EEE Intra-Department Programming Contest 2025

Organized by: EEE Dept

Problem A: Who Better?

Description

To test the fresh brains of some of the brightest juniors from NSTU EEE Department, some of the seniors have decided to organize a tricky number-game competition. The seniors will provide a number n to the juniors.

If n is divisible by 3, they have to shout "EEE Better". If n is divisible by 5, they must shout "CSE Better". But if n is divisible by both 3 and 5, the seniors expect the juniors to shout "Both Better".

However, if the number doesn't satisfy any of these conditions, the juniors must simply shout "None Better", proving that they are smart enough not to be tricked.

As one of the brightest juniors, your task is to figure out what you have to shout.

Input Format

The first line contains an integer t — the number of test cases. The next t lines each contain a single integer $n(1 \leq n \leq 10^9)$. For each n, you must print the appropriate shout based on the rules described above.

Output Format

For each test case, print exactly one line containing one of the following: `"EEE Better"`, `"CSE Better"`, `"Both Better"`, or `"None Better"`.

Sample Test Case

Sample Input/Output:

Input:

```
4
2
6
10
15
```

Output:

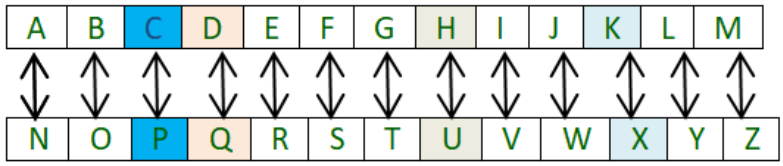
```
None Better
EEE Better
CSE Better
Both Better
```

Problem B: Will you be able to read the letters?

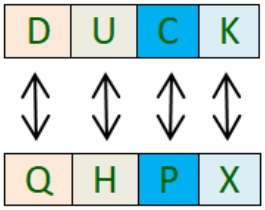
Description

Seniors write letters to each other in a ciphered format so that juniors don't understand the message while carrying the letter from one senior to another. Being a curious junior with some knowledge of ciphers, you suspect that it might be a ROT cipher. **A ROT cipher is a ciphering technique where each letter in the plaintext is replaced by another letter shifted by a fixed number of positions down the alphabet**. That fixed number is your **key**.

For example, if the key is 13, then A is replaced by N, B by O, and so on. The following diagram illustrates the mapping:



ROT13



For instance, if the original message is `HELLO`, after applying the cipher it becomes `URYYB`.

Additionally, your friend—who also carries letters for seniors and is knowledgeable in this field—discovered that the **seniors use one or more lowercase letters in place of every whitespace**. So `HELLO WORLD` may become something like `URYYBsaJBEYQ`.

Since decrypting the messages manually is difficult, you both decide to write a program to automate this process. Your task is to perform the **decipher** operation: retrieve the original message from its encrypted form. You will be given the key and the encrypted message. You must restore the original uppercase message, and every sequence of lowercase letters should be converted back into a single space.

Input Format

The first line contains an integer t — the number of test cases. Each test case consists of:

- An integer k — the cipher key.
- A single line containing the encrypted string, consisting of:
  - Uppercase letters (A–Z) representing shifted letters.
  - Lowercase letters (a–z) representing encrypted whitespaces.

Your task is to reverse the shifting of uppercase letters by k positions and replace every continuous sequence of lowercase letters with a single space.

Output Format

For each test case, output the original deciphered message in uppercase letters, with proper spaces restored.

Sample Test Case

Sample Input/Output:

Input:

```
4
5 XYTU
13 URYYBsaJBEYQ
13 URYYBsaeepJBEYQ
7 LLLpppppppppILAALY
```

Output:

```
STOP
HELLO WORLD
HELLO WORLD
EEE BETTER
```

Problem C: Palindrome or Armstrong?

Description

In this problem, you are given a positive integer n. Your task is to check two separate properties of this number:

1. Palindrome Number: A number is called a palindrome if it reads the same forwards and backwards. For example: 121, 1331, and 5 are palindromes, while 123 is not.

2. Armstrong Number: A number with d digits is called an Armstrong number (or Narcissistic number) if

$$n = \sum_{i=1}^{d} (\text{digit}_i)^d$$

For example: 153 is Armstrong since $1^3 + 5^3 + 3^3 = 153$. Same way 370 and 9474 are also Armstrong numbers. But 1253 is not Armstrong since $1^4 + 2^4 + 5^4 + 3^4 = 311 \neq 1253$

Your job is to determine:

- Whether n is a palindrome.
- Whether n is an Armstrong number.

If n is a palindrome, print `YES`, otherwise print `NO`. Then, in the next line, print `YES` if n is an Armstrong number, otherwise print `NO`.

Input Format

The first line contains an integer t — the number of test cases. The next t lines each contain a single integer n. For each n, you must print YES/NO seperated by a space based on whether n is palindrome/armstrong. n could be large.($1 \leq n \leq 10^{18}$). You should use `long long` instead of `int`.

Output Format

Print one line per testCase seperated by "Space":

- `YES` or `NO` indicating whether n is a palindrome.
- `YES` or `NO` indicating whether n is an Armstrong number.
- **50% of the score for determining if n is a palindrome, and 50% for determining if it is an Armstrong number.**

Sample Test Case

Sample Input/Output:

Input:

```
3
153
12321
8
```

Output:

```
NO YES
YES NO
YES YES
```

Problem D: CHAR Code Decoder

Description

The NSTU EEE department noticed that resistors mostly use round values like 10, 20, 100, 500 instead of 453, 8, 67, etc., and mostly lower-valued resistors are used for educational purposes. To simplify teaching, they recently published a new **resistor CHAR code decoder** scheme, which uses characters instead of colors to efficiently describe resistance values in Ohms. This scheme uses **2 characters** instead of 3 colors. The CHAR code scheme works as follows: The first character represents the digit and the second one represents the multiplier.

Character	Digit	Multiplier
A	1	1
B	2	10
C	3	100
D	4	1000
E	5	10000

Examples:

- `AA` $= 1 \times 1 = 1$ Ohm
- `BC` $= 2 \times 100 = 200$ Ohms
- `EB` $= 5 \times 10 = 50$ Ohms

One of the seniors asked you to write a program to calculate the resistance value in Ohms, so that no one has to go through the hassle of manual calculations.

Input Format

The first line contains an integer t — the number of test cases. The next t lines each contain a 2-character string representing the resistor code according to the CHAR code scheme described above.

Output Format

For each test case, print exactly one line containing the resistance value in Ohms corresponding to the 2-character code.

Sample Test Case

Sample Input/Output:

Input:

```
4
AB
BB
BE
CC
```

Output:

```
10
20
20000
300
```