

UNIVERSITI TUNKU ABDUL RAHMAN

ACADEMIC YEAR 2024/2025



Wholly owned by UTAR Education Foundation
(Co. No. 578227-M)
DU012(A)

UCCD 1004 PROGRAMMING CONCEPTS AND PRACTICES**ASSIGNMENT 2**

Group No.	22		
	Member 1	Member 2	Member 3
Name:	WONG WEI QI	LIANG TIM LOK	LIN, GUO YI
Student ID:	2302302	2301094	2301206
Programme:	CS	CS	CS
Email:	wqwong@lutar.my	timlok0319@lutar.my	linguoyi124@gmail.com

Content of Report:

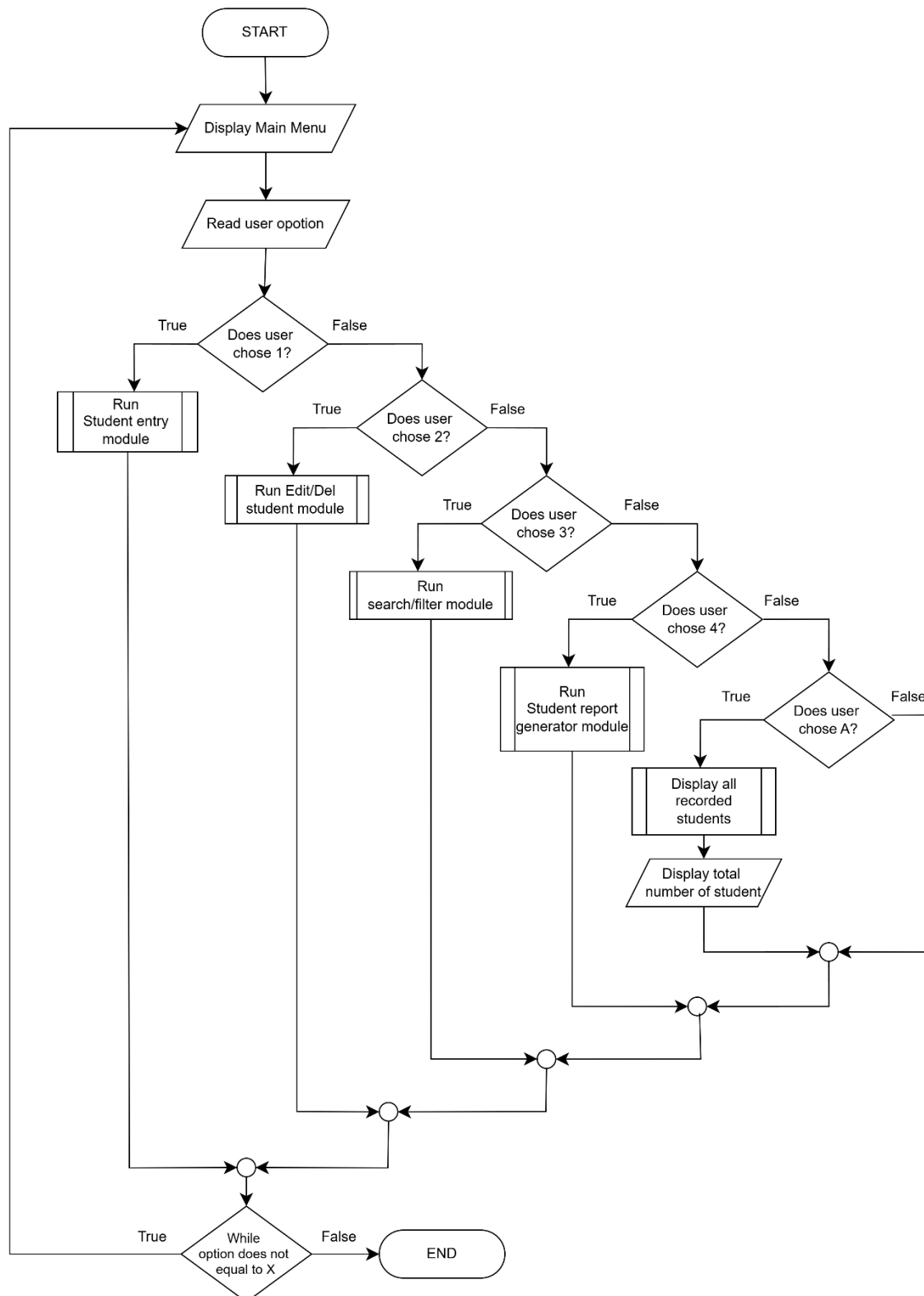
Content	Page(s)
Task Division	1
Flowchart	2-12
Pseudocode	13-25
Test Cases	26-45
Appendix	46-80

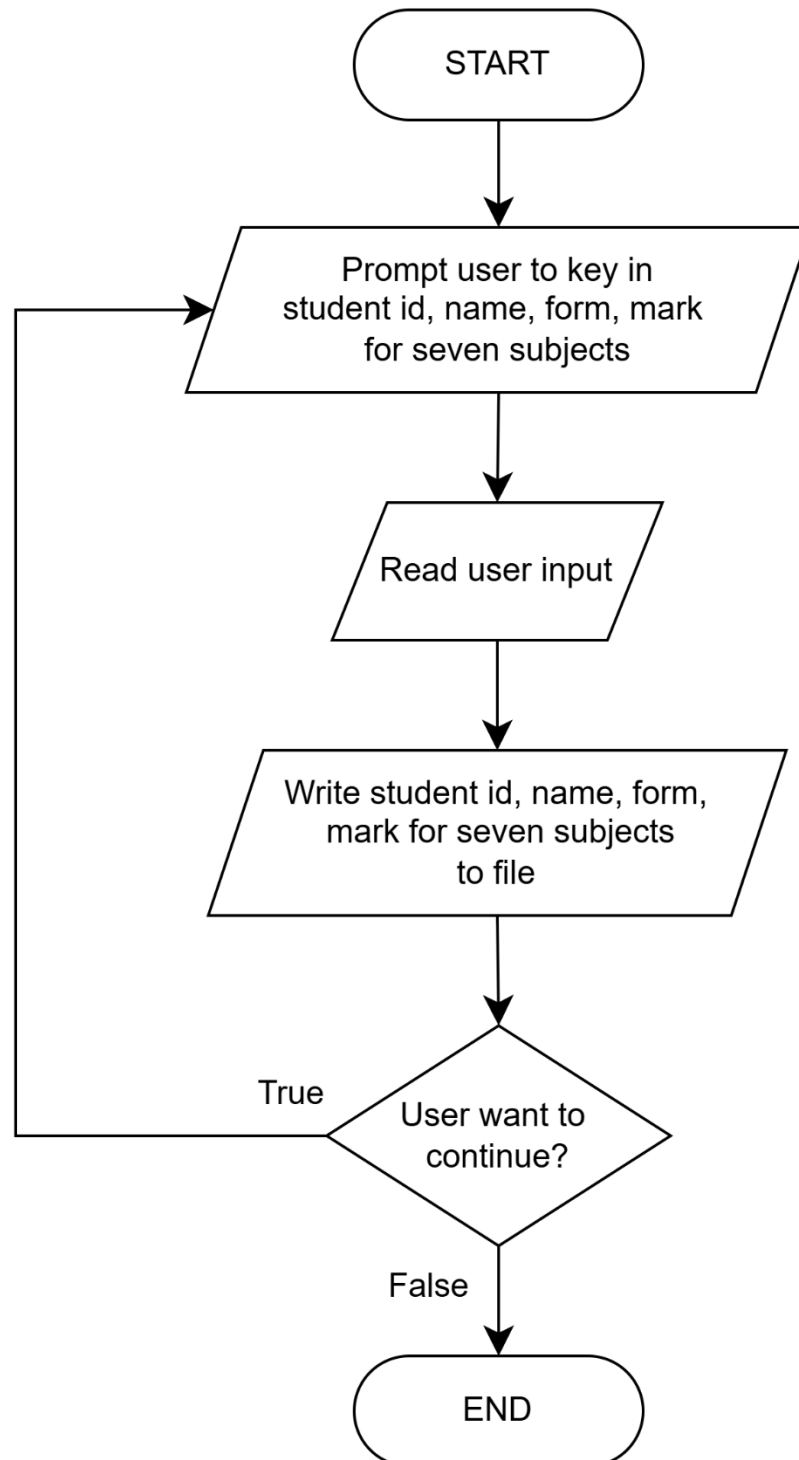
Task Division

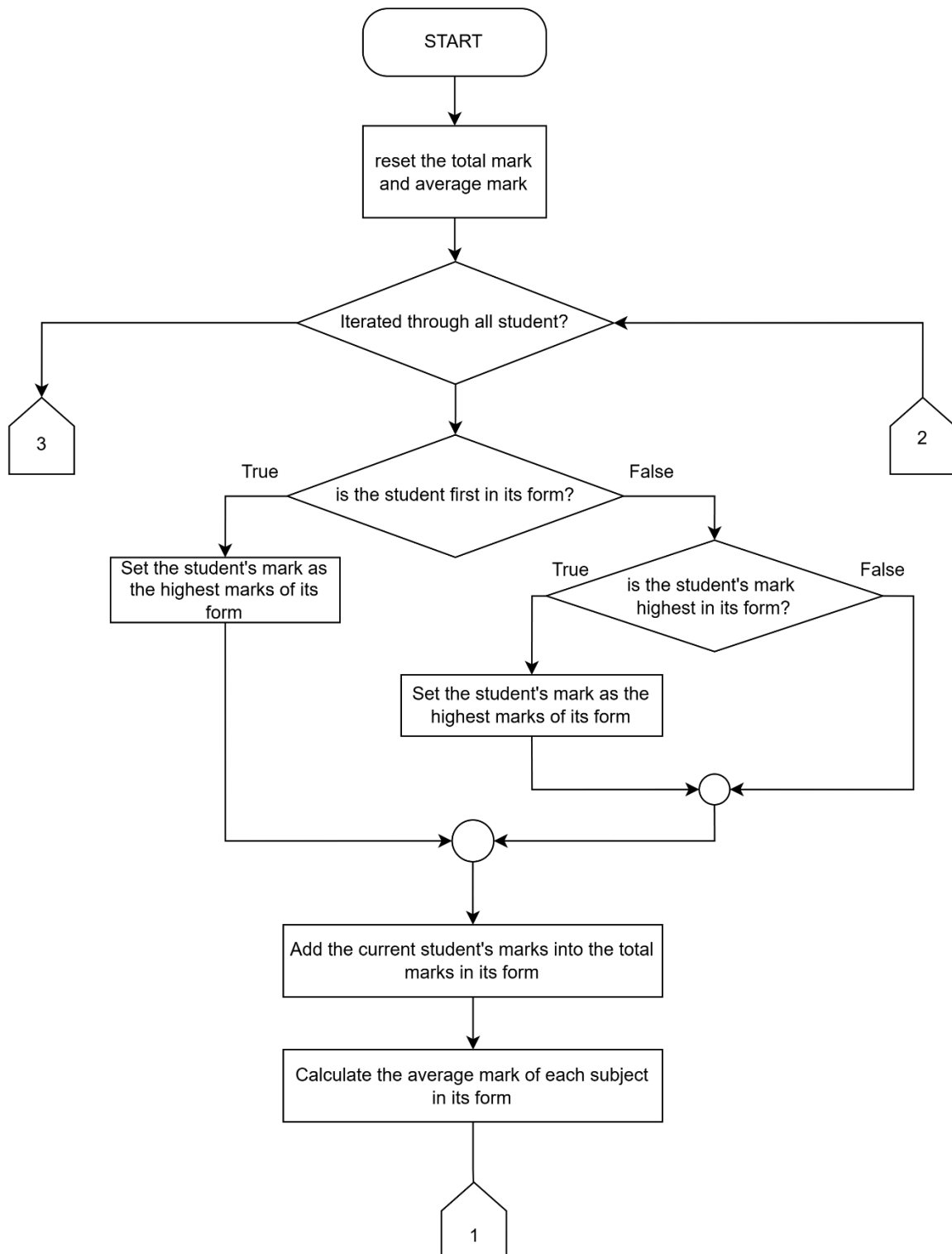
	Member 1	Member 2	Member 3
Name:	WONG WEI QI	LIANG TIM LOK	LIN, GUO YI
Student ID:	2302302	2301094	2301206
Task	Add/Calculation	Search/Filter	Edit/Delete

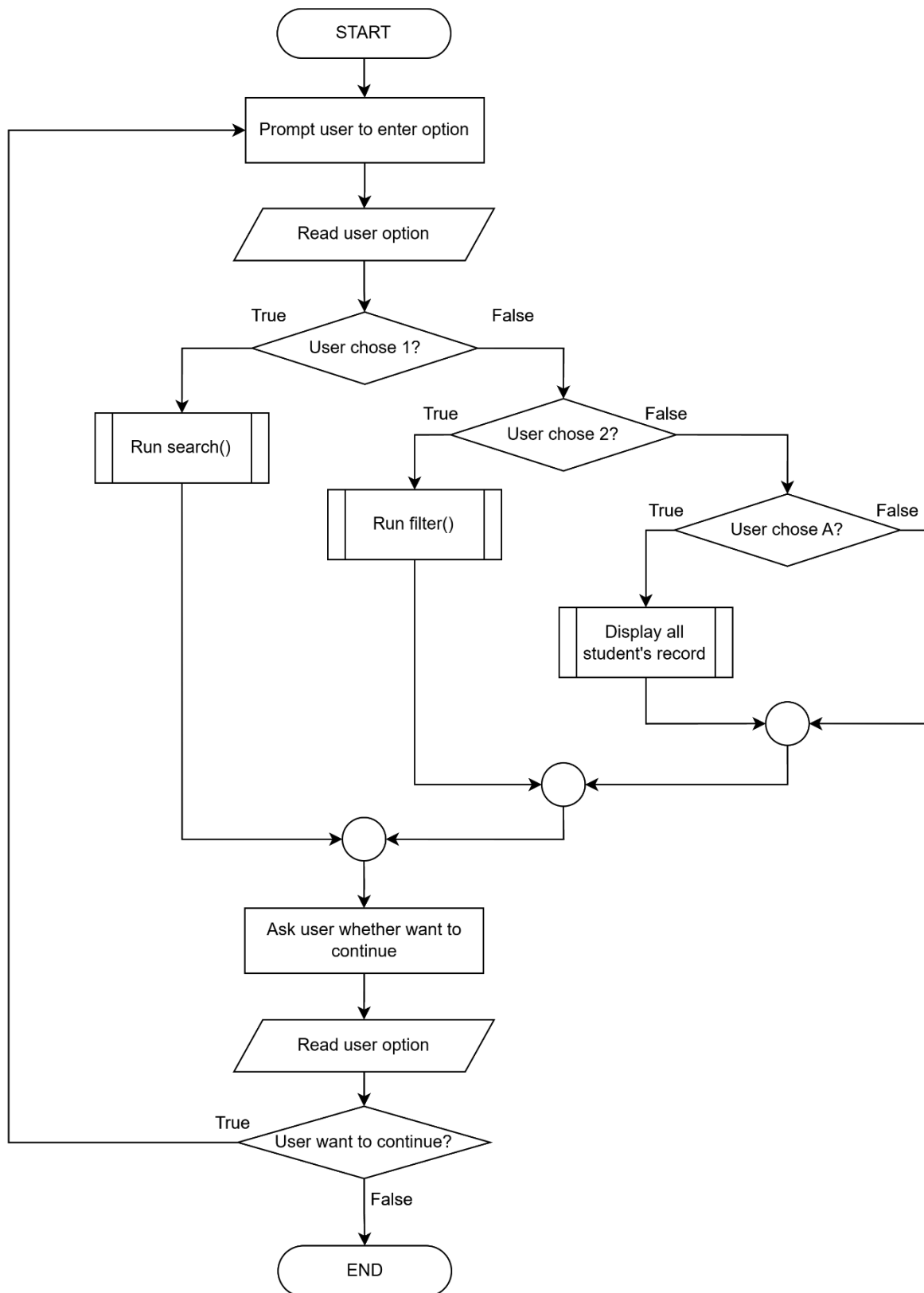
Flowchart

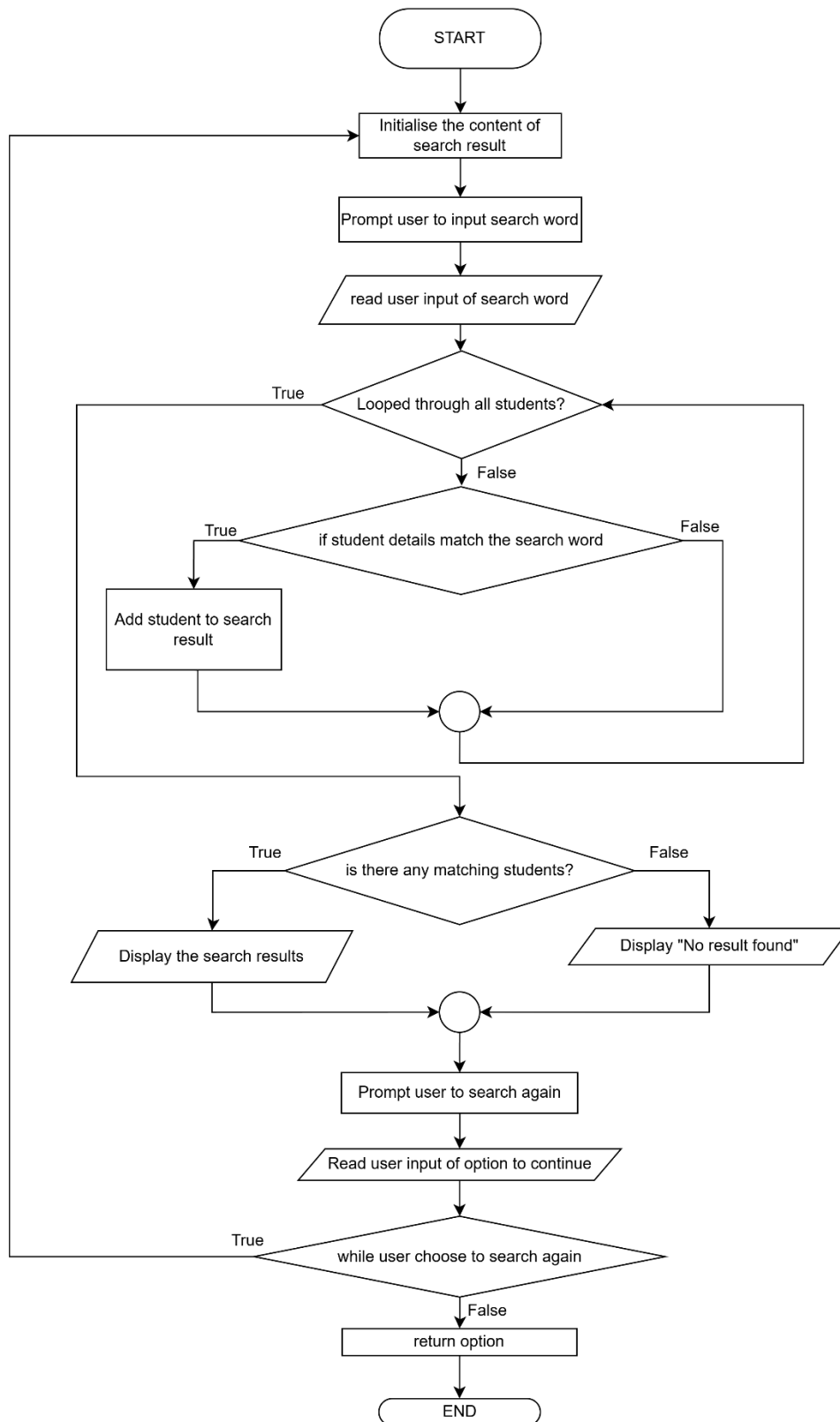
Main:

