National Taiwan Normal University
CSIE Computer Programming I

Instructor: Po-Wen Chi
Due Date: 2023.xx.xx PM 11:59

Assignment 1

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.

1.1 Print Colorful Words (20 pts)

In this class, I have shown you how to print strings on your screen. However, it is very boring to use the same color all the time. Can I print strings with different colors? Absolutely yes! But how to do this? The keyword is **ANSI escape codes**.

Please write a program to show the following message on the screen, including punctuation marks. Then you should color all quoted texts with red and all human names with blue.

```
$ ./hw0101
```

```
Now in those days there was in the land of Helsinki a young scholar named Linus the Torvald. Linus was a devout man, a disciple of RMS and mighty in the spirit of Turing, von Neumann and Moore. One day as he was meditating on the Architecture, Linus fell into a trance and was granted a vision. And in the vision he saw a great Penguin, serene and well-favoured, sitting upon an ice floe eating fish. And at the sight of the Penguin Linus was deeply afraid, and he cried unto the spirits of Turing, von Neumann and Moore for an interpretation of the dream.

And in the dream the spirits of Turing, von Neumann and Moore answered and spoke unto him, saying, "Fear not, Linus, most beloved hacker. You are exceedingly cool and froody. The great Penguin which you see is an Operating System which you shall create and deploy unto the earth. The ice-floe is the earth and all the systems thereof, upon which the Penguin shall rest and rejoice at the completion of its task. And the fish on which the Penguin feeds are the crufty Licensed codebases which swim beneath all the earth’s systems. The Penguin shall hunt and devour all that is crufty, gnarly and bogacious; all code which wriggles like spaghetti, or is infested with blighting creatures, or is bound by grave and perilous Licences shall it capture. And in capturing shall it replicate, and in replicating shall it document, and in documentation shall it bring freedom, serenity and most cool froodiness to the earth and all who code therein."

Linus rose from meditation and created a tiny Operating System Kernel as the dream had foreshewn him; in the manner of RMS, he released the Kernel unto the World Wide Web for all to take and behold. And in the fulness of Internet Time the Kernel grew and replicated, becoming most cool and exceedingly froody, until at last it was recognised as indeed a great and mighty Penguin, whose name was Tux. And the followers of Linus took refuge in the Kernel, the Libraries and the Utilities; they installed Distribution after Distribution, and made sacrifice unto the GNU and the Penguin, and gave thanks to the spirits of Turing, von Neumann and Moore, for their deliverance from the hand of Microsoft. And this was the beginning of the Fourth Age, the age of Open Source.
```

For your convenience, there are **5 human names** in this article. You should also submit a writing note about their introductions. For your reference, the source of the above messages is from the following link.

https://www.linux.com/training-tutorials/gospel-tux/

1.2 Multiplication (20 pts)

Please write a program for a user to input two **integers**, where the range is from 0 to 99 and print the multiplication process.

```
$ ./hw0102
Please enter the first  number: 34
Please enter the second number: 12
    3 4
*)  1 2
-------
    6 8
  3 4
-------
  4 0 8
$ ./hw0102
Please enter the first  number: 5
Please enter the second number: 24
      5
+)  2 4
-------
    2 0
  1 0
-------
  1 2 0
```

If there is any invalid input, print an error message and terminate your program. For your simplicity, I promise that all inputs are integers and can be put in a 32-bits memory space.

1.3 Area (20 pts)

Given two distinct points A and B, you can easily build a line $\overline{AB}$. Please calculate the area bounded by $\overline{AB}$, x-axis and y-axis. Figure. 1.1 is an example.

```
$ ./hw0103
Please enter the point A (x,y): 4,1
Please enter the point B (x,y): 1,4
Area: 12.5
```

I promise that x, y are all 32-bit integers. For any error cases, you need to print an error message and terminate the program.

1.4 Bidding System for Contract Bridge (20 pts)

Do you know how to play the contract bridge game? Contract bridge, or simply bridge, is a trick-taking card game using a standard 52-card deck. In

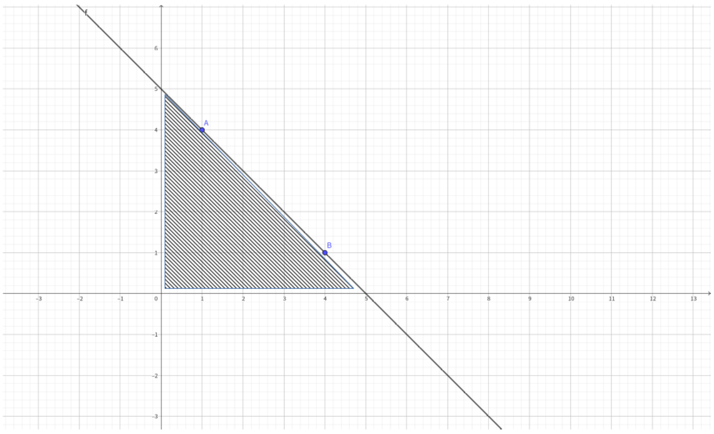

FIGURE 1.1: You need to calculate the marked area.

its basic format, it is played by four players in two competing partnerships, with partners sitting opposite each other around a table.

