Experiment 7 (String)

Student Name: SANSKAR AGRAWAL **UID: 20BCS5914**
Branch: BE CSE **Section/Group: 806 B**
Semester: 5th **Date of performance: 18.10.22**
Subject: Competitive Coding **Subject Code: 20CSP_314**

1. **Aim/Overview of the Practical:**
   a. Strong Password.
   b. Camelcase.

2. **Task to be done / Which logistics used:**

a. Louise joined a social networking site to stay in touch with her friends. The signup page required her to input a name and a password. However, the password must be strong. The website considers a password to be strong if it satisfies the following criteria:

Its length is at least 6.
It contains at least one digit.
It contains at least one lowercase English character.
It contains at least one uppercase English character.
It contains at least one special character. The special characters are:
!@#$%^&*()-+
She typed a random string of length n in the password field but wasn't sure if it was strong. Given the string she typed, can you find the minimum number of characters she must add to make her password strong?

b. There is a sequence of words in CamelCase as a string of letters, , 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.

Steps for experiment/practical/Code:

a. Strong Password:

```
#include <bits/stdc++.h>

using namespace std;

int check(string s, string w) {
for (char c : s) {
   if (w.find(c) != string::npos) {
     return 0;
   }   }
return 1;
}

int main() {
int n;   cin
>> n;
string s;
  cin >> s;
  string numbers = "0123456789";
  string lower_case = "abcdefghijklmnopqrstuvwxyz";   string
upper_case = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";   string
special_characters = "!@#$%^&*()-+";   int ans = 0;
  ans += check(s, numbers);
ans += check(s, lower_case);
ans += check(s, upper_case);
ans += check(s,
special_characters);   ans =
```

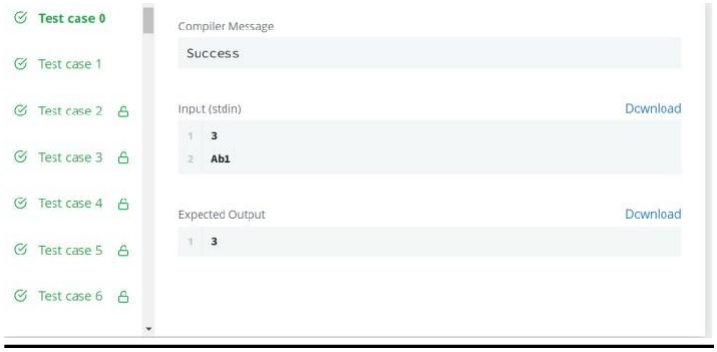
```
max(ans, 6 - n);
printf("%d\n", ans);   return 0;
}
```

Camel Case:

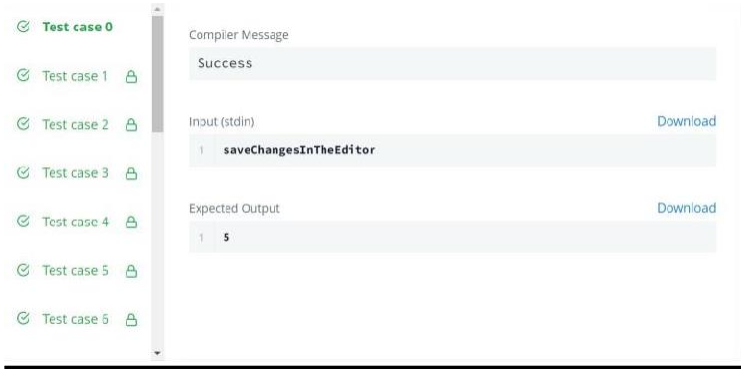
```
#include <bits/stdc++.h>

using namespace std;
int main(){     string s;     cin
>> s;     int t=1;     for (int
i=0;i<s.length();i++)        if
(isupper(s[i]))         t++;
cout<<t<<endl;     return 0; }
```

Result/Output/Writing Summary:

a. Strong Password:



b. Camel Case:



Learning outcomes (What I have learnt):

a. Learnt about strings.
b. Got an overview of the implementation of strings.
c. Get to know about crucial test cases.