**Lab Assignment-3**

**UEC 613 Data Structures and Algorithms**

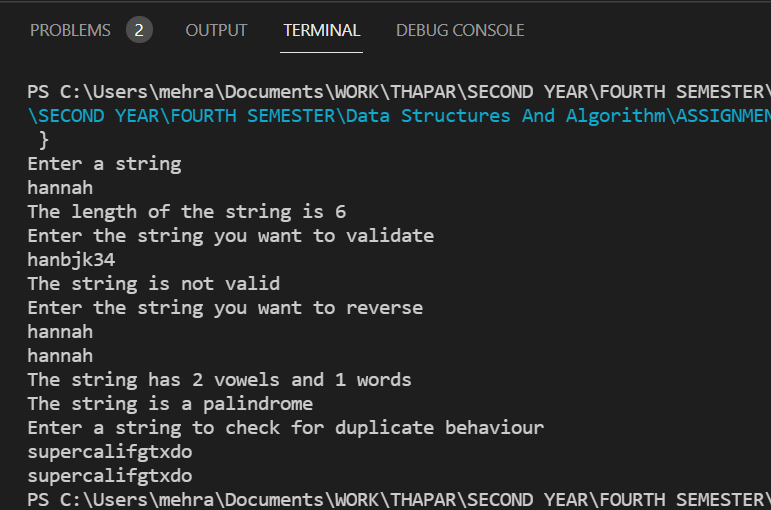
**Note: Use C/C++/JAVA/Python programming language.**

**Q1. Write a function to perform following operations on the string:**

**(Note: You can make single function for all operations/independent function for each problem)**

1. **Finding length of a string**
2. **Converting a string in lowercase**
3. **Counting number of words and vowels in a string**
4. **Validating a string (Note: Valid string does contain only alphabets)**
5. **Reversing a string (Eg. Code 🡪 edoC)**
6. **Checking if a string is palindrome.**
7. **Finding duplicate characters in a string (Note: print the duplicate characters only once, irrespective of the number of times it occurred)**
8. #include<iostream>
9. #include<string>
10. using namespace std;
11. void string\_length(string s)
12. {
13. //int n = sizeof(s)/sizeof(s[0]);
14. // cout<<n<<endl;
15. int count = 0;
16. int i =0;
17. while(s[i]!='\0')
18. {
20. count++;
21. i++;
22. }
23. cout<<"The length of the string is "<<count<<endl;
24. }
25. void string\_lowercase(string s)
26. {
27. //lower case alphabets start from 97 to 122 inclusive
28. //upper case alphabets start from 65 to 90 inclusive
29. }
30. string string\_validate(string s)
31. { int check;
32. for(int i =0;s[i]!='\0';i++)
33. {
34. if(((int(s[i])>=65)&&(int(s[i])<=90))||((int(s[i])>=97)&&(int(s[i])<=122))||(int(s[i]==32)))
35. //ASCII VALUE FOR whiteSPACE IS 64
36. {
37. check =1;
38. }
39. else{
40. check =0;
41. break;
42. }
44. }
45. if(check==1)
46. {
47. return "The string is valid\n";
48. }
49. else
50. return "The string is not valid\n";
52. }
53. string string\_reversal(string s)
54. {
55. string temp;
56. int start =0;
57. int end = s.length()-1;
58. while(start<end)
59. {
60. int temp=s[start];
61. s[start]=s[end];
62. s[end]=temp;
63. start++;
64. end--;
65. }
66. return s;
67. }
68. void string\_count(string s)
69. {  int vowel=0, word=1;
70. for(int i =0;s[i]!='\0';i++)
71. {
72. if(s[i]=='A'||s[i]=='E'||s[i]=='I'||s[i]=='O'||s[i]=='U'||s[i]=='a'||s[i]=='e'||s[i]=='i'||s[i]=='o'||s[i]=='u')
73. {
74. vowel++;
75. }
76. if(int(s[i]) == 32)
77. {
78. word++;
79. }
80. }
81. cout<<"The string has "<<vowel<<" vowels and "<<word<<" words\n";
82. }
83. string string\_palindrome(string s)
84. {
85. if(s==string\_reversal(s))
86. {
87. return "The string is a palindrome\n";
88. }
89. return "The string is not a palindrome\n";
90. }
91. char\* string\_duplicate(char s[],int n)
92. {
93. int index = 0;
95. //transverse through all the characters in the array
96. for(int i =0;i<n;i++)
97. {   int j;
98. //checking to see if s[i] is present before it
99. for(j=0;j<i;j++)
100. {
101. if(s[i]==s[j])
102. {
103. break;
104. }
105. //if the character doesn't repeat and goes through all the iterations of the innermostfor loop,
106. //at the end the value of j is, j=i. Hence the upcoming if statement will m ake sure that it is added to the array.
107. //But if we break out of the loop, j  is never equal to i and the character would not be added to the string.
108. }
110. if(j==i)
111. {
112. s[index++]=s[i];
114. }
116. }
118. return s;
119. }
120. int main()
121. {
122. cout<<"Enter a string \n";
123. string str,str1,str2;
124. getline(cin,str);
125. //String Length
126. string\_length(str);
127. //String validation
128. cout<<"Enter the string you want to validate\n";
129. getline(cin,str1);
130. cout<<string\_validate(str1);
131. //Reversing a string
132. cout<<"Enter the string you want to reverse\n";
133. getline(cin,str2);
134. str2 = string\_reversal(str2);
135. cout<<str2<<endl;
136. //counting the words and vowels
137. string\_count(str2);
138. //checking if string is a palindrome
139. cout<<string\_palindrome(str2);
140. //finding duplicate elements
141. cout<<"Enter a string to check for duplicate behaviour\n";
142. char str3[]={"supercalifragilistisexpialidocious"};
143. int n=sizeof(str3)/sizeof(str3[0]);
144. cout<<string\_duplicate(str3,n);
145. cout<<endl<<str3;
147. return 0;
148. }

