Problem 1

Alloys

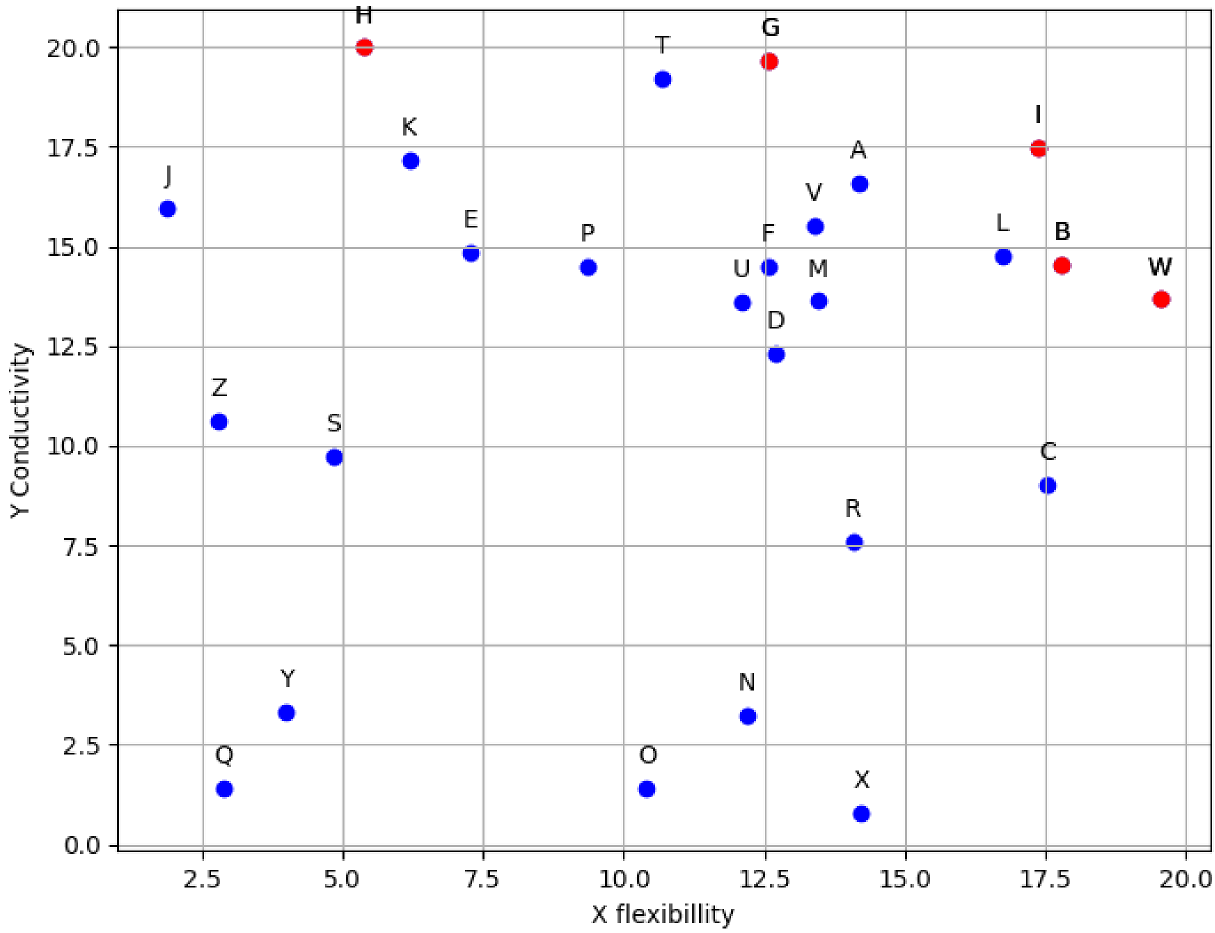
Time limit: 15 seconds

A chemical scientist is attempting to create a new metal alloy that possesses both flexibility and conductivity. To achieve this, the scientist blends various types of metals and then assesses their capabilities. Subsequently, the experiments yield varying levels of conductivity and flexibility over time. Certain metal alloys exhibit high flexibility but low conductivity, while others show the opposite, and finally, a few have both attributes in a diminished capacity. To further his experiments, he is solely focused on alloys that are not dominated by others, allowing for more in-depth investigations.