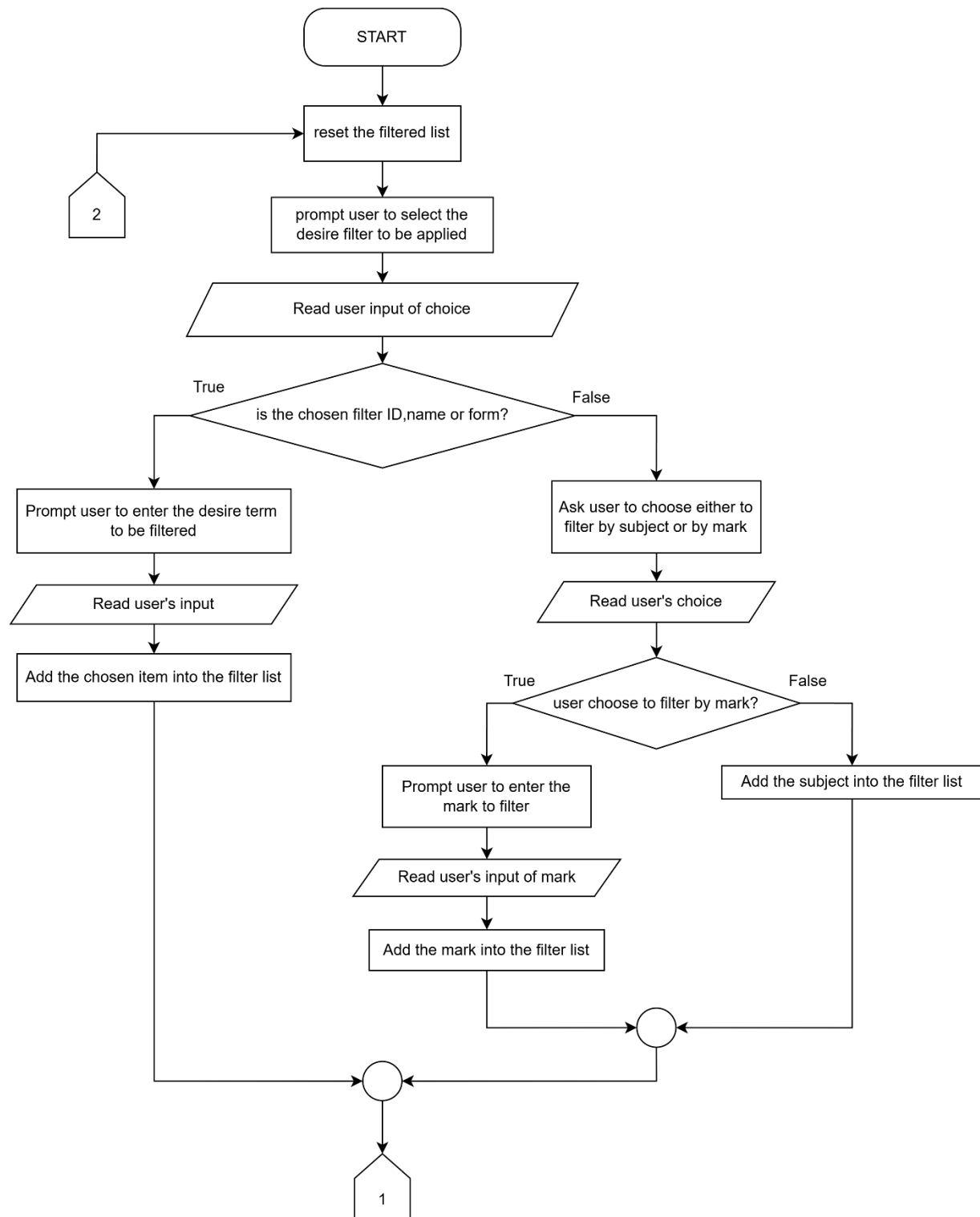


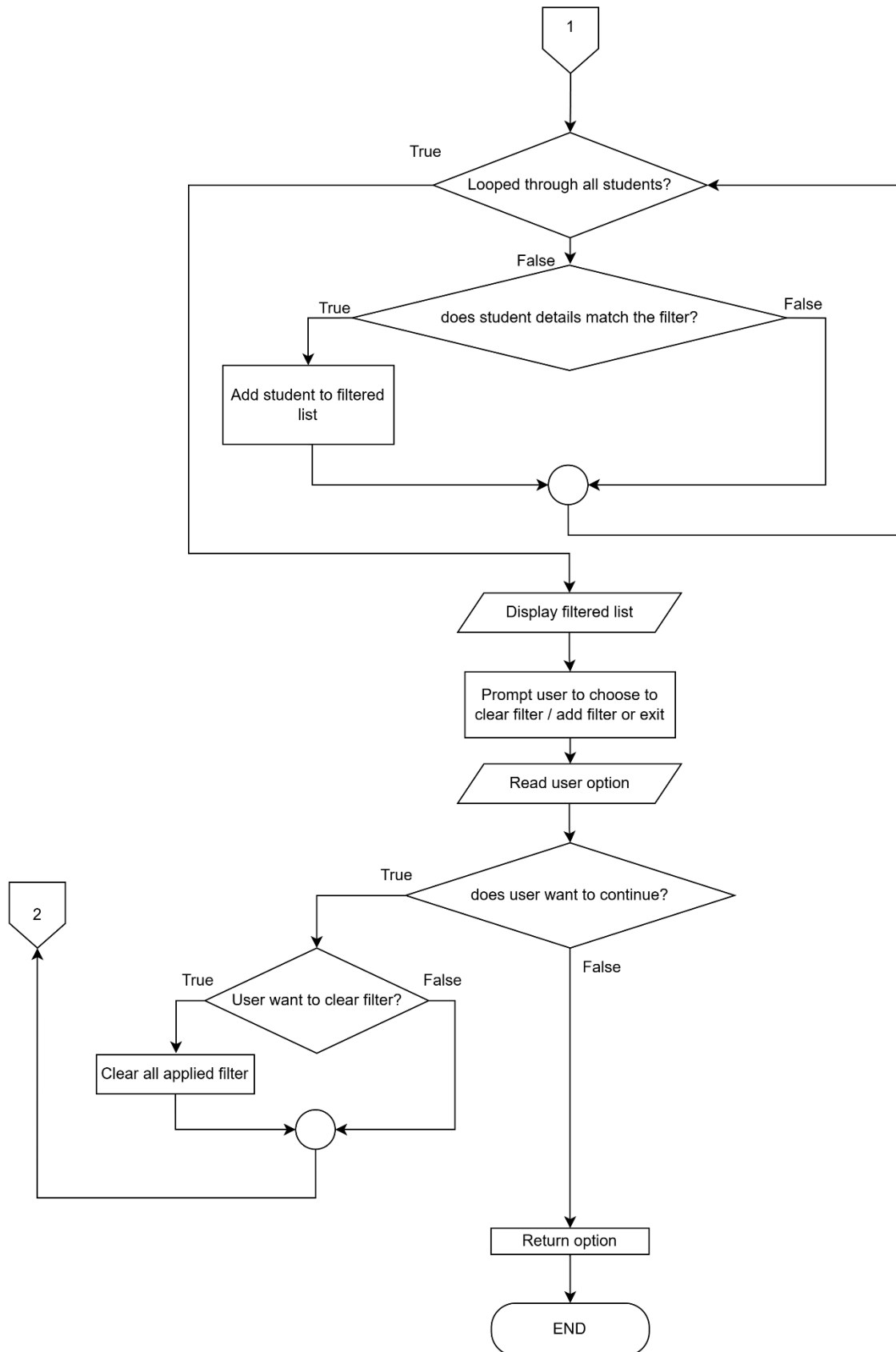
Add new student module (void addstudent())

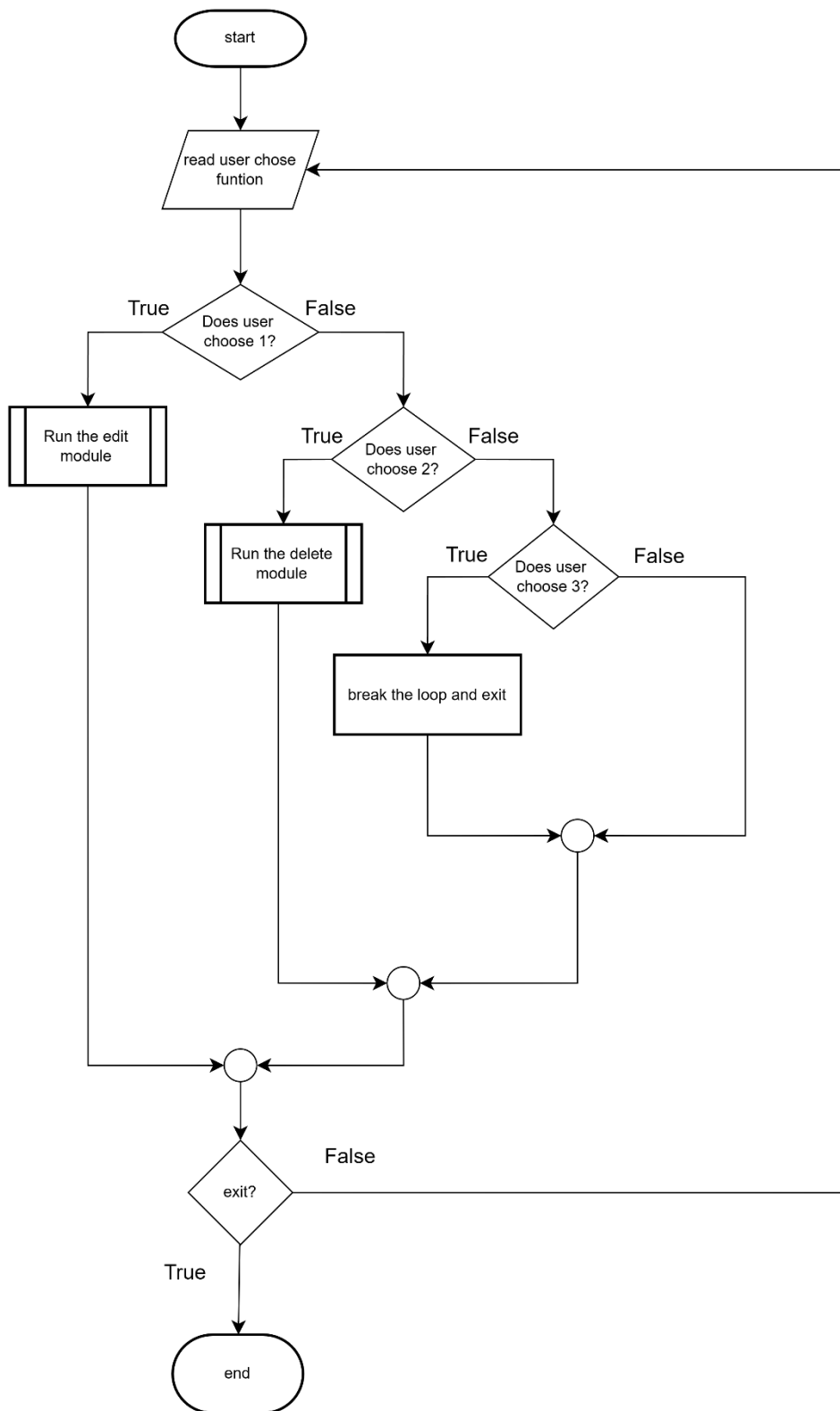
Student's Mark report calculation module (void calculate(double f1, double f2, double f3))

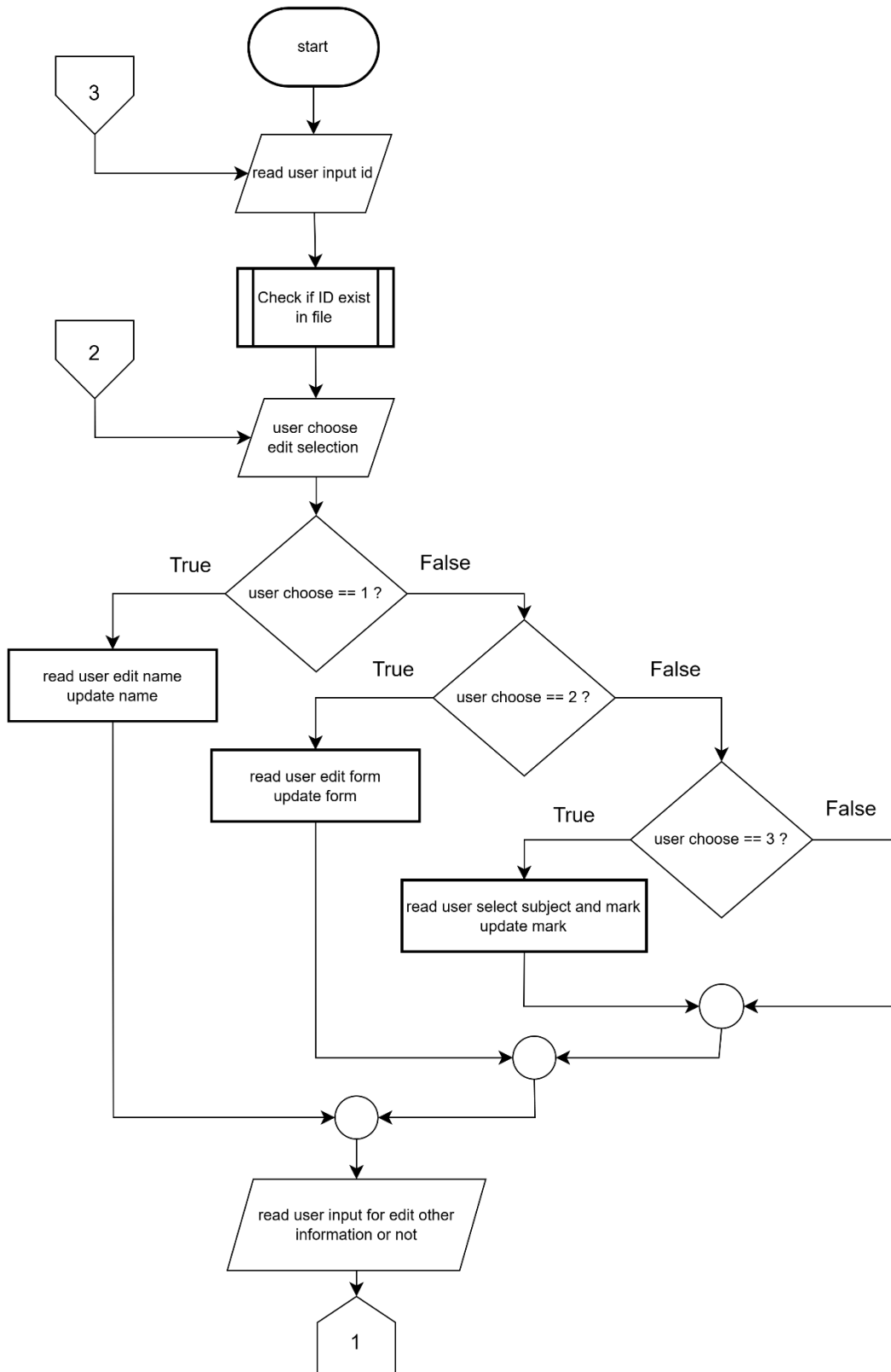
Search/Filter Module (*void searchFilterMenu()*)

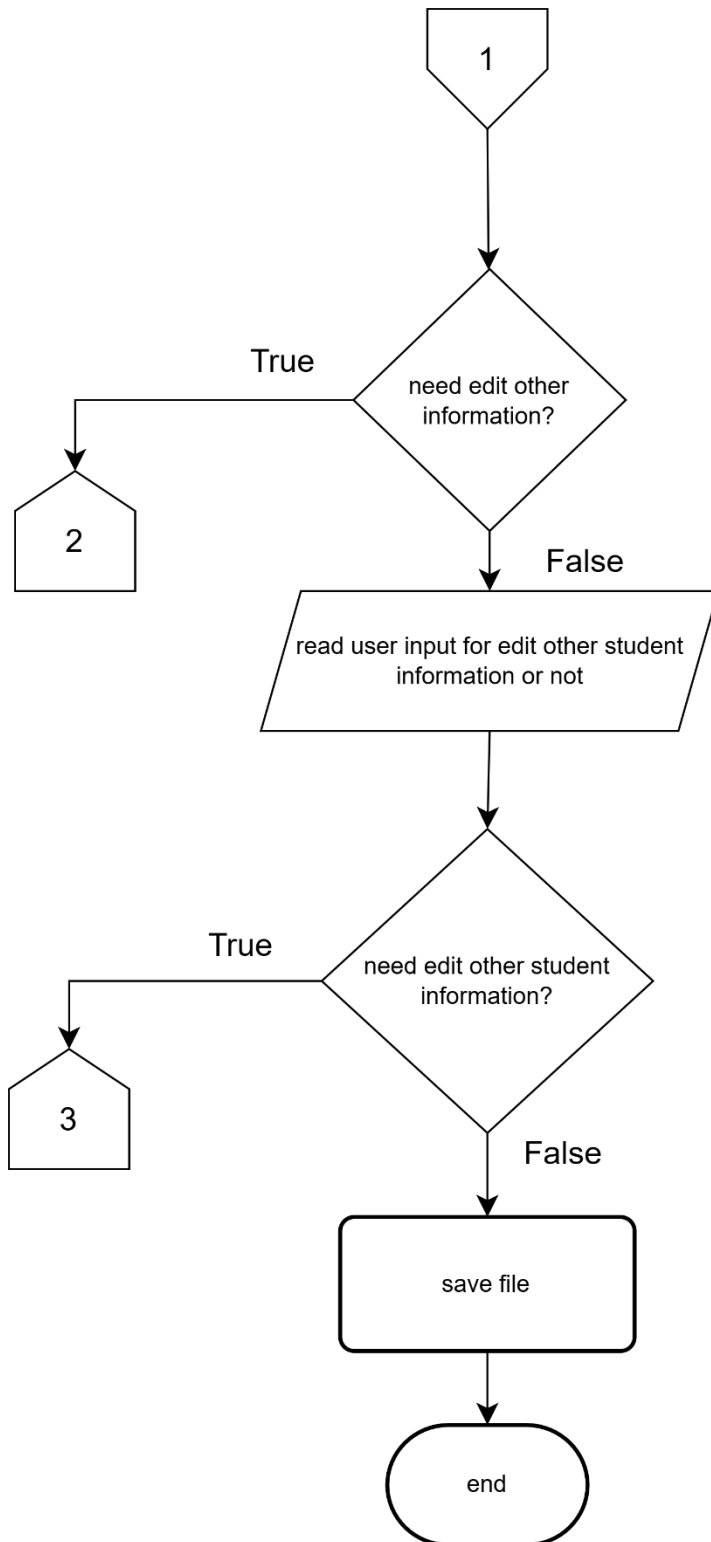
Search Module (*string search()*)

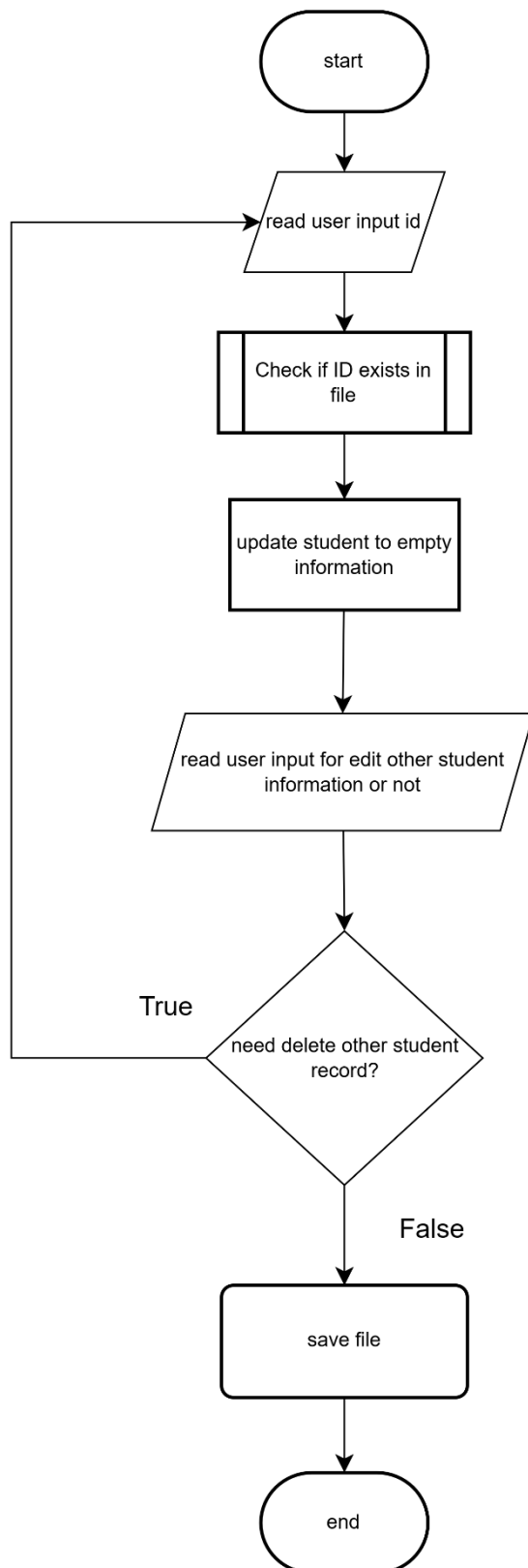
Filter module (*string filter()*)



Edit/Delete Menu Module (void *edit_and_del()*)

Edit Module (void edit(string id))



Delete module (*void del(string id)*)

Pseudocode**Main:**

Calculate the number of student recorded in student.txt

Prompt user to choose from add/delete/search/filter/View student's mark report/display all student/terminate program.

