**To Do’s for Lab Week 2**

**Part 1**

**Exercise 2:**

**Follow the next step and record and submit in a separate text file the items in italics in addition to the C code for the exercise**.

1. What steps would solve the problem?

*Describe in common language the solving of the exercise*.

**Getting the numbers from the user that represent the required pieces, scan for it and do it in such a way that Ignores “/“. Then use an if statement to check all invalid values, and combination of month and day values. Print the values.**

1. Identify the variables that are needed to solve the exercise.

*Indicate the needed variables and specify their purpose.*

**Int month**

**Int day**

**Int year all represent the number of the respective areas**

1. Identify the type of the variables.

*Indicate the type for each variable and explain your choice.*

**Int, i chose it cause you can’t have a decimal value for a year**

1. Read the input information from the keyword and assign the information to the related variables.

*Describe your understanding of the used C instructions*.

**Scanf(“%d%\*c%d%\*c%d” , &month, &day, &year) this takes 3 integers from the user with a “/“ in between the numbers and assigns it to the respective variable**

1. Separately display the values of each variable.

*Explain if the displayed value was correct. If it was not, what change did you do to your program, and why?*

**Yes it was correct.**

1. Display the required output.

*Describe the output. If it is not, what error did you get, and how did you fix it?*

**It was wrong initially because i didnt write Mo , Day , Ye behind the intended values initially.**

*How similar was the solution at Step 1 to the final solution?*

**I'm not sure what this question is asking at all, so i'm assuming if its asking part 2 and part three they’re identical code essentially.**

**Exercise 3:**

**Follow the next step and record and submit in a separate text file the items in italics in addition to the C code for the exercise**.

1. What needs to be changed to solve the exercise starting from the solution for exercise 2? How does this exercise relate to the pocket calculator discussed in class? How is it different?

*Explain in common language your solution for repetitive displaying and compare it to the exercise discussed in class.*

**All i did was put the code the user sees and interacts with within a while loop that is always true. You can just put it in a while loop, they’re different cause in one case your performing calculations based on an input and reading it, and this case were simply reporting back has been told to us, and making sure that these values are in fact valid. In computation there’s not too much that is invalid**

1. What C instructions encode the extra requirements for this exercise starting from the similar instructions used for the pocket calculator program?

*Explain the actions that you expect each of the C instructions will perform.*

*Explain if the displayed values were correct. If it was not, what change did you do to your program, and why?*

**The code will define the variables, it will print and then scan the input, and then it will check all the relevant cases which may make it invalid in a real life application. and it also has all the test cases outlined. it then prints it all back if it passses all the checks**

**Part 2**

**Exercise 1**:

**Follow the next step and record and submit in a separate text file the items in italics in addition to the C code for the exercise**.

1. How can this exercise be solved? What is the concept for the solution?

*Describe in common language the solving of the exercise*.

**All this is is an infinite loop of addition of floating point numbers to 3 decimal places.**

1. What variables are needed for the solution? What is their type?

*Indicate the needed variables and specify their purpose. Indicate the reasons for the selected variable types.*

**You need 3 float one of which being a holding place for the sum to be held. A will hold the input b will hold the sum and bucket will be the temp holding place for the sum float is the only one able to float**

1. Read one number from the keyboard.

*Describe your understanding of the used C instructions*.

**If you scanf a number youre taking an input and then setting it to a variable that can hold your type of numbe**r

1. Add the value of the number to the sum.

*Describe your understanding of the used C instructions*.

**You add the two numbers using sum = num + num**

1. Display the sum.

*Describe your understanding of the used C instructions*.

