

Week 10

Roll no : 240801177

Name : Logaranjan R J

Status	Finished
Started	Friday, 27 December 2024, 12:40 AM
Completed	Friday, 27 December 2024, 12:41 AM
Duration	1 min 12 secs

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

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

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 #include <string.h>
3
4 int main() {
5     char num[1001]; // Input string (max length 1000)
6     int digit_count[10] = {0}; // Array to count frequency of digits 0-9
7
8     // Input the string
9     scanf("%s", num);
10
11     // Iterate over each character in the string
12     for (int i = 0; num[i] != '\0'; i++) {
13         if (num[i] >= '0' && num[i] <= '9') {
14             digit_count[num[i] - '0']++; // Increment count for the respective digit
15         }
16     }
17
18     // Print the frequencies of digits 0 to 9
19     for (int i = 0; i < 10; i++) {
20         printf("%d ", digit_count[i]);
21     }
22
23     return 0;
24 }

```

	Input	Expected	Got	
✓	a11472o5t6	0 2 1 0 1 1 1 1 0 0	0 2 1 0 1 1 1 1 0 0	✓
✓	lw4n88j12n1	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! ✓

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:

$$1 \leq T \leq 10$$

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

SAMPLE INPUT

2

nBBZLaosnm

JHklsnZtTL

SAMPLE OUTPUT

2

1

Explanation

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

Answer:(penalty regime: 0 %)

```
1 #include <stdio.h>
2 #include <string.h>
3 #include <ctype.h>
4
5 int is_vowel(char ch) {
6     // Convert character to lowercase
7     ch = tolower(ch);
8     // Check if the character is a vowel
9     return (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u');
10 }
11
12 int main() {
13     int T; // Number of test cases
14     scanf("%d", &T);
15     getchar(); // To consume the newline after T
16
17     while (T--) {
18         char str[100001];
19         int count = 0;
20
21         // Read the input string
22         scanf("%s", str);
23
24         // Iterate through the string and count vowels
25         for (int i = 0; str[i] != '\0'; i++) {
26             if (is_vowel(str[i])) {
27                 count++;
28             }
29         }
30
31         // Print the result for the current test case
32         printf("%d\n", count);
33     }
34
35     return 0;
36 }
```

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

Passed all tests! ✓

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

	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! ✓

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

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 #include <string.h>
3
4 int main() {
5     char a[101], b[101]; // Input strings (assuming maximum length of 100)
6
7     // Read the input strings
8     scanf("%s", a);
9     scanf("%s", b);
10
11    // Step 1: Print the lengths of a and b
12    int len_a = strlen(a);
13    int len_b = strlen(b);
14    printf("%d %d\n", len_a, len_b);
15
16    // Step 2: Print the concatenation of a and b
17    printf("%s%s\n", a, b);
18
19    // Step 3: Swap the first characters of a and b, then print the modified strings
20    char temp = a[0];
21    a[0] = b[0];
22    b[0] = temp;
23    printf("%s %s\n", a, b);
24
25    return 0;
26 }
```

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

Passed all tests! ✓