Validate the user's input

If not valid,

 Display error message and prompt user to enter again.

Does user choose to add new student?

If yes,

 Run the add new student module

 Loop back to prompt for menu option

If no,

 Does user choose to delete/edit recorded student?

 If yes,

 Run the delete/edit module.

 Loop back to prompt for menu option

 If no,

 Does user choose to search/filter?

 If yes,

 Run the search/filter module.

 Loop back to prompt for menu option

 If no,

 Does user choose to view the mark's report?

 If yes,

 Display the report modules.

 Loop back to prompt for menu option

 If no,

Does user choose to display all recorded students?

If yes,

Display all recorded students.

Loop back to prompt for menu option

If no,

Does user choose to exit the program?

If yes,

Terminate the program.

Add new student module (*void addstudent()*):

Open student.txt in appending mode.

// Entering Student information: ID

Prompt user to input student's ID

Check if user input is in valid format of ID.

If is valid,

 Proceed to prompt user to input student's name.

If not valid,

 Display error message and prompt user to enter ID again.

//Entering Student Information: Name

Prompt user to input student's name.

Check if user input consist of alphabet only

If yes,

 Proceed to enter student's form.

If no,

 Display error message and prompt user to enter name again.

//Entering Student Information: Form

Prompt user to input student form.

Check if user input only consist of 1,2 or 3

If yes,

 Proceed to enter student mark of each subject

If no,

 Display error message and prompt user to enter again.

//Entering Student Information: Mark of each subject.

Prompt user to enter marks for each subject

Check if user input is within range of 1-100.

If yes,

Proceed to append student information into student.txt
If no,
 Display error message and prompt user to enter again.

Close student.txt.

Prompt user to select to add more new student or not
If yes,
 Loop back to start of the function
If no,
 Terminate this function.

Student's Mark report calculation module (*void calculate(double f1, double f2, double f3)*)

Reset the total mark of overall student

Iterate through every recorded student and all subject

//Compute the highest mark and average mark for form 1

Current student's form is 1?

If yes,

First student iterated in form 1?

If yes,

Set the student's mark of current subject as the highest mark in form 1.

If no,

Is the student's mark higher than the previous highest mark in form 1?

If yes,

Set the student's mark of current subject as the highest mark in form 1.

Add the current student's mark of current subject into the total mark of the subject
in form 1

Calculate the average mark of the subject in form 1

If no,

//Compute the highest mark and average mark for form 2

Current student's form is 2?

If yes,

First student iterated in form 2?

If yes,

Set the student's mark of current subject as the highest mark in form 2.

If no,

Is the student's mark higher than the previous highest mark in form 2?

If yes,

Set the student's mark of current subject as the highest mark in
form 2.

Add the current student's mark of current subject into the total mark of the subject in form 2

Calculate the average mark of the subject in form 2

If no,

//Compute the highest mark and average mark for form 3

Current student's form is 3?

If yes,

First student iterated in form 3?

If yes,

Set the student's mark of current subject as the highest mark in form 3.

If no,

Is the student's mark higher than the previous highest mark in form 3?

If yes,

Set the student's mark of current subject as the highest mark in form 3.

Add the current student's mark of current subject into the total mark of the subject in form 2

Calculate the average mark of the subject in form 2

//Finding the highest mark for overall student

Iterate through every student and every subject

Is the current student first to be iterated in the whole record??

If yes,

Set the every mark of the student as the highest mark overall in each subject.

If no,

Is the student mark in the current subject higher than the pervious highest mark?

If yes,

Set the current student mark as the highest mark overall in the current subject

Add the current student's mark into the overall subject total mark

Calculate the overall average mark of the subject.

//Compute the Standard deviation

Iterate through every student and subject.

If the student is form 1

 Compute the numerator value of the standard deviation for each subject

 Add the calculated into the total numerator value of form 1 for each subject

If the student is form 2,

 Compute the numerator value of the standard deviation for each subject

 Add the calculated into the total numerator value of form 2 for each subject

If the student is form 3,

 Compute the numerator value of the standard deviation for each subject

 Add the calculated into the total numerator value of form 3 for each subject

Compute the numerator value of the standard deviation for each subject for all student

Add the calculated value into the overall total numerator value for each subject.

Iterate through every student and each student.

Calculate the standard deviation value for each subject of each form and overally.

Search/Filter Module (*void searchFilterMenu()*)

Prompt user to choose to search/filter/display all student

Does user choose to search?

If yes,

Run the search module

If no,

does user choose to filter?

If yes,

Run the filter module

Loop back to prompt user for menu option

If no,

Does user choose to display all recorded students?

If yes,

Display all recorded student

Loop back to prompt user for menu option

If no,

Does user choose to exit this function?

If yes,

Terminate this function

Search Module (*string search()*)

Reset the search result list.

Reset the number of search result.

Prompt user to enter the search word.

Iterate through every student.

Is the current student set as found?

If yes,

Continue to next student

If no,

Does the current student name match the search word?

If yes,

Add the student details into the search result list

Set the student as found.

If no,

Does the current student ID match the search word?

If yes,

Add the student details into the search result list

Set the student as found.

If no,

Does the current student form match the search word?

If yes,

Add the student details into the search result list

Set the student as found.

If no,

Continue to next student

Is the number of search result more than 0?

If yes,

Display the search result list

Display the number of result found

If no,

Display "No student is found"

Prompt user to choose either to search again or not.

If yes,

Loop back to prompt user for search word

If no,

Return user option

Terminate this function.

Filter module (*string filter()*)

Reset the list that store the filtered result.

Reset the number of result found.

Prompt user to select which filter to apply or to exit/proceed

Does user choose to filter by ID, name or form?

If yes,

 Prompt user to enter the information for the selected attribute

 Add the attribute and its value into the applied filter list

 Prompt user to make another option

If no,

 Does user choose to any of the seven subject?

 If yes,

 Prompt user to choose either to filter by subject or mark

 If by subject,

 Add the subject into the applied filter list

 Prompt user to make another option

 If by mark,

 Prompt user to enter the mark

 Add the subject's mark into the applied filter list

 Prompt user to make another option

 If no,

 Does user choose to proceed to filter?

 If yes,

 Compare all the student and its details with the applied filters

 Does the information of the students match the applied filters?

 If yes,

 Add the student into the filtered result list

 Continue to check all other students

 If no,

 Continue to check all other students

 Display all the student in the filtered result list

 Prompt user to choose from add another filter/clear all filter/exit this function

 Does user choose to add another filter?

 If yes,

 Clear the applied filter list

 Loop back to the filter menu

 If no


```
    Does user choose to clear all applied filter?
    If yes,
        Reset the applied filter list
        Loop back to the filter menu
    If no,
        Does user choose to exit the function?
        If yes,
            Return option
            Terminate this function
    If no,
        Does user choose to exit this function?
        If yes,
            Return option
            Terminate this function
```

Edit/Delete Menu Module (void *edit_and_del()*)

Prompt user to choose from edit, delete or exit this function

Does user choose to edit?

 If yes,

 Run the edit module

 Loop back to prompt user for menu option

 If no,

 Does user choose to delete?

Loop back to prompt user for menu option

 If yes,

 Run the edit module

 Loop back to prompt user for menu option

 If no,

 Terminate this function

Edit Module (void *edit(string id)*)

Read user input id

Read user choose edit selection

If selection is 1

 Read user edit name

 Update name

Else if selection is 2

 Read user edit form

 Update form

Else if selection is 3

 Read user select subject and mark

 Update mark to according subject

Read user input for edit other information or not

While need edit other information ,if yes go back to read user choose edit selection

Read user input for edit other student information or not

While need edit other student information, if yes go back to read user input id

Save file for user edit

Delete module (void *del(string id)*)

Read user input id

update student record to empty

save file

Prompt user to select whether to delete another student

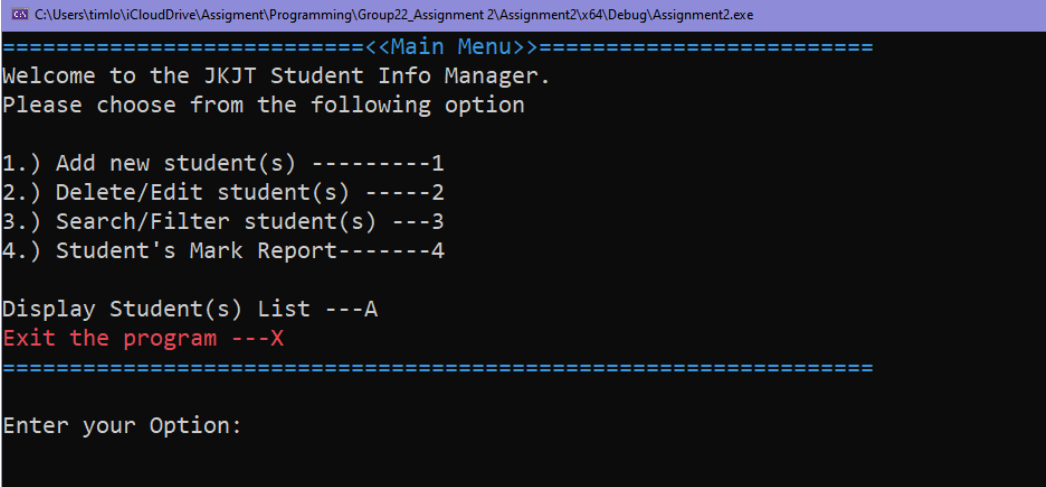
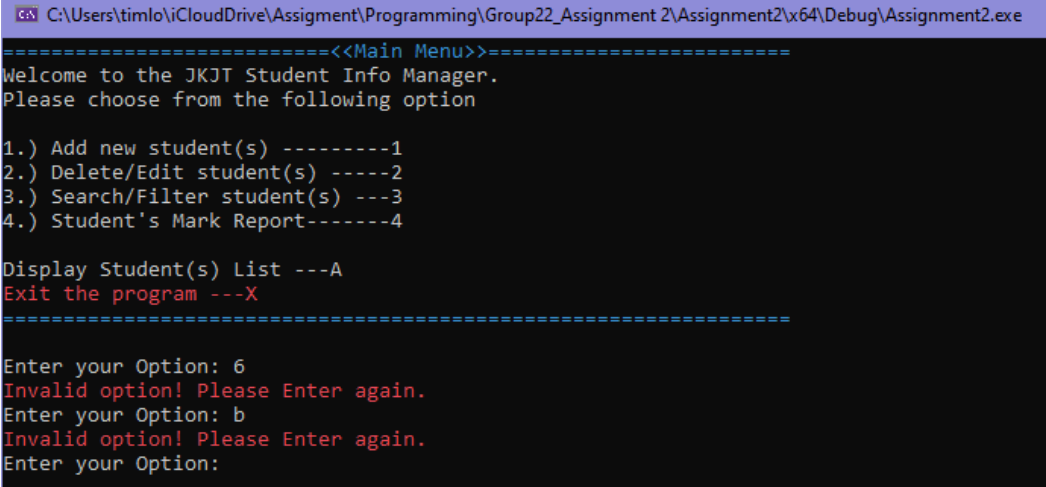
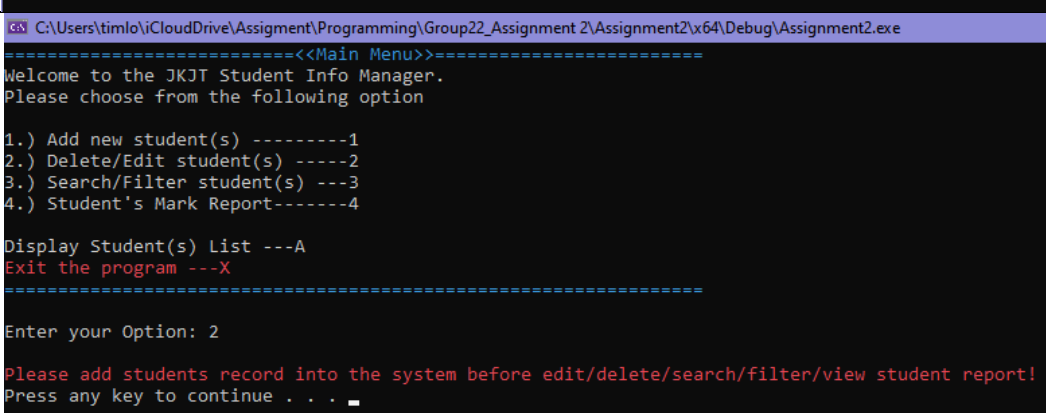
If yes,

 Loop back to read ID

If no,

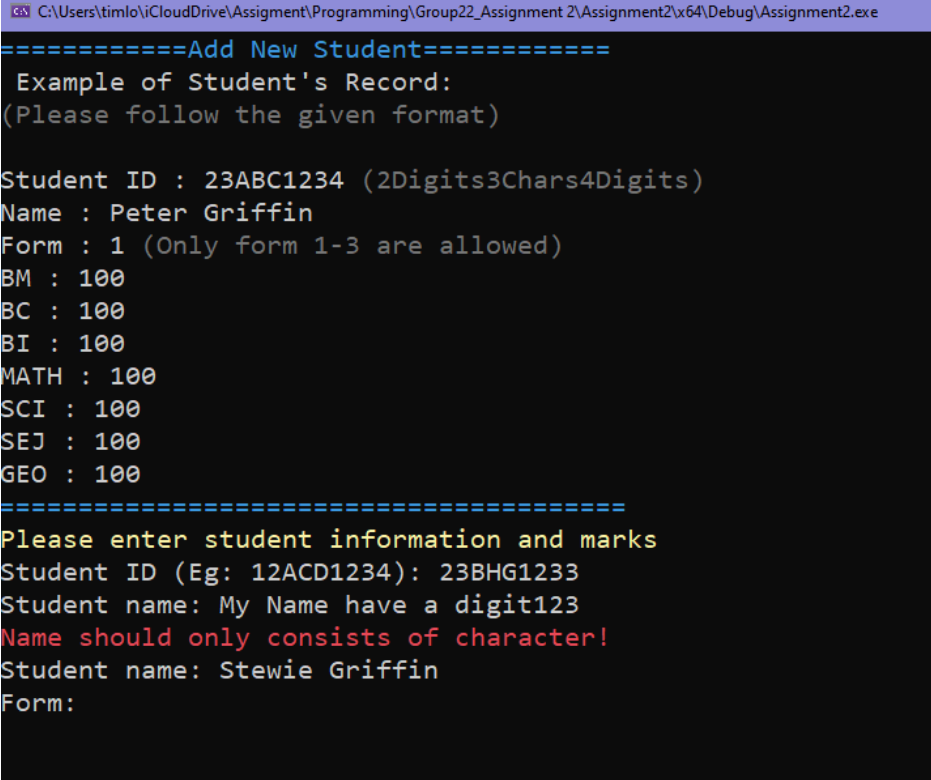
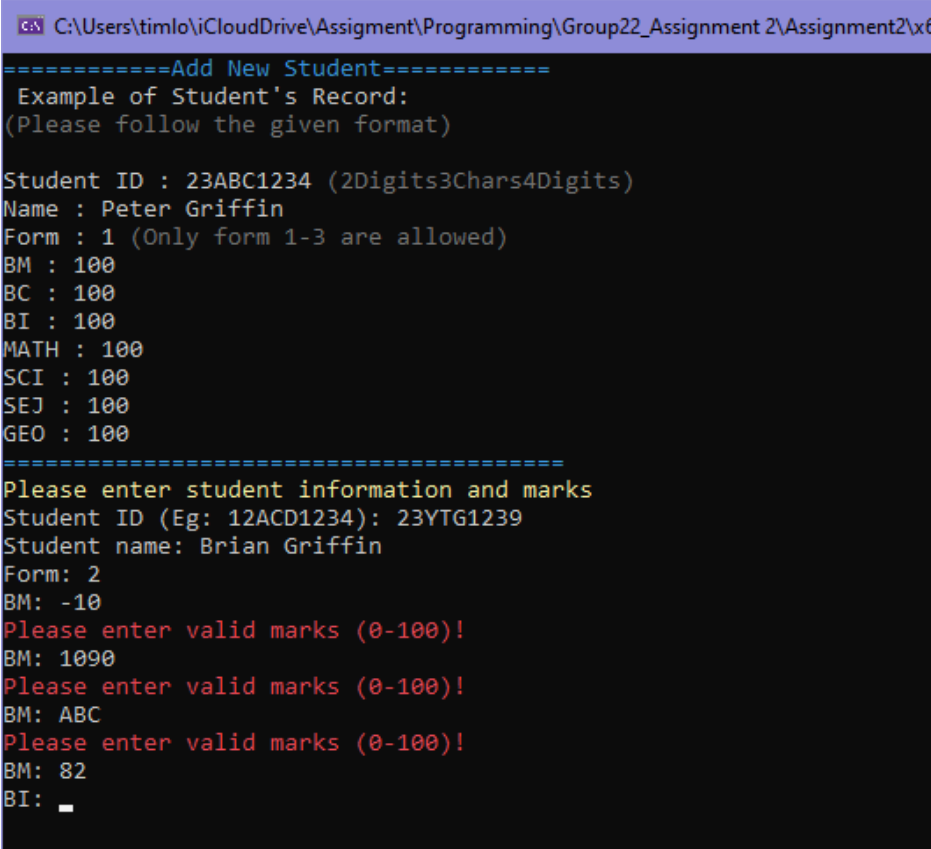
 Terminate the function

