1. **Problem statement:**

The program should be able to create a database which can store students’ information including name, student ID, email address, 6 assignment scores and an average mark. The database should be stored in a text file. The user can either generate a new file or open an existing file. After the user opens a file successfully, the program should provide three operations including adding, searching and sorting the information of students. The program should also be able to detect whether the input is correct or not and let the user enter the input again if not.

1. **Analysis:**

***Inputs:***

An integer number for choosing to create a new database or to open an existing database, the file name (character string), an integer number for choosing to add, search, sort the information of students or quit.

Adding [Family name (character string), Given name (character string), student ID (integer number), email address (character string), 6 assignment scores (integer number), an average mark (float number)], an integer number for choosing to return or quit

Searching [student ID (integer number)], an integer number for choosing to return or quit

Sorting [an integer number for choosing to return or quit]

***Outputs:***

To display the result when user enters the file name, the options provided for users to select, the information needed for users to input, the information of one student who is being searched, the entire content of database when being sorted.

***Additional requirements or constraints:***

To detect whether the input is correct or not and let the user enter the input again if not.

1. **Design:**

Algorithm (inside the main function)

1. Declare a file pointer pfile, an integer a and a character string b
2. Ask user to enter 1 or 2 representing the choice and store it as a.
3. Use if statement. If a=1, ask user to enter the file name and open it.

If a=2, ask user to enter an existing file name and open it with ’r’ mode. If failed opening, exit the program. Else, close the file and reopen it with ‘a+’ mode.

1. Enter into function menu.

Algorithm (outside the main function)

First declare a structure type named information including Family name (character string), Given name (character string), student ID (integer number), email address (character string), 6 assignment scores (integer number), an average mark (float number). Then declare seven functions named menu, add, search, sort, input1, input, option.

**The menu function**

1. Get the file’s address and declare a pointer for it
2. Use switch statement for users to select options
3. enter into different functions (add, search, or sort) based on the input

**The add function**

1. Get the file’s address and declare a pointer for it
2. move the file pointer to the end of file
3. Declare an information structure named IN
4. Ask user to input information and store it in IN
5. Use fwrite to put information to the file
6. Use if statement and the right return value of fwrite (should be 1 if succeed) to detect the result
7. Enter into function option

**The search function**

1. Get the file’s address and declare a pointer for it
2. Declare an array of information structure named IN[100]
3. Ask user to input a student ID and store it for comparing
4. Move the file pointer to the start of the file
5. Use fread and for loop to read the entire data
6. Compare the input with each student ID in the file
7. If there is a same pair, display the information of that student
8. If not, tell user that it is not in the database
9. Enter into function option

**The sort function**

1. Get the file’s address and declare a pointer for it
2. Declare an array of information structure named IN[100] and an information structure named t
3. Move the file pointer to the start of the file
4. Use fread and for loop to read the entire data
5. Fread returns 1 if it works successfully so that it can count the number of student’ information
6. Use 2 for loops to compare the student ID. If the former is bigger, exchange each other.
7. use for loop to display the entire database
8. enter into function option

**The input function**

1. Declare variables
2. Use while loop for circulation
3. Use if else statement and the return value of scanf to decide whether the input is correct. If not, ask the user to enter again. If correct, return the input value and jump out off the loop.

**The input1 function**

1. Declare variables
2. Use while loop for circulation
3. Use if else statement and the return value of scanf to decide whether the input is correct. If it is not an integer, ask the user to enter again. If it is integer but not 1 or 2, ask the user to enter again. If correct, return the input value and jump out off the loop.

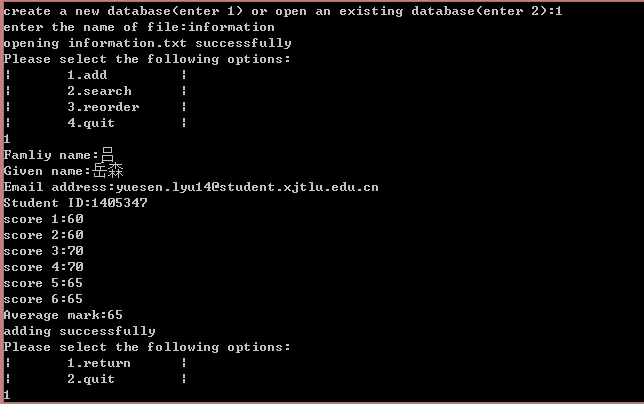
**The option function**

1. Use switch statement for users to select options
2. If user enter 1, enter into function menu
3. If user enter 2, close the file and exit the program

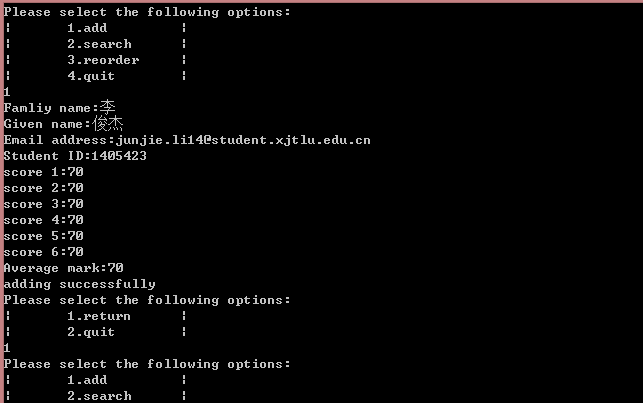
**4. Implementation**: see C code in file 1405347\_5-1.c with comments.

