SEEM3460/ESTR3504 (2014)
Programming Assignment 1

Due on Nov 3 noon (11:59AM), 2014

General Information

- ✧ 30% or more mark penalty for uninformed late submission.
- ✧ You must follow the guideline in this file, otherwise there will be a mark penalty.
- ✧ You should develop your program under Linux environment.
- ✧ Your assignment will be compiled by gcc and executed in Linux.

Faculty Guideline for Plagiarism

If a student is found plagiarizing, a very heavy punishment will be imposed. The definition of plagiarism includes copying of the whole or parts of programming assignments from other's work or the Web. The penalty will apply to both the one who copies the work and the one whose work is being copied.

Problem Overview

The objective of this assignment is to write a program to perform a magic show. You **MUST ONLY USE** the provided module (display.o) for handling the graphical user interface (GUI). You must follow the specification below.

Problem Specification

Flow of the magic show

In the lecture, you should have seen a full demo of the expected program. The texts displayed in the program are listed in message.txt that is given to you. You must keep the logical flow of the magic. In the specified points, you should display some cards on the window. You should display cards according to the requirement in the "Card Display Randomness" session below.

Card Display Randomness

In this magic, the trick is to hide all cards displayed in the first time. Here are the requirements on the card displays:

1. You must show **6 cards** in the **first** display and **5 cards** in the **second** display
2. You must **not repeatedly display any card** in the two card displays
3. The card that you never display must be randomly selected from all 12 picture cards (this means by repeatedly running your program, every card will be the hidden card in some trials)
4. The color of the cards shown must be balanced (i.e., you must display 3 red cards and 3

black cards in the first display)

5. The letter of the cards shown must be balanced (i.e, you must display 2 Jack’s, 2 Queen’s and 2 King’s in the first display)

Provided Module – display.o

In this assignment, you are required to show everything on the GUI. To simplify your work, a module – display.o has been written for you. Here is the content of display.h:

```
#define MAX_CARD_TO_HOLD 6

void init_display(int *argc, char *argv[]);
void display_text(char *message, int x, int y, int size);
void display_card(int position, int card_id);
void display_david(int x, int y);
void display_card_clear();
void move_card_set(int x, int y);
void wait_for_a_click();
void exit_display();
void wait_time(double seconds);
```

Function	Description	Parameters
init_display	Initialize the window	You need not manage argc and agrv. Just pass the same value as you get from main
display_text	Show a message	message: the message to show x, y: the position offset of the text size: text size (16 is suggested)
display_card	Show a card	position: 0 to 5 index of cards in a hand card_id: a number from 0 to 11. You should refer to the filename of the card pictures
display_david	Show photo of David When you use this, all card display will be cleared	x, y: the position offset of the display
display_card_clear	Clear all card display	
move_card_set	Set the position of the card set	x, y: the position offset of the display
wait_for_a_click	Finish of part one and wait for a click to continue with action_on_click()	
exit_display	Close the window	
wait_time	Wait for a time while keeping the GUI display	seconds: number of seconds you want to wait

Two-Part Program Flow

Since the program needs to wait for a click to continue, the whole program flow is split into two parts. The first part is main() and the second part is action_on_click(). You **MUST NOT** use any other names for these two functions. At the end of main(), you should call the function wait_for_a_click(). At the end of action_on_click(), you should call the function exit_display() to end the display.

Compilation Constraint

Since only display.h and display.o will be given, you are not allowed to change anything for the display module. All your work **MUST ONLY** be done in **magic.c** and the final program must be able to be compiled by the make file - make_magic. This make file is given to you but you should revise it for your own compilation test so that it will not need display.c be present.

Submission

Please follow the submission procedures, so that we could ensure your assignment is received properly.

- Only submit your assignment **USING YOUR OWN** UNIX account!
- Type in the following command to submit your source code file (magic.c):
  > ~seem3460/submit asgn1 magic.c
- When the program asks your student ID, type in your FULL student ID as follows:
  What is your full student ID? 1155001122
- Then the program will summarize your personal information and ask for your confirmation. Type in “Y” to confirm your information; otherwise, type “N” to exit the program and start again
  Your student ID: 1155001122; Your account name: lwai
  Is the above information correct? (Y/N)   Y
- Then you should see the following message:
  Connecting to SEEM3460 Submission Server...
  sftp channel opened and connected.
  Uploading...
  Done! Thanks.
- If you see any additional error message apart from the above, then your assignment may not have submitted properly. Try to submit once again. If the problem persists, log down the error message and send an email to *kli@se.cuhk.edu.hk* to clarify.
- Multiple submissions are not encouraged, but allowed when necessary. Only the last submission will be graded.

Additional Requirements for ESTR3540

1. You have to read the sequence of messages from the message file (e.g., “message.txt”), instead of hardcoding the messages in your program.

2. If you use another message file (i.e., not use “message.txt”), you have to submit it as well as your source code file.
3. (*Optional*) Each time you may read a set of one or multiple lines of messages, which can be separated by a blank line.
4. (*Tips*) The following functions maybe useful for you (you can google for the details):

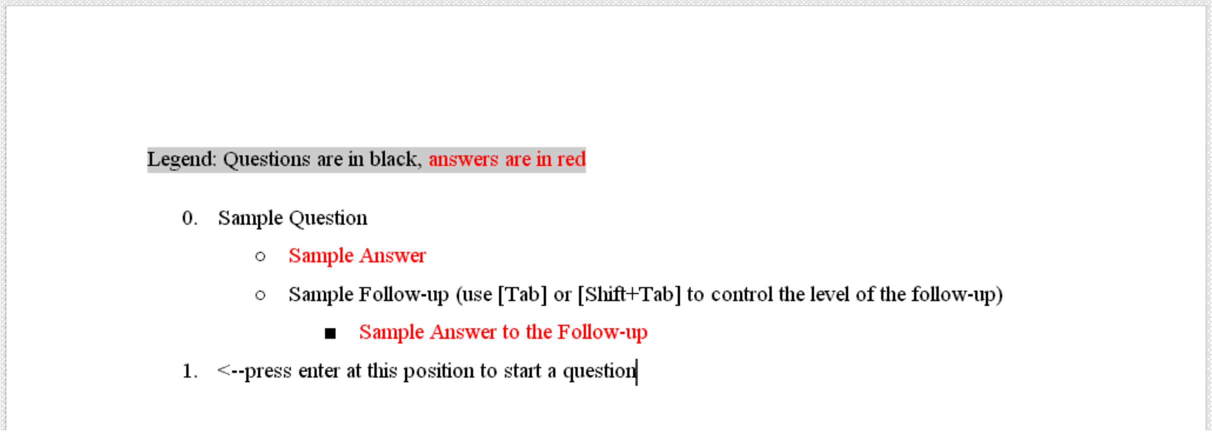
    3.1. To open a file: **FILE *fopen(const char *filename, const char *mode)**

    3.2. To read a line: **char *fgets(char *str, int n, FILE *stream)**

    3.3. To close a file: (you can **google** by yourself)

Question & Answer

With any questions about the assignment, please check the following Google Document for Q&A first and write only new questions that have not been asked yet:
https://docs.google.com/document/d/1SMBxjx13ART9b5NKlg8JHvpxV6VKgP86eTw3jrXMx_I/edit?pli=1



Legend: Questions are in black, answers are in red

0. Sample Question
    ○ Sample Answer
    ○ Sample Follow-up (use [Tab] or [Shift+Tab] to control the level of the follow-up)
        ■ Sample Answer to the Follow-up
1. <--press enter at this position to start a question