National Taiwan Normal University
CSIE Computer Programming I

Instructor: Po-Wen Chi
Due Date: 2023.11.05 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 hwXXYY, 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** or **TEXT**. MS Word is not allowed.
- Do not forget your Makefile. For convenience, each assignment needs only one Makefile.
- Please provide a README file. The README file should have at least the following information:
  - Your student ID and your name.
  - How to build your code.
  - How to execute your built programs.
  - Anything that you want to notify our TAs.
- **DO NOT BE A COPYCAT!!** You will get ZERO if you are caught.

3.1 Circle Functions (20 pts)

A circle is a shape consisting of all points in a plane that are at a given distance from a given point, the centre. The distance between any point of the circle and the centre is called the radius. This time, I want you to develop a series of circle related functions for a circle where its centre is at $\mathcal{O}$.

```
// Setup the radius r of the circle.
// This function must be called before all other functions.
// If r <= 0, return -1; otherwise, return 0.
int32_t set_radius( double r);

// Return the circumference of the circle
// If the radius is not set, return a negative number.
double get_circle_circumference();

// Return the area of the circle
// If the radius is not set, return a negative number.
double get_circle_area();

// Given x, (x,y) is a point on the upper circle.
// Return the area bounded by the tangent line at (x,y), x-axis
    and y-axis.
// If the radius is not set, return a negative number.
// If it cannot form a triangle, return a negative number.
// If x is not a reasonable input, return a negative number.
double get_tangent_area( double x );

// Return the inner and outer regular polygon area.
// If the radius is not set, return a negative number.
// If n < 3, return a negative number.
double get_inner_regular_polygon_area( int32_t n );
double get_outer_regular_polygon_area( int32_t n );
```

For the inner and outer regular polygon, please see Fig. 3.1 for reference.

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

3.2 Control Game Character (20 pts)

Do you like to play video games? In most games, controlling game character is a basic feature. This time, I want you to develop a series of controlling function. For your simplicity, in this problem we only focus on moving.

```
// Setup the initial position to (x,y) and the moving direction
    .
// a is the angle counterclockwise from x axis.
void initialize( double x, double y, double a);

// From the current, move forward length with the given
    direction.
// If the initial position is not set, return -1.
int32_t forward( double length );

// From the current direction, turn clockwise/counterclockwise
    x.
```

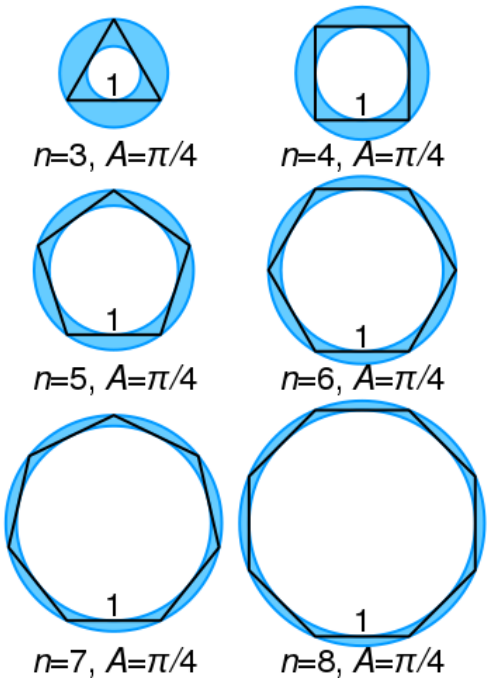

FIGURE 3.1: The area bounded by the circumcircle and incircle.

```
// If the initial position is not set, return -1.
int32_t clock_turn( double x );
int32_t counterclock_turn( double x );

// Print the current location and the direction.
// The print format is "position: (x,y), angle: a".
// a is the angle counterclockwise from x axis and 0 <= a < 2pi
// If the initial position is not set, return -1.
int32_t print();
```

Figure 3.2 is an example.