The game consists of a number of deals, each progressing through four phases. The cards are dealt to the players, and then the players **call** (or **bid**) in an auction seeking to take the contract, specifying how many tricks the partnership receiving the contract (the declaring side) needs to take to receive points for the deal. During the auction, partners endeavor to exchange information about their hands, including overall strength and distribution of the suits. The cards are then played, the declaring side trying to fulfill the contract, and the defenders trying to stop the declaring side from achieving its goal. The deal is scored based on the number of tricks taken, the contract, and various other factors which depend to some extent on the variation of the game being played.

The purpose of bidding during the auction phase of each deal is to disclose information which one’ s partner may employ in order to arrive at an optimal contract while concurrently contending with the opponents’ attempts to do likewise. A bidding system is a set of agreements about the meanings of the different bids that the players use. Each bid provides information about the hand’ s high-card strength and suit distribution based on hand evaluation techniques.

In this assignment, I want you to implement the opening biding rule for Modified Precision in Taiwan (中華精準制). The user inputs 13 cards and your program will output the opening choice. Do not worry, I will simplify the biding system and introduce the system for you.

First I will show the poker card encoding rule here:

- 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

Then, I will teach you how to compute High Card Points (HCP):

- A: 4 points.
- K: 3 points.
- Q: 2 points.
- J: 1 points.

The simplified Modified Precision in Taiwan biding system for the opening is as follows:

- 1♣: HCP $\geq$ 16.
- 1♢: 11 $\leq$ HCP $\leq$ 15 and you have at least 4 ♢ cards.
- 1♠, 1♡: 11 $\leq$ HCP $\leq$ 15 and you have at least 5 cards in that suit. If the numbers of these two suits are equal, call 1♠. This rule is prior to 1♢, 2♣, 2♢ rule.
- 1NT: 13 $\leq$ HCP $\leq$ 15 and is a balanced hand, which means 4-3-3-3[1]. This rule is prior to 1♢ rule.
- 2♣: 11 $\leq$ HCP $\leq$ 15 and you have at least 6 ♣ cards.
- 2♢: 11 $\leq$ HCP $\leq$ 15 and you have at least 5 ♣ cards without any ♢ card.
- 2♡: 8 $\leq$ HCP $\leq$ 10 with 6 ♡ cards.
- 2♠: 8 $\leq$ HCP $\leq$ 10 with 6 ♠ cards.
- 2NT: 22 $\leq$ HCP $\leq$ 24 and is a balanced hand, which means 4-3-3-3[2]. This rule is prior to 1♣ rule.
- 3♣, 3♢, 3♡, 3♠: 8 $\leq$ HCP $\leq$ 11 and you have 7 cards of the corresponding suit.
- 3NT: HCP $\leq$ 16 and has 7 cards of a major suit (♠, ♡) with A,K,Q.
- Pass: Otherwise.

There is an example for the following card:

- ♠: A, K, Q, 4

[1] One suit is four while others are three.

[2] One suit is four while others are three.

- ♡: 2, 3, 4
- ♢: K, 2, 3
- ♣: A, 6, 7

```
$ ./hw0104
1st card: 1
2nd card: 13
3rd card: 12
4th card: 4
5th card: 15
6th card: 16
7th card: 17
8th card: 39
9th card: 28
10 th card: 29
11 th card: 40
12 th card: 45
13 th card: 46
---
HCP: 16 pts
Suit: 4-3-3-3
The bidding choice : 1C
```

Note that if there is any problematic input, please print an error message and terminate the program. Note that the inputs may not be in order. For your convenience, currently you can ignore the following cases:

1. The input is not an integer.
2. Repeated cards.

1.5 How to Write an Email to TA? (20 pts)

Teaching assistants for the programming course are under a lot of stress.

As is well-known, the programming course at NTNU is very popular. Every year, more than a hundred students enroll in it (and possibly thousands more if we count online attendees). Unfortunately, due to limited resources, this prestigious course only has five hardworking teaching assistants. These assistants are also busy with their own courses and research projects, often having to work late into the night to complete all their tasks.

One of the primary responsibilities of the TAs is to respond to student emails. These emails are generally about homework queries or complaints about grading. While the TAs diligently reply to each email, the sheer volume of emails and often incomplete information in the queries lead to unnecessary back-and-forth, making it hard for the TAs to respond to all students in a timely manner.

Thankfully, GPT (Great Problem Triage) is smart enough to analyze the content of student emails and calculate a ”Good Question Index”. A higher index indicates that the email is likely a well-framed query. This means that the email contains all the necessary details, making it easier for TAs to understand and respond quickly. We don’ t need to know the specifics of this calculation.

