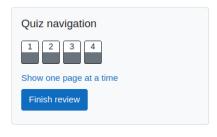
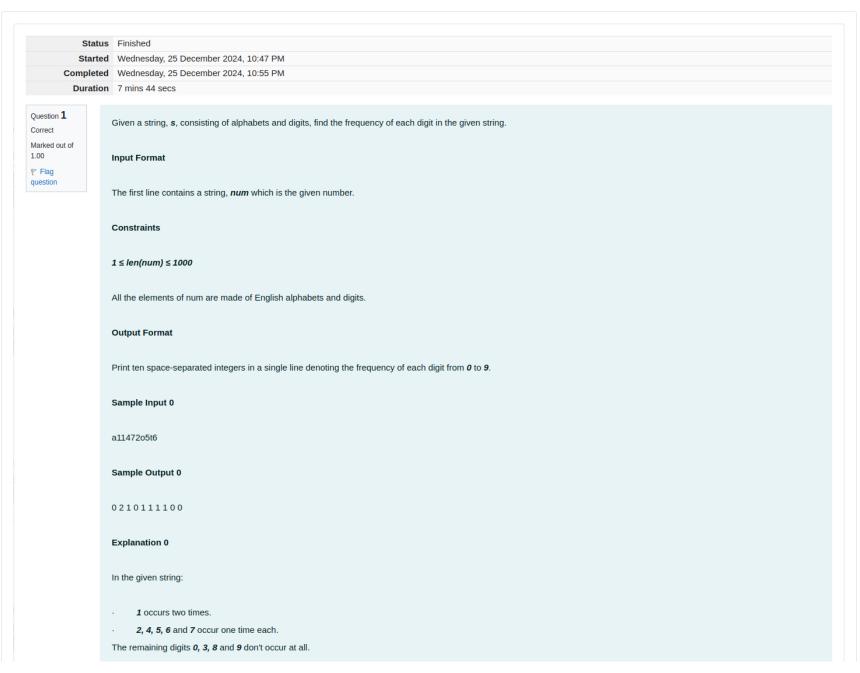
GE23131-Programming Using C-2024





```
Answer: (penalty regime: 0 %)
   1 #include <stdio.h>
   3 #include <string.h>
   5 v int main() {
   7 char num [1001]; // Input string (max length 1000)
       int digit_count [10] = {0}; // Array to count frequency of digits 0-9
  10
       // Input the string
  11
  12
  13
       scanf("%s", num);
  14
  15
       // Iterate over each character in the string
   16
  17 - for (int i = 0; num[i] != '\0'; i++) {
   18
  19 * if (num[i] >= '0' && num[i] <= '9') {
  20
  21 digit_count[num[i] - '0']++; // Increment count for the respective d
  22
  23
  24
  25
  26
  27
       // Print the frequencies of digits 0 to 9
  28
  29 * for (int i = 0; i < 10; i++) {
  30
  31
       printf("%d ", digit_count[i]);
  32
  33
  34
  35
       return 0;
  36
  37
```

		Input		Expected							Got												
	~	a11472o5t6	0	2	1	0	1	1	1	1	0	0	0	2	1	0	1	1	1	1	0	Θ	~
	~	lw4n88j12n1	0	2	1	0	1	0	0	0	2	0	0	2	1	0	1	0	0	0	2	Θ	~
	~	1v88886l256338ar0ekk	1	1	1	2	Θ	1	2	Θ	5	Θ	1	1	1	2	Θ	1	2	0	5	Θ	~

Passed all tests! <

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

5 5

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 JHklsnZtTL

SAMPLE OUTPUT

2

Explanation

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

Answer: (penalty regime: 0 %)

```
1 #include <stdio.h>
    #include <string.h>
4 v int main() {
       int T; // Number of test cases
      scanf("%d", &T);
       // Define the vowels
9
       char vowels[] = "aeiouAEIOU";
10
11 v
       while (T--) {
       char tree_string[100001]; // String for the tree (maximum length 10^5)
12
13
      scanf("%s", tree_string);
14
      int vowel_count = 0; // Initialize vowel count to 0
15
16
17
      // Loop through the string
     for (int i = 0; tree_string[i] != '\0'; i++) {
18 v
      // Check if the character is a vowel
19
20 ₹
     if (strchr(vowels, tree_string[i]) != NULL) {
     vowel_count++;
}
21
22
23
     }
24
      // Print the result for this test case
```

```
26 | printf("%d\n", vowel_count);

27     }

28     return 0;

30 }
```

	Input	Expected	Got	
~	2 nBBZLaosnm JHkIsnZtTL	2	2	~
~	2 nBBZLaosnm JHkIsnZtTL	2	2	~

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 ≤ 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

С

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>
    #include <string.h>
    int main()
5
6
7 ₹
9
    char str[1000];
10
11
    scanf("%[^\n]s",str);
12
    for(int i=0;str[i]!= '\0';i++){
13 ,
14
    if(str[i]==' ')
15
16
    printf("\n");
17
18
19
    else
20
21
    printf("%c", str[i]);
22
23
24
25 }
```

	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! 🗸

Question **4**Correct
Marked out of 1.00

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
42
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>
   4 v int main() {
          char a[101], b[101]; // Declare strings a and b with sufficient length
          scanf("%s", a);
          scanf("%s", b);
          // Print the lengths of a and b
   9
  10
          printf("%ld %ld\n", strlen(a), strlen(b));
  11
  12
          // Print the concatenated string a + b
  13
          printf("%s%s\n", a, b);
  14
  15
          // Swap the first characters of a and b
  16
          char temp = a[0];
  17
          a[0] = b[0];
  18
          b[0] = temp;
  19
  20
          // Print the swapped strings
  21
          printf("%s %s\n", a, b);
  22
  23
          return 0;
  24 }
```

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

Passed all tests! 🗸

Finish review