

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

$$1 \leq \text{len}(num) \leq 1000$$

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

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.

Answer: (penalty regime: 0 %)

```

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

	Input	Expected	G
✓	a11472o5t6	0 2 1 0 1 1 1 1 0 0	0
✓	lw4n88j12n1	0 2 1 0 1 0 0 0 2 0	0
✓	1v888861256338ar0ekk	1 1 1 2 0 1 2 0 5 0	1

Passed all tests! ✓

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:

$$1 \leq T \leq 10$$
$$1 \leq \text{length of string} \leq 10^5$$

SAMPLE INPUT

```
2
nBBZLaosnm
JHkIsnZtTL
```

SAMPLE OUTPUT

```
2
1
```

Explanation

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

Answer: (penalty regime: 0 %)

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

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

Passed all tests! ✓

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

$$1 \leq \text{len}(s) \leq 1000$$

Output Format

Print each word of the sentence in a new line.

Sample Input 0

This is C

Sample Output 0

This
is
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.

Answer: (penalty regime: 0 %)

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

	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! ✓

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

Explanation

a = "abcd"

b = "ef"

|a| = 4

|b| = 2

a + b = "abcdef"

a' = "ebcd"

b' = "af"

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main()
3 {
4     char str1[10],str2[10],t;
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        count2++;
17        j++;
18    }
19    printf("%d %d\n",count1,count2);
20    printf("%s%s\n",str1,str2);
21    t=str1[0];
22    str1[0]=str2[0];
23    str2[0]=t;
24    printf("%s %s",str1,str2);
25    return 0;
26 }
27 }
```

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

Passed all tests! ✓