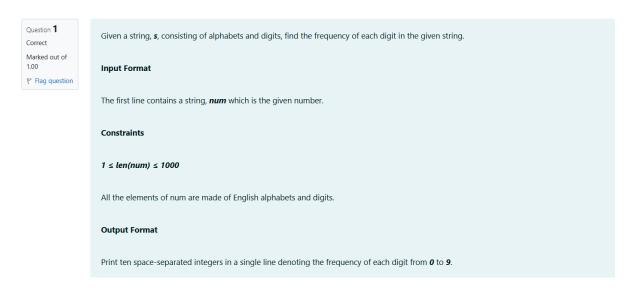
Week-10-Character Arrays and Strings





Source code:

```
1 #include<stdio.h>
    int main()
 2
3 ▼ {
        char str[1000];
scanf("%s",str);
4
5
        char num[10]={'0','1','2','3','4','5','6','7','8','9'};
 6
        for(int i=0;i<=9;i++)</pre>
7
8 🔻
9
          int count=0;
10
         for(int j=0;str[j]!='\0';j++)
11 •
             if(str[j]==num[i])
12
13 v
                  count++;
14
15
16
         printf("%d ",count);
17
18
19 }
```



Question **2**Correct
Marked out of 1.00

F 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 ${\it T}$ denoting the number of test cases.

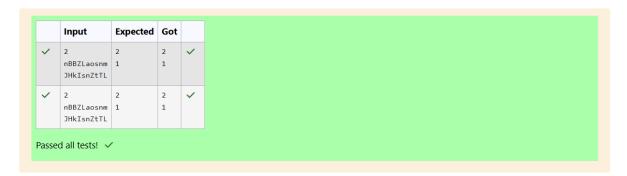
Each test case consists of only one string, each character of string denoting the alphabet (may be lowercase or uppercase) on a tree in the garden.

Output:

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

Source code:

```
#include<stdio.h>
                              int main()
       2
       3 ▼ {
                                                             int T;
       4
       5
                                                            char ch;
                                                            scanf("%d",&T);
       6
                                                            while(T--)
       8
                                                                                         char str[100];
                                                                                       int count=0;
scanf("%s",str);
 10
11
                                                                                         for(int i=0;str[i]!='\0';i++)
12
13
                                                                                                                      \begin{array}{l} ch=str[i];\\ if((ch=-'a'))||(ch=-'e')||(ch=-'i')||(ch=-'u')||(ch=-'A')||\\ |(ch=-'E')||(ch=-'E')||(ch=-'I')||(ch=-'U')||\\ |(ch=-'B')||(ch=-'B')||(ch=-'B')||\\ |(ch=-'B')||(ch=-'B')||(ch=-'B')||\\ |(ch=-'B')||(ch=-'B')||(ch=-'B')||\\ |(ch=-'B')||(ch=-'B')||\\ |(ch=-'B')||(ch=-'B')||\\ |(ch=-'B')||(ch=-'B')||\\ |(ch=-'B')||\\ |(ch=-'B')||\\
14
15
16 1
17
                                                                                                                                                  count++;
18
19
20
                                                                                         printf("%d\n",count);
21
22
                                                          return 0;
                          n
23
```



```
Question 3
Correct
Marked out of 1.00

F 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

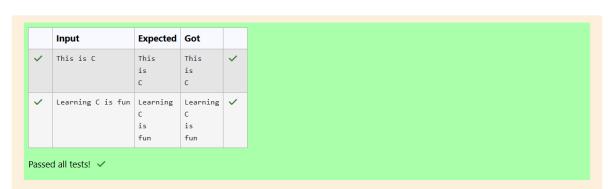
 $1 \leq len(s) \leq 1000$

Output Format

Print each word of the sentence in a new line.

Source code:

```
#include<stdio.h>
2 int main()
3 * {
         char str[1000];
fgets(str,1000,stdin);
for(int i=0;str[i]!='\0';i++)
4
5
6
7 🔻
              if(str[i]!=' ')
8
9 🔻
                  printf("%c",str[i]);
10
11
12
              else
13 🔻
              {
14
                  printf("\n");
15
16
         return 0;
17
18 }
```



```
Question 4
Correct
Marked out of 1.00
F 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 \boldsymbol{a} and \boldsymbol{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.

Source code:

```
1 #include<stdio.h>
  2 int main()
3 * {
  4
            char arr[100],brr[100],temp;
            int count1=0,count2=0;
scanf("%s",arr);
scanf("%s",brr);
   5
   6
   8
            for(int i=0;arr[i]!='\0';i++)
   9
  10
                 count1++;
  11
            for(int j=0;brr[j]!='\0';j++)
  12
  13 •
            {
                 count2++;
  14
  15
            printf("%d %d \n",count1,count2);
printf("%s%s\n",arr,brr);
  16
  17
  18
            temp=arr[0];
  19
            arr[0]=brr[0];
            brr[0]=temp;
printf("%s %s",arr,brr);
  20
  21
  22
            return 0;
  23 }
```

