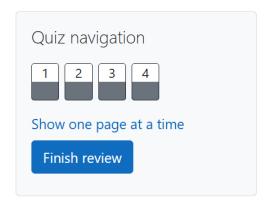
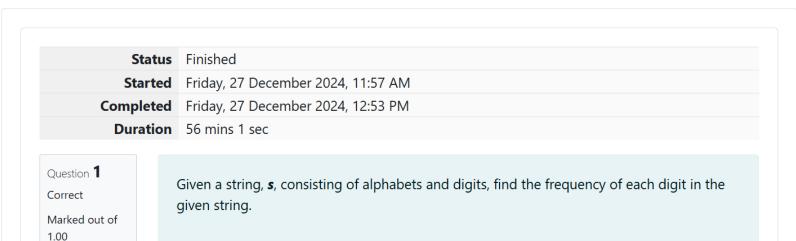
# GE23131-Programming Using C-2024





#### **Input Format**

Flag question

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

#### Constraints

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

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

## Sample Input 0

a11472o5t6

### **Sample Output 0**

0210111100

#### **Explanation 0**

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)
int main()
{char s[1000];
scanf("%s",s);
int i=0,a[10]={0,0,0,0,0,0,0,0,0};
while(s[i]!='\0')
{if(s[i]=='0')
a[0]+=1;
else if(s[i]=='1')
a[1]+=1;
11 else if(s[i]=='2')
12 a[2]+=1;
13 else if(s[i]=='3')
```

```
14 |a[⊃]+=1,
15 | else if(s[i]=='4')
16 a[4]+=1;
17 | else if(s[i]=='5')
18 a[5]+=1;
19 | else if(s[i]=='6')
20 a[6]+=1;
21 | else if(s[i]=='7')
22 a[7]+=1;
23 | else if(s[i]=='8')
24 a[8]+=1;
25 | else if(s[i]=='9')
26 a[9]+=1;
27 i++;}
    for(int i=0;i<10;i++)</pre>
28
   {printf("%d ",a[i]);}}
29
30
```

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

JHklsnZtTL

#### **SAMPLE OUTPUT**

2

1

# **Explanation**

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

```
1 #include<stdio.h>
 2 int main()
 3 v {int a,i,q;
        scanf("%d",&a);
 5 for(int j=0;j<a;j++)</pre>
 6 \neq \{i=0, q=0;
 7 char s[100];
 8
        scanf("%s",s);
 9
       while(s[i]!=0)
    {if(s[i]=='a'||s[i]=='e'||s[i]=='i'||s[i]=='o'||s[i]=='u'
10
        ||s[i]=='A'||s[i]=='E'||s[i]=='I'||s[i]=='0'||s[i]=='U')
11
12
           q+=1;
      i++; }
13
14
       printf("%d\n",q);
15 }
16 }
```

	Input	Expected	Got	
<b>~</b>	2 nBBZLaosnm JHkIsnZtTL	2	2	~
<b>~</b>	2 nBBZLaosnm JHkIsnZtTL	2	2	~

Question **3** 

Correct

Marked out of 1.00

Flag question

Given a sentence, **s**, print each word of the sentence in a new line.

# **Input Format**

The first and only line contains a sentence, s.

#### Constraints

 $1 \leq len(s) \leq 1000$ 

# **Output Format**

Print each word of the sentence in a new line.

# Sample Input 0

This is C

# Sample Output 0

This

is

 $\mathcal{C}$ 

# **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.

```
#include<stdio.h>
int main()
    {char s[100];
    scanf("%[^\n]",s);
    int i=0;
    while(s[i]!='\0')
    {if(s[i]=='')
        printf("\n");
    else
    printf("%c",s[i]);
    i++;
    }}
```

	Input	Expected	Got	
~	This is C	This is C	This is C	~
<b>~</b>	Learning C is fun	Learning	Learning	~

_	С	С
	is	is
	fun	fun

Passed all tests! <

Question 4

Correct

Marked out of 1.00

Flag question

## **Input Format**

You are given two strings,  $\boldsymbol{a}$  and  $\boldsymbol{b}$ , 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  $\boldsymbol{a}$  and  $\boldsymbol{b}$  ( $\boldsymbol{a} + \boldsymbol{b}$ ).

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, except that their first characters are swapped.

### **Sample Input**

abcd

ef

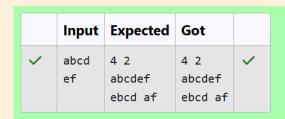
# **Sample Output**

```
abcdef
ebcd af
```

# **Explanation**

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

```
1 #include<stdio.h>
 2 #include<string.h>
 3 int main()
 4 {char a[100],b[100];
 5 scanf("%[^\n]",a);
 6 getchar();
 7 scanf("%[^\n]",b);
 8 getchar();
9 int c=strlen(a);
10 int d=strlen(b);
11 | printf("%d %d\n",c,d);
12 printf("%s%s\n",a,b);
13 char temp=a[0];
14 a[0]=b[0];
15 b[0]=temp;
16 | printf("%s %s",a,b);
17 }
```



Passed all tests! 🗸

Finish review