Here, "dominating" refers to an alloy with flexibility `x` and conductivity `y`, where no other alloy exhibits both greater flexibility and conductivity.

For instance, consider the testing of five different alloys `A`, `B`, `C`, `D`, `E` resulting in the following flexibility and conductivity values: `A=(1,5)`, `B=(2,2)`, `C=(3,4)`, `D=(5,2)`, and `E=(4,1)`. In this scenario, the scientist is particularly interested in alloys A, C, and D.

Another graphical example is depicted in the below image, where red dots represent dominant alloys.



Input:

The first line of the input contains a single integer n ($0 < n < 10^7$). The remaining n lines of the input contain points in the form of `(id flexibility conductivity)` separated with space. Both flexibility and conductivity must be considered to be floating point values.

Output:

The IDs of dominant points separated with space in alphanumeric order.

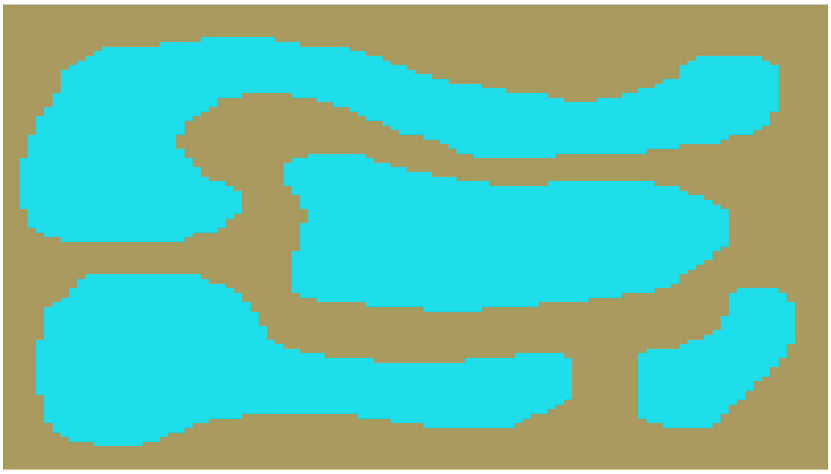
Sample Input 1	Sample Output 1
5 A 1.0 5.0 B 2.0 2.0 C 3.0 4.0 D 5.0 2.0 E 4.0 1.0	A B D
Sample Input 2	**Sample Output 2**
5 4 3.0 4.0 13 1.0 1.0 32 3.0 6.0 111 5.0 5.0 23 1.0 4.0	111 32

Problem 2

A-maze-ing Lakes

Time limit: 10 second

An environmental agency is asked to perform a survey of the lakes found in various places. This effort is part of a program to estimate the size and distribution of the lakes depending on the place they're located. At the initial stage of this project, the agency is going to use aerial photographs to perform the estimation. Due to the large number of lakes that will be investigated they decided to use a computer program to help with the task. More specifically, given a 'two color' image that looks like this:



The task is to find the number of the lakes (marked with blue) as well as the surface area of each one of them. The encoded image is given as a 2D array of tiles that can be either a ground tile or a water tile. Each tile in the image corresponds to one square meter of area. Given a single water tile, a lake is defined to cover all the water tiles that are reachable from the initial tile by performing an arbitrary sequence of up/down/left or right moves on water tiles (not diagonally). You can assume that the border tiles found at the edge of the image are always ground tiles.

Input:

The first line of the input contains two integers `N` and `M` separated with a single whitespace character ($1 \leq N \leq 10^3, 1 \leq M \leq 10^3$). These define an `N` x `M` 2D array of tiles. The next `N` lines of the input are strings of exactly `M` characters, which are a combination of `0` to mark a ground tile and `1` to mark a water tile.

Output:

The first line of the output should be a single integer `K` that is the number of distinct lakes found in the data. The second line should have `K` space separated integers that are the sizes in square meters of each lake. The integers in the second line (lake sizes) must be in sorted order, from the smallest to the largest.

