ICS 103 (Computer Programming in C)
Lab Project for 101 Term

Introduction:

The purpose of this project is to develop and test your programming skills by asking you to design and implement a software tool that solves a small-scale problem. This project should give you the taste of how large-scale software systems are designed and implemented. You should expect writing between 200 – 300 lines of code involving a blend of most programming constructs you learned in this course, e.g. Loops, if Constructs, Functions, File Processing and Arrays.

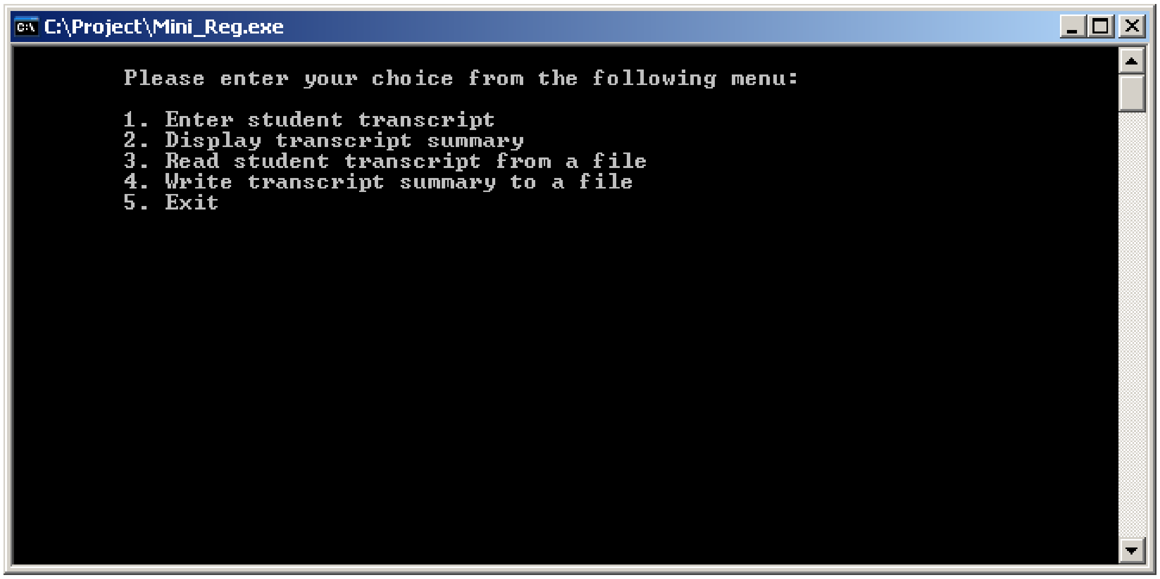
Although we didn't cover all topics needed to finish this project at this time (the time of project assignment), the project is designed in a way that it could be divided into phases, and you should be able to start working on the first phase by now.

Project Description:

This project is about implementing a mini-registrar's system. In this project, you will learn and practice the following programming skills:

1. Designing an interactive software system
2. Processing of collected data
3. Parsing and reading data from files
4. Saving processed results and outcomes into files

Upon execution, your project should display the following menu:



