ITSC 2181 – Introduction to Computer Systems
Spring 2024
Module 03 – Unit 2: Lab

Objectives

- Practice how to write small C programs, including identifying and correcting `C` compiler error messages and warnings.
- Practice the use of **loops** in C to process data.
- Practice the use of **functions** in C to process data.
- Practice the use of **arrays** in C to process data.

General Instructions

- Please do not write your email or user ID anywhere in the program. To identify your code, you may use your UNC Charlotte 800#

- In this lab, you will write a few short C programs. Each program needs to be in its own source code file, i.e., a file with the *`.c`* extension.

- **You need to test your code thoroughly before submitting it**.

- **To earn any credit, a program has to compile.** *You may comment-out lines that have errors to obtain partial credit for work done.*

- Programs need to compile cleanly to receive full credit, i.e., they do not produce any errors or warnings.

- It is essential that you **do your own work.**

    - **Do not use any external resources** (Internet, AI, friends, etc.). All cases of cheating will be taken very seriously.

    - **If you need help, please ask the instructor, TA/IA or CCI Tutoring center.**

Copyright © 2019-2024, Julio C. Bahamón, UNC Charlotte.
All Rights Reserved.

Program 1 (50 points)

1. Write a function named `capitalize` that capitalizes all letters (i.e., alphabetic characters) in its argument.

2. The argument will be a null-terminated string containing arbitrary characters, not just letters.

3. You **must use array scripting to access each character** in the string individually.

4. To check if a character is alphabetic, you can use the `isalpha` function from the C Standard Library, see: https://en.wikibooks.org/wiki/C_Programming/ctype.h/isalpha

5. To convert a character to uppercase, you can use the `toupper` function from the C Standard Library, see: https://en.wikibooks.org/wiki/C_Programming/ctype.h/toupper

6. You cannot use any other C Standard Library functions, except for `printf` and `scanf`.

Use the following code to test your function:

```
char the_str[] = "test";

capitalize(the_str);
printf("%s\n", the_str);

char the_str2[] = "This IS a tesT!";

capitalize(the_str2);
printf("%s\n", the_str2);
```

Sample output is shown below:

```
TEST
THIS IS A TEST!
```

Implement a program named `strings_practice.c` that uses the function you wrote. You may use the code shown above. However, we recommend testing with other strings as well.

Copyright © 2019-2024, Julio C. Bahamón, UNC Charlotte.
All Rights Reserved.

Program 2 (50 points)

Write a program, named **weather.c**, that does the following:

1. Reads 10 temperatures from the console (standard input). Each value represents the high temp for a given day in a 10-day period.
2. Keeps track of the highest (max) temperature entered by the user.
3. Displays the day when the highest (max) temperature was recorded. If two or more days tie for the highest temperature, the program will report the last day with the highest temperature.
4. You need to use a loop in your implementation.
5. You **do** **not** need to use arrays for this program.

Hint: To save time, create a text file with test data and use input redirection when testing your code.

Two sample runs are provided below:

```
You will be asked to enter the daily high temperature for 10
consecutive days.
Enter a high temperature: 91
Enter a high temperature: 93
Enter a high temperature: 92
Enter a high temperature: 97
Enter a high temperature: 90
Enter a high temperature: 88
Enter a high temperature: 95
Enter a high temperature: 97
Enter a high temperature: 96
Enter a high temperature: 96

The highest recorded temperature in the 10-day period was: 97
degrees
Last recorded on: Day 8
```

```
You will be asked to enter the daily high temperature for 10
consecutive days.
Enter a high temperature: 91
Enter a high temperature: 93
Enter a high temperature: 92
Enter a high temperature: 97
Enter a high temperature: 90
Enter a high temperature: 100
Enter a high temperature: 95
```

Copyright © 2019-2024, Julio C. Bahamón, UNC Charlotte.
All Rights Reserved.

```
Enter a high temperature: 97
Enter a high temperature: 96
Enter a high temperature: 96

The highest recorded temperature in the 10-day period was: 100
degrees
Last recorded on: Day 6
```

Submission Instructions

1. Create a folder (directory) on your computer.

2. Name the folder **`ITSC_2181_M03_U2_Lab_`*`student-id`***
   Replace *student-id* with your UNCC student ID (800#), e.g. *8001231234*

3. Download and copy your program (source code) files into this folder. You should keep the files for every lab in a separate folder (directory) from your other course materials.

4. Upload your C files (individually) to *Canvas* and submit the assignment.

Grading Rubric

- This lab is worth a total of **100 points**.

- **Do your own work. Do not use any external resources** (Internet, friends, etc.). All cases of cheating will be taken very seriously. **If you need help, please ask the instructor, TA/IA or CCI Tutoring center.**

- Programs need to compile to earn any credit. *You may comment-out lines that have errors to obtain partial credit for work done.*

- Your work will be graded on three (3) major components: Logic and flow of program, output, and formatting/organization. Refer to the following table for details.

Logic and Flow of Program - 60%	
Fully Correct and code compiles without errors.	Full Credit
Minor Errors and code compiles without errors OR some required functionality is missing.	75% Credit

Copyright © 2019-2024, Julio C. Bahamón, UNC Charlotte.
All Rights Reserved.

Major Errors and/or code compiles with warnings OR significant portions of the required functionality are missing.	50% Credit
Completely incorrect/missing/does not compile.	No Credit
Output - 30%	
Fully Correct and matches formatting and layout of sample output provided.	Full Credit
Minor Errors.	75% Credit
Major Errors.	50% Credit
Completely incorrect output.	No credit
Formatting/Organization of Code - 10%	
Code is clear, easy to read and formatter according to the guidelines. Whitespace has been used appropriately, including indentation and blank lines. Comments are used when needed.	Full Credit
Needs minor improvement.	75% Credit
Needs major improvement.	50% Credit
No formatting/organization at all.	No Credit

Additional Deductions

- Code that produces warnings: -10% per program
- Incorrectly named files: -2 points per file
- *Students who cheat on any course assignment, lab, test or other activity will have their course grade reduced by one letter grade, regardless of the activity's point value, if it is their first offense at UNC Charlotte.*

Copyright © 2019-2024, Julio C. Bahamón, UNC Charlotte.
All Rights Reserved.