Object Oriented Programming Lab
Lab Manual – Cstrings

Exercise 1:

i. Run following piece of code and paste the output in space given below:

```
#include<iostream>
using namespace std;

void main()
{
	//Exercise 1
	char charArray[] = { 'H', 'e', 'l', 'l', 'o', ' ', 'W', 'o', 'r', 'l', 'd' };
	cout << charArray << endl;
}
```

Output:

Why is it printing garbage at the end?

ii. Run following piece of code and paste the output in space given below:

```
	char myStr1[] = { 'H', 'e', 'l', 'l', 'o', ' ', 'W', 'o', 'r', 'l', 'd', '\0' };
	cout << myStr1 << endl;
```

Output:

What is the difference between two arrays declared above? Which one is a c-string?

iii. **[StringLenght]** Write a function **int StringLenght(char*)** that takes a cstring and returns it length. Remember that length of c-string does not include null character. Length of myStr1 should print 11.

iv. Comment the declaration of myStr1 and declare it as the code segment given below. Run and test your program now.

```
//char myStr1[] = { 'H', 'e', 'l', 'l', 'o', ' ', 'W', 'o', 'r', 'l', 'd', '\0' };
char myStr1[] = "Hello World";
```

v. Comment the declaration of myStr1 and declare it as the code segment given below. Run and test your program now.

```
//char myStr1[] = { 'H', 'e', 'l', 'l', 'o', ' ', 'W', 'o', 'r', 'l', 'd', '\0' };
//char myStr1[] = "Hello World";
char myStr1[20] = "Hello World";
```

vi. Run the code given below and check its behavior:

```
        char myStr1[5] = "Helo";
        cout << "Initial String:\t" << myStr1 << endl;

        int len = 0;
        for (; myStr1[len] != '\0'; len++);
        cout << "Lenght of myStr1:\t" << len << endl << endl << endl;

        cout << "Enter another string of size 4 :";
        cin >> myStr1;


        cout << "String entered by the user is:" << myStr1 << endl;


        for (len = 0 ; myStr1[len] != '\0'; len++);
        cout << "Lenght of myStr1:\t" << len << endl << endl << endl;


        cout << "Enter another string of size 5 or Greater :";
        cin >> myStr1;

        cout << "String entered by the user is:" << myStr1 << endl;


        for (len = 0; myStr1[len] != '\0'; len++);
        cout << "Lenght of myStr1:\t" << len << endl << endl << endl;
        cout << "Program is going to terminate.\n";
```

Why did the program crash?

In the exercises given below, dynamically allocate char array of size 80 to save data. Make sure your program does not cause any memory leakage.

Exercise 2(a) (Basic C-String): **[GetCharacterCount]** Write a function **int GetCharacterCount(char* myString, char c)** that takes a character ***ch*** and a c-string ***myString*** and returns total number of occurrences of *ch* in *myString*.

Sample Output:

myString: Pakistan
ch: a
Total No of Occurances: 2

Exercise 2(b) (Basic C-String): Update the program written in above Exercise to accept a sentence in myString and test it. You need to replace **"`cin>>myString;`"** with **"`cin.getline(myString,50);`"**

Sample Output:

myString: I am Pakistani
ch: a
Total No of Occurances: 3

Exercise 3 (C-String): **Char** ReadStudentsListFromFile()**
Write a function that reads "StudentsList.txt" (data for file is given below) and saves these names in memory and returns char** pointing to list of students. Make sure you do not consume any extra single byte.

