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

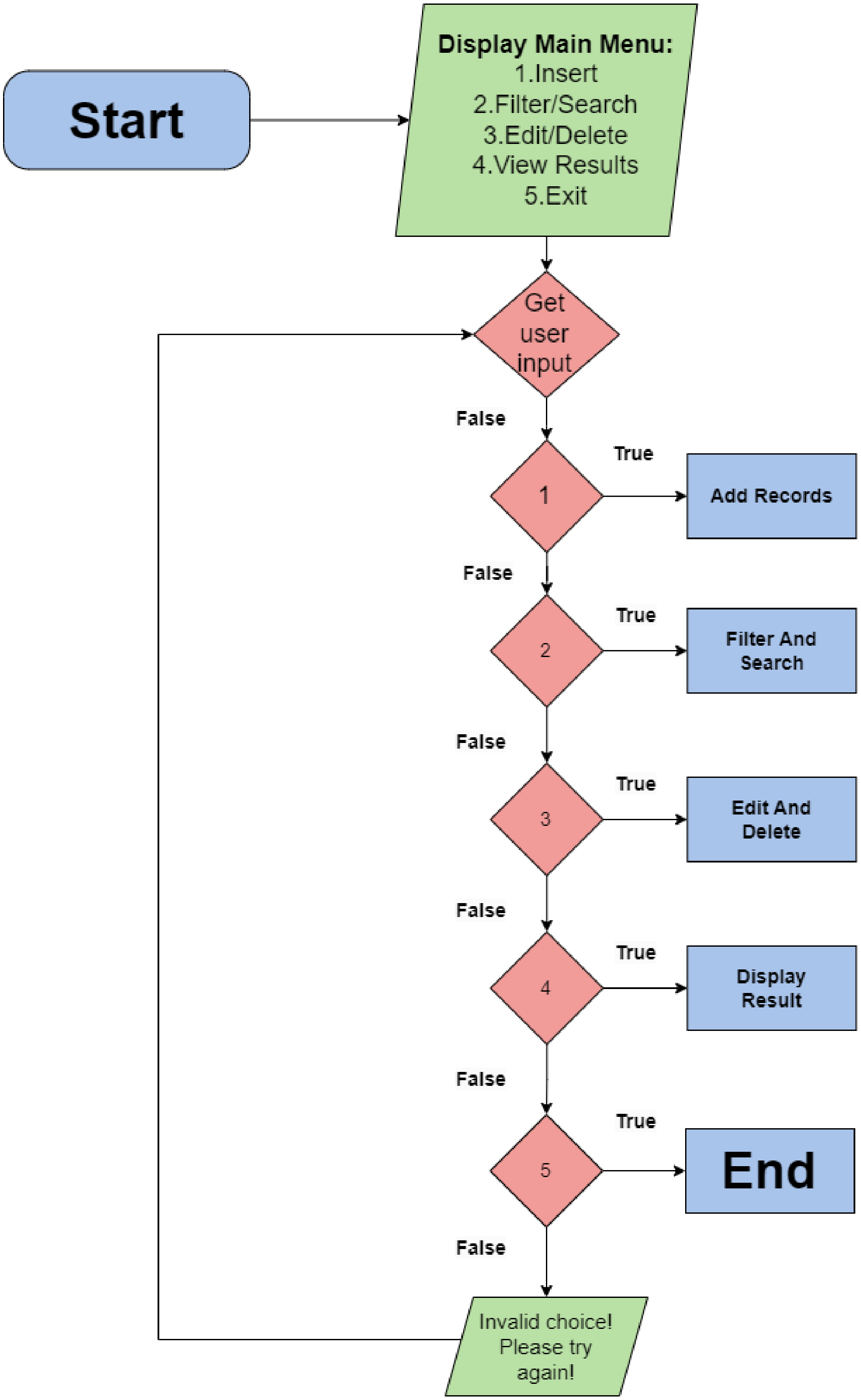
Group No.	**35**		
	Member 1	**Member 2**	**Member 3**
Name:	**Chum Dao Shen**	**Kong Jun Yuan**	**Lim Zhi Hui**
Student ID:	2302347	2302554	2206389
Programme:	Bachelor of Computer Science	Bachelor of Computer Science	Bachelor of Computer Science
Email:	chumdaoshen@1utar.my	kongjunyuan2005@1utar.my	zhihuii@1utar.my

