Department of Information Systems and Technologies

CTIS 152 – Data Structures and Algorithms

Spring 2024 - 2025

Lab Guide #8 – Week 5-1

OBJECTIVE : String Library Functions. Exercises

Instructors : Serpil TIN
Assistants : Berk ÖNDER, Hatice Zehra YILMAZ

<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)`
- `char * strncat ( char * destination, const char * source, size_t num );`
- `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**.

Project Name: LG8_Q1a
File Name: Q1a.cpp

Example Run#1:
```
Enter a sentence: This is so good

Enter a string: is
The first occurence of the str <is> is 2
```

Example Run#2:
```
Enter a sentence: We remember Prof. Dogramaci with deep respect, profound gratitude and great affection

Enter a string: are
The sentence does NOT contain the string <are>
```

b) Modify the program **Q1a.cpp**, so that the program replaces the **first** occurrence of a given string in the sentence with the specified string. Then the new form of the sentence is displayed.

Write the following function;

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

Project Name: LG8_Q1b
File Name: Q1b.cpp

Example Run#1:
```
Enter a sentence: What are we going to do today ?
Enter a String to search: is

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

Example Run#2:
```
Enter a sentence: Sustainability is the balance between the environment, equity, and economy
Enter a String to search: the
Enter a String to replace: XXXXX

New form of the sentence:
Sustainability is XXXXX balance between the environment, equity, and economy
```

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, display 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.**

Project Name: LG8_Q2a
File Name: Q2a.cpp

Example Run#1:
```
Enter a sentence: I thought a thoughtful thief threw three thick things through the theatre

Enter a string: ate
The sentence does NOT contain the string <I thought a thoughtful thief threw three thick things through the
theatre>
```

Example Run#2:
```
Enter a sentence: I thought a thoughtful thief threw three thick things through the theatre

Enter a string: thought
The new form of the sentence after deletion: I  a thoughtful thief threw three thick things through the
theatre
```

b) Modify the **Q2a.cpp**, so the program inputs a sentence, finds and **deletes all occurrences 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, and deletes all occurrences of the given string if it exists.
  Hint: Use the **findFirst** and **deleteFirst** functions.

Project Name: LG8_Q2b
File Name: Q2b.cpp

Example Run#1:
```
Enter a sentence: She sells seashells by the seashore
Enter a string: sea
Final format of the str : She sells shells by the shore
```

Example Run#2:
```
Enter a sentence: She sells seashells by the seashore
Enter a string: sold
Word is not found in the string..
```

Q3. Write a C program that inputs a sentence, finds and displays **the indexes of all occurrences 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.

Project Name: LG8_Q3
File Name: Q3.cpp

Example Run#1:
```
Enter a sentence: A curious courier quickly carried quirky queries to the quiet quarry

Enter a string: qui
All occurences of the str <qui>: 18 34 56
```

Example Run#2:
```
Enter a sentence: If two witches were watching two watches, which witch would watch which watch

Enter a string: watc
All occurences of the str <watc>: 20 33 60 72
```

Example Run#3:
```
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>
```

ADDITIONAL QUESTIONS

AQ1.

Write a C program that will read a sentence and a key string from the user, find the **last occurrence of the given key string**, and display the sentence until the key string's last occurrence.

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.

Project Name: LG8_AQ1
File Name: AQ1.cpp

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
```

AQ2.

Write a C program that **deletes all occurrences of a given WORD** in a sentence and displays 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, and returns the necessary information.
- **`delAllOccur:`** takes a string and a sentence, and deletes all occurrences of the given WORD if it exists.

Project Name: LG8_AQ2
File Name: AQ2.cpp

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>.
```