Logic
Workshop 7

In this workshop, you will code a C-language program that analyzes data logically.

LEARNING OUTCOMES
Upon successful completion of this workshop, you will have demonstrated the abilities:
- to create a simple interactive program
- to code a decision using a selection construct
- to code repetitive logic using an iteration construct
- to nest a logical block within another logical block

Instructions

Part 1 (IN-LAB 30%):
Code a program in `temps.c` that does the following:

1- Before the declaration of main define NUMS as 3: #define NUMS 3

2- Print the title of the application.
>---=== IPC Temperature Analyzer ===---<

3- Using a ***for loop***, prompt the user to enter the high and low values for each of ***NUMS*** days. The values entered must be between -40 and 40, and high must be greater than low.

Print the following messages:
 >**Enter the high value for day 1:** **< (or day 2, or day 3)**
* Read the high value.

>**Enter the low value for day 1:** **< (or day 2, or day 3)**
* Read the low value.

4- Use a nested while (or do-while) loop to analyze the results, high must be greater than low, high must be less than 41, low must be greater than -41

*If any entry is incorrect, prompt the user to enter again until the entries pass the tests:
> Incorrect values, temperatures must be in the range -40 to 40, high must be greater than low. <

Then prompt again for the high and low temperatures for the day.

5- When the user has correctly entered the high and low temperatures, add them to variables that will store the total high and total low temperatures for NUMS days.

6- When the loop finishes calculate the average (mean) temperature for NUMS days and display:
> The average (mean) temperature was: <

OUTPUT EXAMPLE WITH ERRORS HANDLED *(use this data for submission)*
```
---=== IPC Temperature Analyzer ===---
Enter the high value for day 1: 8
Enter the low value for day 1: -2
Enter the high value for day 2: 41
Enter the low value for day 2: -4
Incorrect values, temperatures must be in the range -40 to 40, high must be greater than low.

Enter the high value for day 2: 9
Enter the low value for day 2: -4
Enter the high value for day 3: 5
Enter the low value for day 3: 11

Incorrect values, temperatures must be in the range -40 to 40, high must be greater than low.

Enter the high value for day 3: 11
Enter the low value for day 3: 5
The average (mean) temperature was: 4.50
```

Submission

To test and demonstrate execution of your program use the same data as the output example above. Submissions are due by 23:59 on the day after your class. Failure to use the submitter will incur a 40% penalty.

If not on matrix already, upload your `temps.c` to your matrix account. Compile and run your code and make sure everything works properly.

Then run the following script from your account: (replace profname.proflastname with your professors Seneca userid)

`~profname.proflastname/submit btm200/w7_lab <ENTER>`

and follow the instructions.

Part 2 (AT-HOME 40%):

Upgrade temps.c to
- process a 4-day period using a single change to your temps.c code
- display the highest temperature, and on which day it occurred
- display the lowest temperature, and on which day it occurred
- Calculate and display the mean temperature for the 4-day period.

OUTPUT EXAMPLE
```
---=== IPC Temperature Analyzer ===---
Enter the high value for day 1: 8
Enter the low value for day 1: -2
Enter the high value for day 2: 9
Enter the low value for day 2: -4
Enter the high value for day 3: 11
Enter the low value for day 3: 5
Enter the high value for day 4: 10
Enter the low value for day 4: 3
```

```
The average (mean) temperature was: 5.00
The highest temperature was 11, on day 3
The lowest temperature was -4, on day 2
```

AT-HOME REFLECTION (30%)

1. Within main, to how many levels did you have to nest the loops and if statements?
2. Briefly describe the algorithm you used to find the day with the highest temperature.
3. What values did you use to initialize the variables to store the highest and lowest temperatures and why did you select these values?
4. Describe the loop structure you used to ensure that the input temperatures were valid before you used them in calculations.

Put your answers in a file called **reflect.txt.**

Submission

To test and demonstrate execution of your program use the same data as the output example above.

If not on matrix already, upload your **temps.c and reflect.txt** to your matrix account. Compile and run your code and make sure everything works properly.

Then run the following script from your account: (replace profname.proflastname with your professors Seneca userid)

```
~profname.proflastname/submit btm200/w7_home <ENTER>
```

and follow the instructions.

SUBMISSION POLICY

The workshop is **due within 4 days following the in-lab assigned date by 23:59.**

All your work (all the files you create or modify) must contain your name, Seneca email and student number.

You are responsible for regularly backing up your work.

Please Note

☐ A successful submission does not guarantee full credit for this workshop.

☐ If the professor is not satisfied with your implementation, your professor may ask you to resubmit. Resubmissions will attract a penalty.