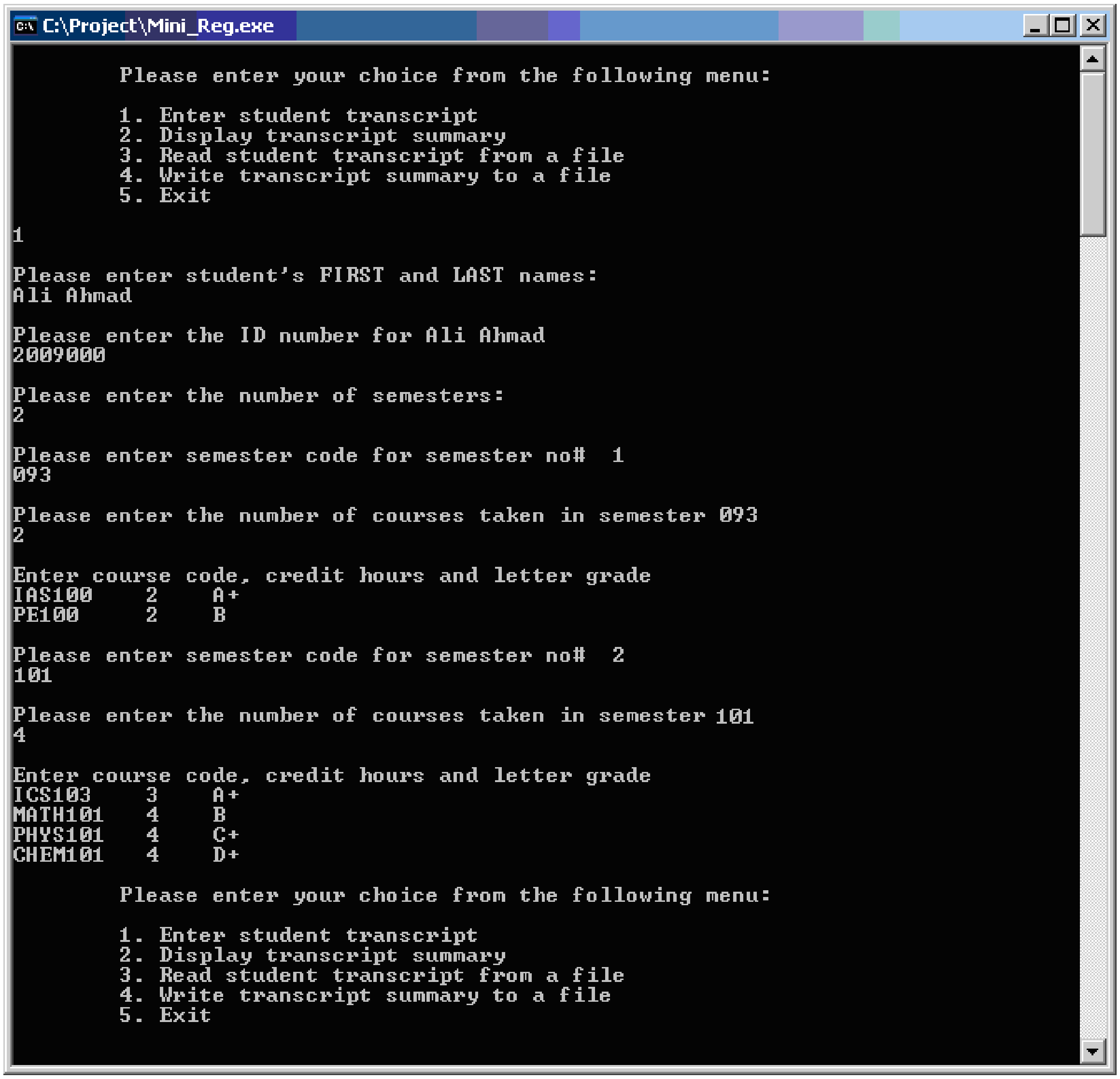
```
        Please enter your choice from the following menu:

        1. Enter student transcript
        2. Display transcript summary
        3. Read student transcript from a file
        4. Write transcript summary to a file
        5. Exit
```

When the user selects an option from the menu, he will be directed to some actions (will be defined shortly), after that, the menu should be displayed again. The menu should repetitively display until the user chooses that he no longer wants to use the program by choosing option 5 (Exit).

Option 1: Enter student transcript:

When this option is selected, your program should interact with the user to get data directly, the following screen snapshot is an example:



```
        Please enter your choice from the following menu:

        1. Enter student transcript
        2. Display transcript summary
        3. Read student transcript from a file
        4. Write transcript summary to a file
        5. Exit

1

Please enter student's FIRST and LAST names:
Ali Ahmad

Please enter the ID number for Ali Ahmad
2009000

Please enter the number of semesters:
2

Please enter semester code for semester no#  1
093

Please enter the number of courses taken in semester 093
2

Enter course code, credit hours and letter grade
IAS100    2    A+
PE100     2    B

Please enter semester code for semester no#  2
101

Please enter the number of courses taken in semester 101
4

Enter course code, credit hours and letter grade
ICS103    3    A+
MATH101   4    B
PHYS101   4    C+
CHEM101   4    D+

        Please enter your choice from the following menu:

        1. Enter student transcript
        2. Display transcript summary
        3. Read student transcript from a file
        4. Write transcript summary to a file
        5. Exit
```

Once this option is selected, your program should ask the user to enter his first and last name (assume that first and last name are no longer than 10 characters each).