Task Division

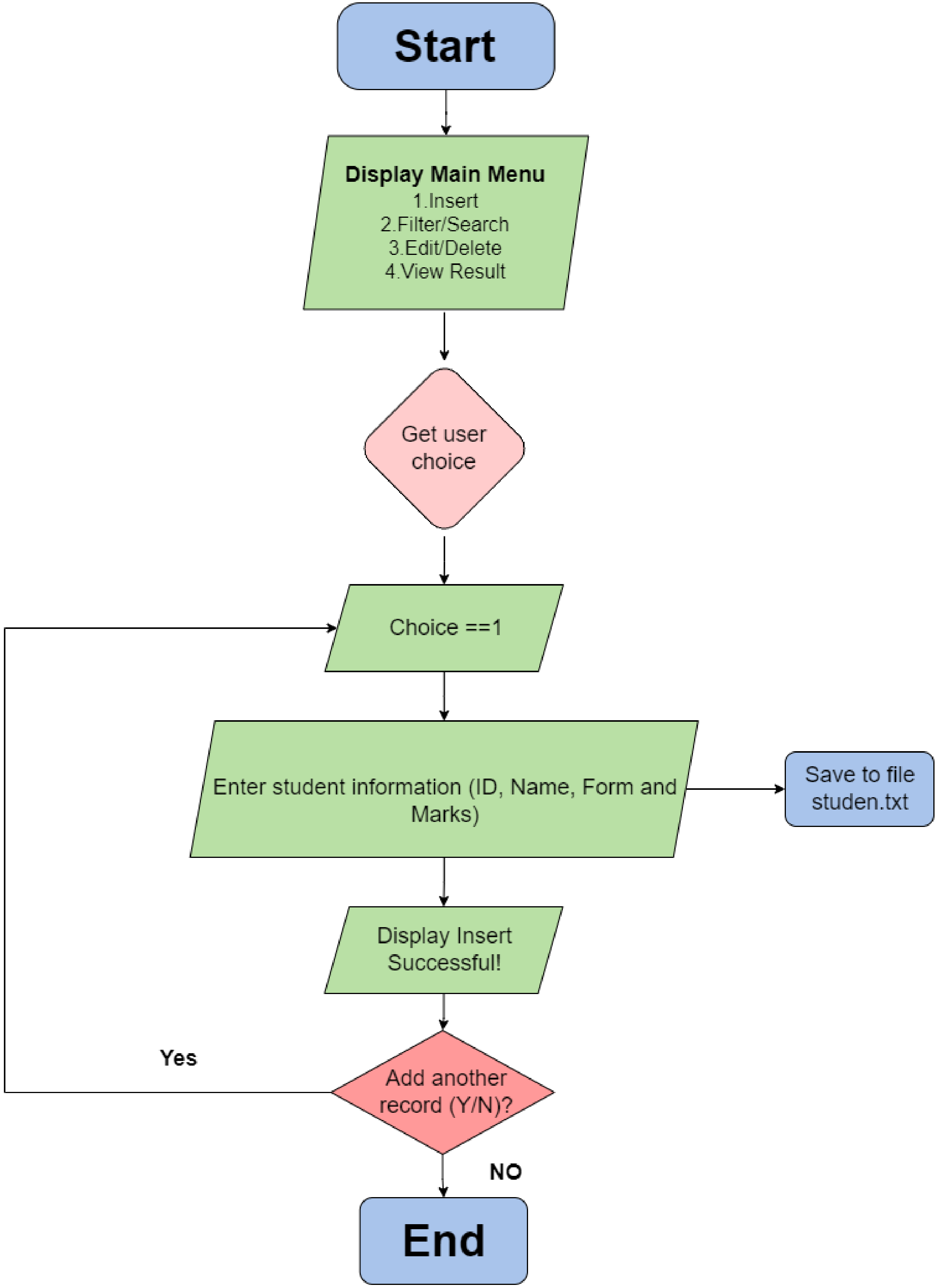
Group Members	Function (modules)	Details
Member 1: **Lim Zhi Hui**	1. Add 2. Calculation	1. Add marks into the system for a list of subjects that taken by the students with student id, student name and form. Check if the student id already exists before adding. 2. Compute average, standard deviation and find the highest score for each subject in each form. 3. Display the results.
Member 2: **Chum Dao Shen**	1. Filter 2. Search	1. Teacher should be able to filter the list through attribute of subjects and form. 2. Basic search function should be provided for teacher to search based on any keyword. 3. Display the results
Member 3: **Kong Jun Yuan**	1. Delete 2. Edit	1. Teacher should be able to delete the record of the selected students. 2. Teacher should be able to edit a particular input except student id. 3. Display the results.