Test Cases

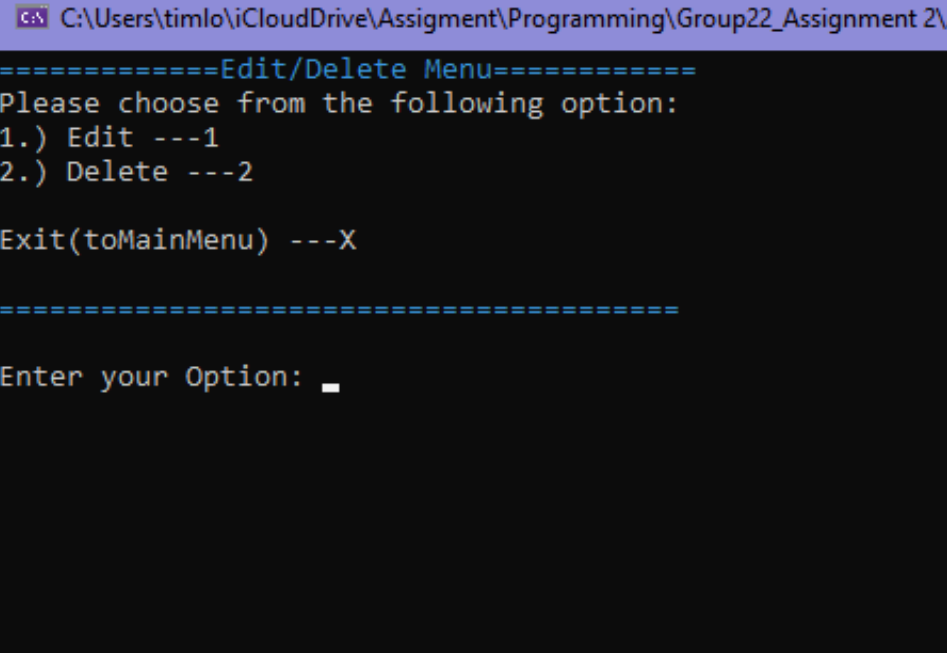
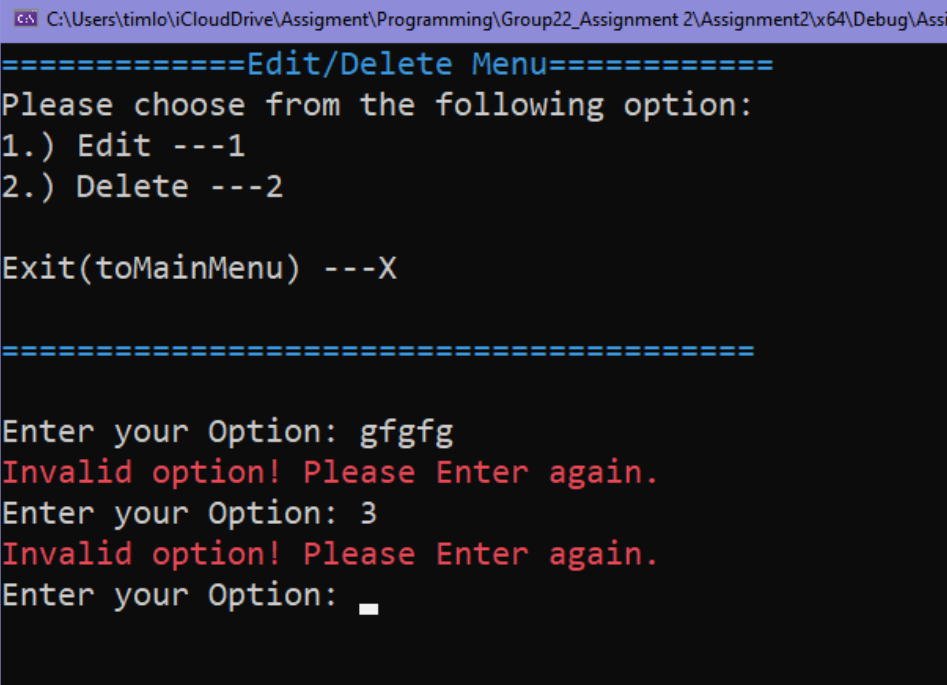
Module	Description	Images
Main	Main Menu that allow user to access all the function of the program	 <pre> C:\Users\timlo\iCloudDrive\Assignment\Programming\Group22_Assignment 2\Assignment2\64\Debug\Assignment2.exe =====<<Main Menu>>===== Welcome to the JKJT Student Info Manager. Please choose from the following option 1.) Add new student(s) -----1 2.) Delete/Edit student(s) -----2 3.) Search/Filter student(s) ---3 4.) Student's Mark Report-----4 Display Student(s) List ---A Exit the program ---X ===== Enter your Option: </pre>
Main	Proper input validation which display an error message when user enter an invalid option	 <pre> C:\Users\timlo\iCloudDrive\Assignment\Programming\Group22_Assignment 2\Assignment2\64\Debug\Assignment2.exe =====<<Main Menu>>===== Welcome to the JKJT Student Info Manager. Please choose from the following option 1.) Add new student(s) -----1 2.) Delete/Edit student(s) -----2 3.) Search/Filter student(s) ---3 4.) Student's Mark Report-----4 Display Student(s) List ---A Exit the program ---X ===== Enter your Option: 6 Invalid option! Please Enter again. Enter your Option: b Invalid option! Please Enter again. Enter your Option: </pre>
Main	Error Handling when user try to perform any of the module other than add student when student.txt is empty or does not exist	 <pre> C:\Users\timlo\iCloudDrive\Assignment\Programming\Group22_Assignment 2\Assignment2\64\Debug\Assignment2.exe =====<<Main Menu>>===== Welcome to the JKJT Student Info Manager. Please choose from the following option 1.) Add new student(s) -----1 2.) Delete/Edit student(s) -----2 3.) Search/Filter student(s) ---3 4.) Student's Mark Report-----4 Display Student(s) List ---A Exit the program ---X ===== Enter your Option: 2 Please add students record into the system before edit/delete/search/filter/view student report! Press any key to continue . . . </pre>

Main	<p>Display Student list function to allow user to check every recorded student and total number of students in the system</p> <p>The list is automatically sort by alphabetic order to ease the reading of user</p>	<pre> C:\Users\timlo\iCloudDrive\Assignment\Programming\Group22_Assignment 2\Assignment2\Debug\Assignment2.exe =====Student(s)===== Name ID Form BM BI BC MATH SCI SEJ GEO ----- 1) AIDEN CARTER 17MNO1017 2 76 82 80 88 85 79 90 2) ALEX ALBON 21ACD2021 3 12 24 54 23 56 87 68 3) ALI MALOU 23JGF1234 1 12 12 12 12 12 12 12 4) ALICE BELL 47ABC2047 3 80 85 90 95 70 75 80 5) AMELIA KING 13YZG2013 3 80 85 90 95 70 75 80 6) AMELIA YOUNG 45VWX2045 2 90 85 88 84 87 92 80 7) AVA CLARK 11MNO2011 2 76 84 80 85 90 91 75 8) BENJAMIN SCOTT 14ABC1014 1 76 80 85 88 90 92 70 9) CHARLES WANG 33ABC2033 3 65 70 75 80 85 90 95 10) CHARLOTTE HARRIS 09GHI2009 3 82 88 94 90 76 72 80 11) CHARLOTTE SCOTT 50JKL1050 3 85 90 92 94 89 88 87 12) CHECO PEREZ 23HDB2345 3 54 56 56 56 56 56 56 13) CHECO PEREZ 23HFG1234 3 34 34 34 34 34 34 34 14) CHRIS JOHNSON 26ABC1026 3 82 79 85 88 90 91 84 15) CORLAS SIZ 12BFD1234 3 45 45 45 45 45 45 45 *Total of 15 student(s) recorded. Press any key to continue . . . </pre>
Add	<p>Module that allow user to add student into the system with clear instruction of format that need to be follow by user when entering student record</p>	<pre> C:\Users\timlo\iCloudDrive\Assignment\Programming\Group22_Assignment 2\Assignment2\Debug\Assignment2.exe =====Add New Student===== Example of Student's Record: (Please follow the given format) Student ID : 23ABC1234 (2Digits3Chars4Digits) Name : Peter Griffin Form : 1 (Only form 1-3 are allowed) BM : 100 BC : 100 BI : 100 MATH : 100 SCI : 100 SEJ : 100 GEO : 100 Please enter student information and marks Student ID (Eg: 12ACD1234): </pre>

Add	<p>User will be prompt to enter every attribute that are needed before writing the details in the file</p> <p>User are allowed to key in another student record after key in a student successfully</p>	<pre> C:\Users\timlo\iCloudDrive\Assignment\Programming\Group22_Assignment 2\Assignment2\Debug\Assignment2.exe =====Add New Student===== Example of Student's Record: (Please follow the given format) Student ID : 23ABC1234 (2Digits3Chars4Digits) Name : Peter Griffin Form : 1 (Only form 1-3 are allowed) BM : 100 BC : 100 BI : 100 MATH : 100 SCI : 100 SEJ : 100 GEO : 100 Please enter student information and marks Student ID (Eg: 12ACD1234): 12GHG7653 Student name: Meg Griffin Form: 3 BM: 29 BI: 58 BC: 12 Math: 58 Sci: 0 Sejarah: 100 Geo: 90 Do you want to key in another record? (Y/N) Enter your Option: </pre>
Add	<p>Input validation that displays error message which tell the user the incorrect portion of the ID enter by the user</p> <p>Prompt user to enter the again if incorrect format of ID are detected</p> <p>Only will proceed to ask for the next attribute when ID with correct format is entered</p>	<pre> C:\Users\timlo\iCloudDrive\Assignment\Programming\Group22_Assignment 2\Assignment2\Debug\Assignment2.exe =====Add New Student===== Example of Student's Record: (Please follow the given format) Student ID : 23ABC1234 (2Digits3Chars4Digits) Name : Peter Griffin Form : 1 (Only form 1-3 are allowed) BM : 100 BC : 100 BI : 100 MATH : 100 SCI : 100 SEJ : 100 GEO : 100 Please enter student information and marks Student ID (Eg: 12ACD1234): Some wrong ID ID must be exactly 9 characters long. Student ID (Eg: 12ACD1234): 123456789 Characters 3 to 5 must be letters. Student ID (Eg: 12ACD1234): 12BHG10P0 Last four characters must be digits. Student ID (Eg: 12ACD1234): AZBGH1233 First two characters must be digits. Student ID (Eg: 12ACD1234): 12BHB9298 Student name: </pre>

Add	<p>Input validation for name. Program will display an error message and prompt user to enter again when the program detect any name that consist of numeric digit</p> <p>Will only proceed to next attribute when name without digit are entered</p>	 <pre> C:\Users\timlo\iCloudDrive\Assignment\Programming\Group22_Assignment 2\Assignment2\Assignment2.exe ====Add New Student==== Example of Student's Record: (Please follow the given format) Student ID : 23ABC1234 (2Digits3Chars4Digits) Name : Peter Griffin Form : 1 (Only form 1-3 are allowed) BM : 100 BC : 100 BI : 100 MATH : 100 SCI : 100 SEJ : 100 GEO : 100 Please enter student information and marks Student ID (Eg: 12ACD1234): 23BHG1233 Student name: My Name have a digit123 Name should only consists of character! Student name: Stewie Griffin Form: </pre>	
Add	<p>Input Validation for mark of every subject.</p> <p>Only integer in valid range (0-100) are accepted</p> <p>Error Message will be displayed for invalid marks and user will be prompted to enter mark again</p>	 <pre> C:\Users\timlo\iCloudDrive\Assignment\Programming\Group22_Assignment 2\Assignment2\Assignment2.exe ====Add New Student==== Example of Student's Record: (Please follow the given format) Student ID : 23ABC1234 (2Digits3Chars4Digits) Name : Peter Griffin Form : 1 (Only form 1-3 are allowed) BM : 100 BC : 100 BI : 100 MATH : 100 SCI : 100 SEJ : 100 GEO : 100 Please enter student information and marks Student ID (Eg: 12ACD1234): 23YTG1239 Student name: Brian Griffin Form: 2 BM: -10 Please enter valid marks (0-100)! BM: 1090 Please enter valid marks (0-100)! BM: ABC Please enter valid marks (0-100)! BM: 82 BI: </pre>	

Add	Input Validation for invalid option entered by user.	<pre> C:\Users\timlo\iCloudDrive\Assignment\Programming\Group22_Assignment 2\Assignment2 =====Add New Student===== Example of Student's Record: (Please follow the given format) Student ID : 23ABC1234 (2Digits3Chars4Digits) Name : Peter Griffin Form : 1 (Only form 1-3 are allowed) BM : 100 BC : 100 BI : 100 MATH : 100 SCI : 100 SEJ : 100 GEO : 100 ===== Please enter student information and marks Student ID (Eg: 12ACD1234): 12GHF1234 Student name: Stewie Griffin Form: 1 BM: 101 Please enter valid marks (0-100)! BM: -90 Please enter valid marks (0-100)! BM: 82 BI: 78 BC: 82 Math: 100 Sci: 90 Sejarah: 52 Geo: 10 Do you want to key in another record? (Y/N) Enter your Option: h Invalid option! Please Enter again. Enter your Option: U Invalid option! Please Enter again. Enter your Option: 1233 Invalid option! Please Enter again. Enter your Option: </pre>
-----	--	---

Edit/ Delete	Edit/Delete Menu which allow user to perform the option to the recorded student	 <pre> C:\Users\timlo\iCloudDrive\Assigment\Programming\Group22_Assignment 2\ =====Edit/Delete Menu===== Please choose from the following option: 1.) Edit ---1 2.) Delete ---2 Exit(toMainMenu) ---X ===== Enter your Option: _ </pre>
	Input Validation for the menu. Error message will be displayed and user will be asked to enter the option again	 <pre> C:\Users\timlo\iCloudDrive\Assigment\Programming\Group22_Assignment 2\Assignment2\x64\Debug\Assi =====Edit/Delete Menu===== Please choose from the following option: 1.) Edit ---1 2.) Delete ---2 Exit(toMainMenu) ---X ===== Enter your Option: gfgfg Invalid option! Please Enter again. Enter your Option: 3 Invalid option! Please Enter again. Enter your Option: _ </pre>

Edit	Edit module allow user to edit any student that are recorded in the system	<pre> C:\Users\timlo\iCloudDrive\Assignment\Programming\Group22_Assignment 2\Assignment2\Debug\Assignment2.exe =====Student(s)===== Name ID Form BM BI BC MATH SCI SEJ GEO ----- 1) AIDEN CARTER 17MNO1017 2 76 82 80 88 85 79 90 2) ALEX ALBON 21ACD2021 3 12 24 54 23 56 87 68 3) ALI MALOU 23JGF1234 1 12 12 12 12 12 12 100 4) ALICE BELL 47ABC2047 3 80 85 90 95 70 75 80 5) AMELIA KING 13YZG2013 3 80 85 90 95 70 75 100 6) AMELIA YOUNG 45VWX2045 1 90 85 88 84 87 92 80 7) AVA CLARK 11MNO2011 2 76 84 80 85 90 91 75 8) BENJAMIN SCOTT 14ABC1014 1 76 80 85 88 90 92 70 9) CARLOZ SAINZ 128FD1234 3 45 45 45 45 45 45 45 10) CHARLES WANG 33ABC2033 3 65 70 75 80 85 90 95 11) CHARLOTTE HARRIS 09GHI2009 3 82 88 94 90 76 72 80 12) CHECO PEREZ 23HDB2345 3 54 56 56 56 56 56 56 ===== EDIT >> Please key in the student id that you want to edit: _ </pre>
Edit	<p>By entering the ID of the student, user can choose to edit any attribute of the student except of ID</p> <p>User can choose to edit another attribute or edit another student afterward</p>	<pre> C:\Users\timlo\iCloudDrive\Assignment\Programming\Group22_Assignment 2\Assignment2\Debug\Assignment2.exe =====Student(s)===== Name ID Form BM BI BC MATH SCI SEJ GEO ----- 1) AIDEN CARTER 17MNO1017 2 76 82 80 88 85 79 90 2) ALEX ALBON 21ACD2021 3 12 24 54 23 56 87 68 3) ALI MALOU 23JGF1234 1 12 12 12 12 12 12 100 4) ALICE BELL 47ABC2047 3 80 85 90 95 70 75 80 5) AMELIA KING 13YZG2013 3 80 85 90 95 70 75 100 6) AMELIA YOUNG 45VWX2045 1 90 85 88 84 87 92 80 7) AVA CLARK 11MNO2011 2 76 84 80 85 90 91 75 8) BENJAMIN SCOTT 14ABC1014 1 76 80 85 88 90 92 70 9) CARLOZ SAINZ 128FD1234 3 45 45 45 45 45 45 45 10) CHARLES WANG 33ABC2033 3 65 70 75 80 85 90 95 11) CHARLOTTE HARRIS 09GHI2009 3 82 88 94 90 76 72 80 12) CHECO PEREZ 23HDB2345 3 54 56 56 56 56 56 56 ===== EDIT >> Please key in the student id that you want to edit: 23HDB2345 >>Chosen Student<< Student ID: 23HDB2345 Name: CHECO PEREZ Form: 3 BM: 54 BI: 56 BC: 56 MATH: 56 SCI: 56 SEJ: 56 GEO: 56 Please select the attribute that you want to edit: (Name ---1 Form ---2 Marks ---3) User Input: 1 Please update the name: Daniel Ricchardo Do you still have anything to edit for this student?(y/n): n Edit for another student(s)? (y/n) : n Student List updated successfully! Press any key to continue . . . _ </pre>
Edit	<p>Input validation for the edit module.</p> <p>Error Message for unknown ID will be</p>	

	<p>display if the entered ID did not match any ID of student recorded.</p> <p>Invalid input for attribute such as input of 4 for student form or -10 for marks will be ask to entered again</p> <p>Any invalid option entered by user will be ask to enter again after display of an error message</p>	<pre> C:\Users\timlo\iCloudDrive\Assignment\Programming\Group22_Assignment 2\Assignment2\64\Debug\Assignment2.exe =====Student(s)===== Name ID Form BM BI BC MATH SCI SEJ GEO ----- 1) AIDEN CARTER 17MNO1017 2 76 82 80 88 85 79 90 2) ALEX ALBON 21ACD2021 3 12 24 54 23 56 87 68 3) ALI MALOU 23JGF1234 1 12 12 12 12 12 100 4) ALICE BELL 47ABC2047 3 80 85 90 95 70 75 80 5) AMELIA KING 13YZG2013 3 80 85 90 95 70 75 100 6) AMELIA YOUNG 45VMX2045 1 90 85 88 84 87 92 80 7) AVA CLARK 11MNO2011 2 76 84 80 85 90 91 75 8) BENJAMIN SCOTT 14ABC1014 1 76 80 85 88 90 92 70 9) CARLOZ SAINZ 12BFD1234 3 45 45 45 45 45 45 45 10) CHARLES WANG 33ABC2033 3 65 70 75 80 85 90 95 11) CHARLOTTE HARRIS 09GHI2009 3 82 88 94 90 76 72 80 12) DANIEL RICCHARDO 23HDB2345 3 54 56 56 56 56 56 56 ===== EDIT >> Please key in the student id that you want to edit: 23HDB234 Student record not found! Please enter again EDIT >> Please key in the student id that you want to edit: 23HDB2345 >>Chosen Student<< Student ID: 23HDB2345 Name: DANIEL RICCHARDO Form: 3 BM: 54 BI: 56 BC: 56 MATH: 56 SCI: 56 SEJ: 56 GEO: 56 Please select the attribute that you want to edit: (Name ---1 Form ---2 Marks ---3) User Input: 4 Invalid input,please enter again Please select the attribute that you want to edit: (Name ---1 Form ---2 Marks ---3) User Input: 3 Please enter the subject that you want to change the mark(BM/BI/BC/MATH/SCI/SEJ/GEO): Computer Science Invalid Input! Please enter again. Please enter the subject that you want to change the mark(BM/BI/BC/MATH/SCI/SEJ/GEO): GEO Please enter the mark: 101 Invalid input,please enter again, mark should between 0 to 100 Please enter the mark: -10 Invalid input,please enter again, mark should between 0 to 100 Please enter the mark: 100 Do you still have anything to edit for this student?(y/n): g Invalid input,please enter again Do you still have anything to edit for this student?(y/n): n Edit for another student(s)? (y/n) : 1 Invalid input,please enter again Edit for another student(s)? (y/n) : n Student list updated successfully! Press any key to continue . . . </pre>
Delete	Delete Module allow user to delete recorded student in the system	<pre> C:\Users\timlo\iCloudDrive\Assignment\Programming\Group22_Assignment 2\Assignment2\64\Debug\Assignment2.exe =====Student(s)===== Name ID Form BM BI BC MATH SCI SEJ GEO ----- 1) AIDEN CARTER 17MNO1017 2 76 82 80 88 85 79 90 2) ALEX ALBON 21ACD2021 3 12 24 54 23 56 87 68 3) ALI MALOU 23JGF1234 1 12 12 12 12 12 100 4) ALICE BELL 47ABC2047 3 80 85 90 95 70 75 80 5) AMELIA KING 13YZG2013 3 80 85 90 95 70 75 100 6) AMELIA YOUNG 45VMX2045 1 90 85 88 84 87 92 80 7) AVA CLARK 11MNO2011 2 76 84 80 85 90 91 75 8) BENJAMIN SCOTT 14ABC1014 1 76 80 85 88 90 92 70 9) CARLOZ SAINZ 12BFD1234 3 45 45 45 45 45 45 45 10) CHARLES WANG 33ABC2033 3 65 70 75 80 85 90 95 11) CHARLOTTE HARRIS 09GHI2009 3 82 88 94 90 76 72 80 12) CHARLOTTE SCOTT 50JKL1050 3 85 90 92 94 89 88 87 13) CHECO PEREZ 23HDB2345 3 54 56 56 56 56 56 56 14) CHECO PEREZ 23HFG1234 3 34 34 34 34 34 34 34 15) MEG GRIFFIN 12GHG7653 3 29 58 12 58 0 100 90 ===== DELETE >> Please key in the student id that you want to delete : </pre>