*Explain if the displayed value was correct. If it was not, what change did you do to your program, and why?*

**Printf(“%.3f”, sum) i had to include the .3 to tell the program to remember and include the 3 decimals.**

*Is there any similarity to Exercise 2? Explain. What is different?*

**Yes the similarity was that i had to take variables from the user in a special way or else it wouldnt fit the test cases.**

1. How can you modify the program to read a second number?

*Explain in common language your solution for repetitive displaying and compare it to the exercise discussed in class.*

**You can modify it to be in a loop so it’ll keep reading a number and set variables back to 0 and by using a b and bucket you can keep adding forever. This is similar to the calculator that it keep running but our classroom calc doesnt carry over the value like this ones ita a continuous calc. The classroom was 1 time use and then reset**

1. Make the changes to the program and display the sum.

*Explain the actions that you expect each of the C instructions will perform.*

*Explain if the displayed values were correct. If it was not, what change did you do to your program, and why?*

**I expect it to work, because it does, all of it is right, i expect all of the c program instructions to complete their respective functions as their programmed**

1. How can you modify the program to work for any number of input values? How many values can your program handle?*Explain in common language your solution for repetitive displaying and compare it to the exercise discussed in class.*

*Explain the actions that you expect each of the C instructions will perform.*

*Explain if the displayed values were correct. If it was not, what change did you do to your program, and why?*

**You can ask what type of number you want based on a switch case and in that case each of them have their own separate loops and addition patterns, or just use limited a size float numbers and use that for all computations and define to the user what the limits of their inputs are. Repetitive displaying just comes from the while loops they’re all correct because the code is right because %.3f worked to display the right decimals**

1. How do you display three decimal numbers for the sum as required by the exercise?

*How similar was the solution at Step 1 to the final solution?*

*I have no idea what this means*

**Exercise 2:**

**Follow the next step and record and submit in a separate text file the items in italics in addition to the C code for the exercise**.

1. What are the special characters in the indicated message?

*Explain what a special character is.*

**Any special chars that have place in the code like “ / \* & all thee may have an effect on the code itself so it's important to have a ways to print them**

1. What are the formats to display special characters?

*Identify the required formats to display the required message.*

**You need a \” for quotes that’s it really**

1. Encode your idea as a C program.

*Describe the output. If it is not, what error did you get, and how did you fix it?*

***I got the exact message in the right format.***

**Part 3**

**Exercise 1**

**Follow the next step and record and submit in a separate text file the items in italics in addition to the C code for the exercise**.

1. How is this exercise like the pocket calculator exercise discussed in class? What is the same and what is different?

*Describe in common language the solving of the exercise, and how it relates to the pocket calculator exercise discussed in class*.

**The solving of this first you have to define what operation you’re doing and the numbers you want to compute on, you need a switch chase and the operations within each one of the and the respective float specific operators, then you need to put it all in a loop for it to keep requesting and completing , what’s different is not much quite honestly**

1. Identify the variables required in the program.

*Indicate the needed variables and specify their purpose. Indicate the reasons for the selected variable types.*

**You need a char variable for the operator choice, floats doubles and more ints in order to have the while loop correctly and the numbers stored and added correctly**

Read a value from the keyboard.

*Describe your understanding of the used C instructions*. *How do you check if the value was read correctly?*

**I can know if the value is read correctly if i just print the value read later on or check the value after all the calculations are finished then im good**

1. Read a character representing the operation.

*How is this read instruction different from the previous read instruction?* *How do you check it the value was read correctly?*

**This read instruction is different because there’s very few operators we are going to allow to do anything with. We have to make a default case for anything but our intended operators. You can check if you just print it or the code works**

1. Perform the operation corresponding to the operation.
2. Display the result.

*Describe your understanding of the C instructions used in Steps 5 and 6*.

*Explain if the displayed value was correct. If it was not, what change did you do to your program, and why?*

***The displayed value initially was not right for absolute value i had to use fabs() for that since a floating value might be used in the operation.***

1. Modify your code so that it can execute steps 3-6 for another value and character.

*Explain in common language your solution for the repetitive actions.*

*Explain the actions that you expect from each C instruction to achieve the repetitions.*

*Explain if the displayed values were correct. If it was not, what change did you do to your program, and why?*

**The displayed values were all the same since its the same working code, and the solution is to just put it in a while loop that repeats just once.C will continue to do the code as long as while is true. As long as we make while not true were good**

1. Modify your code so that it executes for 10 values and characters read from the keyboard.

*What must be changed as compared to Step 7?*

*Explain if the displayed values were correct. If it was not, what change did you do to your program, and why?*

**We can’t just use a minus 1 from our while value anymore because we need it to be utiliziable over and over again for different levels of while. Just some simple algebra and we’re all good.make while true 10 times.**