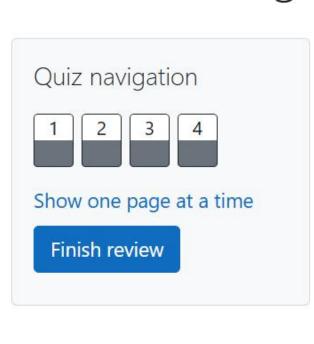
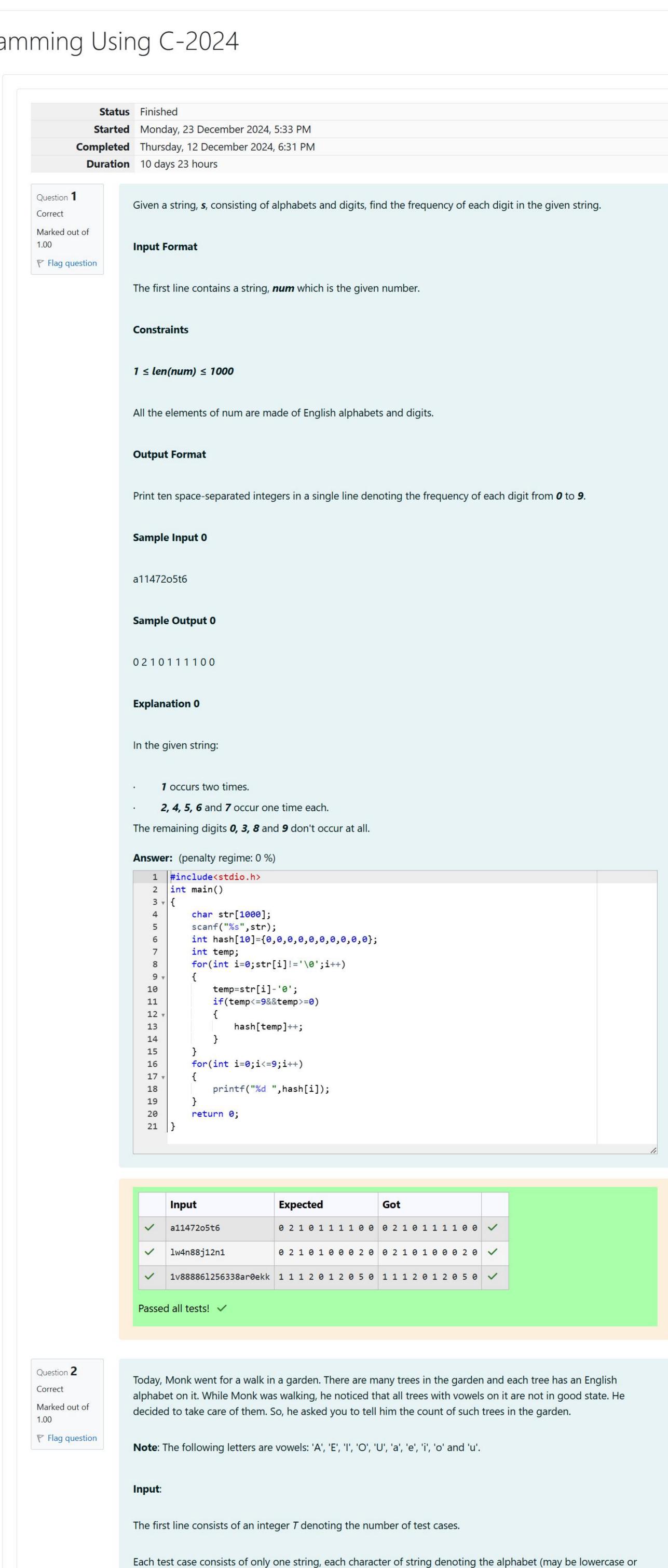
GE23131-Programming Using C-2024





```
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

2

nBBZLaosnm

Explanation

3 *

7 🔻

8

10

Answer: (penalty regime: 0 %)

1 #include<stdio.h>

int t;

while(t--)

scanf("%d",&t);

int main()

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

char str[100000];

scanf("%s",str);

int count=0;

SAMPLE OUTPUT

JHklsnZtTL

for(int i=0;str[i]!='\0';i++) 11 12 13 char c=str[i]; if((c=='a')||(c=='e')||(c=='i')||(c=='o')||(c=='u')||(c=='A')||(c=='E')||(c=='I 14 15 count++; 16 printf("%d\n",count); 17 18 19 return 0; 20 Input **Expected Got** nBBZLaosnm 1 JHkIsnZtTL ~ 2 2 nBBZLaosnm 1 1 JHkIsnZtTL Passed all tests! <

```
Given a sentence, \mathbf{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.
```

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>

char s[1000];

else

return 0;

scanf("%[^\n]s",s);

if (s[i]!=' ')

printf("\n");

for(int i=0;s[i]!='\0';i++)

printf("%c",s[i]);

int main()

2

4

6 7 ▼

8

9 10

11 12 13

14 }

3 ₹ {

Sample Input 0

Sample Output 0

This is C

This

is

```
Expected Got
       Input
      This is C
                        This
                                  This
                                           /
                        is
                                  is
                                  C
                        C
       Learning C is fun Learning
                                  Learning <
                                  is
                        is
                        fun
                                  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

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 \boldsymbol{a} and \boldsymbol{b} (\boldsymbol{a} + \boldsymbol{b}).
In the third line print two strings separated by a space, 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>
       int main()
    2
    3 ₹ {
            char str1[10],str2[10],t;
            int i=0,j=0;
            int count1=0, count2=0;
    6
            scanf("%s",str1);
            scanf("%s",str2);
    8
            while(str1[i]!='\0')
   10
   11
                 count1++;
                 i++;
   12
   13
   14
            while(str2[j]!='\0')
   15
   16
   17
                 count2++;
                 j++;
```

```
Input Expected Got
            4 2
                       4 2
     abcd
            abcdef
                       abcdef
             ebcd af
                       ebcd af
Passed all tests! <
                                                                                          Finish review
```

Question 3

Marked out of

Flag question

Correct

1.00

Question 4

Marked out of

Flag question

('a'-'z').

Output Format

Correct

1.00

18 19 printf("%d %d\n",count1,count2); 20 printf("%s%s\n",str1,str2); 21

22

23

24

25 26

27 }

t=str1[0];

str2[0]=t;

return 0;

str1[0]=str2[0];

printf("%s %s",str1,str2);