Delete	<p>User are allow to choose the student to delete by entering the Student ID</p> <p>After deleting a student, user can choose to delete another student or not.</p>	<pre> C:\Users\timlo\iCloudDrive\Assigment\Programming\Group22_Assignment 2\Assignment2\Debug\Assignment2.exe =====Student(s)===== Name ID Form BM BI BC MATH SCI SEJ GEO ----- 1) AIDEN CARTER 17MNO1017 2 76 82 80 88 85 79 90 2) ALEX ALBON 21ACD2021 3 12 24 54 23 56 87 68 3) ALI MALOU 23JGF1234 1 12 12 12 12 12 12 100 4) ALICE BELL 47ABC2047 3 80 85 90 95 70 75 80 5) AMELIA KING 13YZG2013 3 80 85 90 95 70 75 100 6) AMELIA YOUNG 45VWX2045 1 90 85 88 84 87 92 80 7) AVA CLARK 11MNO2011 2 76 84 80 85 90 91 75 8) BENJAMIN SCOTT 14ABC1014 1 76 80 85 88 90 92 70 9) CARLOZ SAINZ 12BFD1234 3 45 45 45 45 45 45 45 10) CHARLES WANG 33ABC2033 3 65 70 75 80 85 90 95 11) CHARLOTTE HARRIS 09GHI2009 3 82 88 94 90 76 72 80 12) CHARLOTTE SCOTT 50JKL1050 3 85 90 92 94 89 88 87 13) CHECO PEREZ 23HDB2345 3 54 56 56 56 56 56 56 14) CHECO PEREZ 23HFG1234 3 34 34 34 34 34 34 34 ===== DELETE >> Please key in the student id that you want to delete : 23HFG1234 >>Chosen Student<< Student ID: 23HFG1234 Name: CHECO PEREZ Form: 3 BM: 34 BI: 34 BC: 34 MATH: 34 SCI: 34 SEJ: 34 GEO: 34 <<* The chosen Student's Record will be deleted *>> Do you want to continue to delete other record?(y/n): n Student List updated successfully! Press any key to continue . . . </pre>
Delete	<p>Input validation for delete module.</p> <p>If an unknown or un-recorded ID are entered, an error message will be display and user will be ask to enter again.</p> <p>Any invalid option will cause the program to display an error message</p>	<pre> C:\Users\timlo\iCloudDrive\Assigment\Programming\Group22_Assignment 2\Assignment2\Debug\Assignment2.exe =====Student(s)===== Name ID Form BM BI BC MATH SCI SEJ GEO ----- 1) AIDEN CARTER 17MNO1017 2 76 82 80 88 85 79 90 2) ALEX ALBON 21ACD2021 3 12 24 54 23 56 87 68 3) ALI MALOU 23JGF1234 1 12 12 12 12 12 12 100 4) ALICE BELL 47ABC2047 3 80 85 90 95 70 75 80 5) AMELIA KING 13YZG2013 3 80 85 90 95 70 75 100 6) AMELIA YOUNG 45VWX2045 1 90 85 88 84 87 92 80 7) AVA CLARK 11MNO2011 2 76 84 80 85 90 91 75 8) BENJAMIN SCOTT 14ABC1014 1 76 80 85 88 90 92 70 9) CARLOZ SAINZ 12BFD1234 3 45 45 45 45 45 45 45 10) CHARLES WANG 33ABC2033 3 65 70 75 80 85 90 95 11) CHARLOTTE HARRIS 09GHI2009 3 82 88 94 90 76 72 80 12) CHARLOTTE SCOTT 50JKL1050 3 85 90 92 94 89 88 87 13) CHECO PEREZ 23HDB2345 3 54 56 56 56 56 56 56 ===== DELETE >> Please key in the student id that you want to delete : 50JKL105 Student record not found! Please enter again DELETE >> Please key in the student id that you want to delete : 50JKL1050 >>Chosen Student<< Student ID: 50JKL1050 Name: CHARLOTTE SCOTT Form: 3 BM: 85 BI: 90 BC: 92 MATH: 94 SCI: 89 SEJ: 88 GEO: 87 <<* The chosen Student's Record will be deleted *>> Do you want to continue to delete other record?(y/n): z Invalid input,please enter again Do you want to continue to delete other record?(y/n): </pre>

Search/ Filter	Search/Filter Menu which ease user to look for specific student(s)	<pre> C:\Users\timlo\iCloudDrive\Assigment\Programming\Group22_Assignment 2\Assignment2\x64\Debug\Assignment2.e =====Search/Filter Menu===== Please choose from the following option: 1.) Search ---1 2.) Filter ---2 DisplayAllStudent ---A Exit(toMainMenu) ---X ===== Enter your Option: █ </pre>
Search/ Filter	Input validation for menu.	<pre> C:\Users\timlo\iCloudDrive\Assigment\Programming\Group22_Assignment 2\Assignment2\x64\Debug\Assignment2 =====Search/Filter Menu===== Please choose from the following option: 1.) Search ---1 2.) Filter ---2 DisplayAllStudent ---A Exit(toMainMenu) ---X ===== Enter your Option: c Invalid option! Please Enter again. Enter your Option: █ </pre>

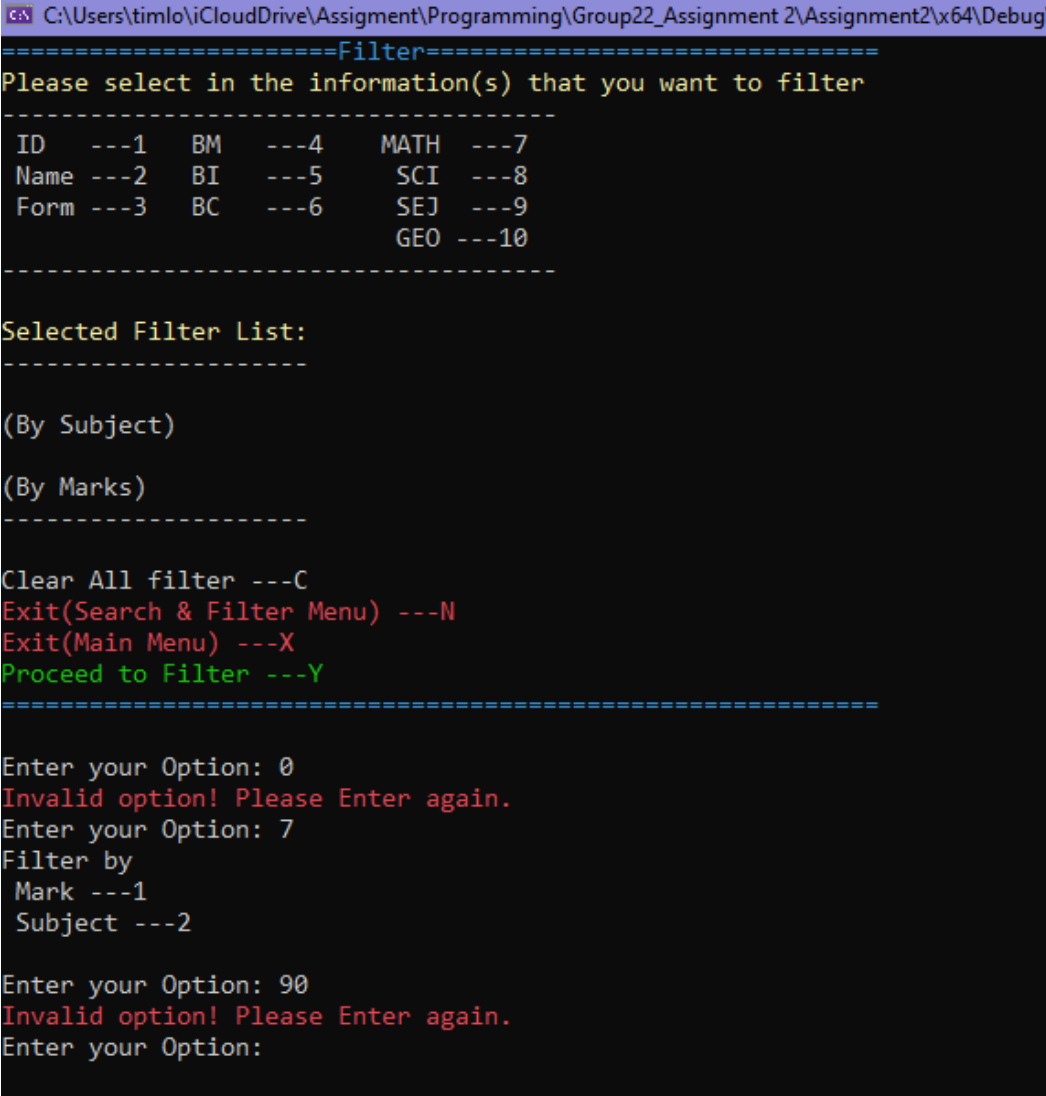
Search	Search module which allow user to search for students using any keyword/search word	<pre> C:\Users\timlo\iCloudDrive\Assignment\Programming\Group22_Assignment 2\Assignment2\64\Debug\Assignment2.exe =====Search===== Please key in the search word (Name,ID,Form): </pre>
Search	<p>Student that match the search word will be display by the program</p> <p>Number of total search result will be displayed</p> <p>User are allow to go to another function(Filter), search again with another search word or exit the module</p>	<pre> C:\Users\timlo\iCloudDrive\Assignment\Programming\Group22_Assignment 2\Assignment2\64\Debug\Assignment2.exe Search Result for keyword: A ===== Name ID Form BM BI BC MATH SCI SEJ GEO ===== 1) AIDEN CARTER 17MNO1017 2 76 82 80 88 85 79 90 2) ALEX ALBON 21ACD2021 3 12 24 54 23 56 87 68 3) ALI MALOU 23JGF1234 1 12 12 12 12 12 12 12 4) ALICE BELL 47ABC2047 3 80 85 90 95 70 75 80 5) AMELIA KING 13YZG2013 3 80 85 90 95 70 75 80 6) AMELIA YOUNG 45VWX2045 2 90 85 88 84 87 92 80 7) AVA CLARK 11MNO2011 2 76 84 80 85 90 91 75 8) BENJAMIN SCOTT 14ABC1014 1 76 80 85 88 90 92 70 9) CARLOZ SAINZ 12BFD1234 3 45 45 45 45 45 45 45 10) CHARLES WANG 33ABC2033 3 65 70 75 80 85 90 95 11) CHARLOTTE HARRIS 09GHI2009 3 82 88 94 90 76 72 80 12) CHARLOTTE SCOTT 50JKL1050 3 85 90 92 94 89 88 87 ===== *Total of 12 search result (s) found. >> Search Again ? << Search ---1 Filter ---2 Exit(toSearchEngineMenu) ---N Exit(toMain Menu) ---X Enter your Option: </pre>

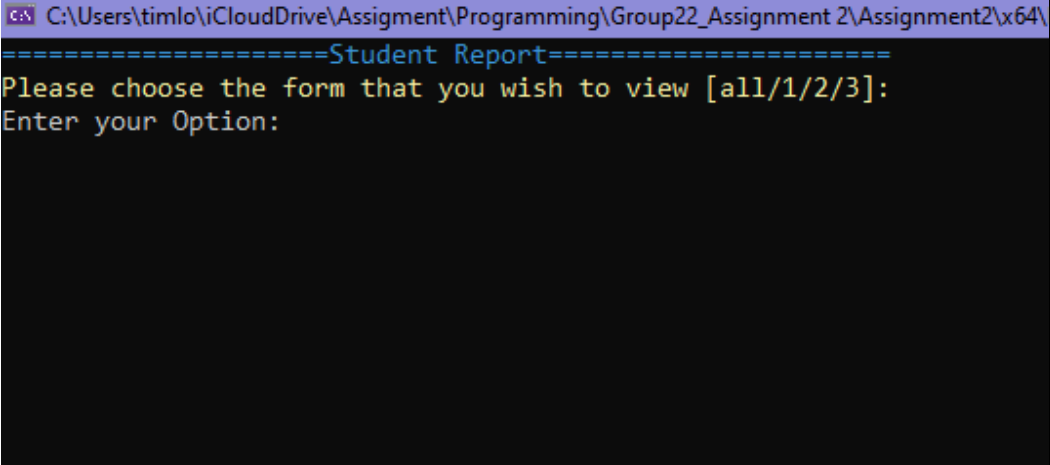
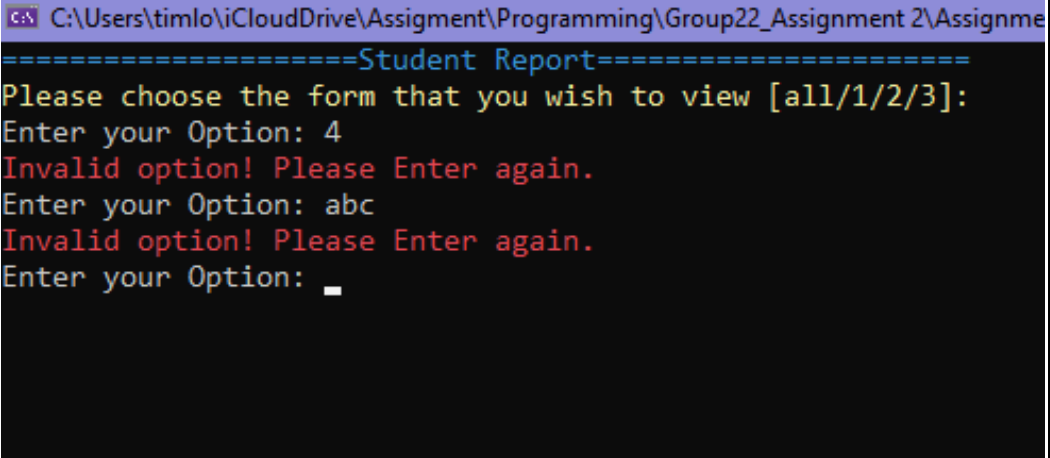
Search	Substring of the student that match the search word will be display in green color	<pre> C:\Users\timlo\iCloudDrive\Assignment\Programming\Group22_Assignment 2\Assignment2\Debug\Assignment2.exe Search Result for keyword: 1 ===== Name ID Form BM BI BC MATH SCI SEJ GEO ===== 1) AIDEN CARTER 17MNO1017 2 76 82 80 88 85 79 90 2) ALEX ALBON 21ACD2021 3 12 24 54 23 56 87 68 3) ALI MALOU 23JGF1234 1 12 12 12 12 12 12 12 4) AMELIA KING 13YZG2013 3 80 85 90 95 70 75 80 5) AVA CLARK 11MNO2011 2 76 84 80 85 90 91 75 6) BENJAMIN SCOTT 14ABC1014 1 76 80 85 88 90 92 70 7) CARLOZ SAINZ 12BFD1234 3 45 45 45 45 45 45 45 8) CHARLOTTE SCOTT 50JKL1050 3 85 90 92 94 89 88 87 9) CHECO PEREZ 23HFG1234 3 34 34 34 34 34 34 34 10) MEG GRIFFIN 12GHG7653 3 29 58 12 58 0 100 90 ===== *Total of 10 search result (s) found. >> Search Again ? << Search ---1 Filter ---2 Exit(toSearchEngineMenu) ---N Exit(toMain Menu) ---X Enter your Option: _ </pre>
Search	<p>When no student has the matching details with the search word, a error message will be display</p> <p>Input Validation for user option.</p>	<pre> C:\Users\timlo\iCloudDrive\Assignment\Programming\Group22_Assignment 2\Assignment2\Debug\Assignment2.exe Search Result for keyword: JKJKJ ===== (No result found!) ===== *Total of 0 search result (s) found. >> Search Again ? << Search ---1 Filter ---2 Exit(toSearchEngineMenu) ---N Exit(toMain Menu) ---X Enter your Option: L Invalid option! Please Enter again. Enter your Option: </pre>

Filter	Filter module that allow user to look for student with specific attributes values	<pre> C:\Users\timlo\iCloudDrive\Assignment\Programming\Group22_Assignment 2\Assignment2\64\Debug\Assignment2.exe =====Filter===== Please select in the information(s) that you want to filter ----- ID ---1 BM ---4 MATH ---7 Name ---2 BI ---5 SCI ---8 Form ---3 BC ---6 SEJ ---9 GEO ---10 ----- Selected Filter List: ----- (By Subject) (By Marks) ----- Clear All filter ---C Exit(Search & Filter Menu) ---N Exit(Main Menu) ---X Proceed to Filter ---Y ===== Enter your Option: </pre>	
Filter	<p>User are allow to choose any of the filter listed</p> <p>Multiple filter at the same filter operation are allowed</p> <p>User can also choose to clear all the applied filter if needed</p>	<pre> C:\Users\timlo\iCloudDrive\Assignment\Programming\Group22_Assignment 2\Assignment2\64\Debug\Assig =====Filter===== Please select in the information(s) that you want to filter ----- ID ---1 BM ---4 MATH ---7 Name ---2 BI ---5 SCI ---8 Form ---3 BC ---6 SEJ ---9 GEO ---10 ----- Selected Filter List: ----- ID: LI Form: 2 (By Subject) MATH (By Marks) GEO: 100 ----- Clear All filter ---C Exit(Search & Filter Menu) ---N Exit(Main Menu) ---X Proceed to Filter ---Y ===== Enter your Option: 6 Filter by Mark ---1 Subject ---2 Enter your Option: </pre>	

