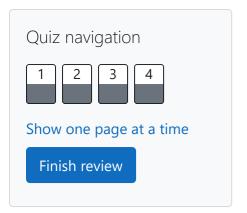
# GE23131-Programming Using C-2024

# mohamed rifai 240901059



Status	Finished
Started	Tuesday, 14 January 2025, 7:54 AM
Completed	Tuesday, 14 January 2025, 8:11 AM
Duration	16 mins 43 secs

# Question **1**

Correct

Marked out of 1.00

Flag question

Given a string, **s**, consisting of alphabets and digits, find the frequency of each digit in the given string.

# **Input Format**

The first line contains a string, *num* which is the given number.

#### **Constraints**

#### $1 \leq len(num) \leq 1000$

All the elements of num are made of English alphabets and digits.

### **Output Format**

Print ten space-separated integers in a single line denoting the frequency of each digit from **0** to **9**.

### Sample Input 0

a11472o5t6

# **Sample Output 0**

0210111100

In the given string:

- · **1** occurs two times.
- **2, 4, 5, 6** and **7** occur one time each.

The remaining digits **0**, **3**, **8** and **9** don't occur at all.

```
#include<stdio.h>
 2
    int main()
 3 ▼ {
         char str[1000];
 4
         scanf("%s",str);
 5
         int hash[10]={0,0,0,0,0,0,
 6
 7
         int temp;
 8
         for(int i=0;str[i]!='\0'
 9 •
10
             temp=str[i]-'0';
             if(temp<=9&&temp>=0)
11
12 🔻
                 hash[temp]++;
13
14
15
        for(int i=0;i<=9;i++)</pre>
16
17 🔻
             printf("%d ",hash[i]
18
19
         return 0;
20
21
```

	Input	Expected
<b>~</b>	a11472o5t6	0 2 1 0 1 1 1 1 0 0
~	lw4n88j12n1	0210100020
~	1v888861256338ar0ekk	1 1 1 2 0 1 2 0 5 0

#### Passed all tests! <

Question **2** 

Correct

Marked out of 1.00

Flag question

Today, Monk went for a walk in a garden. There are many trees in the garden and each tree has an English alphabet on it. While Monk was walking, he noticed that all trees with vowels on it are not in good state. He decided to take care of them. So, he asked you to tell him the count of such trees in the garden.

**Note**: The following letters are vowels: 'A', 'E', 'I', 'O', 'U', 'a', 'e', 'i', 'o' and 'u'.

#### Input:

The first line consists of an integer *T* denoting the number of test cases.

Each test case consists of only one string, each character of string denoting the alphabet (may be lowercase or uppercase) on a tree in the garden.

# Output:

For each test case, print the count in a new line.

#### **Constraints**:

 $1 \le T \le 10$  $1 \le length of string \le 10^5$ 

#### **SAMPLE INPUT**

2

nBBZLaosnm

#### **SAMPLE OUTPUT**

2

1

# **Explanation**

In test case 1, a and o are the only vowels. So, count=2

```
#include<stdio.h>
 2
    int main()
 3 ▼ {
        int t;
 4
        scanf("%d",&t);
 5
        while(t--)
 6
 7 🔻
 8
             char str[100000];
 9
             int count=0;
            scanf("%s",str);
10
             for(int i=0;str[i]!=
11
12 🔻
13
                 char c= str[i];
                 if((c=='a')||(c=
14
15
                 count++;
16
17
            printf("%d\n",count)
18
        return 0;
19
20
```

	Input	Expected	Got	
~	2	2	2	~
	nBBZLaosnm	1	1	
	JHkIsnZtTL			
~	2	2	2	~
	nBBZLaosnm	1	1	
	JHkIsnZtTL			
4 4				
<b>■</b> Passed	d all tests! 🗸	,		

Marked out of 1.00

▼ Flag question

tne sentence in a new line.

### **Input Format**

The first and only line contains a sentence, **s**.

#### **Constraints**

# $1 \le len(s) \le 1000$

## **Output Format**

Print each word of the sentence in a new line.

# Sample Input 0

This is C

# **Sample Output 0**

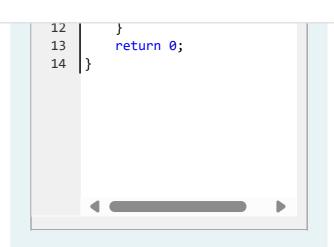
This

is

C

### **Explanation 0**

In the given string, there are three words ["This", "is", "C"]. We have to print each of these words in a new line.



✓ L	earning	This is C Learning	This is C
✓ L	earning	С	С
		Learning	Learning
С			
	is fun	С	С
		is	is
		fun	fun
4			
Deced	all tests!	,	

Question **4** 

Correct

Marked out of 1.00

▼ Flag question

# **Input Format**

You are given two strings, **a** and **b**, separated by a new line. Each string will consist of lower case Latin characters ('a'-'z').

# **Output Format**

In the first line print two space-separated integers, representing the length of **a** and **b** respectively.

In the second line print the string produced by concatenating **a** and **b** (**a** + **b**).

In the third line print two strings separated by a space, **a'** and **b'**. **a'** and **b'** are the same as **a** and **b**, respectively,

# **Sample Input**

abcd ef

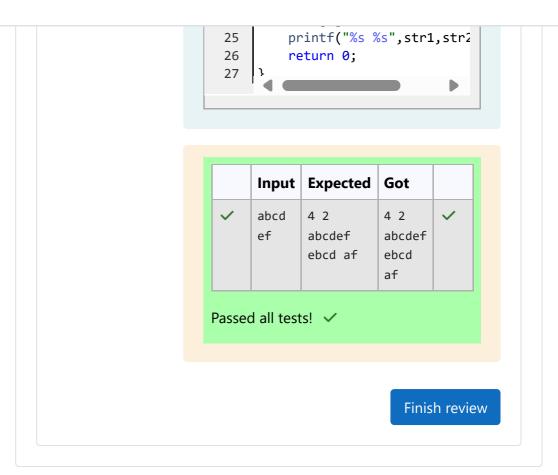
# **Sample Output**

4 2 abcdef ebcd af

### **Explanation**

```
a = "abcd"
b = "ef"
|a| = 4
|b| = 2
a + b = "abcdef"
a' = "ebcd"
b' = "af"
```

```
1 #include<stdio.h>
 2
    int main()
3 ▼ {
        char str1[10],str2[10],t
4
5
        int i=0,j=0;
 6
        int count1=0, count2=0;
7
        scanf("%s",str1);
 8
        scanf("%s",str2);
 9
        while(str1[i]!='\0')
10 🔻
11
            count1++;
12
            i++;
13
14
        while(str2[j]!='\0')
15
16 🔻
17
            count2++;
18
            j++;
19
        printf("%d %d\n",count1,
20
        printf("%s%s\n",str1,str
21
```



1.