

National Taiwan Normal University
CSIE Computer Programming I

Instructor: Po-Wen Chi
Due Date: 2022.11.15 PM 11:59

Assignment 3

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.

3.1 Quadratic Function (20 pts)

Given a quadratic function $f(x) = ax^2 + bx + c$, please implement the following functions:

```
1 // Setup a quadratic function.
2 // This function must be called before all other functions.
3 // If a == 0, return -1; otherwise, return 0.
4 int32_t setup( int32_t a, int32_t b, int32_t c);
5
6 // Return f(x)
7 double value( double x );
8
9 // Return the minimum value of f(x) between m and n
10 // m may be greater than/equal to/smaller than n
11 double min( double m, double n );
12
13 // Return the maximum value of f(x) between m and n
14 // m may be greater than/equal to/smaller than n
15 double max( double m, double n );
16
17 // Return slope of the tangent line of f(x) at x=t
18 double slope( double t );
```

You should also prepare a header file called **myfunc.h**. Our TAs will prepare hw0301.c which includes myfunc.h and uses these functions. Do not forget to make hw0301.c to hw0301 in your Makefile.

3.2 Standard Deviation (20 pts)

In statistics, the standard deviation is a measure of the amount of variation or dispersion of a set of values. If you do not know what it is, you can check it from wikipedia¹.

https://en.wikipedia.org/wiki/Standard_deviation

Now, given a series of numbers, please implement a function that outputs its standard deviation. The function should be as follows. You can see that the function only takes one input argument. Do not worry. The user can call this function multiple times for inputting a series of numbers.

```
1 #include "mystddev.h"
2
3 // The output is standard deviation for input numbers.
4 // EX: get_stddev(1); --> return standard deviation for {1}
5 //     get_stddev(2); --> return standard deviation for {1,2}
6 //     get_stddev(3); --> return standard deviation for {1,2,3}
7 double get_stddev( int32_t number );
```

If the user wants to clear the input number series, the user should call this function three times where the input are **154,-321,965**. You should also prepare a header file called **mystddev.h**. Our TAs will prepare hw0302.c which includes mystddev.h and uses this function. Do not forget to make hw0302.c to hw0302 in your Makefile.

3.3 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.

¹You can read the Chinese version directly.

I will teach you how to calculate the equivalent resistance in series and parallel resistor combinations. Please see the figure 3.1.

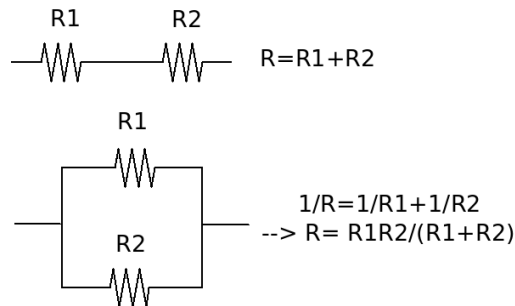


FIGURE 3.1: Equivalent Resistance.

Now, given a circuit as figure 3.2. 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 $ ./hw0303
2 Please enter the resistance (1-100): 1
3 Please enter n (1-100): 1
4 Ans: 2
```

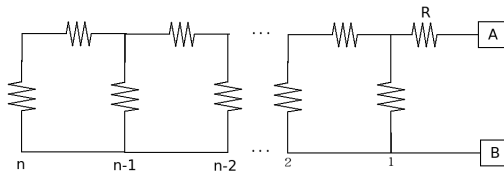


FIGURE 3.2: Equivalent Resistance Problem.

Again, precision is not our concern.

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

3.4 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 3.3 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:

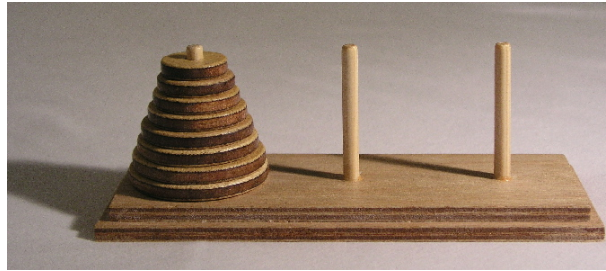


FIGURE 3.3: Tower of Hanoi

- 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 $1, 2, \dots, n$ 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. hw0304-1 is the **recursive** version and hw0304-2 is the **iterative** function. Good Luck.

