CNIT 31500 – Lab 2, v2

Purdue University

Fall 2023

1 Goals

The primary goal of this project is to write, test and complete a c program which uses functions and recursion. It is critical that you know how to program and use recursion. While you are given the code for some of these functions, make sure you know how the functions execute! There will be some functions that you must write from scratch. All functions will be called by a numbered user menu, prompting the user for the correct inputs for each one. Then, you will print the output in a neatly formatted and understandable manner. The menu must ask the user if they want to run more functions and also ask them if they want to quit the program, and act accordingly.

2 Specifications

- For each of these, write a simple function. In some cases the code is given to you. Just place it in context of a program.
- You will have a number of functions in your program. You will create a basic menu system that allows a user to select which one of the functions they want to execute.
- Once the user makes their choice, the program must ask them for the correct parameters. After the parameters are entered, the program will execute and echo the results to to command line.
- Each function must be properly formatted, with a prototype.
- When the program is finished, return the number of selections that were executed in your program to the operating system. For example, if you executed all 5 function selections, the return would be 5. If you executed all of them two times, the return value would be 10. You may assume that Pool/Hottub counts as 1 selection even though it involves multiple functions.

3 Submission Guidelines

Submit your code on Gradescope, NOT Brightspace. Do not indicate your name or username in the comments to allow for anonymous grading.

3.1 Checkoff requirements

Pool/hottub and Factorials problems should be checked off based on the specifications. Checkoff of this component must be completed **in lab** by the end of week 4.

4 Functions

4.1 Pool and Hottub calculation

You are to use code from Lab 1 and convert it to multiple functions. The first one will asks a user whether they want to calculate a volume of a pool or a hottub and ask for dimensions (verify them as well). The second one will calculate volume of a hottub and how much water is needed to fill it. The third one will calculate volume of a pool and how much water is needed to fill it. Most of your code should be already written, you will just move it to multiple functions. Feel free to use more than three functions that are listed here.

Bonus: Refactor the code that does parameter checks into a (somewhat) recursive problem. In other words, the function should call itself until correct parameters are obtained. Each function should check a single parameter. Return this parameter if it is valid. You may return -1 if it is not.

4.2 Factorials

Write a function in your C program based on the code given below. The factorial function is an example of direct recursion and is defined by:

$$n! = \begin{cases} 1, & \text{if n} = 0 \\ n(n-1)!, & \text{if n} > 0 \end{cases}$$

The code to use is:

```
int factorial(int n)
{
    if (n==0)
        return 1;
    else
        return n * factorial(n - 1);
}
```

Output: Print out a message and the input and output numbers.

4.3 Towers of Hanoi

Copy the following function into your program and complete the code with a printf statement that prints all moves.

```
void runHanoi(int n, char x, char y, char z)
{
    if (n == 1)
    {
        //printf...
    }
    else
    {
        runHanoi(n-1,x,z,y);
        runHanoi(1,x,y,z);
        runHanoi(n-1,y,x,z);
    }
}
```

Output: Print all of the moves made to the discs on the pegs.

4.4 Fibonacci numbers

Use the following definition to create a C function which takes a int as a parameter and gives the Fibonacci sequence for that number. [Hint: the function header will be: *int fibonacci(int n)*]. The Fibonacci function is defined by:

$$F_n = \begin{cases} 0, & \text{if n} = 0 \\ 1, & \text{if n} = 1 \\ F_{n-1} + F_{n-2}, & \text{if n} > 1 \end{cases}$$

Output: Print out the list of Fibonacci numbers.

4.5 Sum of N

Using recursion, compute the sum of all positive integers until (and including) N, where N is a user input. For example, if N=4, your sum would be 10.
Output: Print the N and the sum of all integers until N.