Sample Input 1	Sample Output 1
6 13 0000000000000 0111101111110 0001101000100 0001101000110 0011000011110 0000000000000	2 10 15

Sample Input 2	Sample Output 2
6 10 0000000000 0111111110 0100000010 0100000010 0111111110 0000000000	1 20

Problem 3

Course Prerequisites

Time limit: 15 seconds

As a student at a university, you want to choose a set of courses for the next semester. Each course may or may not have a set of prerequisite courses. Therefore, before making your final decision, you want to ensure that you have satisfied all the prerequisites required for the courses you intend to take.

For instance, suppose a student has successfully completed courses `courseA`, `courseB`, `courseC`, `courseE`, `courseG`, and `courseH`. The course she wishes to enroll in has specific prerequisites, namely `courseA`, `courseC`, and `courseG`. In this scenario, the student meets the prerequisites because she has already completed all of them - `courseA`, `courseC`, and `courseG`.

Input:

The first line of the input contains a single integer `m`, the size of set A (1 ≤ `m` ≤ 10,000,000). The second line of input contains the courses of set A separated by spaces. The third line of the input contains a single integer `n`, the size of set B (1 ≤ `n` ≤ 10,000,000). The fourth line of input contains the courses of set B separated by spaces.
Note: All cources are strings.

Output:

Output 0 if prerequisites are not satisfied, else 1 if prerequisites are satisfied.

Sample Input 1	Sample Output 1

`4`
`CourseA CourseB CourseC CourseD CourseE`
`2`
`CourseC CourseD`
	1

Sample Input 2	**Sample Output 2**

`5`
`CourseA CourseB CourseD CourseE CourseO CourseI`
`4`
`CourseD CourseE CourseO CourseG`
	0