You should also prepare a header file called mycontrol.h. Our TAs will prepare hw0302.c which includes mycontrol.h and uses these functions. **Do not forget to make hw0302.c to hw0302 in your Makefile.**

3.3 Binary Form (20 pts)

Given a **signed** 32-bit integer, please develop a function to print its binary form. There are two requirements here:

1. Loop and Bitwise operators are not allowed.
2. The feature should be implemented as a function.

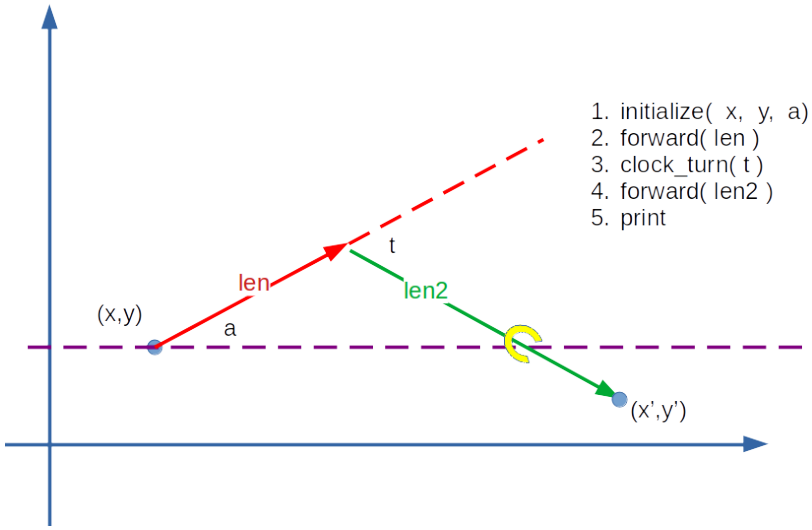

FIGURE 3.2: Example.

```
$ ./hw0303
Please enter the number: 1
00000000 00000000 00000000 00000001
$ ./hw0303
Please enter the number: 2
00000000 00000000 00000000 00000010
```

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.

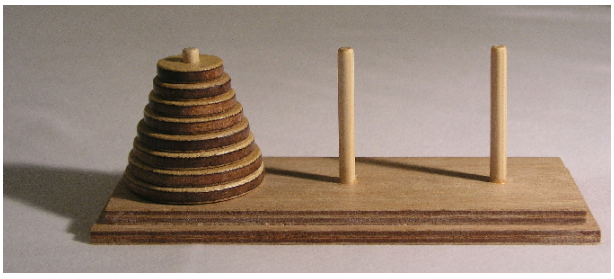

FIGURE 3.3: Tower of Hanoi

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

```
$ ./hw0304-1
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.

3.5 How to roll your dice SE (Simple Edition)? (20 pts)

Your TAs love playing games (replies to 3.2), and one of them loves Role Playing Games the most. He loves to play many kinds of RPGs, and one of them is TRPG (Tabletop Role Playing Game).

TRPGs, unlike their digital counterparts, rely heavily on imagination, improvisation, and yes, the roll of dice. The unpredictability of a dice roll, the anticipation as it tumbles, and the fate it decides - all contribute to the magic of TRPGs.

You are going to create a general dice-rolling simulator to help people play common TRPGs without any physical dice, helping players dive into the TRPG world without any distractions.

3.5.1 Requirements:

Requirements are a must-have. The score will be graded zero if any of it is missed.

- Pray prepare a README to show all features and functions you have implemented. If you use PDF as README, please make sure your file can be opened in a general Linux environment.
- Please show a tutorial when starting the program to help your user how the simulator works.
- DO implement dice roll functions in another C code diceRolls.c and prepare a header file.
- You CANNOT include <math.h> directly inside hw0305.c, but it' s okay to use it outside.
- Using char is prohibited. You can use Array in this problem.
- Make sure your dice will be rolled randomly and independently.
- Please show the result of each die and the sum upon each roll.
- Any program crashes are prohibited.

3.5.2 Basic features (4 pts each):

1. Let users call for a roll of 1d6
2. Let users call for a roll of AdX
3. Let users call for a roll of AdXkY+B
4. Let users call for a roll of AdXkhHklLkcC+B

Explanation:

- **1d6** means **"roll one 6-sided die"**.

- **A** is **the number of dice to be rolled**.
- **X** is **the number of faces of each dice**.
- **B** is**a number to be added to the sum of the rolls**. B can be negative.
- **Y** is **the number of dice kept from the roll**. Users may choose by free will.
- **AdXkhHklLkcC+B** means **"roll A dice with X sides each, keeping the H highest, the L lowest, and C of the player's choice"**.
- **Constraints:**
  - A, B, X, Y, H, L, C are 32-bits integers.
  - $0 \leq A \leq 10$
  - $2 \leq X \leq 100$
  - $-10 \leq B \leq 10$
  - $0 \leq Y, H, L, C \leq 10$

3.5.3 Additional features (up to 10 pts):

You can choose which additional features you want to implement:

- Print error message upon any wrong input (2pts)
- You can design by yourself, or refer to common TRPG rolls (SKILL ROLLS, LUCK ROLLS, SANITY ROLLS, ⋯)

You have good TAs. Every feature below will be a free evaluation of evidence graded by TAs (Please notify and outline in README if you want points here):

- Give your simulator a cool name ($0 \sim +1$)
- Outstanding User Interface and Experience ($0 \sim +2$)
- Anything special (1pts each up to 4pts)

3.5.4 Sample Interaction

- The sample interaction below is the monk result. You can design the interface yourself.
- Please DO NOT use any ascii dice art *exactly* like mine (-5). Please think of any other more interesting. (It' s okay to not look like dice)

