of buying, day of selling and the profit obtained respectively. If there are multiple answers, print any. If you don't trade any stocks, print "NO TRADING".

---

input
7
100 695 260 310 40 535 180

output
1 2 595

---

input
6
100 90 65 43 21 10

output
NO TRADING

---

input
7
10 10 10 10 10 10 10

output
2 5 0

N. Inverses

Consider the lowercase English alphabet. We define the inverse of a letter as the following: inv(a) = z, inv(b) = y, inv(c) = x ... inv(m) = n, inv(n) = m ... inv(y) = b, inv(z) = a. Now given a string S, check if each alphabet in the string contains its inverse within the same string. This problem must be solved in $O(|S|)$.

Input

The only line of input contains the string S ($1 \leq |S| \leq 10^6$)

Output

Print "YES" if each alphabet contains a inverse within the same string and "NO" otherwise. If the output is "NO", print the all the unique alphabets separated by spaces in lexical order that do not have an inverse in the string.

```
input
abcxyz

output
YES
```

```
input
caaaby

output
NO
a c
```

```
input
nnm

output
YES
```

O. Tetris v2

You are given a 2×N board and an infinite number of 2×1 tiles. Find the number of ways this 2×N board can be tiled. A tile can be placed *vertically* or *horizontally*.

Input

The only line of input contains a single integer N ($1 \leq N \leq 10^6$) denoting the breadth of the board.

Output

Print a single integer X, denoting the number of ways the board can be tiled.

---

input
4

output
5

---

input
7

output
21

P. Longest Subsequence

In this task, you are given an array and you need to find the length of the *longest subsequence* such that the elements of the subsequence are *strictly in increasing order*.

Input

The first line of input contains a single integer N ($1 \leq N \leq 10^6$) denoting the size of the given array. The following line contains a sequence of N space-separated integers A ($-10^9 \leq A_i \leq 10^9$).

Output

Print a single integer L, denoting length of the longest strictly increasing subsequence.

```
input
9
7 20 5 33 11 55 43 70 90

output
6

explanation
The longest increasing subsequence is 7, 20, 33, 55, 70, 90
```

```
input
5
-4 10 2 1 0

output
2
```
