C Programming I
2022 Fall
Midterm

Instructor: Po-Wen Chi

Date: 2022.10.29 PM 2:00-6:00

Policies:

- Online test.
- Do not forget to include your Makefile. TA will only use the command make to build your program. If make fails, you will get zero points and no room for bargaining. **So if you do not know how to solve a problem, please, do not include it in your Makefile**.
- I do not care your source code file names, but the executive binary names should be **mid01, mid02, mid03, mid04, mid05**.
- You can ask TA if you do not understand the problems.

1 Line Equation (20 pts)

Given a rectangle and a point P, please find a line equation that passes P and can divide the rectangle equally. The rectangle is given by four points P_1, P_2, P_3, P_4. For your convenience, I promise the first point is the **top-left** point and the order of these four points is clockwise. Figure 1 is an example.

Note that if the input shape is not a rectangle, you need to print an error message and terminate the program. The output form should be one of these:

- $y = a \cdot x + b$
- $x = c$

```
$ ./mid01
P1(x,y): 0.0,2.0
P2(x,y): 4.0,2.0
P3(x,y): 4.0,0.0
P4(x,y): 0.0,0.0
P(x,y): 6.0,3.0
Line: y = 0.50 * x + 0.00
```

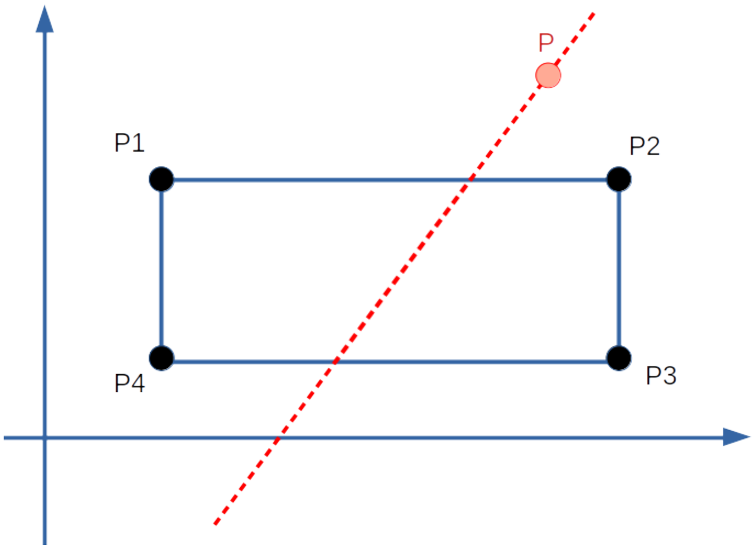

Figure 1: You should find the red dotted line which divides the rectangle equally.

You should use **double**, print to the 2nd decimal place and the precision is not a concern in this problem.

2 Binary Palindrome Number (20 pts)

We call a number is a binary palindrome number if its binary from is the same backward as forward. For example, a **8-bits** integer 36 is a binary palindrome number because

$$(36)_{10} = (00100100)_2$$

- →: 0010 0100
- ←: 0010 0100

Given a **32-bits** integer, please write a function to determine if it is a binary palindrome number.

```c
// Return 1 if x is a binary palindrome number; otherwise, return 0.
int32_t isBinaryPalindrome( int32_t x );
```

You need to implement this function in **binp.c** with a header file **binp.h**. The TA will prepare **mid02.c** for you. Do not forget to build **mid02.c** in your makefile. The following is an example of **mid02.c**.

```c
#include <stdio.h>
#include <stdint.h>
#include <binp.h>

int main()
{
  int32_t x = 36;
  printf( "%d\n", isBinaryPalindrome( x ) );
  return 0;
}
```

It is useless if you always return 0 or 1. For the sake of fairness, **using Array is not allowed!!**

3 GPA Calculator (20 pts)

Grade Point Average (GPA) is a number that indicates how high you scored in your courses on average. The calculation is as follows.

$$\mathrm{GPA} = \frac{\sum\limits_{\forall Courses} \text{Course\_Credit} \times \text{Course\_Grade\_Point}}{\sum\limits_{\forall Courses} \text{Course\_Credit}}$$

Please develop a program to to calculate GPA.

```
$ ./mid03
1) Input the course grade.
2) Remove the course grade.
3) GPA report.
4) Target GPA.
5) Exit.
Your Choice (1-5):
```

For any invalid choice, return back to this main menu. The choice should be **int32_t**. Let's see the function of each choices.

```
$ ./mid03
1) Add the course grade.
2) Delete the course grade.
3) GPA report.
4) Target GPA.
5) Exit.
Your Choice (1-5): 1
Points (0-100): 90
Credits (0-4): 4
```

Note that the input point is 100 points scale. You need to translate it to the grade according to table 1.