Data.txt

```
44      //Total No of Students
Afzaal Amjad
Rashid Mahmood
Abubaker Saleem
Yousaf Khan
Ahsan Zafar
Husnain Rafiq
Afraz Kamal
Taha Tahir
Aqib Javed
Mansoor Hassan
Amber Warsi
M Usman
Hamza Yaseen
Lubaina Zubair
```

Muhammad Ubaidullah
Rabia Noor
Hassan khan
Moin Ali
Ali Hussnain
Rahat mubeen
Fatima Tahira
Ali Awan
Muhammad Ali
Aena Maryam
Hunain Haider
Bilal Ahmad
Saif Ul Islam
Adil Ashraf
Hasan Zahid
Umer Naseer
Hamza Majeed
Saud Ul Hassan
Faiq Rizwan
Hamza Rashid
Mohsin Ali
Usman Ahmed
Fatima Khan
Asma Maqbool
Osama Khan
Farwa Abbas
Tehreem Aftab
Waqas Wasi
Zain Ali
Aalia Nazi

Exercise 4: **void PrintAllNames(char** studentsList, int& size)**

Print all the names saved on heap on the screen.

Exercise 5: **int FirstIndexOfSubString(char* myStr, char* strToFind)**

Write a function that takes a string myStr and a substring strToFind and returns index of string where it finds first occurrence of substring. If it does not find the substring in string, it will return -1.

Sample Output:

```
String: "I am a Pakistani so I support Pakistani Cricket team in Pak-India matches."
Substring: Pak
Index Returned: 7

Substring: Pakii
Index Returned: -1
```

Exercise 6: char** FindNameFromStudentsList (char** list, char* SearchString,…)

Write a function that takes students list and searches string (entered by user) and returns list of pointers pointing to students names having search string. For example
Note: Use function **FirstIndexOfSubString** to search string.

```
Search String: Ali

Result Found:
    1- Moin Ali
    2- Ali Hussnain
    3- Ali Awan
    4- Muhammad Ali
    5- Mohsin Ali
    6- Zain Ali
    7- Aalia Nazi

Search String: Shams

No Result Found
```

Note: You do not need to copy whole strings (names). Just make a copy of their pointers in result array. Call PrintAllNames function to print this result.

Exercise 7: void RemoveStudents(//Decide parameters Yourself. Return type will remain void)

Write a function that takes students list and deletes all the students having searchString in it. For example, if search string is Ali, your updated students list will have only 37 (44-7 = 37) students names in it. Call PrintAllNames again and verify the result. (Do not consume extra memory. There shouldn't be any memory leakage or dangling pointer in your code).

Practice Problems

char* FindAndReplaceString(char* str, char* toFind, char* toReplace)

Write a function that takes a string, replaces all occurances of ***toFind*** with ***toReplace*** *in* a newly created string.

Sample Output:

String: "I am Pakistani so I support Pakistani Cricket team in Pak-India matches."
toFind: Pak
toReplace: Afghan
New String: "I am Afghanistani so I support Afghanistani Cricket team in Afghan-India matches."

String: "I am Pakistani so I support Pakistani Cricket team in Pak-India matches."
toFind: Pakii
toReplace: Afghan
New String: "I am Pakistani so I support Pakistani Cricket team in Pak-India matches." (as Pakii does not exist in the string).

Return original string if find/replace is not possible. Otherwise create new string to make result string.

Q: Write a program that takes a c-string ***myStr*** and two characters ***charToFind*** and ***charToReplace*** from user and replaces all the occurrences of ***charToFind*** with ***charToReplace*** in ***myStr***. Your program should create a space of 50 characters on heap in order to save *myStr*.

Sample output:

InputString: **dd**s**d**fhgrts**d**fhjghjks**dd**
CharToFind: d
CharToReplace: $

ModifiedString: **$$**s**$**fhgrts**$**fhjghjks**$$**

Q: Write a program that takes a character ***ch*** and a CString ***myStr*** from user and removes all the occurrences of ***ch*** from ***myStr***.

Sample Output:

myStr: cabccdefcfdcxyzcc
ch: 'c'

Modified String: abdeffd xyz

TrimStart(char* str)

Write a function that takes a string and removes all the space in start of the string.

Sample Output:

Before TrimStart
str: " Hello How are you?"

After TrimStart

str: “Hello How are you?”

TrimEnd(char* str)

Write a function that takes a string and removes all the space at the end of the string.

Sample Output:

Before Trim End str: **“**I love programmin g. **”** After Trim End str: “I love programmin g.”

Hint: Traverse the array from end