After that, the program should ask the user to enter his ID number, followed by the number of semesters in which the user has finished.

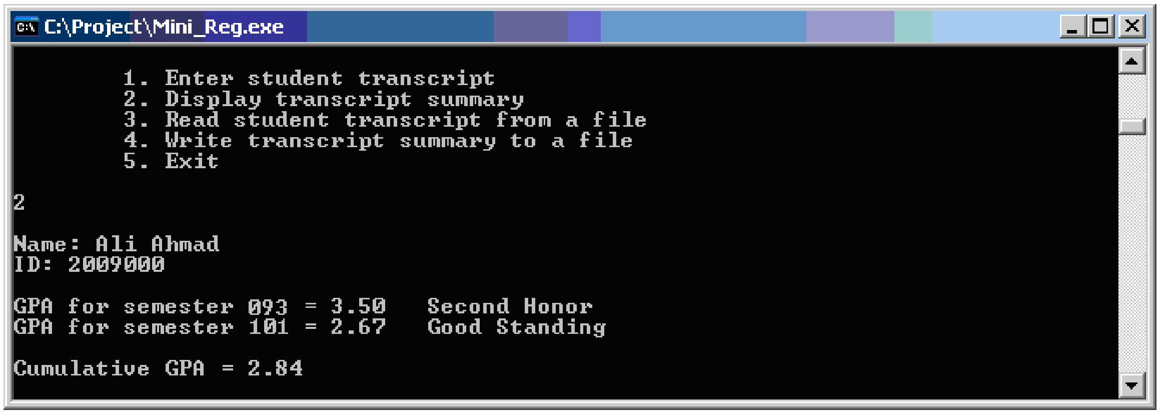
Next, the program will ask for information for every semester in which the user is enrolled in. This includes the semester code, which is the three numbers that identifies a given semester at KFUPM e.g. 092, 093, 101 … etc. After that, the program will ask the user to enter the number of courses attended for that semester, then for every course, the user is asked to enter information, this includes the course code (the course name identified by KFUPM, e.g. ICS103), then the number of credit hours for that course and then the letter grade obtained. This is repeated for the number of semesters that the user entered in the beginning.

Option 2: Display Transcript summary:

NOTE: This option should work only when there is information entered already through option 1 or option 3, otherwise, upon choosing this option without entering data first, your program should display a proper error message asking the user to enter transcript information first through option 1 (manually) or option 3 (from a file).

Once this option is selected and there is already entered data, your project should display a report that shows some analysis of collected data representing a student academic record. You should print the current student full name, ID number, semesters' list with the GPA for every displayed semester. Also you should print an academic standing note on every semester, indicating whether a student is a first honor, second honor, third honor, good standing or in academic warning in that semester. Finally, you should print the cumulative GPA of the current student.

The following screen is a snapshot example:



```
        1. Enter student transcript
        2. Display transcript summary
        3. Read student transcript from a file
        4. Write transcript summary to a file
        5. Exit

2

Name: Ali Ahmad
ID: 2009000

GPA for semester 093 = 3.50   Second Honor
GPA for semester 101 = 2.67   Good Standing

Cumulative GPA = 2.84
```

You should print academic standing note based on the following criteria:

GPA	NOTE
3.75 ≤ GPA	First Honor
3.5 ≤ GPA < 3.75	Second Honor
3.0 ≤ GPA < 3.5	Third Honor
2.0 ≤ GPA < 3.0	Good Standing
GPA < 2.0	Academic Warning

For the above example, the student above has a GPA of 3.50 in 093, this is why he got second honor, and he got a GPA of 2.67 in 101, this is why we see that he is in a good standing.

