DSA Final Lab exam

Section B

July 6, 2022

Instructions for the exam

1. Place your Permanent / Temporary Student ID card on the desk during the examination for verification by the Invigilator.
2. Reading material such as books are not allowed inside the examination hall.
3. Borrowing writing material or calculators from other students in the examination hall is prohibited.
4. If any student is found indulging in malpractice or copying in the examination hall, the student will be given 'F' grade for the course and may be debarred from writing other examinations.
5. This is a 4 hour exam.
6. You will be submitting your problems on the OJ
7. All the data structures will be provided in a .c file. You can copy them into your code and use them as required. You can modify them as well. Please go through the template code before using it (at-least the function definitions).
8. Refer to the time limit and memory limit of each problem on OJ.
9. There are 4 problems.
    - **Problem A:** 100 points
    - **Problem B:** 200 points
    - **Problem C:** 200 points
    - **Problem D:** 300 points

Problem A: Key fixes (100 points)

You're typing a long text with a broken keyboard. Well it's not so badly broken. The only problem with the keyboard is that sometimes the *home* key or the *end* key gets automatically pressed (internally).

Home key moves the cursor to the start of the text you are typing, and *End* key moves the cursor to the end of the text you are typing.

You're not aware of this issue, since you're focusing on the text and did not even turn on the monitor! After you finished typing, you can see a text on the screen (if you turn on the monitor).

Your task is to find the final text on the screen.

Constraints

$1 \leq T \leq 10$
$1 \leq |s_i| \leq 10^5$ for all $i = 1, \ldots, T$. Here $|x|$ denotes the length of string x.

Subtasks

Subtask 1(5 points): $T = 1$ $1 \leq |s_i| \leq 100$
Subtask 2(5 points): $T = 1$ $1 \leq |s_i| \leq 1000$
Subtask 3(5 points): $1 \leq T \leq 10$ $1 \leq |s_i| \leq 1000$
Subtask 4(85 points): Original constraints

Input format

First line contains an integer T — the number of strings.
The i^{th} of the next n lines contain a string s_i made up of lowercase latin alphabets and two special characters '(' and ')', where '(' dentotes that the *Home* key is pressed internally, and ')' denotes the *End* key is pressed internally.

Output format

For each input string output the final string seen on the screen, on a new line.

Examples

Sample Testcase 1

Input

4
Thisisa(Beiju)text
(())()()HappyBirthdaytoIIIT
ajdbac(abaxc(hjbwe)akbck)cb
)adkjb(cjkbakac)akbc

Output

BeijuThisisatext
HappyBirthdaytoIIIT
hjbweabaxcajdbacakbckcb
cjkbakacadkjbakbc

Explanation

For the first string, you type in "Thisisa", then *Home* is hit, type "Beiju", *End* is hit and finally type out "text". Thus, you finally get "BeijuThisisatext".
For the second string, *Home* and *End* are hit when nothing is typed. Thus, it has no effect. So finally you type "HappyBirthdaytoIIIT", and the same string appears on the screen.

Problem B: Array Merger

Given an array of n integers $a_1, a_2, \ldots, a_n$. You can perform the following operations as many times as you can:

Take any two unmerged elements of the array, say the elements at positions i and j ($i \neq j$). Then you can merge these two elements only if $a[j] \geq 2 \times a[i]$, i.e., if one of the elements is atleast twice as big as the other. After merging the two elements, the smaller one is deleted and the greater one is marked as *merged*, which means you **cannot** merge the greater element with another element from the array.
Initially all elements are unmerged.
Output the minimum number of elements(merged or unmerged) that remain.

Constraints

$1 \leq n \leq 5 \times 10^5$
$1 \leq a_i \leq 10^5$ For all $i = 1, \ldots, n$

Input format

First line contains an integer n.
Next n lines contain the n integers $a_1, a_2, \ldots, a_n$, one on each line.

Output format

Single integer denoting the minimum number of elements(merged or unmerged) that remain.

Examples

Sample Testcase 1

Input

8
2
5
7
6
9
8
4
2

Output

5

Explanation

You can merge the 1^{st} and 3^{rd} elements (sizes 2 and 7 respectively), 4^{th} and 8^{th} elements (sizes 6 and 2 respectively) and 6^{th} and 7^{th} elements (sizes 8 and 4 respectively). After these merges, the unmerged elements are $[5, 9]$, which cannot be merged. Thus, the minimum number of elements that remain in the array are 5.

