C Programming II
2023 Spring
Homework 02

Instructor: Po-Wen Chi

Due: 2023.04.05 PM 11:59

Policies:

- **Zero tolerance** for late submission.
- **Plagiarism is not allowed.** Both source and copycat will be **zero**.
- You need to prepare a README file about how to make and run your program. Moreover, you need to provide your name and your student ID in the README file.
  - Your Name and Your ID.
  - The functional description for each code.
  - Anything special.
- Please pack all your submissions in one zip file.
- 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.

1 Matrix (20 pts)

You all know what matrix is, right? In this problem, I want you to design a structure for matrix and a set of utility functions.

```
typedef struct _sMatrix
{
  // Your Design
  // All elements are int32_t.
}sMatrix;
```

```

// Memory allocation for a m*n matrix. Fill zeros to all matrix element.
sMatrix * matrix_init( uint8_t m, uint8_t n);

// Set the element in m-th row and n-th column to value.
// Note that m and n start from 0.
// If error, return -1; otherwise, return 0;
int32_t matrix_set( sMatrix *pM, uint8_t m, uint8_t n, int32_t value );

// Print the matrix as follows
// 1 0
// 0 1
void matrix_print( const sMatrix *pM );

// A = B + C
// If error, return -1; otherwise, return 0;
int32_t matrix_add( sMatrix *pA, const sMatrix *pB, const sMatrix *pC );

// A = B * C
// If error, return -1; otherwise, return 0;
int32_t matrix_multiply( sMatrix *pA, const sVector *pB, const sVector *pC );

// A = A^T
// If error, return -1; otherwise, return 0;
int32_t matrix_transpose( sMatrix *pA);

// Determinant
// Determinant should be put to *pAns
// If error, return -1; otherwise, return 0;
int32_t matrix_det(const sMatrix *pA, int32_t *pAns);

// Free
// If error, return -1; otherwise, return 0;
int32_t matrix_free( sMatrix *pA);
```

You need to prepare **matrix.h** and TA will prepare **hw0201.c**. Of course, Makefile is your own business. Do not forget to make **hw0201.c** to hw0201 in your Makefile. If you use math.h, remember to use -lm in your Makefile.

2 Directory (20 pts)

I want you to develop a program to recursively show the files in the current directories.

```
$ ./hw0202
.:
1000px-SDL_Layers.svg.png            driver.png                  file_stream.
    png      lecture.10.File.out        lecture.10.File.toc
256px-SDL_Layers.svg.png             example                     io.mmap.png
          lecture.10.File.pdf          lecture.10.File.vrb
after.png                            example.zip                 lecture.10.
    File.aux  lecture.10.File.snm         NTNU3.jpg
before.png                           figure.odp                  lecture.10.
    File.log  lecture.10.File.synctex.gz  socket.png
```

```
bmp_6x6_24bit_xxd_vim_annotated.png  file_position_pointer.png  lecture.10.File.nav  lecture.10.File.tex  u1revenge-sdl2-screenshot.png

./example:
example.10.binary  example.10.fprintf  example.10.fseek  example.10.mmap  example.10.redirection  example.10.sdl.bmp.example
example.10.fd  example.10.fscanf  example.10.gtk3  example.10.read  example.10.rewind  example.10.sdl.example

./example/example.10.binary:
bmp_read_v1.c  bmp_read_v2_error.c  bmp_write_gray.c  maldives162.bmp  maldives.bmp  maldives_out.bmp
bmp_read_v2.c  bmp_write.c  Makefile  maldives16.bmp  maldives_gray.bmp  maldives_ref.bmp

./example/example.10.fd:
file_size_v2.c  Makefile  maldives.bmp  maldives_copy.bmp  mycopy.c

./example/example.10.fprintf:
main.append.c  main.c  main.v2.c  Makefile

./example/example.10.fscanf:
main.c  Makefile  score.txt

./example/example.10.fseek:
file_size.c  lyrics.txt  Makefile  maldives.bmp  score.txt

./example/example.10.gtk3:
gtest.c  Makefile  maldives.bmp

./example/example.10.mmap:
lyrics_original.txt  lyrics.txt  Makefile  test.c  upper_lower_api.c  upper_lower_api.h

./example/example.10.read:
main.c  Makefile

./example/example.10.redirection:
abs.c  hello.c  hello.v2.c  Makefile  number.txt

./example/example.10.rewind:
main.c  Makefile  score.txt

./example/example.10.sdl.bmp.example:
bmp_display.c  Makefile  maldives.bmp

./example/example.10.sdl.example:
example.c  Makefile  Paper-Mario-icon.png
```

Note that you need to **colorize all directories with blue**.

I know that I have not taught you how to get directory information but I believe that you can learn the related functions yourself. I will give you a small hint: Please **man opendir** as your first step. Remember! These related APIs are OS dependent and our TAs will test

Figure 1: FIFA Game.

your program in Linux.

3 Bible (20 pts)

The Bible is a collection of religious texts, writings, or scriptures sacred in Christianity. It is is widely considered to be the best-selling book of all time. This time, I want you to develop a search function for the English Bible. I provide you a text file, **bible.txt**, which contains all verses. The searching process should be case insensitive. Your program should work as follows:

```
$ ./hw0203
Please enter the search target: in the beginning
Found 1 time(s)
1. Gen 1:1 In the beginning God created the heavens and the earth.
```

Note that the above is just an example and I do not guarantee the search correctness.

4 FIFA Manager (20 pts)

FIFA, also known as FIFA Football from August/September 2023, is a series of association football video games developed and released annually by Electronic Arts under the EA Sports label[1]. Figure 1 is the game picture. FIFA Game series have football player data in detail. This time, I want you to develop a program to build the best team from the given dataset.

