# GE23131-Programming Using C-2024

Da··· / M··· / GE231··· / Week-10-Char··· / C··

# Coding



# Re-attempt quiz

Attempts allowed: 5

Time limit: 1 hour 30 mins

Grading method: Highest grade

# Your attempts

Attem	pt 1	
	Status	Finished
	Started	Wednesday, 15 January 2025, 7:25 PM
(	Completed	Wednesday, 15 January 2025, 7:52 PM
	Duration	26 mins 46 secs
Reviev	V	

# GE23131-Programming Using C-2024

Status	Finished
Started	Wednesday, 15 January 2025, 7:25 PM
Completed	Wednesday, 15 January 2025, 7:52 PM
Duration	26 mins 46 secs

Question **1**Correct

Flag question

Marked out of 1.00

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

# Constraints

1 ≤ len(num) ≤ 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

**Explanation 0** 

0210111100

In the given string:

1 occurs two times.

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

<pre>7    int temp; 8    for(int i=0;str[i]!='\0'; 9</pre>	1	#include <staio.n></staio.n>
<pre>char str[1000]; scanf("%s",str); int hash[10]={0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,</pre>	2	<pre>int main()</pre>
<pre>5</pre>	3 ▼	{
<pre>int hash[10]={0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,</pre>	4	char str[1000];
<pre>int temp; for(int i=0;str[i]!='\0';</pre>	5	scanf("%s",str);
<pre>8 9 v 10 10 11</pre>	6	int hash[10]={0,0,0,0,0,0
<pre>9</pre>	7	<pre>int temp;</pre>
<pre>10 11</pre>	8	<pre>for(int i=0;str[i]!='\0';</pre>
<pre>if(temp&lt;=9&amp;&amp;temp&gt;=0)  if(temp&lt;=9&amp;&amp;temp&gt;=0)  {    hash[temp]++; }  for(int i=0;i&lt;=9;i++)  for(int i=0;i&lt;=9;i++)  return 0;</pre>	9 ▼	{
12 v { 13	10	temp=str[i]-'0';
13	11	if(temp<=9&&temp>=0)
14	12 ▼	{
15 } 16 for(int i=0;i<=9;i++) 17 * { 18 printf("%d ",hash[i]) 19 } 20 return 0;	13	hash[temp]++;
16 for(int i=0;i<=9;i++) 17 * { 18	14	}
17 v { 18	15	}
18	16	for(int i=0;i<=9;i++)
19 } 20 return 0;	17 ▼	{
20 return 0;	18	<pre>printf("%d ",hash[i])</pre>
	19	}
21  }	20	return 0;
	21	}

	Input	E	хp	ec	te	d
~	a11472o5t6	0	2	1	0	1
~	lw4n88j12n1	0	2	1	0	1
~	1v888861256338ar0ekk	1	1	1	2	0

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

1	<stdio.h></stdio.h>
2	)
3 ▼	
	str[1000];
5	("%s",str);
	ash[10]={0,0,0,0,0,0,0,0,0,0,};
7	emp;
	nt i=0;str[i]!='\0';i++)
9 🔻	
	emp=str[i]-'0';
11	f(temp<=9&&temp>=0)
12 ▼	
13	hash[temp]++;
14	
15	
	nt i=0;i<=9;i++)
17 ▼	
	rintf("%d ",hash[i]);
19	
	n 0;
21	

	Input	E	хp	ec	te	d
~	a11472o5t6	0	2	1	0	1
~	lw4n88j12n1	0	2	1	0	1
~	1v888861256338ar0ekk	1	1	1	2	0
Passe	ed all tests!	1		1	2	

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:

# 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

nBBZLaosnm JHklsnZtTL

2

2

1

## SAMPLE OUTPUT

# Explanation

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

Answer: (penalty regime: 0 %)

