Fundamental Structures Lab 08

(Due Date: 03/28/2016 4:30 pm)

Program 1 - 10 pts

Develop a terminal application that accomplish the following requirements:

1. Read a string from console.
2. Make a function to check if all the characters of the string are unique (the string has no repeated characters) using the following algorithm:

   Loop through the characters of the string and compare each character with all the remaining characters. If the character is repeated, stop the loop and return false. Otherwise, return true.
3. The program should show a message indicating if the characters are unique or not.
4. Write a sentence explaining the complexity of the program using Big-O notation. Also, show this message for console.

Program 2 - 15 pts

Develop a terminal application that accomplish the following requirements:

1. Read a string from console.
2. Make a function to check if all the characters of the string are unique (the string has no repeated characters) using the following algorithm:

   Given that all the characters are represented using the ascii code, we can create a boolean array of 256 values where each value represents an ascii character. Initialize the array in false.

   Then, loop through the characters of the string and check: if the array at the character's ascii position is false, change it to true; if not, return false. Otherwise, return true.
3. The program should show a message indicating if the characters are unique or not.
4. Write a sentence explaining the complexity of the program using Big-O notation. Also, show this message for console.

Program 3 - Comparison - 25 pts

- Compare program 1 and program 2 using preset strings with unique characters (Do not ask the strings to the users).
- Use strings of size 5, 10, 20, 50, 80, 100, 150, 200, and 256.
- Compute the time that program 1 and program 2 take to process each one of the strings.
- Show for console the time that each program takes in a table like this:

n	program 1	program 2
5	time	time
10	time	time
20	time	time
50	time	time
80	time	time
100	time	time
150	time	time
200	time	time
256	time	time

To compute running times, retrieve system time before and after execution and compute the difference. Do do not use boost or any other non standard libraries. You can use the function clock() as shown in this example:

```
#include <time.h>

int main()
{
    clock_t start = clock();
    //your code here
    clock_t end = clock();
    printf("Elapsed: %f seconds\n", (double)(start - end) / CLOCKS_PER_SEC);
    return 0;
}
```

This example and a complete reference the function time() can be found at `http://en.cppreference.com/w/cpp/chrono/c/clock`:

Submission Example

Extraction of LastnameFirstnameLab01.zip

```
/Documents
    LastnameFirstnameLab01.zip
    /LastnameFirstnameLab01
        /prog1
            prog1.cpp
        /prog2
            prog2.cpp
            A2Output.txt
       /Bonus
           bonus.cpp
```

Important reminder: Minimum penalty of plagiarism is failing (F) grade in the course.