

A student from ITESM Campus Monterrey plays with a new encryption method for numbers. These method consist of the following steps:

Steps : Example

1. Read the number N to encrypt : $M = 265$
2. Interpret N as a decimal number : $X_1 = 265$ (decimal)
3. Convert the decimal interpretation of N to its binary representation : $X_1 = 100001001$ (binary)
4. Let b_1 be equal to the number of 1's in this binary representation : $b_1 = 3$
5. Interpret N as a Hexadecimal number : $X_2 = 265$ (hexadecimal)
6. Convert the hexadecimal interpretation of N to its binary representation : $X_2 = 1001100101$
7. Let b_2 be equal to the number of 1's in the last binary representation : $b_2 = 5$
8. The encryption is the result of $M \text{ xor } (b_1 * b_2)$: $265 \text{ xor } (3*5) = 262$

This student failed Computational Organization, thats why this student asked the judges of ITESM Campus Monterrey internal ACM programming Contest to ask for the numbers of 1's bits of this two representations so that he can continue playing.

You have to write a program that read a *Number* and give as output the number b_1 and b_2

Input

The first line will contain a number N which is the number of cases that you have to process. Each of the following N Lines ($0 < N \leq 1000$) will contain the number M ($0 < M \leq 9999$, in decimal representation) which is the number the student wants to encrypt.

Output

You will have to output N lines, each containing the number b_1 and b_2 in that order, separated by one space corresponding to that lines number to crypt

Sample Input

```
3
265
111
1234
```

Sample Output

```
3 5
6 3
5 5
```