Flowchart

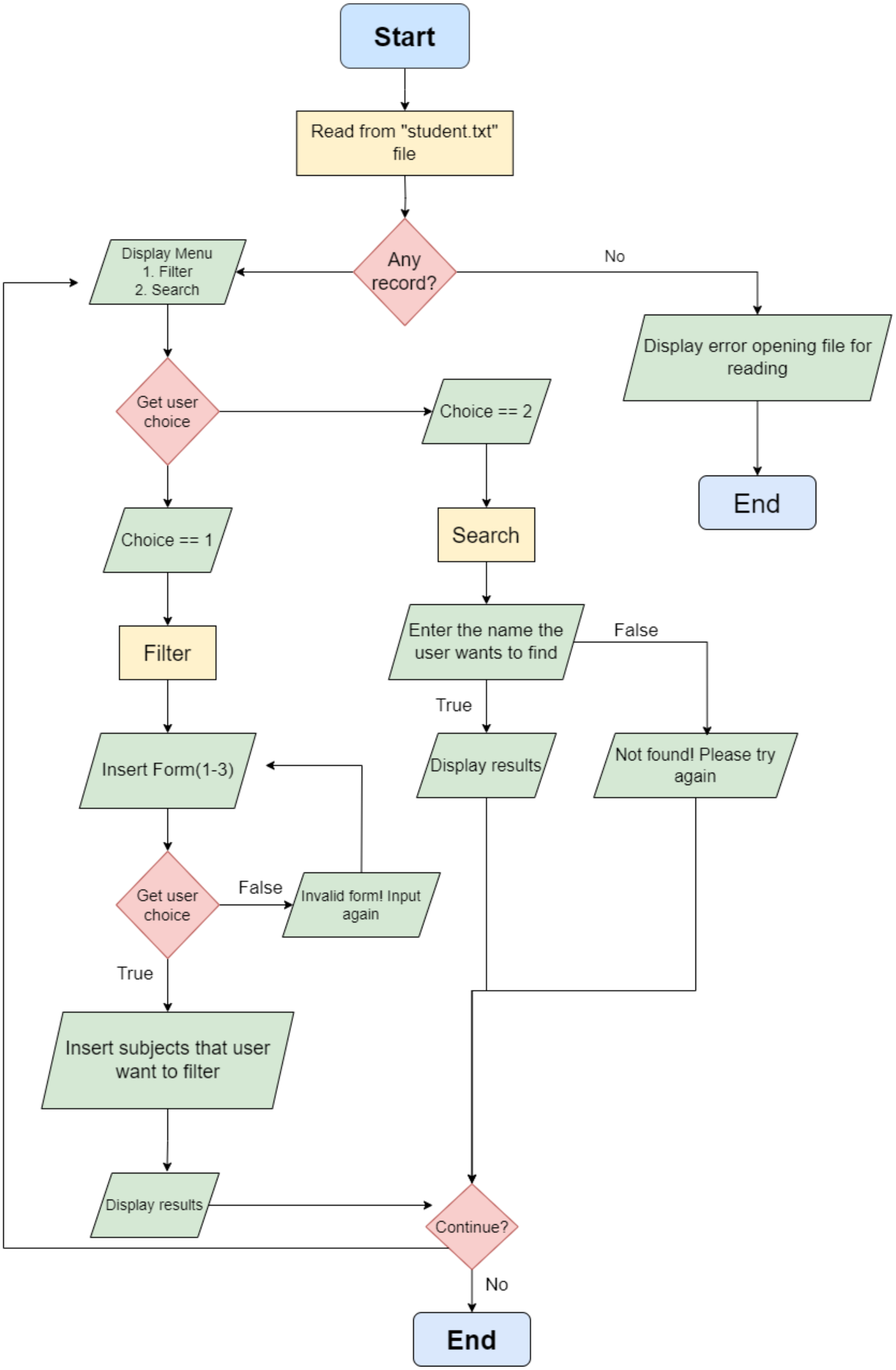
Main Menu Module



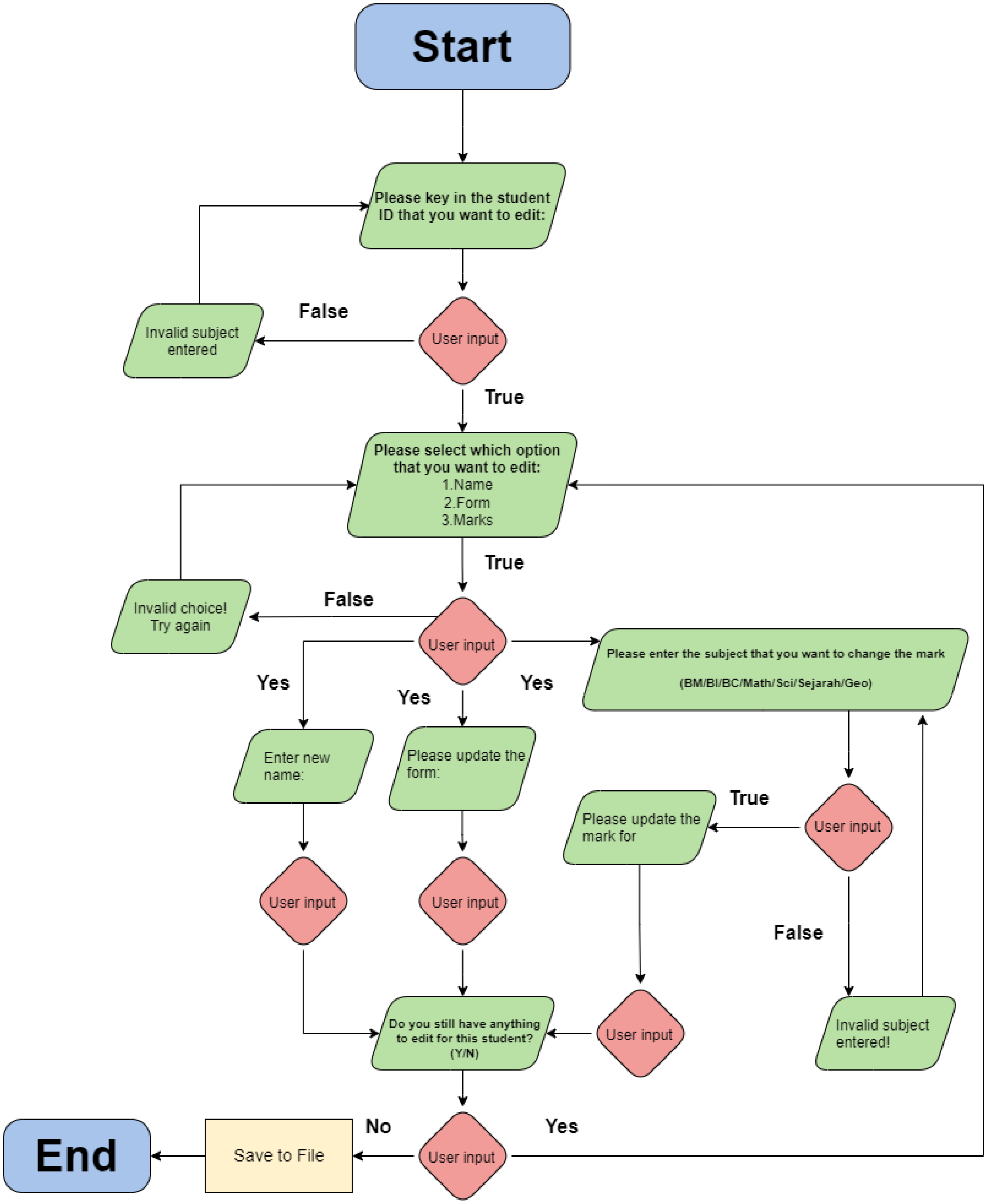
Insert Module



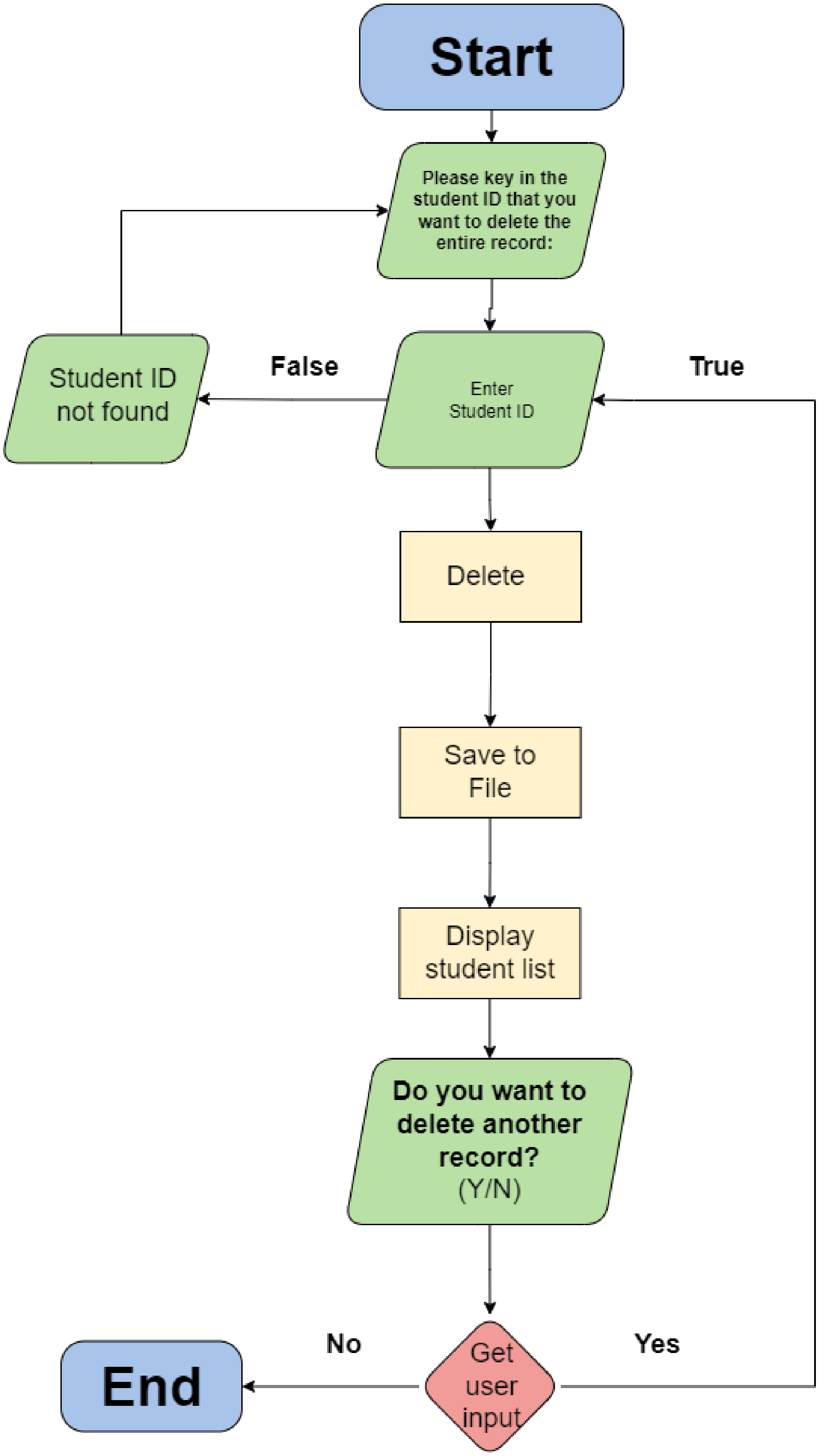
Filter/Search Module


Start
Read from "student.txt" file
Any record?
No
Display error opening file for reading
End
Display Menu
1. Filter
2. Search
Get user choice
Choice == 2
Search
Enter the name the user wants to find
False
True
Display results
Not found! Please try again
Choice == 1
Filter
Insert Form(1-3)
Get user choice
False
Invalid form! Input again
True
Insert subjects that user want to filter
Display results
Continue?
No
End