1 #include <stdio.h>

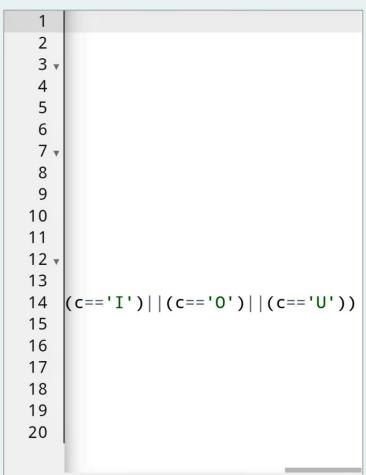
Answer: (penalty regime: 0 %)

```
#include <stdio.h>
 1
 2
    int main()
 3 ▼
    {
 4
         int t;
 5
         scanf("%d",&t);
         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];
14
                  if((c=='a')||(c==
15
                  count++;
16
17
             printf("%d\n",count);
18
         return 0;
19
20
    }
```

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

Passed all tests! ✓

**Answer:** (penalty regime: 0 %)



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

Passed all tests! <

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 ≤ len(s) ≤ 1000

Output Format

Print each word of the sentence in a new line.

Sample Input 0

This is C

This

Sample Output 0

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

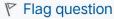
```
1
    #include <stdio.h>
    int main()
 2
 3 ▼
    {
 4
         char s[1000];
 5
         scanf("%[^\n]s",s);
 6
         for(int i=0;s[i]!='\0';i+
 7 🔻
         {
             if(s[i]!=' ')
 8
 9
             printf("%c",s[i]);
             else
10
             printf("\n");
11
12
13
         return 0;
14
    }
```

	Input	Expected	Go
~	This is C	This	Thi is
		C	C
~	Learning C is fun	Learning	Lea
		С	C
		is	is
		fun	fur

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>
 1
 2
   nt main()
 3 •
      char s[1000];
 4
 5
       scanf("%[^\n]s",s);
 6
       for(int i=0;s[i]!='\0';i++)
 7 ▼
       {
           if(s[i]!=' ')
 8
 9
           printf("%c",s[i]);
           else
10
           printf("\n");
11
12
13
       return 0;
14
```

Input	Expected	Go
This is C	This	Thi
	is	is
	С	С
Learning C is fun	Learning	Lea
	С	С
	is	is
	fun	fur
	This is C	This is C  This is C  C  Learning C is fun Learning C  is



### **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  $\boldsymbol{a}$  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

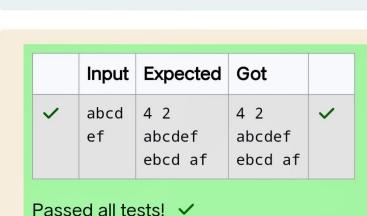
ef

# Sample Output

```
|a| = 4
|b| = 2
a + b = "abcdef"
a' = "ebcd"
b' = "af"
Answer: (penalty regime: 0 %)
       #include <stdio.h>
    1
    2
       int main()
    3 ▼
       {
    4
            char str1[10],str2[10],t;
    5
            int i=0, j=0;
    6
            int count1=0,count2=0;
            scanf("%s", str1);
    7
    8
            scanf("%s", str2);
            while(str1[i]!='\0')
    9
  10 •
            {
  11
                count1++;
  12
                i++;
  13
  14
            while(str2[j]!='\0')
  15 ▼
  16
                count2++;
  17
                j++;
  18
            }
            printf("%d %d\n",count1,c
  19
            printf("%s%s\n",str1,str2
  20
  21
            t=str1[0];
  22
            str1[0]=str2[0];
  23
            str2[0]=t;
            printf("%s %s",str1,str2)
  24
  25
            return 0;
  26
       }
```

b = "ef"

```
4
    r str1[10],str2[10],t;
 5
     i=0, j=0;
 6
     count1=0, count2=0;
    nf("%s",str1);
 7
 8
    nf("%s", str2);
    le(str1[i]!='\0')
 9
10 •
11
     count1++;
12
     i++;
13
    le(str2[j]!='\0')
14
15 •
    count2++;
16
     j++;
17
18
19
    ntf("%d %d\n",count1,count2);
    ntf("%s%s\n", str1, str2);
20
    tr1[0];
21
22
    1[0]=str2[0];
   2[0]=t;
23
    ntf("%s %s",str1,str2);
24
25
    urn 0;
26
```



Finish review