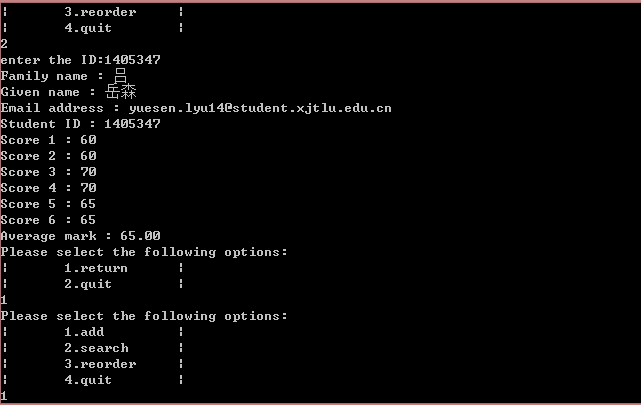
**5. Testing:**

The C program was tested by carrying out a set of experiments and the C program output was verified successfully. However, there still exists a limitation. If user selects to create a new database and then he enters an existing file name, the original file will be overwritten, which means the data is lost. One possible solution can be to use fopen with ‘r’ mode to detect whether the file is existing or not. If existing, ask user to enter another file name. If not, close the ‘r’ mode file and reopen it using ‘w+’ mode. Some screenshot of testing are shown below. For instance,

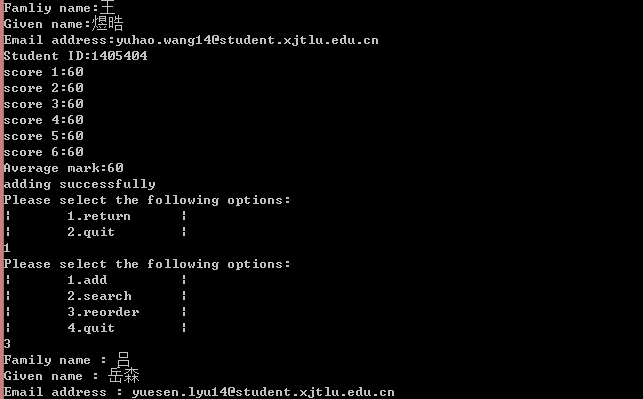


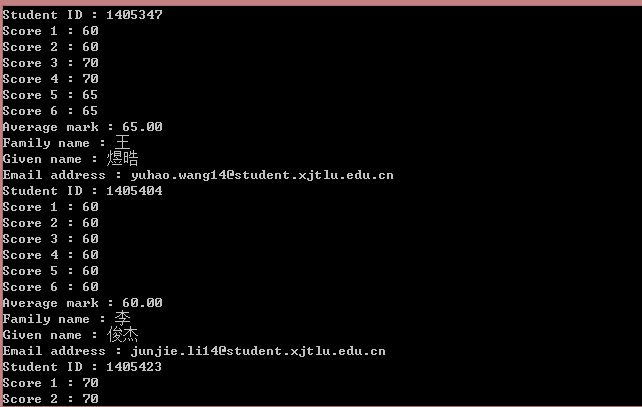
First add two students’ information

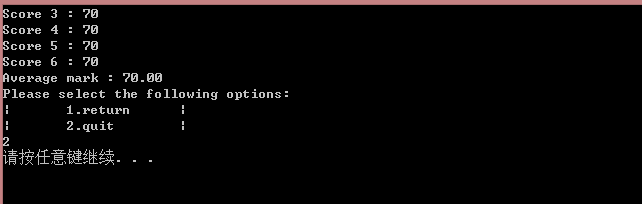




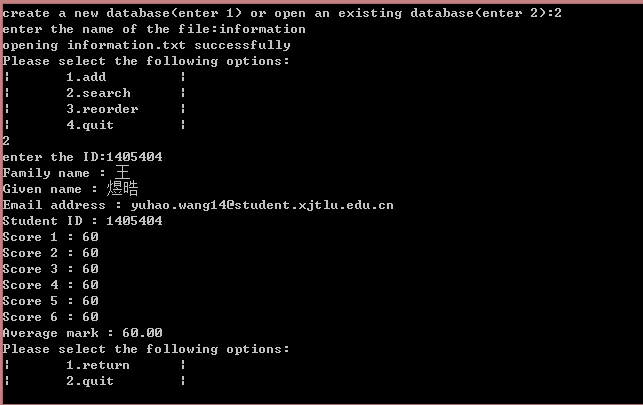
Search the information by entering the student ID







Add the third student‘s information. Sort and Print whole database by student ID.



Open an existing file and search a student’s information.

The following screenshots show what happens if the users enter the wrong input.