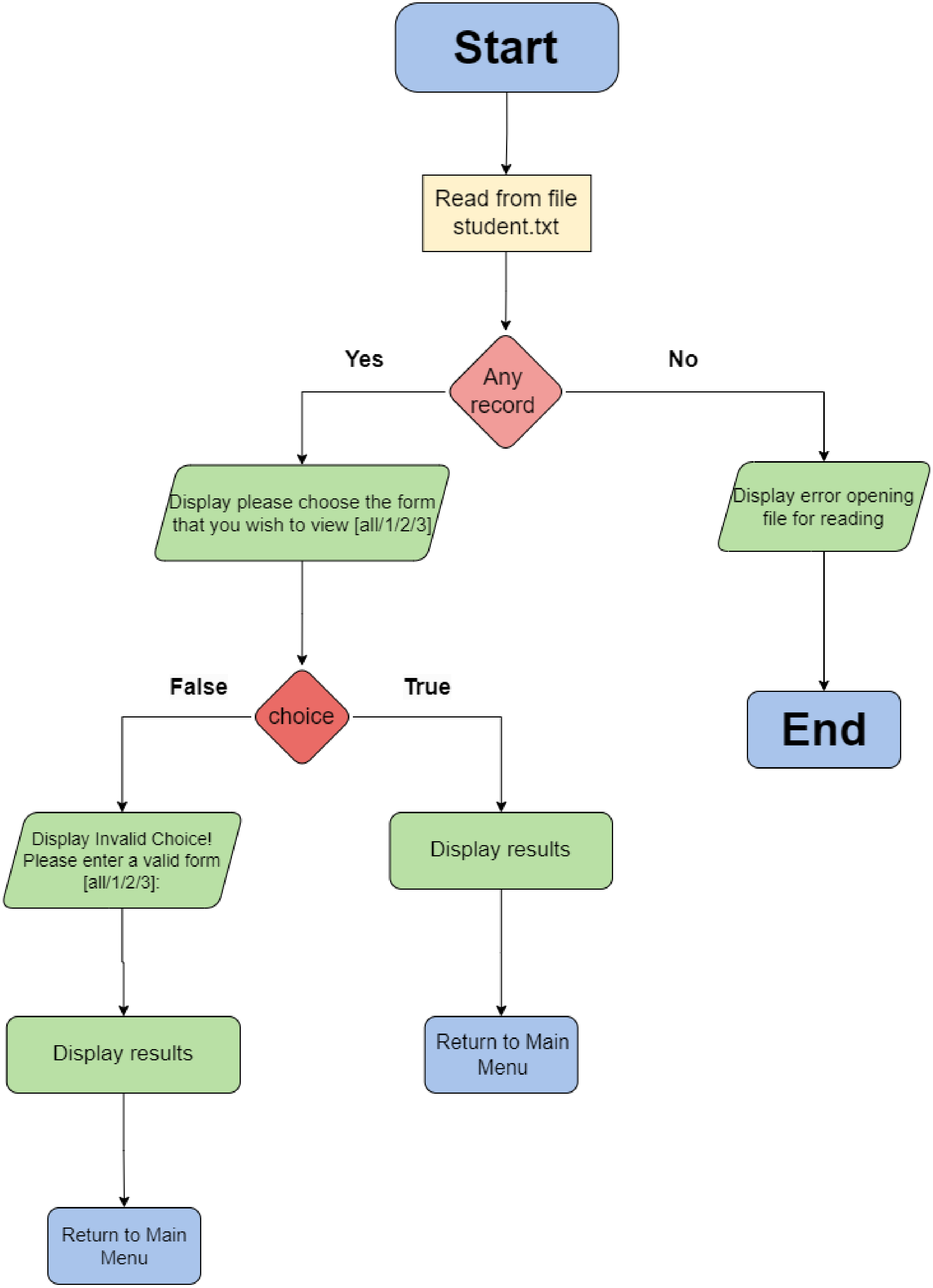
Edit Module


Start
Please key in the student ID that you want to edit:
User input
False
Invalid subject entered
True
Please select which option that you want to edit:
1.Name
2.Form
3.Marks
True
False
Invalid choice! Try again
User input
Yes
Yes
Yes
Please enter the subject that you want to change the mark
(BM/BI/BC/Math/Sci/Sejarah/Geo)
Enter new name:
Please update the form:
Please update the mark for
True
User input
False
User input
User input
User input
Invalid subject entered!
Do you still have anything to edit for this student? (Y/N)
No
Yes
User input
Save to File
End

Delete Module


Start
Please key in the student ID that you want to delete the entire record:
Student ID not found
False
Enter Student ID
True
Delete
Save to File
Display student list
Do you want to delete another record? (Y/N)
No
Get user input
Yes
End

View Result Module


Start
Read from file student.txt
Any record
Yes
No
Display please choose the form that you wish to view [all/1/2/3]
Display error opening file for reading
End
choice
False
True
Display Invalid Choice! Please enter a valid form [all/1/2/3]:
Display results
Display results
Return to Main Menu
Return to Main Menu

Pseudocode

Function for Menu

```
Start
Clear the screen.
Output “Welcome to JKJT”.
Output the main menu with options. 1 for Insert , 2 for Filter/Search, 3 for Modify, 4 for View Result and 5 for Exit.
Prompt the user to enter their choice.
Switch based on user choice:
    Case 1: Call addRecords function.
    Case 2: Call filterAndSearch function.
    Case 3:
    Case 4: Call displayResults function.
    Case 5: Output “Exiting…” then exit the loop.
Repeat the loop until the user chooses option 5 (Exit).
End
```

Function for Insert

```
Start
Prompt user to enter 1 for Insert, 2 for Filter/Search, 3 for Modify, 4 for View Result and 5 for Exit.
//User choose 1 for Insert.
If user selects 1 (Insert);
    Output “Please enter student information and marks.”
    Prompt user to enter student information such as Student ID, Student Name and Form and store in student object.
    If form is not 1, 2, or 3:
        Prompt user to re-enter the correct form such as 1, 2, and 3.
    After entering student information, prompt the user to enter the scores of subjects for each student.
        Output the subjects to the user (BM, BI, BC, Math, Sci, Sejarah, and Geo).
        For loop each subject, starting from 0 up to 7, incrementing by 1 then prompt the user to enter a score.
    Store the student information in the students array at studentCount index then increment studentCount by 1 to store the next student's information.
    Store the subject scores in the student's marks array.
    Prompt the user whether they want to continue add in another record.
    If user inputs 'Y' or 'y'. Continue to add more records.
    If user inputs 'N' or 'n'. Exit the loop.
    After adding records, call saveToFile to save the updated data to a file.
End
```

Function for Filter/Search

```
Start
Prompt user to enter 1 for Filter or 2 for Search
//User choose 1 for Filter
If choices is 1:
    Prompt user to input form (only 1 to 3). Next, prompt user to input subjects (must be space-separated).
    Loop through each character in input, starting from 0 up to input.size(), incrementing by 1.
    Split the input subjects into individual subjects and store in filterSubjects array. Count the number of subjects in subjectCount.
    Call filter function with form, filterSubjects, and subjectCount.
//User choose 2 for Search
If choices is 2:
    Prompt user to input a search keyword.
    Call search function with input.

Prompt user if they want to continue. If yes go back to the sub menu for search and filter, if no go back to menu.
End
```

