

en[i]gma 0x01 Senior

December 17, 2023

Problems

Problem	Time Limit	Memory Limit	Points
MASTERMIND - Annoula	1 sec	16MB	15
Encrypted Messages	1 sec	16MB	20
Crack the Code	2.1 sec	512MB	25
MASTERMIND - Totos	3 sec	256MB	40
Total			100

MASTERMIND - Annoula

Totos and Annoula, have finished their homework and decided to play the board game Mastermind. In Mastermind, one player hides a code of four different colors, and the other player tries to find it. In each round, the guessing player tries a combination of colors (that could be the repeating), and the one who hid the code provides feedback on the attempt. Specifically, they indicate how many colors from the secret code are present in the guessed combination and are in the correct position, and how many colors are present but in the wrong position. There are a total of 8 colors, and the goal of the second player is to find the code in the fewest attempts possible.



In our case, Annoula is the one hiding the code, and Totos is trying to find it. Help Annoula give the correct feedback to Totos by writing a program that calculates the number of correct colors that are also in the correct position, and the number of correct colors that are in the wrong position.

Input

The colors are 8 in total and are represented by one of the numbers 0-7. The format of the input is as follows:

1st line: 4 numbers A, B, C, D, representing the secret code set by Annoula.

2nd line: 1 number N, the number of Totos's attempts.

3rd to (N + 2)th lines: 4 numbers x, y, z, w, representing Totos's attempt.

Output

1st to Nth line: For each attempt by Totos, you will print 2 numbers, f and g, where f is the number of correct colors in the correct position, and g is the number of colors in the wrong position.

Note:

It is not certain that Totis will ever find the secret code, and in the case that he finds it in some attempt before the Nth, he may continue trying other combinations for fun.

Σημείωση:

Δεν είναι σίγουρο ότι ο Τοτός θα βρει τον μυστικό κωδικό και σε περίπτωση που τον βρει σε κάποια προσπάθεια πριν την Nοστή μπορεί να συνεχίσει να δοκιμάζει άλλους συνδυασμούς για πλάκα.

Examples

1st

STDIN

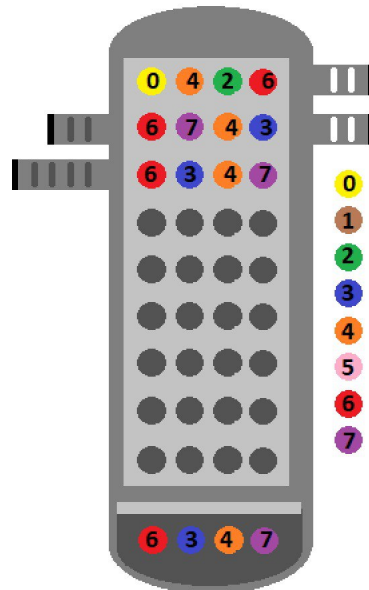
```
6 3 4 7
3
0 4 2 6
6 7 4 3
```

6 3 4 7

STDOUT

0 2
2 2
4 0

The game of the first example can be seen on the image below.



2nd

STDIN

0 3 2 1
12
5 2 6 4
4 2 5 7
6 1 6 4
3 3 3 3
1 7 5 2
5 6 4 7
1 2 3 4
3 4 5 6
5 2 4 2
4 3 5 6
0 3 2 1
4 6 2 1

STDOUT

```
0 1
0 1
0 1
1 0
0 2
0 0
0 3
0 1
0 1
1 0
4 0
2 0
```

Encrypted Messages

It is Friday noon, and Totos is very tired from all this week's classes, so he decides to communicate with his friend Annoula during a class.

Since they can not be heard talking during the class, they decide to exchange messages on small pieces of paper.

However, last time they tried that, they were caught by their teacher who read their messages out loud for the whole class to hear!

Of course they wanted to prevent that from happening again, so from then on they decided to encrypt and decrypt all the messages they exchange!



In order to avoid catching his teachers attention, Totos needs to be able to read and send messages quickly, so he would like to write a program to automate this procedure.

Your task is to write this program for him.

