

National Taiwan Normal University
CSIE Computer Programming I

Instructor: Po-Wen Chi

Due Date: Nov 30, 2021, PM 11:59

Assignment

4

Policies:

- Zero tolerance for late submission.
- Please pack all your submissions in one zip file. **RAR is not allowed!!**
- For convenience, your executable programs must be named following the rule hw**XXYY**, where the red part is the homework number and the blue part is the problem number. For example, hw0102 is the executable program for homework #1 problem 2.
- I only accept **PDF**. MS Word is not allowed.
- **Do not forget your Makefile. For convenience, each assignment needs only one Makefile.**
- Please provide a README.

4.1 K (20 pts)

Please implement a program for the user to input a positive integer and print the character **K** on the screen. If the input is invalid, print an error message and terminate the program.

```
1 $ ./hw0401
2 Please enter n (1-100, n must be odd): 5
3 * *
4 **
5 *
6 **
7 * *
8 $ ./hw0401
9 Please enter n (1-100, n must be odd): 9
10 *   *
11 *   *
12 *   *
13 **
14 *
15 **
16 *   *
```

```
17 *  *
18 *  *
```

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

4.2 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 4.1 is an example with 8 disks.

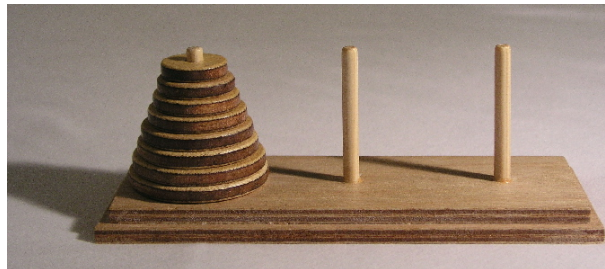


FIGURE 4.1: Tower of Hanoi

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

- 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 $1, 2, \dots, n$ 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.

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

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

4.3 Triangle (40 pts)

Undoubtedly, you know what a triangle is, right? You have learned lots of triangle properties since you were a child. Now, I want you to implement a series of triangle related functions.

```
1 // Setup three points for a triangle.
2 // Success: return 1; Fail: return 0
3 int set_1_point( double x, double y);
4 int set_2_point( double x, double y);
5 int set_3_point( double x, double y);
6
7 // Check if the current setting is a valid triangle.
8 // Yes: return 1; No: return 0
9 int check( void );
10
11 // Return the perimeter of the triangle.
12 // If the current setting is not a triangle, return < 0
13 double get_perimeter( void );
14
15 // Return the area of the triangle.
16 // If the current setting is not a triangle, return < 0
17 double get_area( void );
18
19 // Get the degrees (0-360) of three angles
20 // If the current setting is not a triangle, return < 0
21 double get_1_degree( void );
22 double get_2_degree( void );
23 double get_3_degree( void );
```

```

24
25 // Inscribed circle and Circumscribed circle of the triangle.
26 // If the current setting is not a triangle, the area functions
    return < 0
27 // and the center functions return 0.
28 double get_inscribed_center_x( void );
29 double get_inscribed_center_y( void );
30 double get_inscribed_center_area( void );
31 double get_circumscribed_center_x( void );
32 double get_circumscribed_center_y( void );
33 double get_circumscribed_center_area( void );

```

You need to prepare a header file called **triangle.h**. TA will prepare **hw0403.c** for you. **You MUST build hw0403.c to hw0403 in your Makefile!!** The objective of this problem is to train you how to develop a library.

4.4 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 4.2.

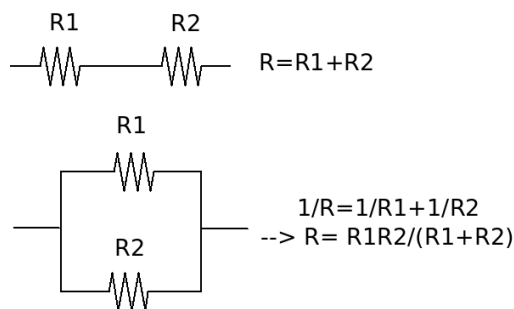


FIGURE 4.2: Equivalent Resistance.

Now, given a circuit as figure 4.3. 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.

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

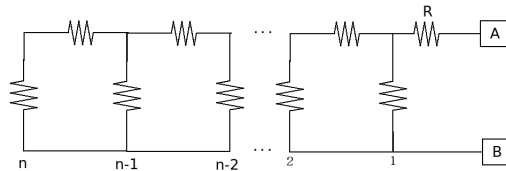


FIGURE 4.3: Equivalent Resistance Problem.

Again, precision is not our concern.

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

4.5 Bonus: printf and scanf (5 pts)

In many C programming classes, **printf** and **scanf** are the first two functions that a student learn. Students use these two functions to show messages on screens and get inputs from users. In most examples, we do not care the return values of these two functions. Do they have return values or just return void? We can read their manuals and see the following prototypes.

```
1 int printf(const char *format, ...);
2 int scanf(const char *format, ...);
```

Integers? Why? Please find the meanings of their return values. Not only write down your description, you should also provide **example codes** to show you are right.