MIDDLE EAST TECHNICAL UNIVERSITY, NORTHERN CYPRUS CAMPUS
CNG140 C Programming – Worksheet 20

Purpose:
In this work sheet, you will get hands-on experience with strings (arrays of characters).

Tasks:

a. Write a C program that includes a function to count the number of sentences entered. Assume a sentence ends in either a period, question mark or exclamation mark (Hint: You won't be able to use scanf() to input sentences, because scanf() ends input on receipt of the first space character). You **can** use string.h library for this exercise.

   A sample run would be as follows:

```
Please enter a sentence: How many dogs do you have? 3 dogs!
You entered 2 sentences
```

b. Modify the function written in part (b) to count the number of words as well as the number of sentences. Assume that each word is separated with a space. The function needs to return the average words per sentence. You **can** use string.h library for this exercise.

   A sample run would be as follows:

```
Please enter a sentence: How many dogs do you have? 3 dogs!
You entered 2 sentences
Average words per sentence is 4
```

   In this sample run, please note that we added all words in all sentences (6 words in sentence 1 and 2 words in sentence 2) and then divided by the total number of sentences (2 sentences) to calculate the average (8/2=4).

c. Write a program that reads a line of C statement and store it in a string, and removes the comments given as /*comment*/ from that line of statement. You need to write a function called remove-comment() that takes a string and removes the comment. You need to only consider comments that are in this format: /**/. You **can** use string.h library for this exercise.

   A sample run would be as follows:

```
Please enter a C statement: int i; /* comment */
Your statement without comments: int i;
```

   Another sample run would be as follows:

```
Please enter a C statement: int i; /* comment 1 */ i=90; /* comment
2*/

Your statement without comments: int i; i=90;
```

d. Write a C program that finds the maximum occuring character in a string.

   A sample run would be as follows:

```
Enter your string: Online Courses are starting!
Maximum occuring character in the string is s!
```

e. Write a program that takes a list of names and surnames from the user and then sorts them alphabetically. You **can** use string.h library for this exercise. In your program you should have the following functions:
   a. Populate_list(): this function will read names from the user. The user can enter names as long as they do not type exit keyword. When they type exit, your program should stop reading names.
   b. Sort_names(): This function will get the list of names from the main and then sorts them alphabetically().
   c. Display_names(): This function will display the names in alphabetical order.

A sample run would be as follows:

```
Enter the list of names:
Yeliz Yesilada
Idil Candan
Hakan Yekta Yatbaz
Erfan Khalaji
Meryem Erbilek
Naziha Khalil
Exit

The names are sorted as follows:
Erfan Khalaji
Hakan Yekta Yatbaz
Idil Candan
Meryem Erbilek
Naziha Khalil
Yeliz Yesilada
```

f. Write a program that takes a list of course names belonging to CNG and EEE departments from the user and then list the frequency of each course based on department code. You can assume that always the department code and course code is separated with a space. You **can** use string.h library for this exercise. In your program you should have the following functions:
   a. `populate_list()`: This function will read course names from the user. The user can enter names as long as they do not type exit keyword. When they type exit, your program should stop reading course names. If the string that user enters does not have a space then you should give them a warning that the course name should be in this format: department code course code (e.g., CNG 111). If the string also does not start with CNG and EEE, then your program should also give an error. No other mistakes need to be checked.
   b. `count_courses()`: This function will get the list of course names from the main and then counts the frequency of CNG and EEE courses.
   c. `display_frequency()`: This function will take the frequency of CNG and EEE courses and will display the frequency of courses for each department.

A sample run would be as follows:

```
Enter the list of courses:
CNG 111
MECH 224
Wrong course name! Course name should start with EEE or CNG!
CNG 140
CNG 213
EEE201
Wrong course name! Course name format is departmentcode coursecode
EEE 201
Exit
```

```
Number of courses per department:
CNG  : 3
EEE  : 1
```

Recommended Reading: Chapter 9 (pp. 441-481)
Recommended Exercises: Programming Exercises given in the following pages.