Week:10-Character Arrays
Name:Krishna Akhileshwaran AV
Roll no: 241501091

Q1)problem statement:

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.
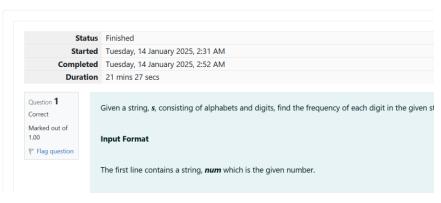
Sample Input 0

a11472o5t6

Sample Output 0

0 2 1 0 1 1 1 1 0 0

GE23131-Programming Using C-2024

Status	Finished
Started	Tuesday, 14 January 2025, 2:31 AM
Completed	Tuesday, 14 January 2025, 2:52 AM
Duration	21 mins 27 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 st

Input Format

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

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

Output:

	Input	Expected	Got	
✓	a11472o5t6	0 2 1 0 1 1 1 1 0 0	0 2 1 0 1 1 1 1 0 0	✓
✓	1w4n88j12n1	0 2 1 0 1 0 0 0 2 0	0 2 1 0 1 0 0 0 2 0	✓
✓	1v888861256338ar0ekk	1 1 1 2 0 1 2 0 5 0	1 1 1 2 0 1 2 0 5 0	✓

Passed all tests! ✓

Q2)Problem statement:

 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 Format:

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 Format:

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

Constraints:

$1 \le T \le 10$

$1 \le$ length of string ≤ 105

Sample Input

2

nBBZLaosnm
JHkIsnZtTL

Sample Output

2

1

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

JHkIsnZtTL

SAMPLE OUTPUT

2

1

```c
#include<stdio.h>

int main()
{ int t;
scanf("%d",&t);
while(t--)
{
char str[100000];
int count =0;
scanf("%s",str);
for(int i=0;str[i]!='\0';i++)
{
char c=str[i];
if((c=='a')||(c=='e')||(c=='i')||(c=='o')||(c=='u')||(c=='A')||(c=='E')||(c=='I')||(c=='C
    count++;

}printf("%d\n",count);

}return 0;}
```

Output:

	Input	Expected	Got	
✓	2 nBBZLaosnm JHkIsnZtTL	2 1	2 1	✓
✓	2 nBBZLaosnm JHkIsnZtTL	2 1	2 1	✓

Passed all tests! ✓

Q3)problem statement:

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

This is C

Sample Output

This

is

C

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

Sample Output 0

This
is
C

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

Output:

	Input	Expected	Got	
✓	This is C	This is C	This is C	✓
✓	Learning C is fun	Learning C is fun	Learning C is fun	✓

Passed all tests! ✓

Q4)problem statement:

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, except that their first characters are swapped.

Sample Input

abcd
ef

Sample Output

4 2
abcdef
ebcd af

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, except that their first characters are swapped.

Sample Input

abcd

ef

Sample Output

4 2

abcdef

ebcd af

```c
#include<stdio.h>
int main()
{
    char str1[10],str2[10],t;
    int i=0,j=0;
    int count1=0,count2=0;
    scanf("%s",str1);
    scanf("%s",str2);
    while(str1[i]!='\0')
    {
        count1++;
        i++;
    }
    while(str2[j]!='\0')
    {
        count2++;
        j++;
    }
    printf("%d %d \n",count1,count2);
    printf("%s%s\n",str1,str2);
    t=str1[0];
    str1[0]=str2[0];
    str2[0]=t;
    printf("%s %s",str1,str2);
    return 0;
}
```

Output:

	Input	Expected	Got	
✓	abcd ef	4 2 abcdef ebcd af	4 2 abcdef ebcd af	✓

Passed all tests! ✓

s