Department of Information Systems and Technologies

CTIS 152 – Data Structures and Algorithms

Summer 2020 - 2021

Lab Guide #8 – Week 3 - 2

OBJECTIVE : String Operations	
Instructor : Okyay SAY **Assistant :** Ruşen ASAN	

<string.h> library functions are:

- `int strlen(const char *str)`
- `char *strcpy( char *dest, const char *src)`
- `char *strncpy(char *str1, const char *str2, size_t n)`
- `char *strcat( char *dest, const char *src)`
- `int strcmp(const char *str1, const char *str2)`
- `int strncmp(const char *strl, const char *str2, size_t n)`

Q1. **a)** Write a C program that will input a sentence, find and display the position of the given **string** in the sentence. If the searched string is NOT found, display an appropriate message.

Write the following function;

- **findFirst:** takes a sentence and a string to be searched as input parameters, finds and returns the index of the **first occurrence of the given string** in the sentence. If the sentence does NOT contain the searched string the function should return **-1**.

Example Run #1:

```
Enter a sentence: This watch is expensive
Enter a string: is

The first occurence of the str <is> is 2
```

Example Run #2:

```
Enter a sentence: It sounds good
Enter a string: are

The sentence does NOT contain the string <are>
```

Project Name: LG8_Q1a
File Name: Q1a.cpp

b) Modify the program **Q1a.cpp**, so that the program replaces the first occurence of a given string in the sentence with the specified string. And then displays the new form of the sentence.

Write the following function;

- **replaceFirstOccur:** takes a sentence, a search string, a string to be replaced and the index of the first occurence of the search string as parameters, replaces the first occurence of the given string with the specified string.

Example Run #1:

```
Enter a sentence: Google and Facebook collect information about us and then sell that data to
advertisers
Enter a String to search: and
Enter a String to replace: &

New form of the sentence:
Google & Facebook collect information about us and then sell that data to advertisers
```

Example Run #2:

```
Enter a sentence: Create a different email address for every service you use
Enter a String to search: are

The sentence does NOT contain the string <are>
```

Project Name: LG8_Q1b
File Name: Q1b.cpp

Q2. **a)** Write a C program that inputs a sentence, finds and **deletes** the first occurrence of the searched string, and displays the new sentence. If the searched string is NOT found, displays an appropriate message.

Write the following function;

- **deleteFirst:** takes a sentence, a string and the starting index of the given string in the sentence as parameters. The function deletes the given string in the sentence using the **findFirst** function in **Q1.cpp.**

Example Run #1:

```
Enter a sentence: A proper cup of coffee from a proper copper coffee pot
Enter a string: per

The new form of the sentence after deletion: A pro cup of coffee from a proper copper coffee pot
```

Example Run #2:

```
Enter a sentence: I scream, you scream, we all scream for ice cream

Enter a string: cream
The new form of the sentence after deletion: I s, you scream, we all scream for ice cream
```

Project Name: LG8_Q2a
File Name: Q2a.cpp

b) Modify the **Q2a.cpp**, so the program inputs a sentence, finds and **deletes all occurences** of the searched string, and displays the new sentence. If the searched string is NOT found, display an appropriate message.

Write the following function;

- **deleteAllOccur:** takes a string and a sentence, deletes all occurrences of the given string if it exists.
  Hint: Use **findFirst** and **deleteFirst** functions.

Example Run:

```
Enter a sentence: Peter Piper picked a peck of pickled peppers
Enter a string: pick

Final format of the str : Peter Piper ed a peck of led peppers
```

Project Name: LG8_Q2b
File Name: Q2b.cpp

Q3. Write a C program that inputs a sentence, finds and displays the indexes of all occurences of the given **string** in the sentence. If the searched string is NOT found, display an appropriate message.

Write the following function;

- **findAllOccur:** takes a sentence and a string to be searched as input parameters, finds and returns the indexes of **all occurrences of the given string** in the sentence.

Example Run #1:

```
Enter a sentence: How much wood would a woodchuck chuck if a woodchuck could chuck wood
Enter a string: chuck

All occurences of the str <chuck>:
 26 32 47 59
```

Example Run #2:

```
Enter a sentence: I scream, you scream, we all scream for ice cream
Enter a string: are

The sentence does NOT contain the string <are>
```

Project Name: LG8_Q3
File Name: Q3.cpp

Q4. Gunes company wants to create email addresses for its workers. Therefore, they give their employees' name to the mail server company to create the email addresses. The mail format should be formed with the first letter of the name and surname and "@gunes.com.tr". For example; **GIZEM CINAR** will be **gcinar@gunes.com.tr.**

Write a C program that gets the names and surnames of the employees from the file named **worker.txt**. Then creates the email addresses as explained above and writes the email addresses into the file **mails.txt**.

Write the following function;

- **`convertLower:`** gets an uppercase string and returns same string with lowercase letters.
  Notice: You can use tolower function including this function.

  Hint: All the data in the file are uppercase letters.
  YOU MUST READ THE NAME AND SURNAME OF A PERSON INTO A STRING FROM THE FILE.

worker.txt

```
DOGAN ACAR
AHMET AYDIN
ALI BULUT
AYSE CAGLAR
SAMET DEMIR
SEDA KARAKAYA
VOLKAN OZ
MEHMET SENTURK
METIN TURKOGLU
```

mails.txt

```
dacar@gunes.com.tr
aaydin@gunes.com.tr
abulut@gunes.com.tr
acaglar@gunes.com.tr
sdemir@gunes.com.tr
skarakaya@gunes.com.tr
voz@gunes.com.tr
msenturk@gunes.com.tr
mturkoglu@gunes.com.tr
```

Project Name: LG8_Q4
File Name: Q4.cpp

ADDITIONAL QUESTIONS

AQ1. Write a C program that will read a sentence and a key string from the user, finds the LAST OCCURENCE of the given key string and displays the sentence back until the key string's last occurence.

Write the following function;

- **`findLastOcc:`** takes a sentence and a string to be searched as input parameters, finds and returns the index of the **last occurrence of the given string** in the sentence.

Example Run #1:

```
Enter a sentence: do not go gentle into that gentle good night
Enter a key string: gentle

Result: do not go gentle into that
```

Example Run #2:

```
Enter a sentence: do not go gentle into that good night
Enter a key string: do

Result: That's an empty string, sorry..
```

Example Run #3:

```
Enter a sentence: rage against the dying of the light rage
Enter a key string: rage

Result: rage against the dying of the light
```

Project Name: LG8_AQ1
File Name: AQ1.cpp

AQ2. Write a C program that **deletes all occurrences of a given WORD** in a sentence and display the new sentence. if the word doesn't exist display an appropriate message for the user.

Write the following functions;

- **`findFirstOcc:`** takes a string and a sentence, returns the necessary information.
- **`delAllOccur:`** takes a string and a sentence, deletes all occurrences of the given WORD if it exists.

Example Run #1:

```
Enter a sentence: How much wood would a woodchuck chuck if a woodchuck could chuck wood
Enter a word: wood

Final format of the sentence : How much  would a woodchuck chuck if a woodchuck could chuck
```

Example Run #2:

```
Enter a sentence: She sells seashells by the seashore
Enter a word: sea

The sentence does NOT contain the WORD <sea>..
```

Project Name: LG8_AQ2
File Name: AQ2.cpp