```
$ make
$ ./hw0305
Welcome to "How to Rickroll your dice" !
Please refer to README for more information.
0. Terminate me
1. blah [explain more]
2. blah blah [explain more]
3. blah blah blah [explain more]
4. blah blah blah blah [explain more]
5. [Your cool feature]
----------------------------------------
Your action: 1

+---+
| 3 |
+---+
result: 3

----------------------------------------
Your action: 2
Please enter A, X: 3 100

+---+ +---+ +---+
| 5 | |100| |22 |
+---+ +---+ +---+
result: 5 + 100 + 22 = 157

----------------------------------------
Your action: 3
Please enter A, X, Y, B: 5 10 1 2

1.    2.    3.    4.    5.
+---+ +---+ +---+ +---+ +---+
| 4 | | 8 | | 7 | | 6 | | 3 |
+---+ +---+ +---+ +---+ +---+
Please choose 1 dice from above: 3
+---+
| 7 |
+---+
result: 7 + 2 = 9

----------------------------------------
Your action: 4
```

```
Please enter A, X, H, L, C, B: 10 20 2 1 3 -3

+---+ +---+ +---+ +---+ +---+
| 4 | | 8 | | 7 | | 6 | | 3 |
+---+ +---+ +---+ +---+ +---+
+---+ +---+ +---+ +---+ +---+
|14 | |18 | |17 | |16 | |13 |
+---+ +---+ +---+ +---+ +---+

Highest 2:
+---+ +---+
|18 | |17 |
+---+ +---+
Lowest 1:
+---+
| 3 |
+---+
Choose 3:
1.    2.    3.    4.    5.
+---+ +---+ +---+ +---+ +---+
| 4 | | 8 | | 7 | | 6 | |14 |
+---+ +---+ +---+ +---+ +---+
6.    7.
+---+ +---+
|16 | |13 |
+---+ +---+
Please choose 3 dice from above: 1 2 3

result: 18 + 17 + 3 + 4 + 8 + 7 - 3 = 54

----------------------------------------
Your action: 0
bye
>
```

3.5.5 Reminder:

- Your README should include the following sections: Introduction, Features Implemented, How to Use the Program, and Special Notes.
- Please write a description if you implement EVERY kind of roll variations. Make bullet points or checkboxes if needed.
- The max score of this problem will only be up to 20.
- Rick-rolling your TAs is prohibited.
- **You have good peers**. Feel free to discuss this on the Forum.
- **Your TA is not a TRPG dictionary**. Please refer to the following source for more information:

- https://en.wikipedia.org/wiki/Dice_notation
- https://www.chaosium.com/cthulhu-quickstart
- Please list any source on README if you refer to anything online.

3.6 Bonus: diff and patch (5 pts)

Welcome to the coding hell! From now on, you will write lots of codes. Undoubtedly, it implies that you will update your code very frequently. Sometimes your code has been released to others before your updates. How to deliver your new code to them? Of course you can directly deliver the new code but it is not a good idea when the code size is huge. To solve this problem, you can use **diff** command to create patch files for them and they can use **patch** command to apply patches.

How to do this task? Please write a tutorial with an example.

3.7 Bonus: Plagiarism Detection (5 pts)

Do you know how TAs detect assignments plagiarism? I will show you the way. Please see the following URL.

https://theory.stanford.edu/~aiken/moss/

This is an Internet service. But how does it work? At the bottom of the page, there is a paper to describe the detection idea. Please read the paper and write an introduction report **in Chinese** unless you are a foreign student.