For calculating GPA, use the following table to convert a letter grade to equivalent point:

Letter Grade	Points
A+	4.0
A	3.75
B+	3.5
B	3.0
C+	2.5
C	2.0
D+	1.5
D	1
F	0

GPA calculation for a given semester is found by the following:

1. For every course in a given semester, multiply credit hours of a course by the point value of the grade obtained in that course., then find the summation of all those multiplications. For example, the student above has the following courses and grades for semester 101:

ICS103	3	A+
MATH101	4	B
PHYS101	4	C+
CHEM101	4	D+

   For ICS103, we need to multiply 3 (the credit hours of ICS103) by the point value of A+, which is 4.0 (taken from the above table), in this case the result will be: 3 * 4 = 12.0.

   For MATH101, we need to multiply 4 (the credit hours of MATH101) by the point value of B, which is 3.0 (taken from the above table), in this case the result will be: 4 * 3 = 12.0.

   We do the same thing for PHYS101 and CHEM101 and we will get the following results respectively: 4 * 2.5 = 10 and 4 * 1.5 = 6.

   Finally, we sum all the results that we got from above multiplications:

   12.0 + 12.0 + 10.0 + 6.0 = 40

2. Finally, we divide the sum that we found in the previous step by the total number of credit hours registered in that semester. For example, the GPA for semester 101 of the above student is calculated as follows:

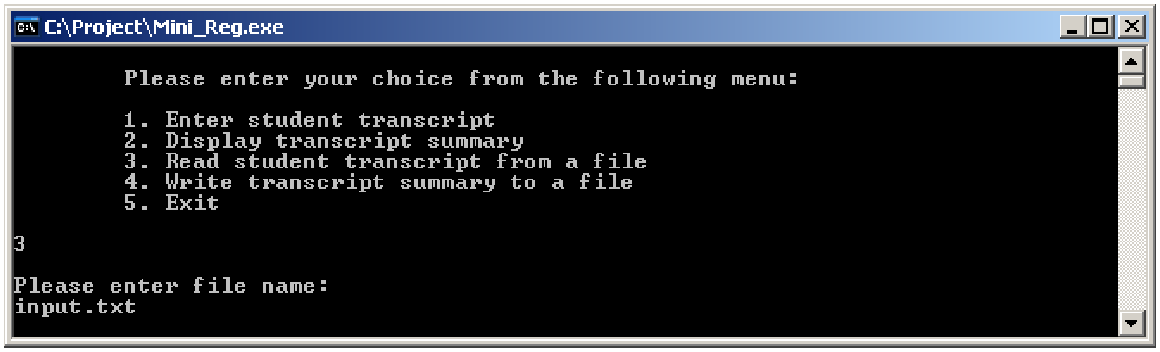
   GPA = 40/(3+4+4+4) = 2.67

(Note that 3+4+4+4 represents the summation of credit hours registered in 101 semester)

The cumulative GPA is calculated the same way, but instead of applying the formula for one semester, you should apply it for all semesters. (i.e. sum the multiplication of all courses in all semester by their grade point equivalents, then divide over the total number of credit hours that the student has finished in all semesters).

<u>Option 3: Read a student transcript from a file:</u>

Once this option is selected, your program should ask the user to enter a filename from which student transcript will be parsed and read:



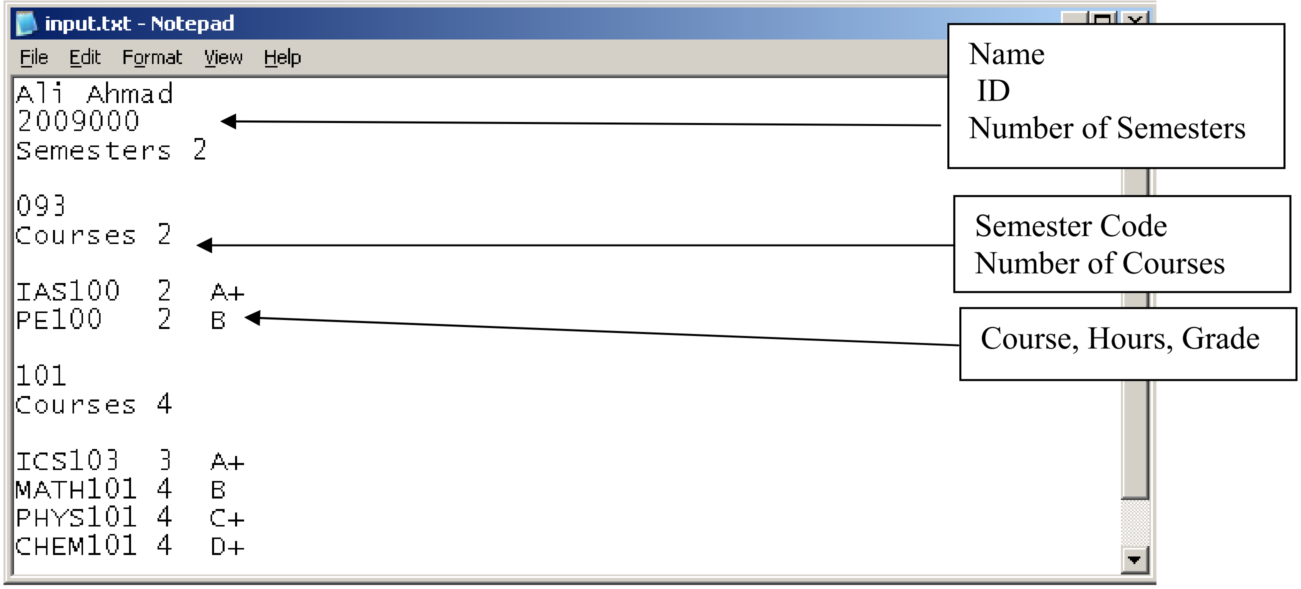
```
        Please enter your choice from the following menu:

        1. Enter student transcript
        2. Display transcript summary
        3. Read student transcript from a file
        4. Write transcript summary to a file
        5. Exit

3

Please enter file name:
input.txt
```

The input file must have a pre-defined format, such a format is explained in the following sample file "input.txt": (Note: the file name could be any name, not only "input.txt")



```
Ali Ahmad
2009000
Semesters 2

093
Courses 2

IAS100  2  A+
PE100   2  B

101
Courses 4

ICS103  3  A+
MATH101 4  B
PHYS101 4  C+
CHEM101 4  D+
```

Name
ID
Number of Semesters

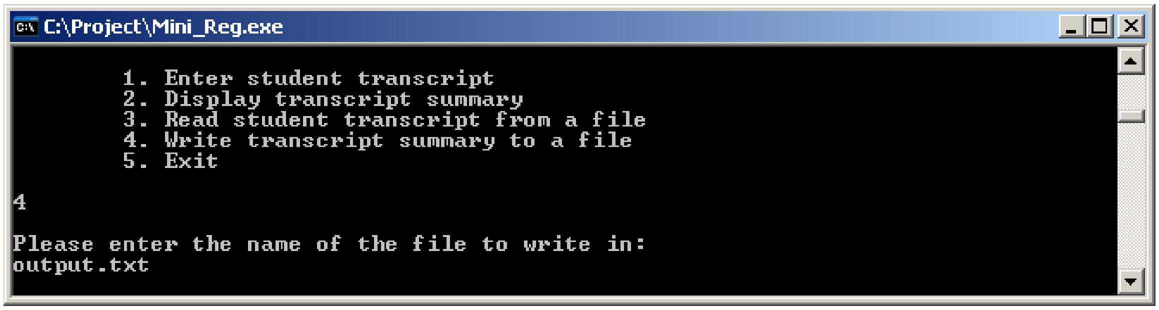
Semester Code
Number of Courses

Course, Hours, Grade

Option 4: Write transcript summary to a file:

NOTE: This option should work only when there is information entered already through option 1 or option 3, otherwise, upon choosing this option without entering data first, your program should display a proper error message asking the user to enter transcript information first through option 1 (manually) or option 3 (from a file).