You will not be given the algorithm for the encryption and decryption of the messages, so you will have to obtain that yourselves from the input and output samples.

For your convenience, the table below contains the ASCII values for the uppercase latin letters and the space character. ASCII is a data-encoding format that assigns standard numeric values to letters, numerals, punctuation marks, and other characters used in computers so that we can process them more easily.

ASCII Values - Part 1

Character	(space)	A	B	C	D	E	F	G	H	I	J	K	L	M	N
ASCII Value	32	65	66	67	68	69	70	71	72	73	74	75	76	77	78

ASCII Values - Part 2

Character	O	P	Q	R	S	T	U	V	W	X	Y	Z
ASCII Value	79	80	81	82	83	84	85	86	87	88	89	90

In Python programming language you can convert a character to its ASCII value using the function `ord()` and an ASCII value to its corresponding character using the function `chr()`. In C/C++/Java you don't need an instruction for that and you can handle the characters as if they are their corresponding ASCII values.

Input

The first line of input contains either `ENCRYPT` or `DECRYPT` which represents whether you need to encrypt or decrypt (respectively) the message that follows

The second line of input contains the message that needs to be encrypted or decrypted which will consist of at most 500 uppercase letters or space characters.

Output

In a single line, output the encrypted or decrypted the message as requested.

Examples

1st

STDIN

```
ENCRYPT  
ENIGMA
```

STDOUT

```
HQLJPD
```

2nd

STDIN

```
ENCRYPT  
THE CAKE IS A LIE
```

STDOUT

```
WKH FDNH LV D OLH
```

3rd

STDIN

```
DECRYPT  
WKH FDNH LV D OLH
```

STDOUT

```
THE CAKE IS A LIE
```

Crack the Code

Over the past year you've been taking a hacking course and now it's time to show what you've learned in a test! The test has 1 problem that is simple (or at least that's what your teacher, Mr. Ergastopoulos, says): you have to break a code to get the PIN for a credit card.



Thankfully, you've been taught this way of coding during your course: You can store a number x within another A with a parameter: the so-called "base" b . To recover x , knowing b, A you perform the following calculation:

- c++ / java etc.: `x = A % b;`
- python: `x = A % b`

x is thus the remainder of the integer division A/b .

But what happens when you want to store various numbers $x_1, x_2 \dots x_n$ within A using b ? Mr. Ergastopoulos hasn't taught you how (you do need to do some things by yourself in the exam!), but has told you that you can "clean" A from x_1 in order to read x_2 :

- c++ / java etc.: `x = A / b;`
- python: `x = A // b`

So you divide A by b and keep the whole part of the division. Now you can get x_2 , as you learned above.

An example for: $b = 10, A = 42$

- To get the first hidden number, you do: $x = A \% b = 42 \% 10 = 2$
- To "clear" the first hidden number, you do: $A' = A / b = 42 / 10 = 4.2 \rightarrow 4$
- To get the second hidden number, you do: $x' = A' \% b = 4 \% 10 = 4$
- and so forth.

Problem

Write a script that reads the base b and the number A and finds the requested hidden PIN digits.

Examples

1st

STDIN

```
10 42172413
5
1 2 8 4 2
```

STDOUT

```
3
1
4
2
1
```

Explanation

We break A into x_1, x_2, \dots :

- x_1 : 3
- x_2 : 1
- x_3 : 4
- x_4 : 2
- x_5 : 7
- x_6 : 1
- x_7 : 2
- x_8 : 4

The requested digits are therefore:

x_1, x_2, x_8, x_4, x_2

(we don't care that x_2 is being requested twice, the answer stays the same)

2nd

STDIN

```
4 343434
3
1 3 5
```

STDOUT

```
2
0
1
```


Restrictions

- $b \geq 2$
- $1 \leq A \leq 2 \cdot 10^9$
- $1 \leq Q \leq 50$
- d_i always expresses a number that is stored in A (eg. the third hidden digit in number 5 with base 10 or the 0th digit of a number will not be requested).

For an extra 20% of the grade, $1 \leq Q \leq 2 \cdot 10^6, 1 \leq A \leq 2 \cdot 10^{18}$ (use `long long` instead of `int` in languages such as C/C++/Java etc.)