Filter	<p>Student with attributes that match all the applied filter will be display</p> <p>If user choose the filter by subject, only the chosen subject will be displayed.</p> <p>User can choose to apply another filter, clear all filter or exit the module</p>	<pre> C:\Users\timlo\iCloudDrive\Assignment\Programming\Group22_Assignment 2\Assignment2\Debug\Assignment2.exe Filtered List ===== Name ID Form BC SCI ===== 1) AIDEN CARTER 17MNO1017 2 80 85 2) ALEX ALBON 21ACD2021 3 54 56 3) ALI MALOU 23JGF1234 1 12 12 4) ALICE BELL 47ABC2047 3 90 70 5) AMELIA KING 13YZG2013 3 90 70 6) AMELIA YOUNG 45VWX2045 2 88 87 7) AVA CLARK 11MNO2011 2 80 90 8) BENJAMIN SCOTT 14ABC1014 1 85 90 9) CARLOZ SAINZ 12BFD1234 3 45 45 10) CHARLES WANG 33ABC2033 3 75 85 11) CHARLOTTE HARRIS 09GHI2009 3 94 76 12) CHARLOTTE SCOTT 50JKL1050 3 92 89 13) CHECO PEREZ 23HDB2345 3 56 56 14) CHECO PEREZ 23HFG1234 3 34 34 15) MEG GRIFFIN 12GHG7653 3 12 0 ===== Filter Applied: ----- (By Subject) BC SCI (By Marks) ----- >> Filter again? << Clear All filter(s) ---1 Add Another Filter ---2 Display All student ---3 Search ---4 Exit to Search/Filter Menu ---N Exit to Main Menu ---X ----- Enter your Option: </pre>
Filter	<p>Filtering by mark allow user to look for any student with the specific mark in the subject</p>	<pre> C:\Users\timlo\iCloudDrive\Assignment\Programming\Group22_Assignment 2\Assignment2\Debug\Assignment2.exe Filtered List ===== Name ID Form BM BI BC MATH SCI SEJ GEO ===== 1) ALI MALOU 23JGF1234 1 12 12 12 12 12 12 100 2) AMELIA KING 13YZG2013 3 80 85 90 95 70 75 100 ===== Filter Applied: ----- (By Subject) (By Marks) GEO: 100 ----- >> Filter again? << Clear All filter(s) ---1 Add Another Filter ---2 Display All student ---3 Search ---4 Exit to Search/Filter Menu ---N Exit to Main Menu ---X ----- Enter your Option: _ </pre>

Filter	<p>By applying multiple filter together, it can help user to look for specific student with matching attribute quickly</p> <p>e.g. Finding Form 1 students that score 100 in Geography</p>	<pre> C:\Users\timlo\CloudDrive\Assigment\Programming\Group22_Assignment 2\Assignment2\Debug\Assignment2.exe Filtered List ===== Name ID Form BM BI BC MATH SCI SEJ GEO ===== 1) ALI MALOU 23JGF1234 1 12 12 12 12 12 12 100 ===== Filter Applied: ----- Form: 1 (By Subject) (By Marks) GEO: 100 ----- >> Filter again? << Clear All filter(s) ---1 Add Another Filter ---2 Display All student ---3 Search ---4 Exit to Search/Filter Menu ---N Exit to Main Menu ---X ----- Enter your Option: </pre>
Filter	<p>If there are no student that match the filter applied, and error message will be displayed</p> <p>Input validation for option.</p>	<pre> C:\Users\timlo\CloudDrive\Assigment\Programming\Group22_Assignment 2\Assignment2\Debug\Assignment2.exe Filtered List ===== (No result found!) ===== Filter Applied: ----- ID: KLLKKLKKLKKLLK (By Subject) (By Marks) ----- >> Filter again? << Clear All filter(s) ---1 Add Another Filter ---2 Display All student ---3 Search ---4 Exit to Search/Filter Menu ---N Exit to Main Menu ---X ----- Enter your Option: 0 Invalid option! Please Enter again. Enter your Option: █ </pre>

Filter	Input Validation. All option in the menu are implemented with proper input validation to avoid any unforeseen scenario in the program	 <pre> C:\Users\timlo\iCloudDrive\Assignment\Programming\Group22_Assignment 2\Assignment2\x64\Debug =====Filter===== Please select in the information(s) that you want to filter ----- ID ---1 BM ---4 MATH ---7 Name ---2 BI ---5 SCI ---8 Form ---3 BC ---6 SEJ ---9 GEO ---10 ----- Selected Filter List: ----- (By Subject) (By Marks) ----- Clear All filter ---C Exit(Search & Filter Menu) ---N Exit(Main Menu) ---X Proceed to Filter ---Y ===== Enter your Option: 0 Invalid option! Please Enter again. Enter your Option: 7 Filter by Mark ---1 Subject ---2 Enter your Option: 90 Invalid option! Please Enter again. Enter your Option: </pre>
--------	--	--

Calculation	Student report module that allow user to view analyzation of student result	 <pre> C:\Users\timlo\iCloudDrive\Assignment\Programming\Group22_Assignment 2\Assignment2\x64\ =====Student Report===== Please choose the form that you wish to view [all/1/2/3]: Enter your Option: </pre>
Calculation	Input validation for the module. Error message will be display and user will be ask to enter again.	 <pre> C:\Users\timlo\iCloudDrive\Assignment\Programming\Group22_Assignment 2\Assignme =====Student Report===== Please choose the form that you wish to view [all/1/2/3]: Enter your Option: 4 Invalid option! Please Enter again. Enter your Option: abc Invalid option! Please Enter again. Enter your Option: _ </pre>

Calculation	By choosing the form, user can review average, standard deviation etc. of the form	<pre> C:\Users\timlo\iCloudDrive\Assigment\Programming\Group22_Assignment 2\Assignment2\x64\Debug\Assign ----- Student Report for Form 2 had been generated. BM --- Average: 76.00 Standard Deviation: 0.00 Highest Score: 76 Student Name: AIDEN CARTER BI --- Average: 83.00 Standard Deviation: 1.00 Highest Score: 84 Student Name: AVA CLARK BC --- Average: 80.00 Standard Deviation: 0.00 Highest Score: 80 Student Name: AIDEN CARTER MATH --- Average: 86.50 Standard Deviation: 1.50 Highest Score: 88 Student Name: AIDEN CARTER SCI --- Average: 87.50 Standard Deviation: 2.50 Highest Score: 90 Student Name: AVA CLARK SEJ --- Average: 85.00 Standard Deviation: 6.00 Highest Score: 91 Student Name: AVA CLARK GEO --- Average: 82.50 Standard Deviation: 7.50 Highest Score: 90 Student Name: AIDEN CARTER ----- Do you want to continue? (Y/N): Enter your Option: </pre>
-------------	--	--

Calculation	User are also allow to review the an overall result which take every recorded student into calculation by choose "all"	<pre> C:\Users\timlo\iCloudDrive\Assigment\Programming\Group22_Assignment 2\Assignment2\x64\Debug\Assign ----- Student Report for All Form(s) had been generated. BM --- Average: 59.73 Standard Deviation: 25.96 Highest Score: 90 Student Name: AMELIA YOUNG (Form 1) BI --- Average: 65.20 Standard Deviation: 24.81 Highest Score: 90 Student Name: CHARLOTTE SCOTT (Form 3) BC --- Average: 65.80 Standard Deviation: 27.65 Highest Score: 94 Student Name: CHARLOTTE HARRIS (Form 3) MATH --- Average: 68.47 Standard Deviation: 27.32 Highest Score: 95 Student Name: ALICE BELL (Form 3) SCI --- Average: 63.00 Standard Deviation: 28.02 Highest Score: 90 Student Name: AVA CLARK (Form 2) SEJ --- Average: 72.53 Standard Deviation: 24.28 Highest Score: 100 Student Name: MEG GRIFFIN (Form 3) GEO --- Average: 76.67 Standard Deviation: 18.79 Highest Score: 100 Student Name: ALI MALOU (Form 1) ----- Do you want to continue? (Y/N): Enter your Option: _ </pre>
-------------	--	--

Calculation	Input validation for the module to avoid unwanted outcome	<pre> C:\Users\timlo\iCloudDrive\Assigment\Programming\Group22_Assignment 2\Assignment2\x64\Debug\Assig ----- Student Report for All Form(s) had been generated. BM --- Average: 62.33 Standard Deviation: 25.46 Highest Score: 90 Student Name: AMELIA YOUNG (Form 1) BI --- Average: 66.33 Standard Deviation: 25.13 Highest Score: 88 Student Name: CHARLOTTE HARRIS (Form 3) BC --- Average: 70.75 Standard Deviation: 23.42 Highest Score: 94 Student Name: CHARLOTTE HARRIS (Form 3) MATH --- Average: 70.08 Standard Deviation: 27.71 Highest Score: 95 Student Name: ALICE BELL (Form 3) SCI --- Average: 68.50 Standard Deviation: 22.23 Highest Score: 90 Student Name: AVA CLARK (Form 2) SEJ --- Average: 72.17 Standard Deviation: 23.02 Highest Score: 92 Student Name: AMELIA YOUNG (Form 1) GEO --- Average: 81.92 Standard Deviation: 15.75 Highest Score: 100 Student Name: ALI MALOU (Form 1) ----- Do you want to continue? (Y/N): Enter your Option: b Invalid option! Please Enter again. Enter your Option: 123 Invalid option! Please Enter again. Enter your Option: </pre>
-------------	---	--

Appendix

```

#include <iostream>
#include <fstream>
#include <cctype>
#include <cstring>
#include <string>
#include <iomanip>
#include <cmath>
using namespace std;

//Global Variables
ifstream inFile;
int studNum = 0;
const int stud_max = 1000;
struct student {
    string name = {}, ID = {}, form = {};
    int markList[7] = {};
    bool found = false;
    int resultType = 0;
};
student studList[stud_max];

double totalMarkArr[4][7] =
{ { 0,0,0,0,0,0,0 }, { 0,0,0,0,0,0,0 }, { 0,0,0,0,0,0,0 }, { 0,0,0,0,0,0,0 } };
double avgArr[4][7];
int highestMarkArr[4][7];
string highestMarkStud[4][7];
string all_highestForm[7];
double stdevNumeratorArr[4][7];
double stdevArr[4][7];

const string studInfo[3] = { "ID","Name","form" }, subject[7] =
{ "BM","BI","BC","MATH","SCI","SEJ","GEO" };
const string cyan = "36";
const string brightRed = "91";
const string brightYellow = "93";
const string grey = "90";
const string brightMagenta = "94";
const string brightGreen = "92";

//Function Prototypes
bool idExistsInFile(const string& ID);
int getValidMarks(const string& subject);
bool isValidID(const string& ID);
bool isNumber(const string& str);

```



```

bool isValidName(string name);
void addStudent();
void displayCalc();
void calculate(double f1, double f2, double f3);
string askOption(string errorType);
string toUpperString(string& input);
string setcolour(string colourcode);
string resetcolour();
void printUI(int type);
int studNum_counter();
void studGroup();
void studDisplay();
void alphabeticSort();
string search();
string filter();
void searchFilterMenu();
int check(string id);
string inputstr(string type);
int inputnum();
string ask(string question);
int mark();
string form();
void save_file(string type);
void edit(string id);
void del(string id);
void edit_and_del();
bool fileExistOrEmpty();

//////////////////////////////////*Main Function*//////////////////////////////////
int main() {
    inFile.open("student.txt");
    string mainMenuOption;

    do {
        inFile.close();
        studNum_counter();
        studGroup();
        system("cls");
        cout << setcolour(cyan) << "=====Main
Menu>>===== " << resetcolour() << endl;
        cout << "Welcome to the JKJT Student Info Manager.\nPlease choose from the
following option" << endl << endl;
        cout << "1.) Add new student(s) -----1" << endl;
        cout << "2.) Delete/Edit student(s) -----2" << endl;
        cout << "3.) Search/Filter student(s) ---3" << endl;
        cout << "4.) Student's Mark Report-----4" << endl;
    } while (mainMenuOption != "4");
}

```

```

        cout << "\nDisplay Student(s) List ---A" << endl;
        cout << setcolour(brightRed) << "Exit the program ---X" << resetcolour() <<
endl;
        cout << setcolour(cyan) <<
"===== " <<
resetcolour() << endl;
        mainMenuOption = askOption("mainMenu");

        if (mainMenuOption == "1") {
            addStudent();
        }
        else if (mainMenuOption == "2") {
            while (fileExistOrEmpty()) {
                edit_and_del();
                break;
            }
        }
        else if (mainMenuOption == "3") {
            while (fileExistOrEmpty()) {
                searchFilterMenu();
                break;
            }
        }
        else if (mainMenuOption == "4") {
            while (fileExistOrEmpty()) {
                displayCalc();
                break;
            }
        }
        else if (mainMenuOption == "A") {
            while (fileExistOrEmpty()) {
                system("cls");
                studDisplay();
                cout << "*Total of " << studNum << " student(s) recorded." <<
endl;
                system("pause");
                break;
            }
        }
    } while (mainMenuOption != "X");
}

//////////////////////////////////*Function
Defining*//////////////////////////////////

```

```
//-----Universal Function-----  
----  
bool fileExistOrEmpty() {  
    inFile.open("student.txt");  
    if (inFile.is_open() && studNum != 0) {  
        inFile.close();  
        return true;  
    }  
    else {  
        cout << setcolour(brightRed) << "\nPlease add students record into the system  
before edit/delete/search/filter/view student report!" << resetcolour() << endl;  
        system("pause");  
        inFile.close();  
        return false;  
    }  
}  
  
string setcolour(string colourCode) {  
    return "\033[" + colourCode + "m";  
}  
  
string resetcolour() {  
    return "\033[0m";  
}  
  
string setColourSubStr(string text, string colourCode, int length, int start) {  
    return text.substr(0, start) + colourCode + text.substr(start, length) + "\033[0m" +  
text.substr(start + length);  
}  
  
void printUI(int type) {  
    if (type == 1) {  
        cout << setcolour(cyan) << "-----  
-----" << resetcolour() << endl;  
    }  
    else if (type == 2) {  
        cout << setcolour(cyan) <<  
"  
===== "  
===== "<< resetcolour() << endl;  
    }  
    else if (type == 3) {  
        cout << setw(13) << right << "Name" << setw(22) << right << "ID" << setw(11)  
<< right << "Form" << setw(5) << right  
            << "BM" << setw(6) << right << "BI" << setw(6) << right << "BC" <<  
setw(7) << right << "MATH" << setw(6) << right << "SCI"
```

```

        << setw(6) << right << "SEJ" << setw(6) << right << "GEO" << endl;
    }
    else if (type == 4) {
        cout << setcolour(cyan) <<
"===== " << resetcolour() << endl;
    }

}

int studNum_counter() {
    string line;
    inFile.open("student.txt");
    int counter = 0;

    if (!inFile.eof()) {
        while (getline(inFile, line))
            counter++;
    }

    studNum = counter / 4;
    inFile.close();
    return(studNum);
}

//-----Student 1-----
-
bool idExistsInFile(const string& ID)
{
    ifstream inFile("student.txt");
    string line;

    while (getline(inFile, line)) {
        if (line == ID) { // Check if this line matches the ID
            inFile.close();
            return true; // ID found, return true
        }
        for (int i = 0; i < 3; ++i) { // Skip the next 3 lines (name, form, marks)
            getline(inFile, line);
        }
    }
    inFile.close();
    return false; // ID not found, return false
}

bool isNumber(const string& str) {

```

```

        for (char const& c : str) {
            if (!isdigit(c)) {
                return false; //Form enter by user not digit, return false
            }
        }
        return true;
    }

bool isValidName(string name)
{
    for (int i = 0; i < name.length(); i++) {
        if (!isalpha(name[i]) && !(name[i] == ' '))
            return false; //If name enter by user consist of digit, return false
    }
    return true;
}

bool isValidID(const string& ID) {
    //Data validation function (Check if the id enter by user is in correct form)
    if (ID.length() != 9) {
        cout << setcolour(brightRed) << "ID must be exactly 9 characters long." <<
resetcolour() << endl;
        return false;
    }

    if (!isdigit(ID[0]) || !isdigit(ID[1])) {
        cout << setcolour(brightRed) << "First two characters must be digits." <<
resetcolour() << endl;
        return false;
    }

    if (!isalpha(ID[2]) || !isalpha(ID[3]) || !isalpha(ID[4])) {
        cout << setcolour(brightRed) << "Characters 3 to 5 must be letters." <<
resetcolour() << endl;
        return false;
    }

    if (!isdigit(ID[5]) || !isdigit(ID[6]) || !isdigit(ID[7]) || !isdigit(ID[8])) {
        cout << setcolour(brightRed) << "Last four characters must be digits." <<
resetcolour() << endl;
        return false;
    }

    return true;
}

```

```

int getValidMarks(const string& subject) {
    //Check if the marks enter by user is an positive integer
    string input;
    int marks;
    cout << subject << ": ";
    while (true) {
        cin >> input;
        if (isNumber(input)) {
            marks = stoi(input);
            if (marks >= 0 && marks <= 100) {
                break;
            }
        }
        cout << setcolour(brightRed) << "Please enter valid marks (0-100)! " <<
resetcolour() << endl;
        cout << subject << ": ";
    }
    return marks;
}

void addStudent() {
    system("cls");
    string name, ID, form;
    int BM, BI, BC, MATH, SCI, SEJ, GEO;
    string option;
    ofstream wfile;

    do {
        wfile.open("student.txt", ios::app);
        system("cls");
        cout << setcolour(cyan) << "=====Add New Student===== "
<< resetcolour() << endl;
        cout << " Example of Student's Record: " << setcolour(grey) << "\n(Please follow
the given format)\n\n" << resetcolour() << "Student ID : 23ABC1234 " << setcolour(grey) <<
"(2Digits3Chars4Digits)" << resetcolour() << "\nName : Peter Griffin\nForm : 1 " <<
setcolour(grey) << "(Only form 1-3 are allowed)" << resetcolour() << "\nBM : 100\nBC :
100\nBI : 100\nMATH : 100\nSCI : 100\nSEJ : 100\nGEO : 100\n";
        printUI(4);
        cout << setcolour(brightYellow) << "Please enter student information and
marks\n" << resetcolour();
        bool validID = false;
        do {
            cout << "Student ID (Eg: 12ACD1234): ";
            getline(cin, ID);

            // Convert to uppercase

```

```

        for (char& c : ID) {
            c = toupper(c);
        }

        // Check if the ID format is valid
        validID = isValidID(ID);

        // Check if the ID already exists
        if (validID && idExistsInFile(ID)) {
            cout << setcolour(brightRed) << "This ID already exists! Please
enter a different ID." << resetcolour() << endl;
            validID = false;
        }

    } while (!validID);

    bool validName = false;
    while (!validName) {
        cout << "Student name: ";
        getline(cin, name);

        // Check if the name consist only alphabet
        if (isValidName(name))
            validName = true;
        else
            cout << setcolour(brightRed) << "Name should only consists of
character!" << resetcolour() << endl;
    }

    // Validate form input
    bool validForm = false;
    do {
        cout << "Form: ";
        cin >> form;
        if (!isNumber(form) || (form != "1" && form != "2" && form != "3")) {
            cout << setcolour(brightRed) << "You can only enter 1/2/3! Please
enter again." << resetcolour() << endl;
        }
        else {
            validForm = true;
        }
    } while (!validForm);

    BM = getValidMarks("BM");
    BI = getValidMarks("BI");
    BC = getValidMarks("BC");

```

```

    MATH = getValidMarks("Math");
    SCI = getValidMarks("Sci");
    SEJ = getValidMarks("Sejarah");
    GEO = getValidMarks("Geo");

    //Write data to file
    if (studNum == 0 || !fileExistOrEmpty())
        wfile << ID << endl;
    else
        wfile << endl << ID << endl;
    wfile << name << endl;
    wfile << form << endl;
    wfile << BM << " " << BI << " " << BC << " " << MATH << " " << SCI << " "
<< SEJ << " " << GEO;

    cout << "\nDo you want to key in another record? (Y/N) \n";
    cin.ignore();
    option = askOption("addMenu");
    cout << endl;
    wfile.close();
    } while (option == "Y");
}

