C Programming I
2020 Fall
Homework 04

Instructor: Po-Wen Chi

Due: 2020.12.04 PM 11:59

Policies:

- Zero tolerance for late submission.
- You need to prepare a README file about how to make and run your program. Moreover, you need to provide your name and your student ID in the README file.
- For the writing assignment, I only accept pdf. MS. doc/docx format is not acceptable. Moreover, please use Chinese instead of English.
- Do not forget your Makefile. For your convenience, each assignment needs only one Makefile.
- The executable programs should be hw0401, hw0402 ....

1 Roman Number (20 pts)

Please write a function to show a given integer (1-3000) in the Roman number form. If you know nothing about the roman numerical system, please reference the following link.

`https://en.wikipedia.org/wiki/Roman_numerals`

You need to implement the function in another C code and prepare a header file. Again, you should print an error message when receiving an invalid input.

2 Tower of Hanoi (20 pts)

The Tower of Hanoi is a mathematical game or puzzle. It consists of three rods and a number of disks of different sizes, which can slide onto any rod. The puzzle starts with the disks in a neat stack in ascending order of size on one rod, the smallest at the top, thus making a conical shape. Figure 1 is an example with 8 disks.

The objective of the puzzle is to move the entire stack to another rod, obeying the following simple rules:

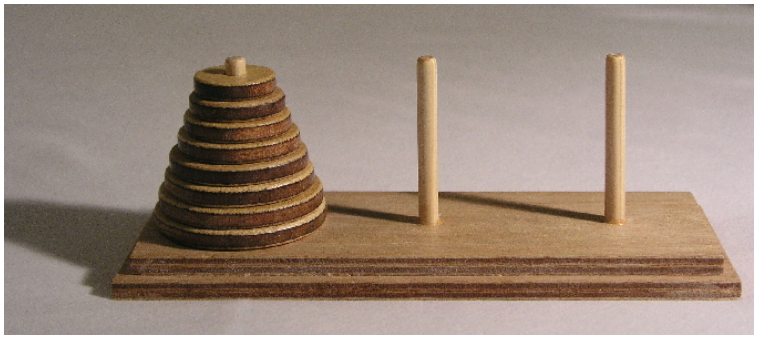

Figure 1: Tower of Hanoi

- Only one disk can be moved at a time.
- Each move consists of taking the upper disk from one of the stacks and placing it on top of another stack or on an empty rod.
- No larger disk may be placed on top of a smaller disk.

Please write a program to list the procedure of moving n disks from one rod to another. We assume there are only 3 rods, and all disks are placed on the first rod with the ascending order of size. Our target is to move these n disks from the first rod to the second rod. The disks are labeled $1, 2, \ldots, n$ from top to bottom.

Note that the Hanoi problem is a very famous recursive problem. For your convenience, I give you a hint as follows:

- Move $m-1$ disks from the source to the spare rod, by the same general solving procedure. Rules are not violated, by assumption. This leaves the disk m as a top disk on the source rod.
- Move the disk m from the source to the target rod, which is guaranteed to be a valid move, by the assumptions.
- Move the $m-1$ disks that we have just placed on the spare, from the spare to the target rod by the same general solving procedure, so they are placed on top of the disk m without violating the rules.

Wait! I have told you that every recursive program can be converted to an iterative program (loop). So this time, I ask you to write **TWO** programs. hw0402-1 is the recursive version and hw0402-2 is the iterative function. Good Luck.

```
$ ./hw0402-1
Please enter the disk number (2-20): 2
move disk 1 to rod 3
move disk 2 to rod 2
move disk 1 to rod 2
```

You need to implement two different functions in another C code and prepare a header file.

3 Taylor's Theorem (20 pts)

I believe you have learned Taylor's Theorem, right? If no, do not worry. I will describe this theorem here.

Theorem 1. *Taylor's Theorem Let $k \geq 1$ be an integer and let the function $f : \mathbb{R} \to \mathbb{R}$ be k times differentiable at the point $a \in \mathbb{R}$. Then there exists a function $hk : \mathbb{R} \to \mathbb{R}$ such that*

$$f(x) = f(a) + f'(a)(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \cdots + \frac{f^{(k)}(a)}{k!}(x-a)^k + h(x)(x-a)^k,$$

and $\lim_{x\to\infty} h(x) = 0$.

Now I want you to use this theorem to calculate the value of e. The approximation will be

$$e^x = 1 + x + \frac{x^2}{2!} + \cdots + \frac{x^k}{k!}$$

The approximation accuracy depends on k. Please write a program to make the user input k and your program calculates e.

```
$ ./hw0403
k-th order Taylor polynomial for e
Please enter k: 1
2
```

What is the k value for the best accuracy? Please write down your answer.