Sample Testcase 2

Input

8
9
1
6
2
6
5
8
3

Output

5

Explanation

You can merge the 2^{nd} and 3^{rd} elements (sizes 1 and 6 respectively), 4^{th} and 5^{th} elements (sizes 2 and 6 respectively) and 7^{th} and 8^{th} elements (sizes 8 and 3 respectively). After these merges, the unmerged elements are $[9, 5]$, which cannot be merged. Thus, the minimum number of elements that remain in the array are 5.

Problem C (200 points)

There are gas stations along the way. The gas stations are represented as an array stations where stations[i] = $[position_i, fuel_i]$ indicates that the i^{th} gas station is $position_i$ miles east of the starting position and has $fuel_i$ liters of gas.

The car starts with an infinite tank of gas, which initially has **startFuel** liters of fuel in it. It uses one liter of gas per one mile that it drives. When the car reaches a gas station, it may stop and refuel, transferring all the gas from the station into the car.

Print the minimum number of refueling stops the car must make in order to reach its destination. If it cannot reach the destination, print -1.

Note that if the car reaches a gas station with 0 fuel left, the car can still refuel there. If the car reaches the destination with 0 fuel left, it is still considered to have arrived.

Constraints

$1 \leq target, startFuel \leq 1e12$
$0 \leq stations.length \leq 1e6$
$0 \leq position_i \leq position_{i+1} < target$
$1 \leq fuel_i < 1e12$

Subtasks

Subtask 1(30 points): $stations.length \leq 10$
Subtask 3(170 points): Original constraints

Input format

First line contains one integer *stations.length*.
Next *stations.length* lines contain a pair of space-separated integers $prosition[i]$ and $fuel[i]$, on each line. Next one line contains two space separated integers $Target$ and $startFuel$.

Output format

The final minimized number of refueling stops.

Examples

Sample Testcase 1

Input

4
10 60
20 30
30 30

60 40
100 10

Output

2

Explanation

We start with 10 liters of fuel. We drive to position 10, expending 10 liters of fuel. We refuel from 0 liters to 60 liters of gas. Then, we drive from position 10 to position 60 (expending 50 liters of fuel), and refuel from 10 liters to 50 liters of gas. We then drive to and reach the target. We made 2 refueling stops along the way, so we return 2.

Problem D: Mazes of (to be renovated) Vindhya (300 points)

Tejas stands at the entrace of A4 yet again confused, where do these hallways of Vindhya lead to?

He holds the map of the entire building in the form of a directed graph. Vindhya forms a **directed acyclic graph** with N nodes and M edges. He stands at node 1 and wants to know if he can visit all the nodes in a single path or not.

Formally, he wants to know whether there exist a path in that graph that starts from node 1 and visits all the nodes in graph or not

Input format

The first line contains 2 integers, N and M.

The following M lines contain 2 integers each, u and v denoting that there is a directed edge from u to v.

It is gauranteed that there are no self loops or multi-edges

Output format

Output "YES" (without quotes) if there exists such a path and "NO" (without quotes) otherwise.

Note: Output is case sensitive

Constraints

Subtask 1(15 points): $N \leq 10, M \leq \frac{N*(N-1)}{2}$
Subtask 2(15 points): $N \leq 10^2, M \leq \frac{N*(N-1)}{2}$
Subtask 3(45 points): $N \leq 10^3, M = N$
Subtask 4(45 points): $N \leq 10^3, M \leq \frac{N*(N-1)}{2}$
Subtask 5(30 points): $N \leq 10^5, M \leq N - 1$
Subtask 6(150 points): $N \leq 10^5, M \leq \min(\frac{N*(N-1)}{2}, 10^6)$

Sample 1

Input

3 2
1 2
1 3

Output

NO

Explanation

There is no way to cover all the nodes in a single path

Sample 2

Input

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

Output

YES

Explanation

Tejas can go on the path $1 \rightarrow 5 \rightarrow 3 \rightarrow 4 \rightarrow 2$