```
1 $ ./hw0304-1
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
```

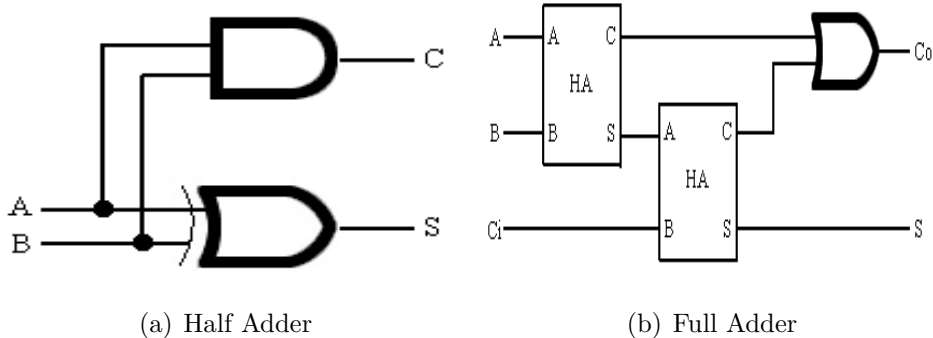
You need to implement two different functions in another C code and prepare a header file.

3.5 Do you have logic? (20 pts)

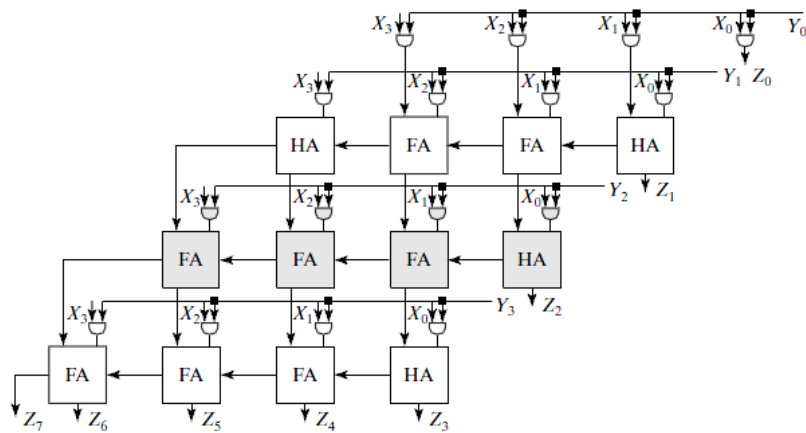
Smallten (小十) , the most powerful artificial intelligence in the world, traveled back to 2022 from the future, and he came back to tell people why "the answer to life the universe and everything is 42".

However, if you want to have the opportunity to communicate with Smallten and gain approval, you need to have some basic knowledge of computers. Are you ready?

1. Please use the provided function called `bool norGate(bool, bool);` to implement the remaining gate related functions in `logic.c`
 - Wiki: [Logic Gate](#)
2. Please implement the half adder and full adder in `logic.c`
 - Wiki: [Adder \(electronics\)](#)



3. Please use **the above library** to implement the 4-bit multiplier in `hw0305.c`
 - Wiki: [Binary multiplier](#)
 - `hw0305.c`
 - The multiplier must be implemented in a function
- ```
1 void multiplier(bool x0, bool x1, bool x2, bool x3,
 bool y0, bool y1, bool y2, bool y3);
```
- You need to print the answer **in decimal**. Please implement in a function. Don't write in `main()` or `multiplier()`.
  - `main()`
    - \* First, call `communicate()` which is declared in `smallten.h`. (TA will prepare `smallten.c`)



(c) Multiplier

- \* Second, get two number for multiplier from TA's input.
- \* Finally, print the answer of multiplier.

- Rules

- If there is any invalid input, print an error message and terminate your program.
- It is **forbidden to declare any variables** when implementing gate, adder, multiplier. You can only use the variables declared in logic.c.
- When implementing function of gate, you can **only use the nor-Gate()**.

```

1 $./hw0305
2 // Result of communicate()
3 Please enter the first number (A) in binary: 1110
4 Please enter the second number (B) in binary: 1011
5 A = 1110 (2) = 14 (10)
6 B = 1011 (2) = 11 (10)
7 A x B = 14 * 11 = 10011010 (2) = 154 (10)

```

## 3.6 Bonus: Where is math.h? (5 pts)

In this class, I have shown you how to use the standard math library. You need to include **math.h** in your code. That is, **math.h** is somewhere in your computer. Where is it? Please find its location in the Linux environment<sup>2</sup>. Is there any path where you can include header files by default?

<sup>2</sup>Actually, its location may be different according to Linux distributions. Do not worry about this issue. Just use your linux environment