You need to implement the function in another C code and prepare a header file.

BTW, if you are interested in how to use e in standard C, you can read **math.h**. We will use this file soon.

4 Equivalent Resistance (20 pts)

The electrical resistance of an object is a measure of its opposition to the flow of electric current. The resistance (R) of an object is defined as the ratio of voltage across it (V) to current through it (I).

$$R = \frac{V}{I}.$$

In a given combination of resistors (series, parallel, or combination of series/ parallel), the equivalent resistance is that value of resistance, which when replaced in place of the combination, will continue to give the same performance for the part of circuit outside this combination. I believe that you know how to calculate the equivalent resistance. If no, do not worry. I will teach you how to calculate the equivalent resistance in series and parallel resistor combinations. Please see the figure 2.

Now, given a circuit as figure 3. Please write a program to calculate the equivalent resistance. You need to let a user to input R and n, which are 32-bits integers. All resistors have the same resistance value.

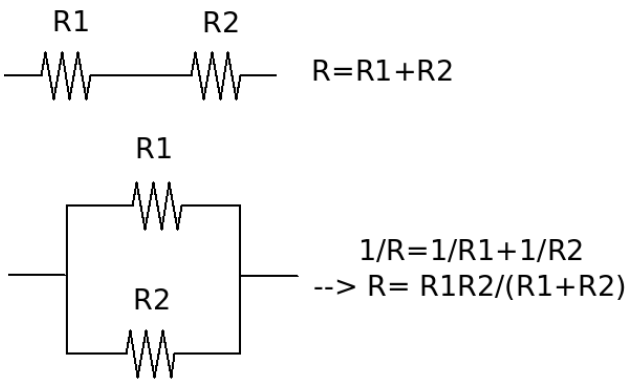

Figure 2: Equivalent Resistance.

```
$ ./hw0404
Please enter the resistance (1-100): 1
Please enter n (1-100): 1
Ans: 2
```

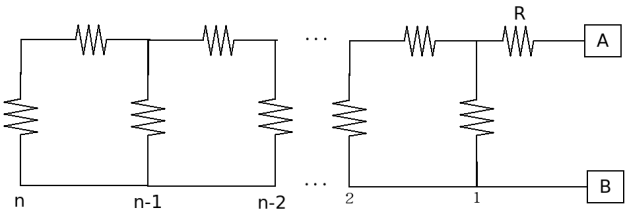

Figure 3: Equivalent Resistance Problem.

Again, precision is not our concern.
You need to implement the function in another C code and prepare a header file.

5 Bulls and Cows (20 pts)

Bulls and Cows (also known as Cows and Bulls or Pigs and Bulls) is an old code-breaking mind or paper and pencil game for two or more players, predating the commercially marketed board game Mastermind.

For the detail rules, please read the following site.

`https://en.wikipedia.org/wiki/Bulls_and_Cows`

You need to develop a game to make a user play with your program. The program ends if and only the user guess the correct answer.

```
$ ./hw0405
Bulls and Cows Game
Round 1 >>>
    Your Guess: 1234
    Response: 1 A 2 B
Round 2 >>>
    Your Guess: 9999
```

```
    Response: Invalid Guess
Round 3 >>>
...
Round 10 >>>
    Your Guess: 4271
    Response: Bingo! Congratulations.
```

Please make the response colorful.

You need to design how to split this program into multiple functions. **Prepare a document to describe the usage of these functions**.

You need to implement your functions in another C code and prepare a header file.

6 Bonus: The Return Value of printf and scanf (5 pts)

printf and **scanf** are the most popular functions you use in this class. In most cases, you do not care the return values of these two functions. Do they have return values? If yes (of course, the answer is yes), what do these values mean? Please check the manuals and write down the answers. Moreover, you need to write some example codes to show your answers.

Again, you need to write down your answer in **Chinese** except some foreign students.

7 Bonus: Fix the Mistakes of Prof. Chi (5 pts)

I have shared a shaming experience of my coding in this class. If you forget the story or miss that class, please ask other classmates and do not ask me. Do not worry, you do not need to know this sad story before answering this problem.

In this class, for the portability issue, I teach you that do not use **short, int, long** since you know nothing about the real memory sizes of these types. Instead, you should use **int32_t** and other explicit definitions.

How about **printf** and **scanf**? In most cases, we use **%d, %ld,** ... in **printf** and **scanf**. Do **%d, %ld,** ... have explicit memory definitions? If no, is there any way to make **printf** and **scanf** use explicit definitions? Please write down the answers.