CS19003: Programming & Data Structure Lab, Section 15
Autumn 2020-21
Lab Test 2, February 18, 2021

PART - 2

Time: 10-40 am to 11-50 am

Instructions (Read carefully)

1. **Your C programs must first contain comments with your name, roll no., and Labtest no. (=2), as done in class.**
2. **Name your C file LT2_2_<your roll no>.c for (For example LT2_2_20ME30006.c)**
3. **Submit through the links (Intermediate and Final) for PART 2 in moodle. MAKE SURE TO VERIFY YOUR SUBMISSION after final submission.**
4. **(Very Important) There are two sets of problems. Problem set 1 is for 100% marks and covers the entire syllabus of what is covered in the lab. Problem set 2 is for 60% marks and covers only up to arrays and functions. You are free to choose which set you will attempt. However, there cannot be any mixing between the sets, you can answer only one of the sets. Submit only one C file for one set.**
5. **All other instructions regarding lab test sent earlier in the slides earlier to be followed strictly.**

PROBLEM SET 1 (FOR 100% (= 10) MARKS)

Consider a string, with each character being one of the digits 0 to 9, terminated by the null character ('\0'). We will call such a string a **Numeric String**.

Write a **recursive** C function

void CountSubstr (char *N, char *p, …..)

that takes as parameter a numeric string **N** and a character **p**, and returns two things: (i) the number of substrings with only the character **p** in N, and the length of the maximum such substring. A "*substring with only the character **p***" is defined as a *maximal consecutive sequence* of the character **p.** For example, if you have the string "12003210450000670", and **p** = '0', then the number of substrings with only **p** is 4 ('00', '0', '0000', '0', also shown underlined in the string) and the maximum length is 4 (that of '0000'). You should add appropriate parameters to the function to return these two things. You cannot change the return value of the function from **void**. You cannot use any string library function. You cannot use any additional arrays.

Write a C function (recursive or non-recursive, your choice)

char *RemoveSubstr(char *N, char *p)

that will take a numeric string **N** and a character **p**, and return the pointer to a string that replaces each substring with only **p** in **N** with a single colon (‘:’) character. For example, for the **N** and **p** above, it should return a pointer to the string “12:321:45:67: “. You cannot use any string library function. The string **N** should remain unchanged, the modified string should be stored in a new string.

Write a **main( )** function that does the following in this order:

1. Read a string **N** from the keyboard (assume it will have maximum 99 characters).
2. Read a character **p** from the keyboard.
3. Call the function **CountSubstr( )** to find and print the number and the maximum length of substrings with only **p** in **N** .
4. Call the function **RemoveSubstr( )** to replace any such substring with a single colon character. Print both the original string **N** and this modified string **after returning from the function**.

Example:

N = “@abc@@@@de@f@@@@ghhh”
p = ‘@’

Your program should output (exact message may vary):

Number of substrings = 4, maximum length = 4
Original string is **@abc@@@@de@f@@@@ghhh**
Modified String is :abc:de:f:ghhh

PROBLEM SET 2 (FOR 60% (= 6) MARKS)

Consider two null–terminated strings **S1** and **S2**.

Write a **recursive** C function

int IsReversed(char *S1, char *S2)

that takes as parameters two null-terminated strings **S1** and **S2**. The function returns 1 if **S2** is exactly the reverse of **S1**, 0 otherwise. For example, if **S1** = “abcd” and **S2** = “dcba”, the function should return 1. But if **S1** = “abcd” and **S2** = “dabc”, the function should return 0. Obviously, the function should return 0 if the two strings have different lengths.

Write a C function (non-recursive)

int CheckStr(char *S1, char *S2)

that returns the number of digit (0 to 9) characters **x** in **S1** such that there exists at least one digit character **y** in **S2** for which the numeric value of **x** is **strictly less** than the numeric value of **y**. For example, if **S1** = "ab3df79" and **S2** = "df6d28", then the function should return 2, as of all the digit characters **x** in **S1** (3, 7, and 9), **x**=3 has a digit character **y**=6 (also 8) in **S2** such that **x** < **y**, and **x** = 7 has a digit character **y**=9 in **S2** such that **x** < **y**, so there are two such characters and the count is therefore 2. But note that **x**=9 in **S1** is not less than any digit character in **S2**, so that is not counted.

Write a **main( )** function that does the following:

1. Read in two strings **S1** and **S2** (assume length < 100).
2. Print the two strings read.
3. Call the function **IsReversed( )** to check if **S1** is the reverse of **S2** or not, and print a suitable message.
4. Call the function **CheckStr( )** to find and print the number of digit (0 to 9) characters **x** in **S1** such that there exists a digit character (0 to 9) **y** in **S2** for which the numeric value of **x** is strictly less than the numeric value of **y.** (The actual digits are not needed)

Example:

S1 = "ab3df79"
S2 = "97fd3ba"

Your program should output (exact message may vary):

The strings are ab3df79 and 97fd3ba
The string ab3df79 is the reverse of 97fd3ba
The number of digit characters is 2

Another Example:

S1 = "a93df76"
S2 = "97fd3ba"

Your program should output (exact message may vary):

The strings are **a93df76** ***and 97fd3ba***
The string ab3df79 is not the reverse of 97fd3ba
The number of digit characters is 3