National Taiwan Normal University
CSIE Computer Programming I

Instructor: Po-Wen Chi
Due Date: Dec 14, 2021, PM 11:59

Assignment 5

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**. MS Word is not allowed.
- Do not forget your Makefile. For convenience, each assignment needs only one Makefile.
- Please provide a README.

5.1 Big Two (20 pts)

Have you ever played the poker game **Big Two**? As the game's name, the point order in this game is

$$2 > A > K > Q > J > 10 > 9 > 8 > 7 > 6 > 5 > 4 > 3$$

In this problem, I want you to sort cards for Big Two players. Again, the card encoding is as follows:

- 1-13: ♠ A, 2, . . ., 9, 10, J, Q, K
- 14-26: ♡ A, 2, . . ., 9, 10, J, Q, K
- 27-39: ♢ A, 2, . . ., 9, 10, J, Q, K
- 40-52: ♣ A, 2, . . ., 9, 10, J, Q, K

The sorting rule is here:

- Ascending order.

- For the same point, the order should be ♠ → ♡ → ♢ → ♣.

You need to implement the following function in **poker.c** with a header file **poker.h**.

```
int32_t big_two_sort( int8_t cards[] );
```

For your convenience, in this problem, I guarantee the element number of the input array is 13. If the input is invalid, please return -1; otherwise, return 0. The TA will prepare **hw0501.c** for you and I promise that **poker.h** will be included. You must build **hw0501.c** to **hw0501** in your Makefile.

5.2 Statistics (20 pts)

Given an 32-bits integer array, please implement the following functions to analyze the input data.

```
double get_mean( const int32_t input[], size_t size );
double get_median( const int32_t input[], size_t size );
int32_t get_mode( const int32_t input[], size_t size );
double get_stddev( const int32_t input[], size_t size );
```

- **mean**: The arithmetic average.
- **median**: The middle value.
- **mode**: The number that occurs most frequently. If there are more than one number that occur the same times, print the max number.
- **stddev**: The standard deviation.

You need to prepare a header file called **data.h** with **data.c**. TA will prepare hw0502.c for you. **You MUST build hw0502.c to hw0502 in your Makefile!!** The objective of this problem is to train you how to develop a library.

Again, precision does not matter.

5.3 Polynomial Calculator (20 pts)

In mathematics, a polynomial is an expression consisting of variables and coefficients. A polynomial involves the operations of addition, subtraction, multiplication. So this time, I want you to develop a polynomial calculator. For your simplicity, this problem guarantees that all polynomials in this problem has only one variable, which we call it x. The degree for each polynomial should be in $[1, 100]$.

```
$ ./hw0503
Please enter p1 degree: 2
Please enter p1 coefficients: 2 3 4
Please enter p2 degree: 2
Please enter p2 coefficients: 1 1 1
p1: 2x^2 + 3x + 4
p2: x^2 + x + 1
p1 + p2: 3x^2 + 4x + 5
p1 - p2: x^2 + 2x + 3
p1 * p2: 2x^4 + 5x^3 + 9x^2 + 7x + 4
```

For your convenience, I guarantee that the coefficient number will be equal to the degree plus one. All coefficients are 32-bits integers.

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

5.4 Banqi (40 pts)

Banqi, also known as Dark Chess or Blind Chess, is a two-player Chinese board game played on a 4×8 grid, or half of the xiangqi board. For your reference, please check the wikipedia:

https://zh.wikipedia.org/wiki/%E6%9A%97%E6%A3%8B

Note that we use the Taiwanese rule here without any variations (變體). This time, I want you to implement this game. Do not worry, you do not need to implement AI. Instead, you simply implement the game interface for two human players. BTW, if you are interested in the game AI research, please contact Prof. Shun-Shii Lin. First, I will show you the piece encoding rule:

將	1	帥	A
士	2	仕	B
象	3	相	C
車	4	俥	D
馬	5	傌	E
包	6	炮	F
卒	7	兵	G

TABLE 5.1: The piece encoding rule.

The game process should be:

```
$ ./hw0504
  1 2 3 4 5 6 7 8
 +---------------+
1|*|*|*|*|*|*|*|*|
 |---------------|
2|*|*|*|*|*|*|*|*|
```

```
 |---------------|
3|*|*|*|*|*|*|*|*|
 |---------------|
4|*|*|*|*|*|*|*|*|
 +---------------+
Player 1 (x,y):
```

Note that $1 \leq x \leq 8$ and $1 \leq y \leq 4$. First, all pieces are faced down and marked as *. When a faced-down piece is chosen, it will be flipped like this:

```
  1 2 3 4 5 6 7 8
 +---------------+
1|*|*|*|*|*|*|*|*|
 |---------------|
2|*|*|*|*|*|*|*|*|
 |---------------|
3|*|*|*|*|*|*|*|*|
 |---------------|
4|*|*|*|*|*|*|*|*|
 +---------------+
Player 1 (x,y): 1, 1
  1 2 3 4 5 6 7 8
 +---------------+
1|B|*|*|*|*|*|*|*|
 |---------------|
2|*|*|*|*|*|*|*|*|
 |---------------|
3|*|*|*|*|*|*|*|*|
 |---------------|
4|*|*|*|*|*|*|*|*|
 +---------------+
Player 2 (x,y):
```

If a faced-up piece is chosen, then you need to move the piece to other place, like this

```
  1 2 3 4 5 6 7 8
 +---------------+
1|B|*|*|*|*|*|*|*|
 |---------------|
2|*|*|*|*|*|*|*|*|
 |---------------|
3|*|*|*|3| | | | |
 |---------------|
4|*|*|*|*|*|*|*|*|
 +---------------+
Player 2 (x,y): 4,3
To (x,y): 5,3
  1 2 3 4 5 6 7 8
 +---------------+
1|B|*|*|*|*|*|*|*|
 |---------------|
```

```
2|*|*|*|*|*|*|*|
 |---------------|
3|*|*|*| |3| | | |
 |---------------|
4|*|*|*|*|*|*|*|
 +---------------+
Player 1 (x,y):
```

If the chosen piece has no possible move, pass the control to the other side. If the chosen piece is wrong or the movement is invalid, please print a warning message and make the player re-input its choice. For other rules, please check the wikipedia. When the game ends, please print the winner.

Good Luck.

5.5 Bonus: An Interesting Code (5 pts)

Please read the following code and guess the output. Compile this code and run it. Is the output the same with what you guess? Please explain the reason of the output.

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

unsigned int ui = 0;
unsigned short us = 0;
signed int si = -1;

int main()
{
    int64_t r1 = ui + si;
    int64_t r2 = us + si;
    printf("%ld %ld\n", r1, r2);
}
```

Hint: You can look up the reason from C11. I have uploaded C11 draft on my website. The keyword: **Integer Conversion** and **Integer Promotion**.