



Experiment 5

Student Name: Gaganjot Singh

UID: 22BCS14843

Branch: BE - CSE

Section/Group: 22BCS-JT-802-B

Semester: 5th

Date of Performance: 09/09/2024

Subject Name: Advance Programming Lab

Subject Code: 22CSP - 314

Question 1.

1. Aim:

A pangram is a string that contains every letter of the alphabet. Given a sentence determine whether it is a pangram in the English alphabet. Ignore case. Return either pangram or not pangram as appropriate.

2. Objective:

The given C++ code determines whether a string is a pangram (contains every letter of the English alphabet at least once). It converts the input to lowercase, stores unique alphabetic characters in a set, and checks if the set contains exactly 26 unique letters.

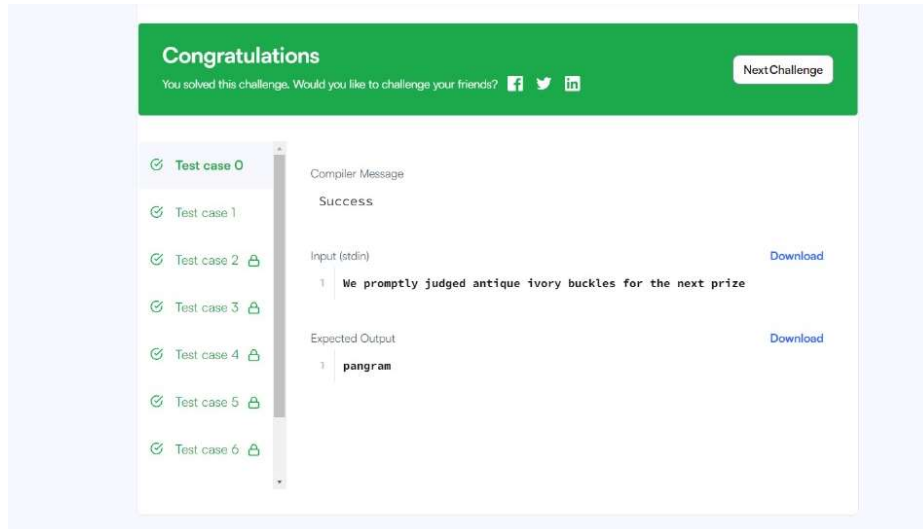
3. Implementation/Code:

```
string pangrams(string s) {
    set<char> letters;

    for (char c : s) {
        if (isalpha(c)) {
            letters.insert(tolower(c));
        }
    }

    if (letters.size() == 26) {
        return "pangram";
    } else {
        return "not pangram";
    }
}
```

4. Output:



5. Learning Outcomes:

- Learn how to use a set to store unique characters efficiently.
- Gain experience in checking alphabetic characters using the isalpha function.
- Learned about Pangram.

Question 2.

1. Aim:

There is a sequence of words in CamelCase as a string of letters, s, having the following properties:

- It is a concatenation of one or more words consisting of English letters.
- All letters in the first word are lowercase.
- For each of the subsequent words, the first letter is uppercase and rest of the letters are lowercase.

Given s, determine the number of words in s.

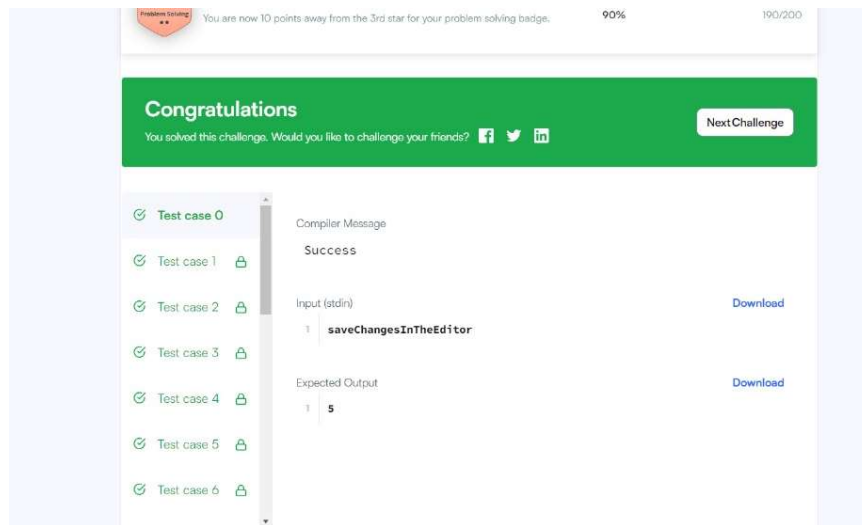
2. Objective:

The given C++ code checks that the string given is in CamelCase and count the number of words used in CamelCase.

3. Implementation/Code:

```
int camelcase(string s) {  
    int ans = 1;  
    for(int i =0;  
        i<s.length();i++)  
    {  
        char ch = s[i];  
        if(isupper(ch))  
        {  
            ans++;  
        }  
    }  
    return ans; }
```

4. Output:



5. Learning Outcomes:

- Learned more about CamelCase way to write the name of variables, functions, etc.
- Learned more about different predefined functions in C++ like `isupper()` and many more.