Problem 4

Heavy Rainfall

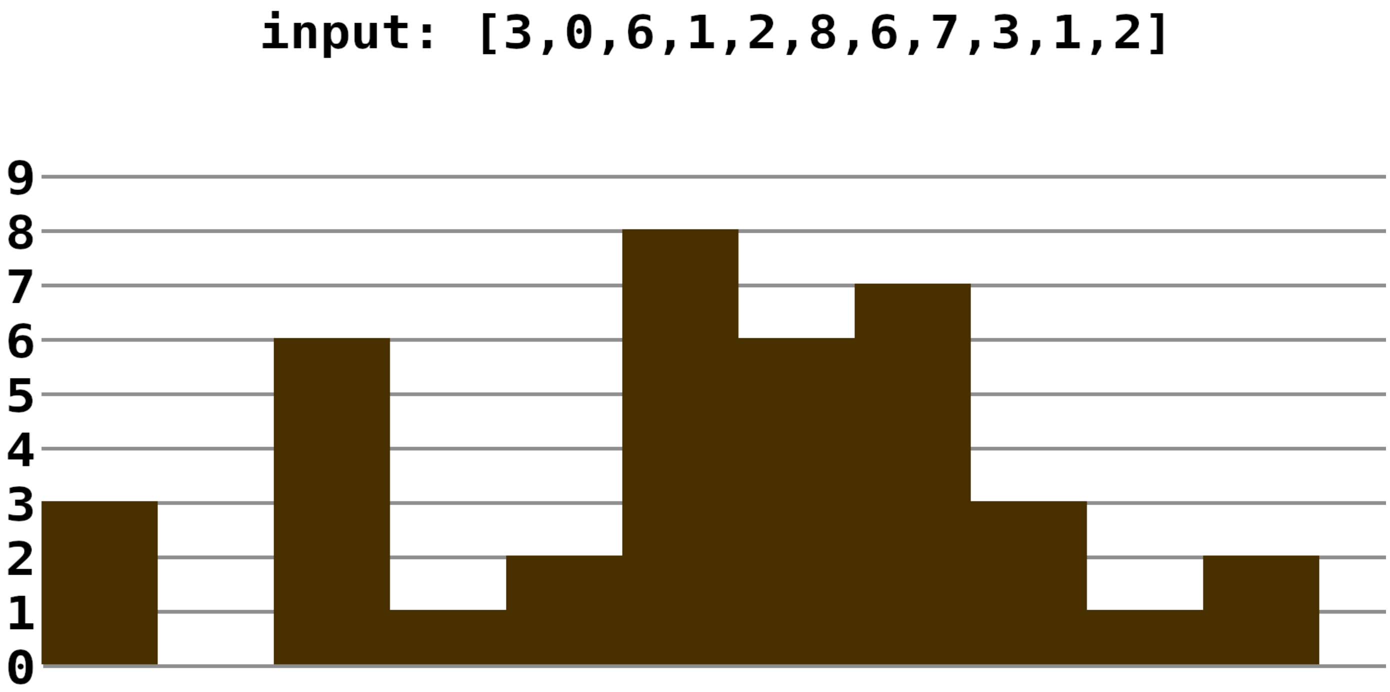
Time limit: 10 second

You're taking part in the development of a novel simulation system that is developed to help with extreme rain in mountainous areas. There have been many occasions of extreme weather recently, and there is a need to calculate the amount of water that will form between mountaintops in the worst-case scenario. This program will be used by experts to aid in predicting the outcome of future extreme weather events.

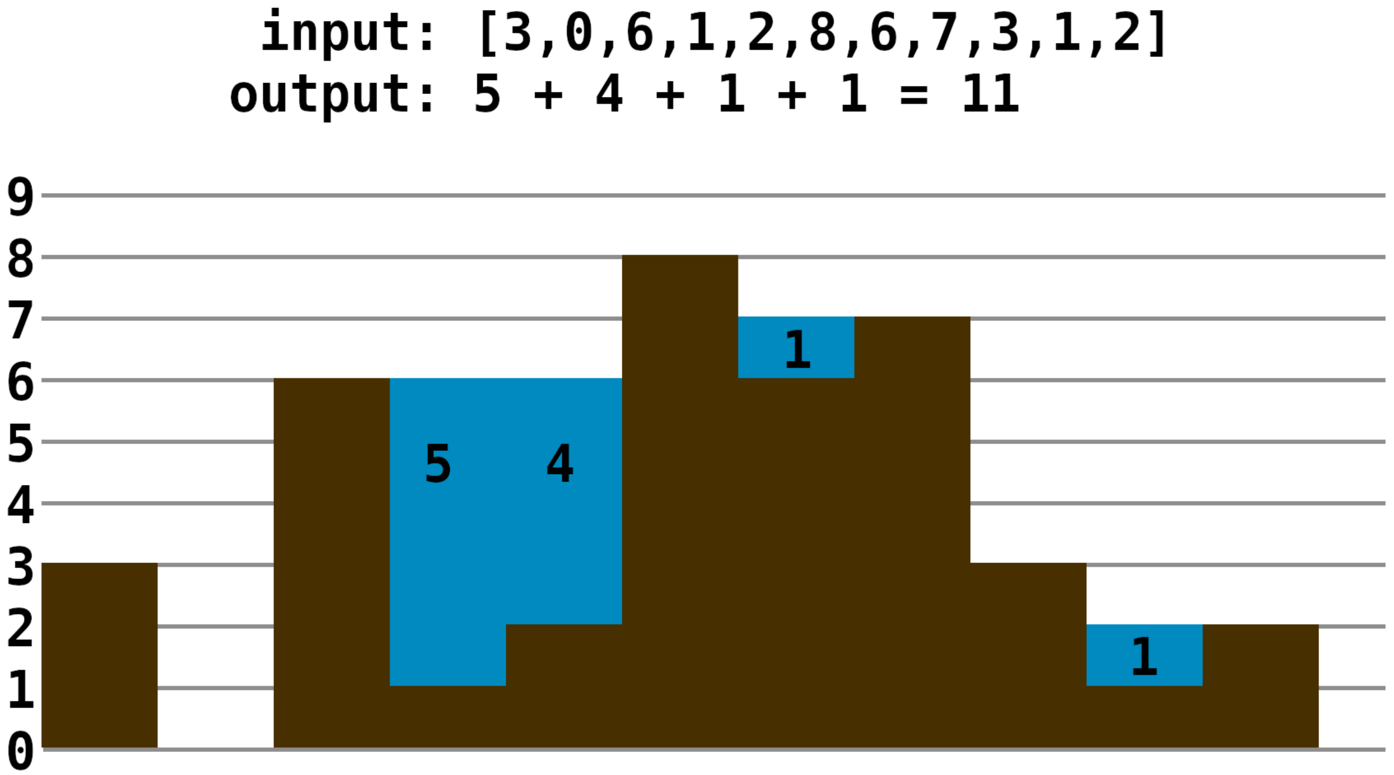
The program will accept a list of heights that result from discrete measurements from the corresponding mountain ranges. A lake can only form if it's surrounded by higher mountains, and its maximum height will be the minimum of the two surrounding mountains. If a height of 0 is given, it represents the ground. A lake cannot form if any of its parts would be connected to the ground; it would quickly drain otherwise. For each lake formed, its size is the number of 'discrete blocks' that it occupies between the mountaintops.