MASTERMIND - Totos

Totos and Annoula, have finished their homework and decided to play the board game Mastermind. In Mastermind, one player hides a password of four different colors, and the other player tries to find it. In each round, the guessing player tries a combination of colors (that could be the repeating), and the one who hid the password provides feedback on the attempt. Specifically, they indicate how many colors from the secret password are present in the guessed combination and are in the correct position, and how many colors are present but in the wrong position. There are a total of 8 colors, and the goal of the second player is to find the password in the fewest attempts possible.



In our case, Annoula is the one hiding the password, and Totos is trying to find it. Sometimes Annoula wants to help Totos by revealing some colors from the password. The task is to help Totos find the secret password in the fewest attempts possible by writing a program that interacts with the computer.

Input - Interaction System

1st line: a single number N , the number of passwords in the testcase.

Next lines: Initially, on a line, there are 4 characters (a, b, c, d), each of which is either a number from 0 to 7 or the character '?', representing Annoula's hint. After each attempt, there are 2 numbers - Annoula's feedback for the attempt. If the feedback is the numbers 4 and 0, the next password is provided.

Your goal is to find the N passwords, specifically the numbers hidden behind the question marks '?' in each password.

The problem is interactive. After reading the 4 characters, the program starts by printing 4 numbers (x, y, z, w) as Totos's first attempt.

Then the grader provides feedback for the attempt (the grader simulates Annoula), indicating the number of correct colors in the correct position and the number of correct colors in the wrong position. The program uses this information to decide on the next combination, and the process repeats until the hidden password is found (i.e., the grader's response is 4 0).

Subsequently, the program reads Annoula's hint for the next password, and so on.

Grading

The scoring in this problem will depend on the number of attempts made by the program. Specifically, the grader will find the password where the program made the most attempts, and according to the following table, the corresponding percentage of points will be given.

Max Attempts	Percentage(%)
≤ 17	100
≤ 32	80
≤ 79	50
≤ 94	40
≤ 326	30
≤ 1680	20
≤ 4096	10
> 4096	0

Examples

1st

STDIN	STDOUT
1	
? ? ? ?	
	0 4 2 6
0 2	
	6 7 4 3
2 2	
	6 3 4 7
4 0	

2nd

STDIN	STDOUT
2	
0 1 2 ?	
	0 1 2 3
4 0	
4 ? ? 7	
	4 5 0 7
3 0	
	4 5 1 7
4 0	

Subtasks

Testcase	Description
1	In Annoula's hints there will be only 1 question mark '?'. This means that only 1 color needs to be found.
2	In Annoula's hints there will be up to 2 question marks '?'.
3	In Annoula's hints there will be up to 3 question marks '?'.
4	In Annoula's hints there will be up to 4 question marks '?'.

Cheat Code ;)

C

```
#include<stdio.h>

void solve(){
    char a,b,c,d;
    scanf("%c %c %c %c", &a, &b, &c, &d);
    getchar();
    while(1){
        printf("0 0 0 0\n");
        int cc, cw;
        scanf("%d %d", &cc, &cw);
        getchar();
        if(cc==4) return;
    }
}

int main(){
    int n;
    scanf("%d", &n);
    getchar();
    for(int i=0; i<n; i++){
        solve();
    }
    return 0;
}
```

C++

```
#include<iostream>

using namespace std;

void solve(){
    char a,b,c,d;
    cin>>a>>b>>c>>d;
    while(1){
        cout<<"0 0 0 0"<<endl;
        int cc,cw;
        cin>>cc>>cw;
        if(cc==4)return;
    }
}

int main(){
```

```
int n;  
cin>>n;  
for(int i=0;i<n;i++){  
    solve();  
}  
}
```

Python

```
def solve():  
    code=input()  
    code_split=code.split(' ')  
    a,b,c,d=code_split[0],code_split[1],code_split[2],code_split[3]  
    while 1:  
        print('0 0 0 0')  
        feedback=input()  
        if feedback.split(' ')[0]=='4':  
            return  
  
n=input()  
n=int(n)  
for i in range(n):  
    solve()
```