In this option, you should dump the same report that your program displays in option 2 into a file of the user’s choice, your project should prompt the user for a filename like the following example:

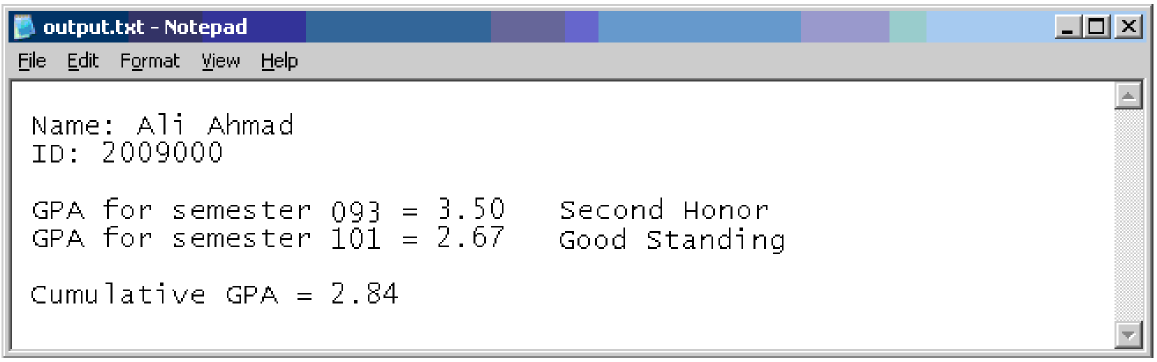


```
        1. Enter student transcript
        2. Display transcript summary
        3. Read student transcript from a file
        4. Write transcript summary to a file
        5. Exit

4

Please enter the name of the file to write in:
output.txt
```

After finishing the write process, the output file should contain the student report. For example, here is the content of the file "output.txt" after writing:



```
Name: Ali Ahmad
ID: 2009000

GPA for semester 093 = 3.50   Second Honor
GPA for semester 101 = 2.67   Good Standing

Cumulative GPA = 2.84
```

Option 5: Exit:

When this option is selected, your program should terminate execution and exit.

Important Notes:

- This is a team project, every team consists of two students at most.
- Each team or group is expected to submit a unique implementation of this project. Although sharing ideas between different group members is permissible, copying code is not allowed. Any similarity between submitted projects will result in an F grade in the course for all related students, and their names will be reported to the ICS chairman, Dean of CCSE college and Dean of students affairs.

- The weight of this project is 4% out of your total course grade. It is not necessary that all members of a team will get the same grade (grade will be based on individual contribution and demonstration of understanding the whole project). In order to score full mark in this project, you should do a significant contribution to it, and show the instructor that you understand all parts of the project even those done by other team members.

- You should submit a project that compiles and executes, otherwise you will be assigned a reduced grade.

- This project is to be submitted into phases with assigned deadlines according to the following schedule:

Phase	Weight	Deadline	Needed Topics
Menu Design & Option 5	10%	04/DEC/2010, 1:50PM	While Loop IF Constructs
Option 1	15%	18/DEC/2010, 1:50PM	Functions Do Loop Arrays
Option 2	35%	01/JAN/2011, 1:50PM	Functions Do Loop Arrays
Option 3 & 4	40% (25% + 15%)	15/JAN/2011, 1:50PM	File Processing

NOTE: Submission is expected at your lab time within scheduled period for every phase. Late submissions will not be accepted.

- The executable file of this project is available for your reference, try to make your project as close as possible to this executable file. Your grade will be relative to the closeness of your project to this executable file.

- Sample input and output files are also included for your reference.

- Grading will consider proper documentation and meaningful variable names. The more readable your program is, the better grade you will score. Try to be as clear as possible, avoid redundancy and unneeded repetitions of written code.

- Make your code modular by using subprograms (Functions) that divide and conquer the complexity of the given project.