Input:

It's easier to understand all of this if we visualize the problem. As an example, consider we're given as input the height values `[3, 0, 6, 1, 2, 8, 6, 7, 3, 1, 2]`. The mountains formed could be visualized as below:



The biggest lakes that can form for this mountain range and their corresponding sizes would be the following:



None of the lakes could be bigger, since the water would fall off either side. Also, a lake cannot form in the first gap, as it drains to the ground.

The first line of the input file contains a single integer `N`, $0 \le N \le 10^5$, and the second a space-separated list of `N` integers that are the heights of the mountains. Each height value `H` given is within the range $[0, 10^5]$.

Output:

The output is a single number `S` (in the range $[0, 10^{10}]$) that is the sum of the maximally sized lakes that can form.

Sample Input 1	Sample Output 1
11 3 0 6 1 2 8 6 7 3 1 2	11

Sample Input 2	Sample Output 2
7 7 5 3 1 3 5 7	18

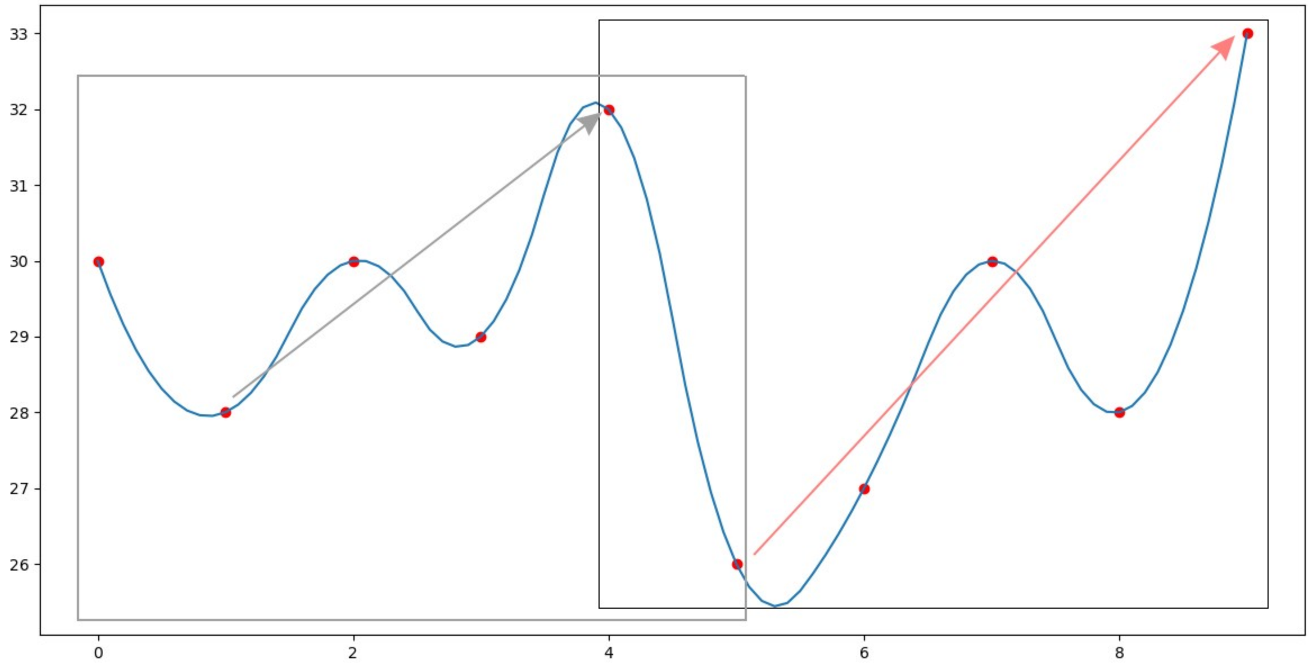
Problem 5

Weather Trends

Time limit: 120 seconds

George is an aspiring meteorologist who is passionate about analyzing temperature data. He is tasked with finding the largest temperature increase that occurs within specific time frames. Your task is to help George develop an efficient algorithm to calculate such temperature increases for any time interval.

Given a sequence of daily temperature measurements over a period of `n` days, George is interested in finding the greatest temperature increase that occurs within a window of `m` days. In other words, he wants to identify the maximum temperature difference between two days, where the second day occurs no more than `m` days after the first one.



Input:

The first line of input contains a single integer `n`, representing the number of available temperature measurements. The second line contains a single integer `m`, indicating the maximum time interval in days. The third line contains `n` temperature values, separated by spaces.

Output:

Output a single real number, representing the maximum temperature increase that occurs within the specified `m`-day interval.

Your task is to implement an efficient algorithm that can solve this problem for George. Additionally, ensure that your output has an absolute or relative error of at most 10^{-6}.