The team construction policy is as follows:

1. The user picks eleven positions from the following list:

[1]On 10 May 2022, it was announced that EA and FIFA's partnership of 30 years would come to an end

- ls, st, rs, lw, lf, cf, rf, rw, lam, cam, ram, lm, lcm, cm, rcm, rm, lwb, ldm, cdm, rdm, rwb, lb, lcb, cb, rcb, rb, gk.
- Positions are separated by space.
- For your simplicity, you do not need to consider the rationality of the squad, but the team must has **one and the only one gk**. None gk or More gks will be treated as wrong inputs.
- For any error inputs, show a warning message and make the user re-input again.

2. For each selected position, the user inputs the budget. **I promise the budget format is correct**.

3. Show the best team you can find.

   - You can find the abilities of a player at all positions from the dataset.
   - Find the best player of the position in your budget. Not the overall ability.
   - **A player can only be chosen once**. The earlier position has higher priority.
   - For more than one player have the same ability, choose the **cheapest** one. If they have the same value, choose the one with the smallest **sofifa_id**.
   - "89+5=94"

```
$ ./hw0204
Please enter the dataset: players_20.csv
Please enter the squad: st lf rf cam ldm rdm lwb rwb cb cb gk
Budget:
    st: 100000000
    lf: 100000000
    ...
Team:
    st: L. Messi/FC Barcelona/94/95500000/560000
    ...
```

The output message includes

- short_name
- club_name
- ability at the given position position
- value
- wage

Note that **gk** must be the last one of the team. If you cannot form a team with the given requirements, print an error message and terminate your program.

5 Follow-up of Software Engineering Course (20 pts)

SubaRya , CSIE major at NTNU , served as the front-end team leader of Group 2 in Professor Chi.'s software engineering course, using Git , revision control, to manage source code of the project he was responsible for. After the project development was completed, SubaRya was very interested in the contributions made by members of the front-end team he led. Therefore, he used the `git log --stat` command to query the dev branch. To his surprise, there were over 700 commit records! As a result, SubaRya provide you **frontend.txt**[2], git log file, and want you to write a program to help him organize the data. And then **write** all retrieved information into the **contribution.txt** file. The data he needs is as follows:
Hint: This link (https://reurl.cc/gZeYgR) is the basic knowledge about git log. You can take a look if you don't know what git log is.

```
// ContributionCalculate.h
// first function
void searchCommitInformationByHashVal(const char *hashVal)
// second function
void searchMonthlyContribution(const char *monthAbbrev)
```

1. **searchCommitInformationByHashVal**

   Please retrieve the following information by querying the hash value (using the first 8 characters) of each commit:

   - (<hash value>)
   - Author's name
   - Author's email
   - Time of this commit (year/month/day/hour:minute:second)
   - The following changes caused by this commit:
     - Number of file changed
     - Number of insertions
     - Number of deletions

   Note that if there are 2 or more search results, please write "(<hash value>) More than two search results" into the file. If no information is found, please write "(<hash value>) Not found" into the file.

2. **searchMonthlyContribution**

   Please retrieve the following information by querying the first three characters (Table 2.1) of month in English:

   - The contribution of each author in that month, including:
     - (<Month Abbrev.>)

[2]For your simplicity, the file name will always be **frontend.txt**.

- Author’ s name
- Total number of commits
- Total number of file changed
- Total number of insertions
- Total number of deletions

Note that All authors involved in the project should be included in the calculation. And if any author did not contribute any commit in that month, please also write their contribution as 0 in the file.

月份	Month	Abbreviation	月份	Month	Abbreviation
一月	January	Jan	七月	July	Jul
二月	February	Feb	八月	August	Aug
三月	March	Mar	九月	September	Sep
四月	April	Apr	十月	October	Oct
五月	May	May	十一月	November	Nov
六月	June	Jun	十二月	December	Dec

Table 1: Table 2.1

```
// example of hw0205.c
...

searchCommitInformationByHashVal("8b120821");
searchCommitInformationByHashVal("1a712222");
searchCommitInformationByHashVal("ca50f778");
searchMonthlyContribution("Sep");
return 0;
```

```
$ ./hw0205
// generate contribution.txt
// the following is the example of contribution.txt content
(8b120821)
- Howard Guo
    - 50100922+toto6038@users.noreply.github.com
    - 2022/10/03/22:09:16
    - 1 file changed
    - 53 insertions
    - 0 deletions

(1a712222) Not found

(ca50f778) More than two search results

(Sep)
```

```
- Birkhoff Lee
    - 2 commits
    - 2 file changed
    - 115 insertions
    - 0 deletions
- Howard Guo
    - 0 commits
    - 0 file changed
    - 0 insertions
    - 0 deletions
- KutsunaSubaRya
    - 0 commits
    - 0 file changed
    - 0 insertions
    - 0 deletions
```

You need to prepare **ContributionCalculate.h** with corresponding C codes and TA will prepare **hw0205.c**. Of course, Makefile is your own business. Do not forget to make **hw0205.c** to hw0205 in your Makefile.

6 Bonus: Bit Operation (5 pts)

I want to write a program to display a 32-bit integer in the binary form. So I write the following code. However, this code has some problem.

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

int main()
{
    int32_t number = 0;

    scanf( "%d", & number );

    int32_t bit = 1;
    bit = bit << 31;

    for( int i = 0 ; i < 32 ; i++ )
    {
        if( bit & number )
            printf( "1" );
        else
            printf( "0" );
        bit = bit >> 1;
    }
    return 0;
}
```

Please explain the reason of the problem of this code and show how to fix it.