Function for Filter

```
Start
For loop through i starting from 0 up to studentCount, where i increase by 1 each iteration.
    If the student form matches the given form.
    Output the student's ID, student's name, and student's form.
    For loop through j starting from 0 up to subjectCount, where j increase by 1 each iteration.
    Set subjectFound to false.
        For loop through k starting from 0 up to 7 which is the number of subjects, where k increase by 1 each iteration.
        If the subject matches the current filter subject.
        Output the subject and with the student's mark.
        Set subjectFound to true.
//Break loop
End
```

Function for Search

```
Start
Set found to false.
For loop through i starting from 0 up to studentCount, where i increase by 1 each iteration.
    If the input keyword matches the student's name.
    Set found to true.
    Output the student's ID, student's name, and student's form.
    For loop through j starting from 0 up to 7 which is the number of subjects, where j increase by 1 each iteration.
    Output every subject and the student's mark.
End
```

Function for Edit/Delete

```
Prompt user to choose between 1. Edit, 2. Delete.
If user choose 1 (Edit):
    Call editStudent() function.
If user choose 2 (Delete):
    Call deleteStudent() function.
Ask user if they wish to continue editing and delete. Given the options (Y/N):
If user choose Y:
    Repeat current function.
If user choose N:
    Exit current function.
End
```

Function for Edit

```
Start
Output the student list.
Prompt user to key in a student ID to edit.
Set found to false.
For loop through i starting from 0 to studentCount, incrementing by 1.
If user input matched existing student ID:
    Set found to true.
    Prompt user to choose between 3 options of 1 for Name, 2 for Form and 3 for Marks.
    If user choose 1 (Name):
        Prompt user to enter a new name.
    If user choose 2 (Form):
        Prompt user to enter a new form to update.
    If user choose 3 (Marks):
        Prompt user to choose between one of the 7 subjects to edit.
        Set validSubject to false.
        For loop through j starting from 0 to 7, increase by 1 each iteration.
        If user chose an existing subject:
            Prompt user to edit the score of the chosen subjects.
            Set validSubject to true.
        Ask the user if the current student requires more edit. Given the options (Y/N) only:
        If user choose Y:
            Repeat the choice 3.
        If user choose N:
            Exit current code.
    Ask user if there's still another student want to be edit. Given the options (Y/N):
    If user choose Y:
        Repeat prompt for user to key-in student ID.
    If user choose N:
        Exit current function.
    Save current changes to the file.
    End
```

Function for Delete

```
Start
Output the student list.
Prompt user to key in a student Id to delete from the file.
Set found to false.
For loop through i starting from 0 to studentCount, increment by 1.
 If user input matched existing student ID:
    Delete chosen Student information from the student.txt file.
    Saves current changes.
Ask user if there's another student to delete. Given the options (Y/N):
```

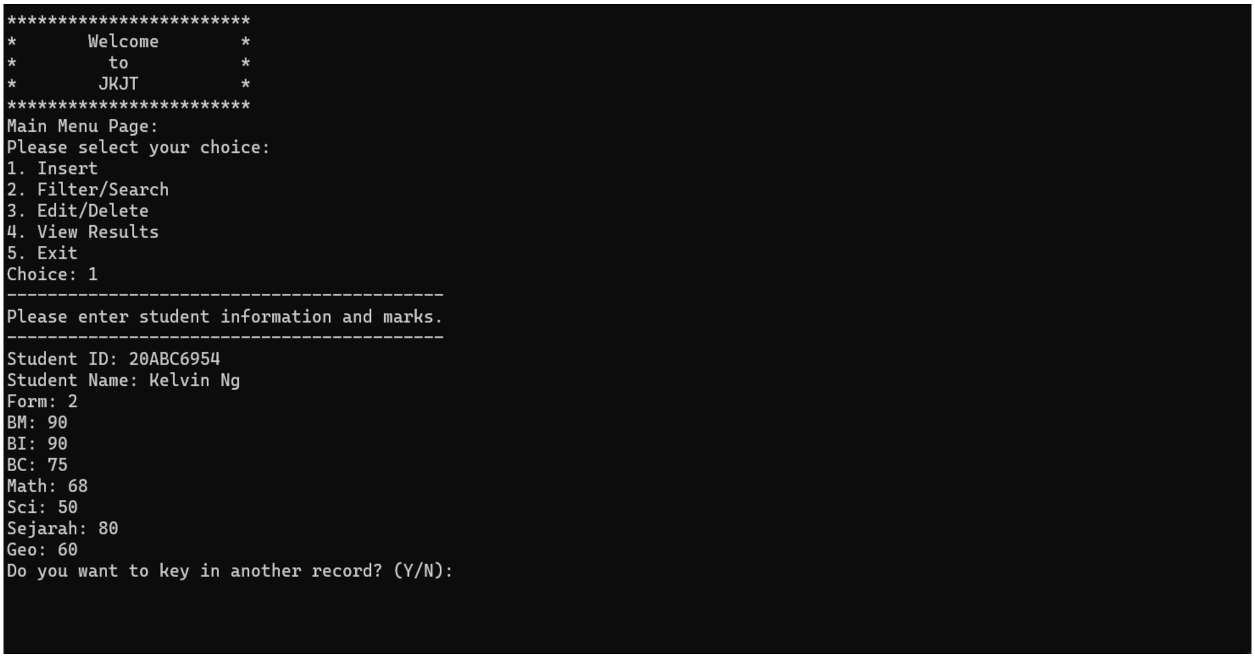
```
If user choose Y:
    Repeat current function.
IF user choose N:
    Exit current function.
End
```