Sample Input 1	Sample Output 1
10 6 30 28 30 29 32 26 27 30 28 33	7.000000

Sample Input 2	Sample Output 2
7 3 31 32 33 25.91 31 30 35	5.090000

Problem 6

Port Robot

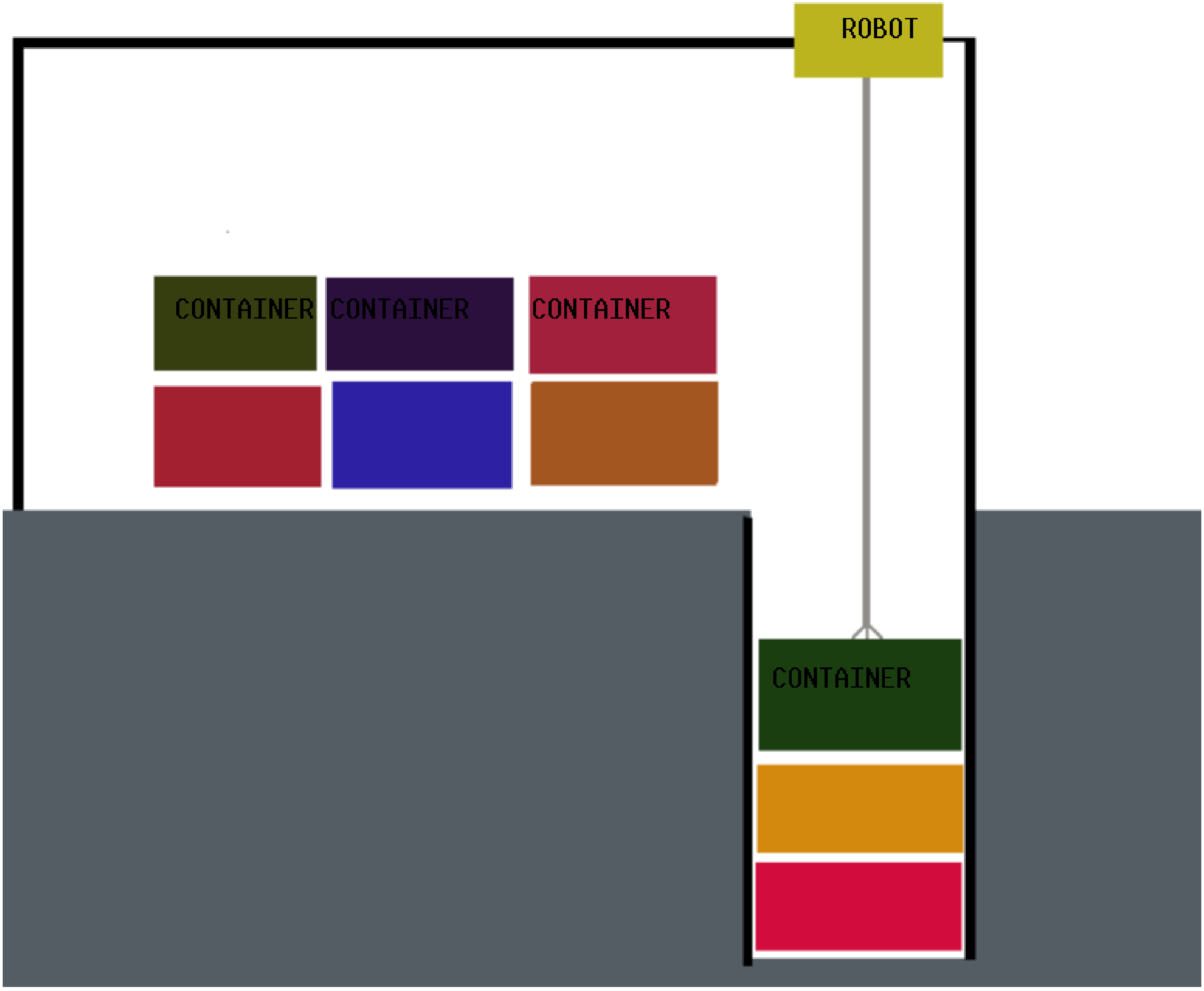
Time limit: 10 seconds

You are developing a robot that helps in storing containers that have arrived at ports before sending them to ships. These robots are responsible for handling these containers using limited space. Specifically, the space can fit multiple containers one over the other, but not side by side, as depicted in the figure.

While operating, the robot performs two actions:

1. Putting a container in storage.
2. Taking a container from storage.

There are 26 types of containers in this port, each one characterized by a letter of the Latin alphabet. Each time an action is performed, the robot logs the type of the container in lowercase when storing a container and in uppercase when taking a container from storage. Your task is to check if the logs produced by a robot indicate stable operation. In stable operation, the logged containers of action type 2 exist in storage and are in the uppermost position. Moreover, in stable operation, we always end with empty storage.



As an example, consider a scenario in which the robot produces the log: `cdDC`. Here, we have a stable operation of the robot, as the robot initially places `c` and then `d` (indicating that `d` is in the upper position). The robot then takes the first `d` out and, finally, takes out `c`. On the other hand, an unstable operation would result from the log: `cdCD`, since the robot cannot take `c` before taking out `d.

Input:

The firsdt line contains the length of the log generated by the robot.
The second line contains the extracted log of the robot operation.

Output:

A single number 0 if the log indicate wrong operation and 1 if the operation is stable.

Sample Input 1	Sample Output 1
8 DdaacCAA	0
Sample Input 2	**Sample Output 2**
14 aabcdDCBdDAbBA	1

Problem 7

Variable Assignments

Time limit: 4 second

Unfortunately, this year, you somehow managed to make your math teacher at school *really* angry with you. He wants to take his revenge but without raising any suspicion. After some thought, he comes up with a plan: he will hand you a list of very boring math exercises that you'll have to solve during the school's Christmas break...

So he comes up with the following exercise for you:

Let x_1, x_2, ..., x_n be variables.

You're given an ordered list of constraints (equations) of the form $x_i = x_j$ or $x_i = -x_j$. The constraints are transitive, so if you're given $x_i = x_j$ and $x_j = x_k$, then $x_i = x_k$ holds true even if not explicitly given in the list.

Then for each constraint between two variables x_i and x_j, you must mark it with:

- The letter N if it's a new constraint, meaning that x_i and x_j are so far not constrained in any way (neither directly or transitively).
- The letter E if it's an existing constraint, meaning that no new information is added (the constraint existed either directly or transitively).
- The letter C it's a contradicting constraint, meaning that it's impossible to fulfill the constraint given the previous ones. If you come across a contradiction the exercise ends and you don't mark the rest of the constraints.

In addition, we consider that a constraint $x_i = x_i$ is always existing and a constraint $x_i = -x_i$ is always contradicting.