After entering a course grade, you should return to the main menu. If there is any invalid input, print an error message and ask the user to re-input the value again. Next is the deletion function.

```
$ ./mid03
1) Add the course grade.
2) Delete the course grade.
3) GPA report.
4) Target GPA.
5) Exit.
Your Choice (1-5): 2
```

Table 1: NTNU Conversion Table

Rank	Grade Point	100 points
A+	4.3	90-100
A	4.0	85-89
A-	3.7	80-84
B+	3.3	77-79
B	3.0	73-76
B-	2.7	70-72
C+	2.3	67-69
C	2.0	63-66
C-	1.7	60-62
D	1.0	50-59
E	0.0	1-49

```
Points (0-100): 86
Credits (0-4): 2
```

For your simplicity, currently you do not need to check if the course grade has been added to the record. After deleting a course grade, you should return to the main menu. If there is any invalid input, print an error message and ask the user to **re-input** the value again. Next is the report function.

```
$ ./mid03
1) Add the course grade.
2) Delete the course grade.
3) GPA report.
4) Target GPA.
5) Exit.
Your Choice (1-5): 3
Total Credits: 120
GPA: 3.72
```

After this function, return to the main menu. Next is the target function.

```
$ ./mid03
1) Add the course grade.
2) Delete the course grade.
3) GPA report.
4) Target GPA.
5) Exit.
Your Choice (1-5): 4
Target GPA: 4.12
Remaining Credits: 18
Your should get at least A+ on average for the remaining credits.
```

After the target grade, you should return to the main menu. If there is any invalid input, print an error message and ask the user to **re-input** the value again. For the choice 5, you can just terminate your program.

4 Tortoise and Hare(10 pts)

Have you ever heard a story about Tortoise and Hare（龜兔賽跑）？Now they want to have a match again. The race distance is 12.3m. Tortoise still runs slowly and each turn it can only move forward 0.3m. As for Hare, each turn it may take a nap or jump forward. Of course, if Hare takes a nap, this turn it will stay at the same place. The speed of Hare and the nap probability is given by the user. You should use **double** for the speed and the probability. Note that the animal must passes over 12.3m to win the game.[1]

```
$ ./mid04
The Hare Speed (m/turn): 1.2
The Nap Probability (0-1): 0.5
Turn 0) Tortoise: 0.000000, Hare: 0.000000
Turn 1) Tortoise: 0.300000, Hare: 1.200000
Turn 2) Tortoise: 0.600000, Hare: 2.400000
Turn 3) Tortoise: 0.900000, Hare: 2.400000 (NAP)
Turn 4) Tortoise: 1.200000, Hare: 2.400000 (NAP)
...
Turn X) Tortoise: 3.900000, Hare: 13.200000 (Winner)
```

Note that if they pass over the line at the same turn, print **Draw** instead of Winner.

5 Triangle (30 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.

```
// Setup three points for a triangle.
// Success: return 1; Fail: return 0
int set_point_1( double x, double y);
int set_point_2( double x, double y);
int set_point_3( double x, double y);

// Check if the current setting is a valid triangle.
// Yes: return 1; No: return 0
int check( void );

// Return the perimeter of the triangle.
// For any errors, return < 0
double get_perimeter( void );

// Return the area of the triangle.
// For any errors, return < 0
double get_area( void );

// Get the degrees (0-360) of three angles
// For any errors, return < 0
double get_degree_1( void );
double get_degree_2( void );
double get_degree_3( void );
```

[1] Stopping at 12.3m does not mean winning the game!

```c

// The incircle （內切圓） of a triangle is the largest circle that can be contained in the triangle.
// Return the area of the incircle of the given triangle.
// For any errors, return < 0
double get_incircle_area( void );
```

You need to prepare a header file called **triangle.h**. TA will prepare mid05.c for you. **You MUST build mid05.c to mid05 in your Makefile!!**

6 Bonus: Your Comments (5 pts)

Please answer the following questions.

- Your comments about this class. Any comments are welcomed. Do not worry about typos or any grammar errors. However, you will get nothing if you leave this question blank.
- The final exam will be 12/24, which is Christmas Eve. I want to move the exam from the afternoon to 9:00-13:00 so that you can enjoy the dinner with your family or have a date. Do you agree? Please vote on the moodle.