Your task is to implement a portion of an automated email classification system. The parent program will input three parameters in sequence: sender’s email address, email subject, and Good Question Index. Your program will run tests on these inputs, output the results of each test, and finally provide a decision on assignment (or rejection) of the email.

1.5.1 Specifications

- **Email Addresses**:
  - Authorized Domains:
    1. ntnu.edu.tw
    2. gapps.ntnu.edu.tw
    3. csie.ntnu.edu.tw
    4. csie.cool
  - Domain Error Messages:
    - **local-part is invalid**: The local-part contains characters other than upper and lowercase English letters and numbers.
    - **domain is not authorized**: The email comes from an unauthorized domain.
- **Email Subject**:
  - Accepted Email Subjects:
    - Subject: [general] <title>
    - Subject: [hw<A>][p<B>] <title>, where A and B are positive integers less than 10.
  - Subject Error Messages:
    - **no category**
    - **title is empty**
- **Email Content Scores**:
  - Accepted Email Content Scores:
    - Non-negative integers

  - Content Score Error Messages:
    - **out of range**: The score is less than 0 or greater than 10^{19}.
    - **too low**: The score is within the range but less than the length of <title> multiplied by 10^{10}.

- **TA Assignment**:
  - Emails with the subject [general] are assigned to TA QB.
  - From the subject [hw<A>][p<B>], calculate C = (A x B) mod 5. Based on the value of C ranging from 0 to 4, the email is assigned to:
    - Kaname Madoka
    - Akemi Homura
    - Miki Sayaka
    - Tomoe Mami
    - Sakura Kyoko

1.5.2 Hint

You have good TAs. They prepare some references for you.

1. Addr-Spec Specification (RFC 5322 3.4.1)
   https://datatracker.ietf.org/doc/html/rfc5322#section-3.4.1
2. scanf(3) - Linux man page
3. Understanding scanf(”%*[a-z]@%n”, &readed)
   https://chat.openai.com/share/14677243-f62a-4a05-846b-0b8a13cac69c
4. Exit Status
   https://en.wikipedia.org/wiki/Exit_status

1.5.3 Examples

```
# Example 1: The email passes all tests and is assigned to TA Akemi Homura.
$ ./hw0105
Please enter the sender address      > ta@csie.cool
Please enter the email subject       > [hw1][p1] I need some help
Please enter the email content score > 200701090941
================================================================================
Sender Address Test : Passed
```

```
Email Subject Test  : Passed
Email Content Test  : Passed
--------------------------------------------------------------------------------
Assigned to Akemi Homura

# Example 2: The email fails the domain test; other tests are skipped, and the email is rejected.

$ ./hw0105
Please enter the sender address      > coolguy@gmail.com
Please enter the email subject       > [general] I want to do computer science
Please enter the email content score > 321
================================================================================
Sender Address Test : Failed, domain is not authorized
Email Subject Test  : -
Email Content Test  : -
--------------------------------------------------------------------------------
Rejected

# Example 3: The email passes the domain test but fails the subject test; the content test is skipped, and the email is rejected.

$ ./hw0105
Please enter the sender address      > ta@csie.cool
Please enter the email subject       > How to use "scanf" in my code?
Please enter the email content score > 123
================================================================================
Sender Address Test : Passed
Email Subject Test  : Failed, no category
Email Content Test  : -
--------------------------------------------------------------------------------
Rejected

# Example 4: The email passes the domain and subject tests but fails the content test; the email is rejected.

$ ./hw0105
Please enter the sender address      > ta@csie.cool
Please enter the email subject       > [general] How to use "scanf" in my code?
Please enter the email content score > 299999999999
================================================================================
Sender Address Test : Passed
```

```
Email Subject Test  : Passed
Email Content Test  : Failed, too low
--------------------------------------------------------------------------------

Rejected
```

1.6 Bonus: Makefile for Multiple files (5 pts)

Please read the following two codes.

```
// a.c
#include <stdio.h>

int main()
{
    printf( "Hello Kitty" )
    return 0;
}

// b.c
 #include <stdio.h>

int main()
{
    printf( "Hello World" );
    return 0;
}
```

I believe that you can see that **a.c** has an error. In this case, if you write a Makefile as follows, you will find that building **a.c** fails and **b.c** will not be built. How to modify this Makefile to make **b.c** be built even **a.c** fails. **You cannot switch the compiling order.**

```
$ cat Makefile
all:
  gcc a.c -o a
  gcc b.c -o b
$ make
gcc a.c -o a
a.c: In function ‘main’:
a.c:5:28: error: expected ‘;’ before ‘return’
    5 |     printf( "Hello Kitty" )
      |                           ^
      |                           ;
    6 |     return 0;
      |     ~~~~~~
make: *** [Makefile:2：all] 錯誤 1
```