**OUTPUT:**



**Q2 Sort the characters of the string entered by user.**

#include<iostream>

using namespace std;

char\* string\_sort(char s[],int n)

{

    for(int i=0;i<n-1;i++)

    {

        for(int j=0;j<n;j++)

        {

            if(int(s[j])>int(s[j+1]))

            {

                char temp;

                temp=s[j];

                s[j]=s[j+1];

                s[j+1] = temp;

            }

        }

    }

    return s;

}

int main()

{

    char str[]={"abhjwcaiwecneicunnaeibercib"};

    int n = sizeof(str)/sizeof(str[0]);

    char\* sorted\_string = string\_sort(str,n);

    cout<<"The sorted string is\n";

    for(int i =0;i<n;i++)

    {

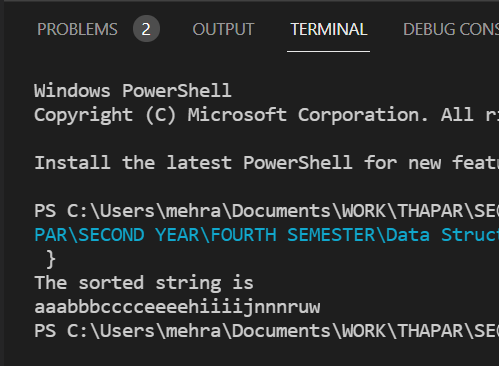
        cout<<\*(sorted\_string+i);

    }

    return 0;

}

**OUTPUT:**

****

**Q3. Implement combination formula nCr using recursion**

#include<iostream>

using namespace std;

int fac(int n)

{

    if(n==0)

        return 1;

    return n\*fac(n-1);

}

float combination(int n,int r)

{

    if(n>=0&&r>=0)

    {   cout<<"The value of all possible combinations is\n";

        float answer = fac(n)/((fac(n-r))\*(fac(r))) ;

        return answer;

    }

    else

    {

        cout<<"Invalid entries";

        return 0;

    }

}

int main()

{   int n,r;

    cout<<"Enter the value of n and r\n";

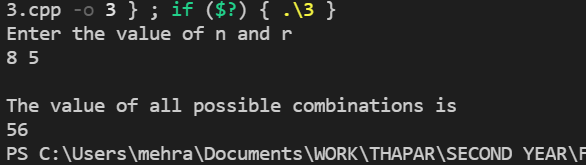
    cin>>n>>r;

    cout<<endl<<combination(n,r);

    return 0;

}

**OUTPUT:**

****

**Q4. Implement the Fibonacci series using recursion**

#include<iostream>

using namespace std;

int fib(int n)

{

    if(n<=0)

    {

        cout<<"Invalid entry\n";

        return 0;

    }

    if(n==1)

        return 0;

    else if(n==2)

        return 1;

    else

        return fib(n-1)+fib(n-2);

}

int main()

{   cout<<"Enter number of terms for which you want the series\n";

int num;

    cin>>num;

    for(int i=1;i<=num;i++)

    {

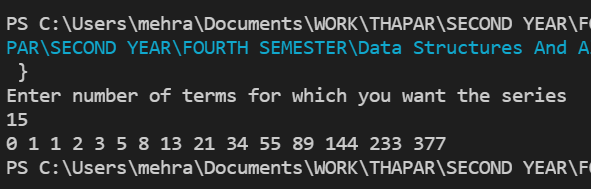
      cout<<fib(i)<<" ";

    }

    return 0;

}

**OUTPUT:**

****

**Q5 Write a program for finding the factorial of a number recursively.**

#include<iostream>

using namespace std;

int fac(int n)

{

    if(n==0)

        return 1;

    return n\*fac(n-1);

}

int main()

{   int x;

    cout<<"Enter the number whose factorial you want\n";

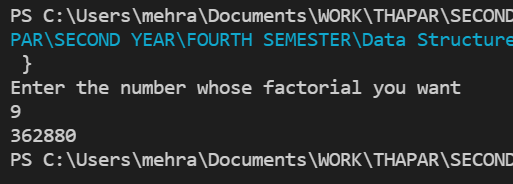
    cin>>x;

    cout<<fac(x)<<endl;

    return 0;

}

**OUTPUT:**

****