VG101 Lab Worksheet

Lab 10

Instructor: Dr. Yifei ZHU
TA: Hangrui CAO
TA: Haoxuan SHAN
TA: Qinhang WU
TA: Muchen XU

Code Quality Requirement

- Ensure proper indentation
- Make naming meaningful
- Include necessary comments
- Split the code over functions
- Test the code as much as you can

Lab 10 Worksheet

Basic Syntax Exercise

FileIO (II)

Write a program that takes inputs from a local file, prints output in the command line and creates a log file locally.

Description:

You will receive m numbers and a lucky number n. For each m_i, you need to see whether it is greater, less or equal to the lucky number. You need to create a log file named `lucky.log` locally on your computer. When m_i is greater to n, log `[i] Greater.` where i is the index of m_i. When m_i is less than n, log `[i] Less.` When m_i is equal to n, log `[i] Equal.`. You also need to count the total number of m_i that is equal to n.

Input:

The first line contains two integers n, m. n indicates the lucky number, and m indicates how many numbers you need to deal with later.

In the next m lines, each line includes a integer.

Output:

A line include a number p, indicating the total number of m_i that is equal to n.

Sample Input

```
7 5
3
7
10
1
7
```

Sample Output

```
2
```

Sample `lucky.log`

```
[1] Less.
[2] Equal.
[3] Greater.
[4] Less.
[5] Equal.
```

Card Displayer

You will get a card represented by two integer number as input: **Suit**, **Spot**.

Suit will be an integer in $\{0, 1, 2, 3\}$, representing `Spades`, `Hearts`, `Clubs`, `Diamonds`. **Spot** will be an integer in $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13\}$, representing `Ace`, `Two`, `Three`, `Four`, `Five`, `Six`, `Seven`, `Eight`, `Nine`, `Ten`, `Jack`, `Queen`, `King`. You task is to output a text description of the card: "`Spotname` of `Suitname`".

Try take the input as standard input first. Then, try to do it as command-line input locally. On JOJ, just submit standard input version.

Sample Input

```
2 1
```

Sample Output

```
Ace of Clubs
```

Monetary Usage Problem

As we know, here're RMB bills with values: 1, 5, 10, 20, 50, 100. Now, suppose you have enough amount for each kind of bill, when given a price, decide how many bills of each kind you should pay so that the total number of bills is minimized. Output the number of each bill you need to pay in the order 1, 5, 10, 20, 50, 100.

Sample Input

```
77
```

Sample Output

```
2 1 0 1 1 0
```

Pointer with Structure

Implement a header file of rational number struct, with which you can perform arithmetics including addition, subtraction, multiplication and division.

```c
//rational.h

#ifndef RATIONAL_H
#define RATIONAL_H

#include <stdio.h>
typedef struct rational_t
{
    int num; //Numerator
    int den; //Denominator, always positive
} RationalInt;
int gcd(int a, int b)
//return the greatest common divisor of a and b, might be negative.
{
    return b ? gcd(b, a % b) : a;
}
void reduce(RationalInt *a)
{
    //TODO: reduce the fraction num/den, save the answer in a
    //TODO: make sure the denominator is always positive
}
void set(RationalInt *a, int num, int den)
{
    //TODO: set a with reduced num/den
}
void add(RationalInt *a, const RationalInt b)
{
    //TODO: add b to a, save the reduced answer in a
}
void subtract(RationalInt *a, const RationalInt b)
{
    //TODO: subtract b from a, save the reduced answer in a
}
void multiply(RationalInt *a, const RationalInt b)
{
    //TODO: mulitply a with b, save the reduced answer in a
}
void divide(RationalInt *a, const RationalInt b)
{
    //TODO: divide a by b, save the reduced answer in a
}
void print(RationalInt *a)
{
    printf("%d/%d\n", a->num, a->den);
```

```
}

#endif
```

After your implementation, design your own test on your struct. Here we provide you with a sample test file:

```c
//main.c

#include <stdio.h>
#include "rational.h"
int main()
{
    RationalInt a, b, c, d;
    set(&a, 1, 5);
    set(&b, 1, 4);
    set(&c, 1, 3);
    set(&d, 1, 2);
    add(&a, b);
    print(&a);
    subtract(&b, c);
    print(&b);
    multiply(&c, d);
    print(&c);
    divide(&d, a);
    print(&d);
}
```

Here the expected output should be

```
9/20
-1/12
1/6
10/9
```

Note: smartly combine the `RationalInt` struct with expression evaluation algorithm, you will derive a simple scientific calculator like `fx-991`.

Grading Rubric

Criteria	Weight	Available Time	Due Time	Entry
Worksheet Exercises	100%	**4:00pm**, July.17	**11:59am**, July.19	JOJ

- For worksheet exercises that are available on JOJ, you need to submit your code to JOJ before Sunday midnight. You'll earn most partial points as long as you submit the code. For those that are not available on JOJ, you don't need to submit them.