



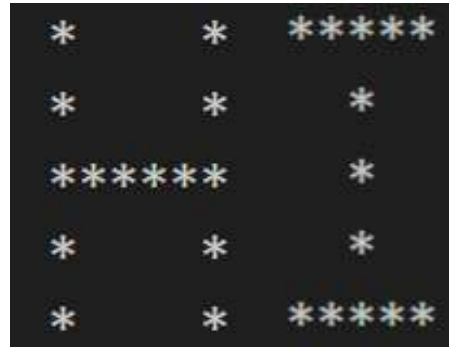
# **ELECTRICAL TEAM TRAINING**

## **TASK 1**

# **TABLE OF CONTENTS**

<b>Task 1.1</b>	<b>3</b>
<b>Task 1.2</b>	<b>4</b>
<b>Task 1.3</b>	<b>6</b>
<b>Task 1.4</b>	<b>8</b>
<b>Notes and submission</b>	<b>11</b>

## Task 1.1



Our robots are very friendly that they welcome every person who gets to know them by saying HI on their LCD screen. Help them welcome all your visitors.

### Required

- Write a code to output “HI” using asterisks ‘\*’
- Make sure you use the same number of asterisks as in the figure shown.

## Task 1.2

Ten years ago, scientists managed to invent a supernatural missile from thermonuclear materials, a prototype of that missile was created and tested in the Pacific Ocean, and that massive explosion woke up evil aquatic creatures settled on earth, fortunately, the military forces have managed to invent mighty robots to help in defeating those creatures, robots are provided with vigorous weapons and armaments.

Initially, the health point of the evil creature is **hp**. When its health points go to 0 or under 0, it will be defeated. To defeat those creatures, a robot can use two types of weapons.

### **Weapon 1: Plasma blast**

After using that weapon the enemy health points (hp) will become  $\text{floor}(\text{hp}/2) + 10$ .

### **Weapon 2: Hellfire force**

After using that weapon the enemy health points (hp) will decrease by 10.

Each robot has a different performance. Each can only use no more than **P** Plasma blast and **H** Hellfire force. The robot can use its weapons in any order and does not have to use all its ammo. You are going to help the robots to find out whether it is possible to defeat the evil creatures or not.

## Input

The input consists of multiple test cases.

The first line contains a single integer  $q$

- $q$ : the number of test cases ( $1 \leq q \leq 30$ )

The following  $q$  lines each contains three integers  $hp, P, H$

- $hp$ : the creature's initial health point ( $1 \leq hp \leq 10^5$ )
- $P$ : the maximum number of Plasma blasts the robot can use ( $0 \leq P \leq 30$ )
- $H$ : the maximum number of Hellfire forces the robot can use ( $0 \leq H \leq 30$ )

## Output

- For each test case: If it is possible to defeat the evil creature, print (without quotes) "YES". Otherwise, print "NO".

## Required

- Take  $q$  as user input. Then take  $hp, P, H$  as user input for each case
- Print "YES" if it is possible to defeat the enemy
- Print "NO" if it is impossible to defeat the enemy

## Example:

```
Input:  2
        100 3 4
        189 3 4
Output: YES
        NO
```

## Task 1.3

Some of our brilliant scientists are kidnapped from those evil aquatic creatures and imprisoned in a high-security prison under the sea, one of the scientists succeeded in decrypting the creature's language and found that the guards will leave their shift 4 at night. You are required to implement a code to help the scientist decrypt any message to help him escape easier.



Each statement is in the form of a string **s** made of digits and '&'. The scientist succeeded in mapping each statement **s** to English Uppercase characters as follows:

('1' to '9') are decoded into ('A' to 'I') respectively.

('10&' to '26&') are decoded into ('J' to 'Z') respectively.

## Input

The input consists of multiple test cases.

The first line contains a single integer  $q$

- $q$ : the number of test cases ( $1 \leq q \leq 30$ )

The following  $q$  lines contains different strings  $s$  for the encrypted messages

- $1 \leq \text{String size} \leq 1000$
- String consists of digits and the '&' letter
- Assume input string is always valid

## Output

- Strings containing the message after decryption for each test case.

## Required

- Take  $q$  as user input. Take different messages as user input strings.
- Decrypt the message
- Print the decrypted messages