The exercise is difficult, but your teacher is kind enough to give an example with explanations:

Let's say we have `N` = 4 variables and the following `M` = 7 constraints:

1. $x_1 = x_2$
2. $x_3 = -x_4$
3. $x_1 = -x_4$
4. $x_1 = x_3$
5. $x_2 = -x_4$
6. $x_2 = -x_3$
7. $x_1 = -x_4$

They're marked as follows:

1. Is a new constraint, as x_1 and x_2 weren't bound in any way. (N)
2. Is a new constraint, as x_3 and x_4 weren't bound in any way. (N)
3. Is a new constraint, as x_1 and x_4 weren't bound in any way. (N)

4. Is an existing constraint due to the transitive property from constraints 2 and 3: From $x_1 = -x_4$ and $-x_4 = x_3$ we can get $x_1 = x_3$. (E)
5. Is an existing constraint too, by using transitivity on constraints 1 and 2. (E)
6. Is a contradiction, because applying the transitive property at constraints 1 and 4 gives us $x_2 = x_3$ which cannot hold at the same time with $x_2 = -x_3$. (C)

Since we've reached a contradiction, the exercise stops.

The only thing that your teacher didn't knew is that you're really good at programming. In order to quickly solve all the exercises and get your time back, you've decided to write a program that automatically solves any instance of this exercise.

Inputs

The first line of the input contains two integers `N` and `M` separated with a single whitespace character ($1 \le N \le 10^6, 1 \le M \le 10^6$). `N` is the number of variables (from x_1 to x_N). After the first line there are `M` more lines, each with two integers `I` and `J` separated with a whitespace character and with $I \in [1, N]$ and $J \in [-N, 1] \cup [1, N]$. Each `(I,J)` pair represents a constraint. If `J>0` then it represents the constraint $x_I = x_J$. If `J<0` then it represents the constraint $x_I = -x_{|J|}$. Although not necessary for the solution, you can assume that `I<|J|` holds for all input data if you want.

Outputs

The output must be `K` lines, each with a single character which can be either `N`, `E` or `C`. The character at line `i` in the output is what you mark the `i`-th constraint of the input. It holds that $1 \le K \le N$, depending on if there's a contradiction.

Examples

The first example represents the same problem as what was given to you by your teacher.

Sample Input 1	Sample Output 1
4 7 1 2 3 -4 1 -4 1 3 2 -4 2 -3 1 -4	N N N E E C

Sample Input 2	Sample Output 2
4 9 1 1 2 2 2 3 1 3 1 2 1 3 3 4 1 4 2 4	E E N N E E N E E

Problem 8

Spy Network

Time limit: 5 seconds

The NIS spy network has decided to change the phone number from its headquarters every day as an extra security measure. To inform the spies about the new number, a sequence of words is given only to the spies (a total of 10 words), which they must then compare to a sequence of words from a newspaper, word by word. The comparison of words is done by finding the minimum number of operations required to make the first word identical to the second word. The possible operations that can be performed on the first word are the following:

- **Operation 1 (INSERT)**: Insert any character before or after any index of str1
- **Operation 2 (REMOVE)**: Remove a character from str1
- **Operation 3 (REPLACE)**: Replace a character at any index of str1 with some other character.

Input:

The first line contains the number (N) of words in the newspaper paragraph that will be used.
The second line contains the 10 positions from the words that must be used from the given paragraph.
The third line contains the 10 initial words of the secret phrase.
The fourth line provides the N words of the paragraph.

Output:

The phone numbers found separated by spaces.

Example:

Sample Input 1	Sample Output 1
15 1 3 2 5 6 8 15 10 11 12 hello I cannot wait to start coding for this problem amidst the tranquil forest a cascade of colors painted the trees in a mesmerizing dance	6 8 6 3 7 5 5 3 3 7

Short explanation: In the above example, the edit distance should be calculated between the words: hello ↔ amidst (6), I ↔ tranquil (8), cannot ↔ the (6), wait ↔ a (3), to ↔ cascade (7), start ↔ colors (5), coding ↔ dance (5), for ↔ the (3), this ↔ trees (3), problem ↔ in (7)