```

```

void calculate(double f1, double f2, double f3) {

    for (int a = 0; a < 4; a++) {
        for (int b = 0; b < 7; b++) {
            totalMarkArr[a][b] = 0;
            avgArr[a][b] = 0;
            stdevArr[a][b] = 0;
            stdevNumeratorArr[a][b] = 0;
            highestMarkArr[a][b] = 0;
        }
    }

    for (int x = 0; x < studNum; x++) {
        if (studList[x].form == "1") {
            for (int i = 0; i < 7; i++) {
                //find highest mark
                if (f1 == 1) {
                    highestMarkArr[0][i] = studList[x].markList[i];
                    highestMarkStud[0][i] = studList[x].name;
                }
                else {
                    if (highestMarkArr[0][i] < studList[x].markList[i]) {
                        highestMarkArr[0][i] = studList[x].markList[i];

```



```

                                highestMarkStud[0][i] = studList[x].name;
                                }
                                }
                                //find average -- save in array
                                totalMarkArr[0][i] += studList[x].markList[i]; //find total marks
for each subject
                                avgArr[0][i] = double(totalMarkArr[0][i]) / f1;
                                }
                                }
                                else if (studList[x].form == "2") {
                                    for (int i = 0; i < 7; i++) {
                                        if (f2 == 1) {
                                            highestMarkArr[1][i] = studList[x].markList[i];
                                            highestMarkStud[1][i] = studList[x].name;
                                        }
                                        else {
                                            if (highestMarkArr[1][i] < studList[x].markList[i]) {
                                                highestMarkArr[1][i] = studList[x].markList[i];
                                                highestMarkStud[1][i] = studList[x].name;
                                            }
                                        }
                                        totalMarkArr[1][i] += studList[x].markList[i];
                                        avgArr[1][i] = double(totalMarkArr[1][i]) / f2;
                                    }
                                }
                                else if (studList[x].form == "3") {
                                    for (int i = 0; i < 7; i++) {
                                        if (f3 == 1) {
                                            highestMarkArr[2][i] = studList[x].markList[i];
                                            highestMarkStud[2][i] = studList[x].name;
                                        }
                                        else {
                                            if (highestMarkArr[2][i] < studList[x].markList[i]) {
                                                highestMarkArr[2][i] = studList[x].markList[i];
                                                highestMarkStud[2][i] = studList[x].name;
                                            }
                                        }
                                        //find average -- save in array
                                        totalMarkArr[2][i] += studList[x].markList[i]; //find total marks
for each subject
                                            avgArr[2][i] = double(totalMarkArr[2][i]) / f3;
                                            }
                                }

                                for (int i = 0; i < 7; i++) {
                                    if (x == 0) {

```

```

        highestMarkArr[3][i] = studList[x].markList[i];
        highestMarkStud[3][i] = studList[x].name;
        all_highestForm[i] = studList[x].form;
    }
    else {
        if (highestMarkArr[3][i] < studList[x].markList[i]) {
            highestMarkArr[3][i] = studList[x].markList[i];
            highestMarkStud[3][i] = studList[x].name;
            all_highestForm[i] = studList[x].form;
        }
    }
    totalMarkArr[3][i] += studList[x].markList[i];
    avgArr[3][i] = double(totalMarkArr[3][i] / studNum); //find average for
f1+f2+f3
    }
    }

    //stdev
    for (int x = 0; x < studNum; x++) { //Find the numerator for standard deviation [Sum of
(x - mean)^2]
        if (studList[x].form == "1") {
            for (int i = 0; i < 7; i++) {
                stdevNumeratorArr[0][i] += pow(studList[x].markList[i] -
avgArr[0][i], 2);
            }
        }
        else if (studList[x].form == "2") {
            for (int i = 0; i < 7; i++) {
                stdevNumeratorArr[1][i] += pow(studList[x].markList[i] -
avgArr[1][i], 2);
            }
        }
        else if (studList[x].form == "3") {
            for (int i = 0; i < 7; i++) {
                stdevNumeratorArr[2][i] += pow(studList[x].markList[i] -
avgArr[2][i], 2);
            }
        }
        for (int i = 0; i < 7; i++) {
            stdevNumeratorArr[3][i] += pow(studList[x].markList[i] - avgArr[3][i],
2);
        }
    }

    for (int i = 0; i < 7; i++) { //Calculate standard deviation
        if (f1 > 1)

```

```

        stdevArr[0][i] = sqrt(stdevNumeratorArr[0][i] / f1);
    if (f2 > 1)
        stdevArr[1][i] = sqrt(stdevNumeratorArr[1][i] / f2);
    if (f3 > 1)
        stdevArr[2][i] = sqrt(stdevNumeratorArr[2][i] / f3);
    if (studNum > 1)
        stdevArr[3][i] = sqrt(stdevNumeratorArr[3][i] / studNum);
    }
}

void displayCalc()
{
    string p4view; //User option to choose which form's student report to generate
    string option = "Y";

    do {
        studNum_counter();
        double f1 = 0, f2 = 0, f3 = 0;
        for (int x = 0; x < studNum; x++) {
            if (studList[x].form == "1") {
                f1++; //total number of form 1 student
            }
            else if (studList[x].form == "2") {
                f2++;
            }
            else if (studList[x].form == "3") {
                f3++;
            }
        }
        system("cls");

        cout << setcolour(cyan) << "=====Student
Report===== " << resetcolour() << endl;
        cout << setcolour(brightYellow) << "Please choose the form that you wish to
view [all/1/2/3]: " << resetcolour();
        p4view = askOption("p4view");
        cout << setcolour(cyan) <<
"===== " <<
resetcolour() << endl;

        system("cls");
        cout << "-----" << endl;
        if (p4view == "ALL")
            cout << "Student Report for " << setcolour("44") << "All Form(s)" <<
resetcolour() << " had been generated." << endl;
        else

```

```

        cout << "Student Report for " << setcolour("44") << "Form " + p4view <<
resetcolour() << " had been generated." << endl;
        cout << endl;
        calculate(f1, f2, f3);
        cout << fixed << setprecision(2);
        if (p4view == "1") { //Form 1 student report
            for (int k = 0; k < 7; k++) {
                cout << setcolour(brightGreen) << subject[k] << endl;
                cout << "---" << resetcolour() << endl;
                cout << setcolour(brightYellow) << "Average: " <<
setcolour(cyan) << avgArr[0][k] << resetcolour();
                cout << setcolour(brightYellow) << "\nStandard Deviation: " <<
setcolour(cyan) << stdevArr[0][k] << resetcolour();
                cout << setcolour(brightYellow) << "\nHighest Score: " <<
setcolour(cyan) << highestMarkArr[0][k] << resetcolour();
                cout << setcolour(brightYellow) << "\nStudent Name: " <<
setcolour(cyan) << highestMarkStud[0][k] << resetcolour() << endl << endl;
            }
        }

        else if (p4view == "2") { //Form 2 student report
            for (int k = 0; k < 7; k++) {
                cout << setcolour(brightGreen) << subject[k] << endl;
                cout << "---" << resetcolour() << endl;
                cout << setcolour(brightYellow) << "Average: " <<
setcolour(cyan) << avgArr[1][k] << resetcolour();
                cout << setcolour(brightYellow) << "\nStandard Deviation: " <<
setcolour(cyan) << stdevArr[1][k] << resetcolour();
                cout << setcolour(brightYellow) << "\nHighest Score: " <<
setcolour(cyan) << highestMarkArr[1][k] << resetcolour();
                cout << setcolour(brightYellow) << "\nStudent Name: " <<
setcolour(cyan) << highestMarkStud[1][k] << resetcolour() << endl << endl;
            }
        }

        else if (p4view == "3") { //Form 3 student report
            for (int k = 0; k < 7; k++) {
                cout << setcolour(brightGreen) << subject[k] << endl;
                cout << "---" << resetcolour() << endl;
                cout << setcolour(brightYellow) << "Average: " <<
setcolour(cyan) << avgArr[2][k] << resetcolour();
                cout << setcolour(brightYellow) << "\nStandard Deviation: " <<
setcolour(cyan) << stdevArr[2][k] << resetcolour();
                cout << setcolour(brightYellow) << "\nHighest Score: " <<
setcolour(cyan) << highestMarkArr[2][k] << resetcolour();
                cout << setcolour(brightYellow) << "\nStudent Name: " <<
setcolour(cyan) << highestMarkStud[2][k] << resetcolour() << endl << endl;
            }
        }
    }
}

```

```

    }
}

else if (p4view == "ALL") { //All form student report
    for (int k = 0; k < 7; k++) {
        cout << setcolour(brightGreen) << subject[k] << endl;
        cout << "---" << resetcolour() << endl;
        cout << setcolour(brightYellow) << "Average: " <<
setcolour(cyan) << avgArr[3][k] << resetcolour();
        cout << setcolour(brightYellow) << "\nStandard Deviation: " <<
setcolour(cyan) << stdevArr[3][k] << resetcolour();
        cout << setcolour(brightYellow) << "\nHighest Score: " <<
setcolour(cyan) << highestMarkArr[3][k] << resetcolour();
        cout << setcolour(brightYellow) << "\nStudent Name: " <<
setcolour(cyan) << highestMarkStud[3][k] << " (Form " + all_highestForm[k] + ")" <<
resetcolour() << endl << endl;;
        //cout << setcolour(brightYellow) << "\nForm: " <<
setcolour(cyan) << all_highestForm[k] << resetcolour() << endl << endl;
    }
}
cout << "-----" << endl;
cout << setcolour(brightYellow) << "Do you want to continue? (Y/N): " <<
resetcolour();
    option = askOption("addMenu");
} while (option == "Y");
}

//-----Student 2-----
-
string askOption(string errorType) {
    // Data Validation function
    string option = {};
    cout << "\nEnter your Option: ";
    getline(cin, option);
    option = toUpperString(option);

    if (errorType == "snfMenu") {
        while (option != "1" && option != "2" && option != "A" && option != "N" &&
option != "X") {
            cout << setcolour(brightRed) << "Invalid option! Please Enter again." <<
resetcolour() << endl;
            cout << "Enter your Option: ";
            getline(cin, option);
            option = toUpperString(option);
        }
    }
}

```

```
        else if (errorType == "filterMenu1") {
            while (option != "1" && option != "2" && option != "3" && option != "4" &&
option != "5" && option != "6" &&
                option != "7" && option != "8" && option != "9" && option != "10" &&
option != "Y" && option != "N" && option != "X" && option != "C") {
                cout << setcolour(brightRed) << "Invalid option! Please Enter again." <<
resetcolour() << endl;
                cout << "Enter your Option: ";
                getline(cin, option);
                option = toUpperString(option);
            }
        }
        else if (errorType == "filterSubject") {
            while (option != "1" && option != "2") {
                cout << setcolour(brightRed) << "Invalid option! Please Enter again." <<
resetcolour() << endl;
                cout << "Enter your Option: ";
                getline(cin, option);
                option = toUpperString(option);
            }
        }
        else if (errorType == "filterMenu2") {
            while (option != "1" && option != "2" && option != "3" && option != "4" &&
option != "N" && option != "X") {
                cout << setcolour(brightRed) << "Invalid option! Please Enter again." <<
resetcolour() << endl;
                cout << "Enter your Option: ";
                getline(cin, option);
                option = toUpperString(option);
            }
        }
        else if (errorType == "mainMenu") {
            while (option != "1" && option != "2" && option != "3" && option != "4" &&
option != "X" && option != "A") {
                cout << setcolour(brightRed) << "Invalid option! Please Enter again." <<
resetcolour() << endl;
                cout << "Enter your Option: ";
                getline(cin, option);
                option = toUpperString(option);
            }
        }
        else if (errorType == "edMenu") {
            while (option != "1" && option != "2" && option != "X") {
                cout << setcolour(brightRed) << "Invalid option! Please Enter again." <<
resetcolour() << endl;
```

```

        cout << "Enter your Option: ";
        getline(cin, option);
        option = toUpperString(option);
    }
}
else if (errorType == "addMenu") {
    while (option != "Y" && option != "N") {
        cout << setcolour(brightRed) << "Invalid option! Please Enter again." <<
resetcolour() << endl;
        cout << "Enter your Option: ";
        getline(cin, option);
        option = toUpperString(option);
    }
}
else if (errorType == "p4view") {
    while (option != "1" && option != "2" && option != "3" && option != "ALL") {
        cout << setcolour(brightRed) << "Invalid option! Please Enter again." <<
resetcolour() << endl;
        cout << "Enter your Option: ";
        getline(cin, option);
        option = toUpperString(option);
    }
}
else if (errorType == "filterSM") {
    while (option != "1" && option != "2") {
        cout << setcolour(brightRed) << "Invalid option! Please Enter again." <<
resetcolour() << endl;
        cout << "Enter your Option: ";
        getline(cin, option);
        option = toUpperString(option);
    }
}
return option;
}

void alphabeticSort() {
    //Array to sort student name in the list from alphabetic order
    student temp[stud_max];
    for (int j = 0; j < studNum; j++) {
        for (int k = 0; k < (studNum - 1); k++) {

            if (studList[k].name.compare(studList[k + 1].name) == 1) {
                temp[k] = studList[k];
                studList[k] = studList[k + 1];
                studList[k + 1] = temp[k];
            }

```

```

    }
}

void studDisplay() {
    //Function to display every recorded student in the file
    cout << setcolour(cyan) <<
    "=====Student(s)=====
===== " << resetcolour() << endl;
    printUI(3);
    printUI(2);

    for (int c = 0; c < studNum; c++) {
        cout << " " << setw(3) << right << c + 1 << " " << setw(22) << left <<
studList[c].name << setw(15) << left << studList[c].ID << setw(5) << left << studList[c].form;

        for (int i = 0; i < 7; i++) {
            cout << setw(6) << studList[c].markList[i];
        }
        cout << endl;
    }
    printUI(2);
    printUI(1);
}

void studGroup() {
    //Function to add every recorded student into the array of struct.
    studNum_counter();
    inFile.open("student.txt");
    for (int x = 0; x < studNum; x++) {
        getline(inFile, studList[x].ID);
        getline(inFile, studList[x].name);
        getline(inFile, studList[x].form);

        for (int j = 0; j < 7; j++) {
            inFile >> studList[x].markList[j];
        }
        inFile.ignore();
    }

    for (int x = 0; x < studNum; x++) {
        studList[x].name = toUpperString(studList[x].name); // Store every student's
name as uppercase
    }
    alphabeticSort();
    inFile.close();
}

```



```

}

string toUpperString(string& input) {
    for (int i = 0; i < input.length(); i++) {
        input[i] = toupper(input[i]);
    }
    return input;
    //Convert string to uppercase
}

void toEmptyArr(student(&arrToEmpty)[stud_max]) {
    //Function to empty a array of struct
    student emptyArr[stud_max];
    for (int i = 0; i < stud_max; i++) {
        arrToEmpty[i] = emptyArr[i];
    }
}

void searchFilterMenu() {

    studGroup();
    string menuOption;

    do {
        system("cls");
        cout << setcolour(cyan) << "=====Search/Filter Menu===== "
<< resetcolour() << endl;
        cout << "Please choose from the following option: \n1.) Search ---1\n2.) Filter ---
2\n\nDisplayAllStudent ---A\nExit(toMainMenu) ---X\n" << endl;
        printUI(4);
        while (menuOption != "1" && menuOption != "2" && menuOption != "A" &&
menuOption != "X")
            menuOption = askOption("snfMenu");

        if (menuOption == "1") {
            menuOption = search();
        }
        else if (menuOption == "2") {
            menuOption = filter();
        }
        else if (menuOption == "A") {
            system("cls");
            studDisplay();
            cout << setcolour(brightYellow) << "\n>> Do you want to go to << " <<
resetcolour() << "\nSearch ---1\nFilter ---2\nExit(toSearch/Filter Menu) ---N\nExit(toMain
Menu) ---X\n";

```

```

        menuOption = askOption("snfmenu");
    }
} while (menuOption != "X"); // X as sentinel to exit the function;
}

string search()
{
    int resultNum;
    string searchWord, menuOption;
    student searchResult[stud_max];

    do {
        system("cls");
        toEmptyArr(searchResult); //initialise an array that store student with matching
searchwords
        resultNum = 0;
        for (int i = 0; i < studNum; i++) { //reinitialise each student's found flag to false to
avoid skipped in search loop
            studList[i].found = false;
        }

        cout << setcolour(cyan) <<
"=====Search===== " <<
resetcolour();
        cout << setcolour(brightYellow) << "\nPlease key in the search word
(Name,ID,Form): " << resetcolour();
        getline(cin, searchWord);
        toUpperString(searchWord);
        system("cls");
        cout << "Search Result for keyword: " << searchWord << endl;

        for (int i = 0; i < studNum; i++) { // Loop through all students in the array
StudList
            if (studList[i].found == false) { // avoid multiple search on same student
                if (studList[i].name.find(searchWord) != string::npos) { // search
for keyword in name of each student
                    resultNum++;
                    searchResult[resultNum - 1] = studList[i];
                    searchResult[resultNum - 1].resultType = 1;
                    studList[i].found = true;
                }

                if (studList[i].ID.find(searchWord) != string::npos) { // search for
keyword in ID of each student
                    if (studList[i].found == true) // if the student have matching
info in both name and ID

```

```

        searchResult[resultNum - 1].resultType = 3;
    else {
        resultNum++;
        searchResult[resultNum - 1] = studList[i];
        searchResult[resultNum - 1].resultType = 0;
        studList[i].found = true;
    }
}

if (studList[i].form.find(searchWord) != string::npos) { // proceed
to copy student details if find() doesnot return string :: npos
    if (studList[i].found == true)// if the student have matching
info in both name and ID
        searchResult[resultNum - 1].resultType = 4;

    else {
        resultNum++;
        searchResult[resultNum - 1] = studList[i]; // copy
the student details into the array searchResult
        searchResult[resultNum - 1].resultType = 2;
        studList[i].found = true; // set found = true to avoid
multiple search in next iteration
    }
}

}

if (resultNum > 0) { // only display the array searchResult if at least one student
have the matching searchword
    printUI(2);
    printUI(3);
    printUI(2);
    for (int x = 0; x < resultNum; x++)
    {
        switch (searchResult[x].resultType) { // display coloured text for
the substring of the information that matches the search word
            case 0:
                cout << " " << setw(3) << right << x + 1 << " " <<
setw(21) << left << searchResult[x].name
                << setw(24) << left <<
setColourSubStr(searchResult[x].ID, "\033[92m", searchWord.length(),
searchResult[x].ID.find(searchWord)) << setw(6) << left << searchResult[x].form;
                break;
            case 1:
                cout << " " << setw(3) << right << x + 1 << " " <<
setw(30) << left << setColourSubStr(searchResult[x].name, "\033[92m", searchWord.length(),
searchResult[x].name.find(searchWord))

```

```

        << setw(15) << left << searchResult[x].ID <<
setw(6) << left << searchResult[x].form;
        break;
        case 2:
            cout << " " << setw(3) << right << x + 1 << " ) " <<
setw(21) << left << searchResult[x].name
            << setw(15) << left << searchResult[x].ID <<
setcolour(brightGreen) << setw(6) << left << searchResult[x].form << resetcolour();
            break;
        case 3:
            cout << " " << setw(3) << right << x + 1 << " ) " <<
setw(30) << left << setColourSubStr(searchResult[x].name, "\\033[92m", searchWord.length(),
searchResult[x].name.find(searchWord))
            << setw(24) << left <<
setColourSubStr(searchResult[x].ID, "\\033[92m", searchWord.length(),
searchResult[x].ID.find(searchWord)) << setw(6) << left << searchResult[x].form;
            break;
        case 4:
            cout << " " << setw(3) << right << x + 1 << " ) " <<
setw(21) << left << searchResult[x].name
            << setw(24) << left <<
setColourSubStr(searchResult[x].ID, "\\033[92m", searchWord.length(),
searchResult[x].ID.find(searchWord)) << setcolour(brightGreen) << setw(6) << left <<
searchResult[x].form << resetcolour();
            break;
    }

    for (int i = 0; i < 7; i++) {
        cout << setw(6) << searchResult[x].markList[i];
    }
    cout << endl;
}
printUI(2);
}

else {
    printUI(2);
    cout << setcolour(brightRed) << setw(50) << right << "(No result
found!)" << resetcolour() << endl;
    printUI(2);
}
cout << "*Total of " << resultNum << " search result (s) found." << endl; //
Display the number of results found