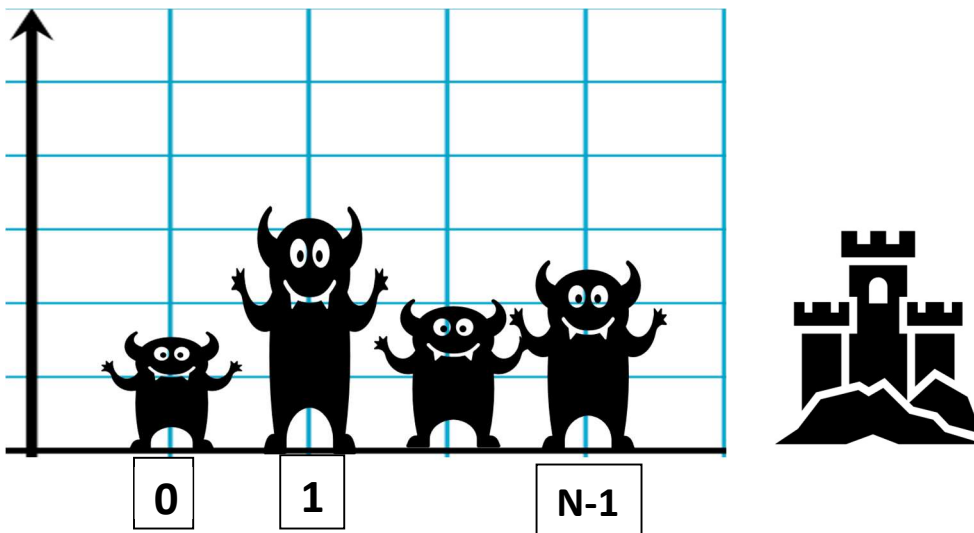
## Example

Input: 2  
10&11&12  
1110&

Output: jkab  
aaj

## Task 1.4

Some of the military spies managed to break into the creature's warehouse, each spy is provided with a shockwave rifle. After they broke in, the spies found that the creatures have different heights and guard the warehouse in queues. You need to know what each creature can see to be able to defeat them.



**N** creatures are standing in a queue to guard the warehouse, numbered from 0 to  $N - 1$  in left to right order. You have the height of each creature as an input.

Each creature can see the other creature to its right in that queue if every creature in between is shorter than both. For more explanation, see the example below.



## Input

The input consists of multiple test cases.

The first line contains a single integer  $q$

- $q$ : the number of test cases ( $1 \leq q \leq 30$ )

The following  $q$  inputs, each contains integer  $N$  then array of size  $N$  which contains each creature height ordered from left to right.

- $N$  equals the number of creatures in a queue. [ $1 \leq N \leq 10^5$ ]
- No more than one creature can have the same height.  
[ $1 \leq \text{height} \leq 10^5$ ]

## Output

Array of size  $N$  for each test case where  $\text{Array}[i]$  is the number of creatures  $i^{\text{th}}$  creature can see to their right.

## Required

- Take  $q$  as user input.
- Take number of creatures in each array then take array containing creatures' heights as user input.
- Find number of creatures each creature can see.
- Output array of the answer for each test case

**Example:**

Input: 1

6

10 6 8 5 11 9

Output: 3 1 2 1 1 0

**Explanation:**

Creature 5 cannot see any other creatures.

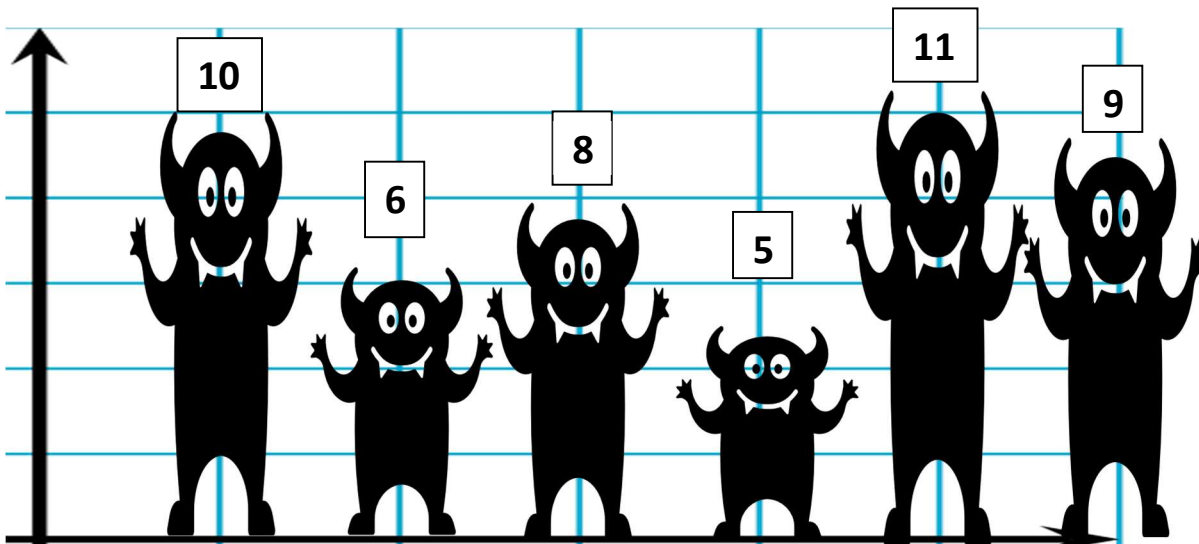
Creature 4 can see creature 5.

Creature 3 can see creature 4.

Creature 2 can see creatures 3 and 4.

Creature 1 can see creature 2.

Creature 0 can see creatures 1, 2, and 4.



## Notes

- Assume input for all problems is valid
- Your codes should be clean and commented thoroughly
- Your codes should compile successfully
- Problems vary in difficulty so don't worry if you get stuck in some of them and solve as much as you can

## Submission

- Submit a separate .cpp file for each task solution
- Each file should follow this naming convention:  
phoneNumber\_problem1.cpp  
phoneNumber\_problem2.cpp (till problem 4)  
\*\*\* make sure all letters in file name are lower case.
- Submit your solution using this form:  
<https://forms.gle/wYPpeDX7WFW9XXt9A>
- DEADLINE Tuesday 19/7 at 11:59PM