```

```

        cout << setcolour(brightYellow) << "\n>> Search Again ? << " << resetcolour()
<< "\nSearch ---1\nFilter ---2\nExit(toSearchEngineMenu) ---N\nExit(toMain Menu) ---X\n ";
        menuOption = askOption("snfMenu"); // Ask user for next operation and loop
again if option != 'X'
        } while (menuOption == "1");

        return menuOption;
    }

string filter() {

    system("cls");
    string filterOption;
    string f_ID = {}, f_form = {}, f_name = {}, menuOption = {};
    int f_markList[7] = { -1,-1,-1,-1,-1,-1,-1 }, f_subjectNum = 0, counter = 0, resultNum;
    string f_subject[7] = {};
    student filterList[stud_max];

    do { // reset the flag in struct array of each student to false to avoid skipping of student
during filter matching
        for (int i = 0; i < studNum; i++)
            studList[i].found = false;
        resultNum = 0; // reset number of result found

        system("cls");
        cout << setcolour(cyan) <<
"=====Filter===== " <<
resetcolour() << endl;
        cout << setcolour(brightYellow) << "Please select in the information(s) that you
want to filter " << resetcolour() << endl;
        cout << "-----" << endl;
        cout << setw(13) << left << " ID ---1 " << setw(13) << left << "BM ---4 " <<
setw(13) << left << "MATH ---7" << endl;
        cout << setw(13) << left << " Name ---2 " << setw(14) << left << "BI ---5 " <<
setw(13) << left << "SCI ---8" << endl;
        cout << setw(13) << left << " Form ---3" << setw(14) << left << "BC ---6" <<
setw(13) << left << "SEJ ---9" << endl;
        cout << setw(36) << right << "GEO ---10" << endl;
        cout << "-----" << endl;

        // Display the selected filter and if nothing is selected, display blank
        cout << setcolour(brightYellow) << "\nSelected Filter List:" << resetcolour() <<
endl;

        cout << "-----" << endl;
        if (f_ID != "")
            cout << " ID: " << f_ID << endl;

```

```

    if (f_name != "")
        cout << " Name: " << f_name << endl;

    if (f_form != "")
        cout << " Form: " << f_form << endl;

    cout << "\n(By Subject)" << endl;
    for (int i = 0; i < 7; i++) {
        if (f_subject[i] == "")
            continue;
        else
            cout << " " << f_subject[i] << endl;
    }

    cout << "\n(By Marks)" << endl;
    for (int i = 0; i < 7; i++) {
        if (f_markList[i] != -1) {
            cout << " " << subject[i] + ": " << f_markList[i] << endl;
        }
    }

    cout << "-----\n" << endl;
    cout << "Clear All filter ---C" << endl;
    cout << setcolour("91") << "Exit(Search & Filter Menu) ---N" << endl;
    cout << "Exit(Main Menu) ---X" << resetcolour() << endl;
    cout << setcolour("92") << "Proceed to Filter ---Y" << resetcolour() << endl;
    cout << setcolour(cyan) <<
    "===== \n" <<
    resetcolour();

    filterOption = askOption("filterMenu1");

    if (filterOption != "N" && filterOption != "X" && filterOption != "C") {
        string markOrsubject = {}; // Variable to store user option

        if (filterOption == "1") { //Prompt user to input the value of chosen filter
            cout << "ID: ";
            getline(cin, f_ID);
            f_ID = toUpperString(f_ID);
        }
        else if (filterOption == "2") {
            cout << "Name: ";
            getline(cin, f_name);
            f_name = toUpperString(f_name);
        }
    }

```

```

        else if (filterOption == "3") {
            cout << "Form: ";
            getline(cin, f_form);
        }
        //If user option is a subject, ask user whether to filter by mark or by whole
subject
        else if (filterOption == "4" || filterOption == "5" || filterOption == "6" ||
filterOption == "7"
                || filterOption == "8" || filterOption == "9" || filterOption == "10")
        {
            cout << "Filter by \n Mark ---1\n Subject ---2" << endl;
            markOrsubject = askOption("filterSM");
            if (markOrsubject == "1") { // if by mark, prompt user to enter the
mark of the subject
                cout << subject[stoi(filterOption) - 4] + ": "; // convert
option of user from string to int and minus by 4 to index to the chosen subject
                cin >> f_markList[stoi(filterOption) - 4];
                //Example: in subject[0] = "BM". To choose BM in the
previous menu,user need to enter "4". To index to the BM,which is 0,minus user option by 4 (4-
4=0)
                cin.ignore();
            }
            else { // if by subject add the subject into the filtered subject array.
                f_subjectNum++;
                f_subject[stoi(filterOption) - 4] = subject[stoi(filterOption)
- 4];
            }
        }
    }

    if (filterOption == "C") { // Clear all applied filter
        for (int i = 0; i < 7; i++) {
            f_markList[i] = -1;
            f_subject[i] = "";
            f_name = ""; f_ID = ""; f_form = "";
        }
    }

    if (filterOption == "Y") { //Proceed to filter
        for (int i = 0; i < studNum; i++) {
            counter = 0; // Every time the student info passes a condition.
counter +1, hence if all condition are passed, counter = 7;
            if (!studList[i].found && studList[i].ID.find(f_ID) != string::npos)
        {

```

```

        if (studList[i].name.find(f_name) != string::npos) { //If no
filter are added for any of the attribute, the students are considered pass for next condition
checking
        if (studList[i].form.find(f_form) != string::npos) {
            for (int j = 0; j < 7; j++) {
                if (studList[i].markList[j] ==
f_markList[j] || f_markList[j] == -1) { // -1 are used to represent empty
                    counter++;
                }
            }
        }
    }
}
if (counter == 7) {
    resultNum++;
    filterList[resultNum - 1] = studList[i]; // If student passed
all the condition, add the student into the result list
    studList[i].found = true;
}
}
system("cls");
cout << "Filtered List" << endl;
int isEmpty = 0;
for (int i = 0; i < 7; i++) {
    if (f_subject[i].empty())
        isEmpty++;
} // variable isEmpty is to check whether user added at least one subject as
filter
if (resultNum > 0)
{
    if (isEmpty == 7) { // Case 1: if no subject are selected, the
program will display all subject marks of the student
        printUI(2);
        printUI(3);
        printUI(1);

        for (int i = 0; i < resultNum; i++) { // display the filtered
student details
            cout << " " << setw(3) << right << i + 1 << " ) " <<
setw(21) << left << filterList[i].name << setw(16) << left << filterList[i].ID << setw(5) << left
<< filterList[i].form;

            for (int j = 0; j < 7; j++) { // display the filtered
student's marks
                cout << setw(6) << left <<
filterList[i].markList[j];

```



```

        }
        cout << endl;
    }
}
else { // Case 2: else the program will only display subject that are
selected.

    printUI(2);
    cout << setw(10) << left << " " << setw(23) << left <<
"Name" << setw(9) << left << "ID" << setw(9) << left << "Form";
    for (int j = 0; j < 7; j++) {
        if (f_subject[j] == subject[j]) // if f_subject[0] (BM)
= subject[0](BM) as if BM is not selected, f_subject[0] will be " " (not equal to subject[0])
        cout << setw(8) << left << f_subject[j];//
Display the label

    }
    cout << endl;
    printUI(2);
    for (int i = 0; i < resultNum; i++) {
        cout << " " << setw(3) << right << i + 1 << " ) " <<
setw(21) << left << filterList[i].name << setw(15) << left << filterList[i].ID << setw(8) << left
<< filterList[i].form;

        for (int j = 0; j < 7; j++) {
            if (f_subject[j] == subject[j])
                cout << setw(8) << left <<
filterList[i].markList[j]; // Display the mark

        }
        cout << endl;

    }

}
printUI(2);
}

else {
    printUI(2);
    cout << setcolour(brightRed) << setw(50) << right << "(No result
found!)" << resetcolour() << endl;
    printUI(2);
}

cout << setcolour(brightYellow) << "\nFilter Applied:" << resetcolour()
<< endl;
cout << "-----" << endl;

if (f_ID != "") // Display the selected filter for clarification during
checking
    cout << " ID: " << f_ID << endl;

```

```

        if (f_name != "")
            cout << " Name: " << f_name << endl;

        if (f_form != "")
            cout << " Form: " << f_form << endl;

        cout << "\n(By Subject)" << endl;
        for (int i = 0; i < 7; i++) {
            if (f_subject[i] == "")
                continue;
            else
                cout << " " << f_subject[i] << endl;
        }
        cout << "\n(By Marks)" << endl;
        for (int i = 0; i < 7; i++) {
            if (f_markList[i] != -1) {
                cout << " " << subject[i] + ": " << f_markList[i] << endl;
            }
        }
        cout << "-----" << endl;

        cout << setcolour(brightYellow) << "\n>> Filter again? << " <<
resetcolour() << endl;
        cout << setw(27) << left << "Clear All filter(s) ---1" << "Add Another
Filter ---2" << "\nDisplay All student ---3"
            << setw(26) << right << "Search ---4" << "\n\nExit to
Search/Filter Menu ---N\nExit to Main Menu ---X" << endl;
        cout << "-----";

        filterOption = askOption("filterMenu2");

        if (filterOption == "1") { // Reset the applied filter list
            for (int i = 0; i < 7; i++) {
                f_markList[i] = -1;
                f_subject[i] = "";
                f_name = ""; f_ID = ""; f_form = "";
            }
        }
        else if (filterOption == "3") {
            menuOption = "A";
        }
        else if (filterOption == "4") {
            menuOption = "1";
        }
    }
}

```

```

        else { // if "2" was chosen, it will loop back to the previous menu without
resetting the applied filter list

```

```

            menuOption = filterOption;

```

```

        }

```

```

    }

```

```

    else if (filterOption == "N" || filterOption == "X")

```

```

        menuOption = filterOption;

```

```

    } while (menuOption != "N" && menuOption != "X" && menuOption != "1" &&
menuOption != "A");

```

```

    return menuOption; // Return the menuOption so to the filer/searchMenu()

```

```

}

```

```

//-----Student 3-----

```

```

-

```

```

int check(string id) {

```

```

    //check id exist or not

```

```

    studGroup();

```

```

    for (int i = 0; i < studNum; i++) {

```

```

        //read studList and compare to user input id if id exist return student number in
list else return no found message

```

```

        if (id == studList[i].ID) {

```

```

            cout << setcolour(brightYellow) << "\n>>Chosen Student<<" <<
resetcolour() << endl;

```

```

            cout << "Student ID: " << studList[i].ID << "\nName: " <<
studList[i].name << "\nForm: " << studList[i].form << endl;

```

```

            for (int j = 0; j < 7; j++)

```

```

            {

```

```

                cout << subject[j] + ": " << studList[i].markList[j] << endl;

```

```

            }

```

```

            cout << endl;

```

```

            return i;

```

```

        }

```

```

    }

```

```

    return -1;

```

```

}

```

```

string inputstr(string type) {

```

```

    //change input string to lower or upper

```

```

    string str;

```

```

    getline(cin, str);

```

```

    for (int i = 0; i < str.size(); i++) {
        if (type == "tolower") {
            //add 32 to accoring lower ascii value
            if (str[i] >= 'A' && str[i] <= 'Z') {
                str[i] += 32;
            }
        }
        else if (type == "toupper") {
            //minus 32 to accoring upper ascii value
            if (str[i] >= 'a' && str[i] <= 'z') {
                str[i] -= 32;
            }
        }
    }
    return str;
}

int inputnum() {
    //check input number is integer or not

    string num;
    int result = 0;
    getline(cin, num);

    for (int i = 0; i < num.size(); i++) {
        //read input number one by one, if input is number conver to corresponding
        //number, else return invalid number message

        if (num[i] >= '0' && num[i] <= '9') {
            result = result * 10 + (num[i] - '0');
        }
        else {
            return -1;
        }
    }
    return result;
}

string ask(string question) {
    //ask user to continue or not

    string continue_option = "None";

```

```

        while (!(continue_option == "y" || continue_option == "n")) {
            cout << question;
            continue_option = inputstr("tolower");
            if (!(continue_option == "y" || continue_option == "n")) { cout <<
setcolour(brightRed) << "Invalid input,please enter again" << resetcolour() << endl; }
        }

        return continue_option;
    }

int mark() {
    //check mark range

    int num;
    do {
        cout << "Please enter the mark: ";
        num = inputnum();
        if (!(num >= 0 && num <= 100)) {
            cout << setcolour(brightRed) << "Invalid input,please enter again, mark
should between 0 to 100" << resetcolour() << endl;
        }
    } while (!(num >= 0 && num <= 100));
    return num;
}

string form() {
    //check form range

    string num;
    do {
        cout << "Please update the form: ";
        getline(cin, num);
        if (!(num == "1" || num == "2" || num == "3")) {
            cout << setcolour(brightRed) << "Invalid input,please enter again, form
should be between 1 to 3" << resetcolour() << endl;
        }
    } while (!(num == "1" || num == "2" || num == "3"));
    return num;
}

void save_file(string type) {
    //save file for edit or delete

    int count = 0;
    ofstream f;

```

```

f.open("student.txt");

if (f.is_open()) {

    if (type == "edit") {
        //save file by according order

        count = 0;
        while (studList[count].form != "") {
            f << studList[count].ID << endl;
            f << studList[count].name << endl;
            f << studList[count].form << endl;
            f << studList[count].markList[0] << " " <<
studList[count].markList[1] << " " << studList[count].markList[2] << " " <<
studList[count].markList[3] << " " << studList[count].markList[4] << " " <<
studList[count].markList[5] << " " << studList[count].markList[6];
            count++;
            if (studList[count].form == "1" || studList[count].form == "2" ||
studList[count].form == "3") { f << endl; continue; }
        }

    else if (type == "delete") {
        //save file by accoring order, when student record is delete in studList it
will no save

        count = 0;
        while ((studList[count].form != "") || (studList[count + 1].form != "")) {

            if (studList[count].form != "") {
                f << studList[count].ID << endl;
                f << studList[count].name << endl;
                f << studList[count].form << endl;
                f << studList[count].markList[0] << " " <<
studList[count].markList[1] << " " << studList[count].markList[2] << " " <<
studList[count].markList[3] << " " << studList[count].markList[4] << " " <<
studList[count].markList[5] << " " << studList[count].markList[6];
                if (studList[count].form != "" && (studList[count +
1].form != "" || studList[count + 2].form != "")) { f << endl; }
            }

            //clean old studList for not read old record
            studList[count].ID = "";
            studList[count].name = "";
            studList[count].form = "";
            studList[count].markList[0] = 0;

```

```

        studList[count].markList[1] = 0;
        studList[count].markList[2] = 0;
        studList[count].markList[3] = 0;
        studList[count].markList[4] = 0;
        studList[count].markList[5] = 0;
        studList[count].markList[6] = 0;
        count++;
    }
}
f.close();
}

void edit(string id) {
    //edit

    int count = -1; bool found = 0;
    string select, subject_option, continue_option;

    while (count == -1) {
        //read user input id and check
        cout << setcolour(brightGreen) << "EDIT >> " << setcolour(brightYellow) <<
        "Please key in the student id that you want to edit: " << resetcolour();
        count = check(inputstr("toupper"));
        if (count == -1) {
            cout << setcolour(brightRed) << "Student record not found! Please enter
again" << resetcolour() << endl;
        }
    }

    do {
        //read user select edit which information

        select = "0";
        while (!(select == "1" || select == "2" || select == "3")) {
            cout << setcolour(brightYellow) << "Please select the attribute that you
want to edit:\n(Name ---1  Form ---2  Marks ---3)\n\n" << resetcolour() << "User Input: ";
            getline(cin, select);
            if (!(select == "1" || select == "2" || select == "3")) { cout <<
setcolour(brightRed) << "Invalid input,please enter again" << resetcolour() << endl; }
        }

        if (select == "1") {
            cout << "Please update the name: ";
            getline(cin, studList[count].name);

```

```

    }

    else if (select == "2") {
        studList[count].form = form();
    }

    else if (select == "3") {

        subject_option = "None";
        found = 0;
        while (subject_option == "None") {
            //read user edit subject

            cout << "Please enter the subject that you want to change the
mark(BM/BI/BC/MATH/SCI/SEJ/GEO): ";
            subject_option = inputstr("toupper");
            for (int i = 0; i < 7; i++) {
                if (subject_option == subject[i])
            { studList[count].markList[i] = mark(); found = 1; }
                }
            if (!found) { cout << setcolour(brightRed) << "Invalid Input!
Please enter again." << resetcolour() << endl; subject_option = "None"; continue; }
        }

        continue_option = ask("Do you still have anything to edit for this student?(y/n):
");
    } while (continue_option == "y");

    save_file("edit");
}

void del(string id) {
    //delete

    int count = -1;
    while (count == -1) {
        //read user input id and check

        cout << setcolour("94") << "DELETE >> " << setcolour(brightYellow) <<
"Please key in the student id that you want to delete : " << resetcolour();
        count = check(inputstr("toupper"));
        if (count == -1) { cout << setcolour(brightRed) << "Student record not found!
Please enter again" << resetcolour() << endl; continue; }
    }
}

```



```

        cout << setcolour(brightRed) << "<<* The chosen Student's Record will be
deleted *>>" << resetcolour() << endl << endl;
    }

    //let delete record equal to empty
    studList[count].form = "";
    save_file("delete");
}

void edit_and_del() {
    //read user choice for edit or delete

    string id, choice;
    string continue_option = "y", subject = "None";

    while (true) {

        //check record is empty or not
        if (studNum_counter() == 0) { cout << "No record found, Please add new student
first" << endl; system("pause"); break; }

        //dispay menu and read user input for edit or delete
        system("cls");
        cout << setcolour(cyan) << "=====Edit/Delete
Menu===== " << resetcolour() << endl;
        cout << "Please choose from the following option: \n1.) Edit ---1\n2.) Delete ---
2\n\nExit(toMainMenu) ---X\n" << endl;
        printUI(4);
        choice = askOption("edMenu");

        if (choice == "1") {
            //go to edit funtion

            do {
                //clean screen, get new studList and display studList
                system("cls");
                studGroup();
                studDisplay();

                edit(id);
                continue_option = ask("Edit for another student(s)? (y/n) : ");
            } while (continue_option == "y");
            cout << setcolour(brightGreen) << "\nStudent List updated
successfully!\n" << resetcolour();
            system("Pause");

```

```

    }

    else if (choice == "2") {
        //go to del funtion

        do {
            //clean screen, get new studList and display studList
            system("cls");
            studGroup();
            studDisplay();

            del(id);
            continue_option = ask("Do you want to continue to delete other
record?(y/n): ");
            if (studNum_counter() == 0) { cout << endl; break; }
        } while (continue_option == "y");
        cout << setcolour(brightGreen) << "\nStudent List updated
successfully!\n" << resetcolour();
        system("Pause");

    }
    else if (choice == "X") { cout << "\nexit" << endl; break; }
    else { cout << setcolour(brightRed) << "Invalid input,please enter again" <<
resetcolour() << endl; }
    cout << endl;
}
}

```