Function for View Results

```
Start
Prompt user to enter 1 for Insert, 2 for Filter/Search, 3 for Modify, 4 for View Result and 5 for Exit
If user selects 4 (View Results):
    Check the students record array.
    If studentCount is 0:
        Output "No records available." Then, return to the main menu.
    Prompt user to choose the form that wish to be view (All/1/2/3).
    Loop until a valid form is entered:
        If user selects "All" or "all":
            Set formChoice = 0.
        If user selects a form between 1 and 3:
            Set formChoice to the selected value.
    After a valid formChoice is selected:
        Call displayStatistics with formChoice (0 for all forms, or the selected forms).
End
```

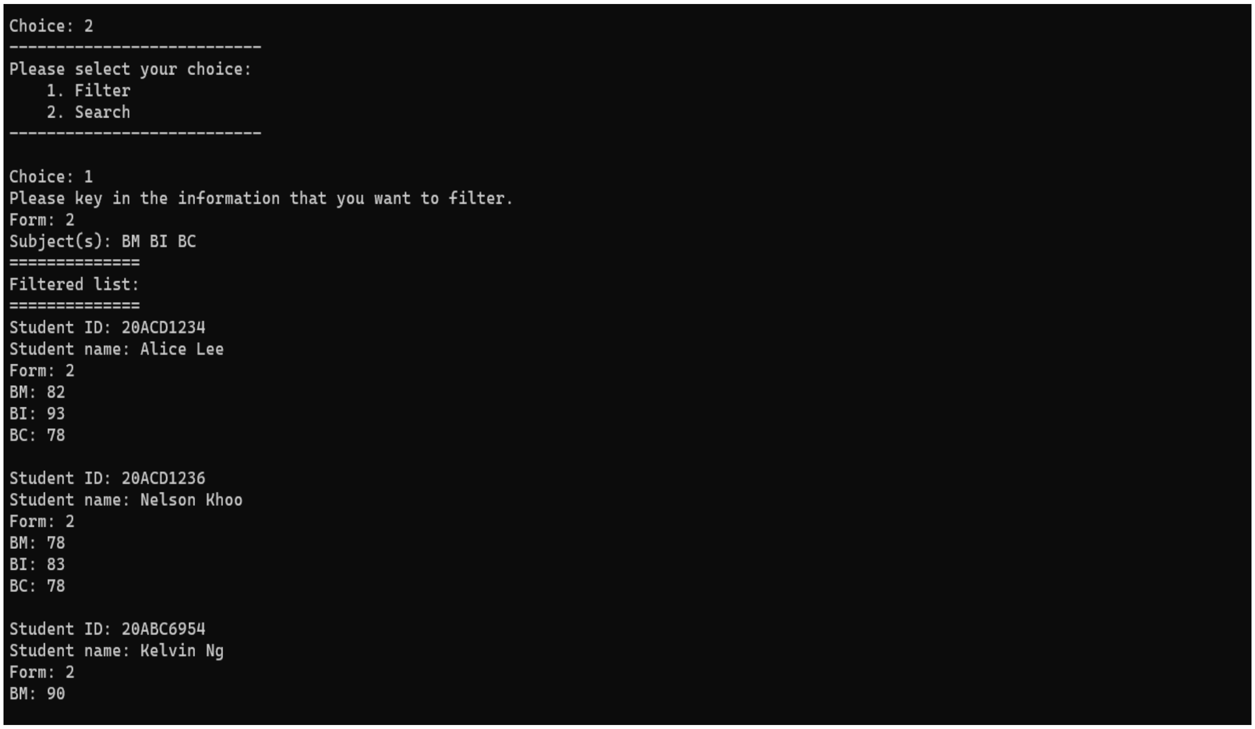
Test cases

1. Insert a new student (Kelvin Ng).



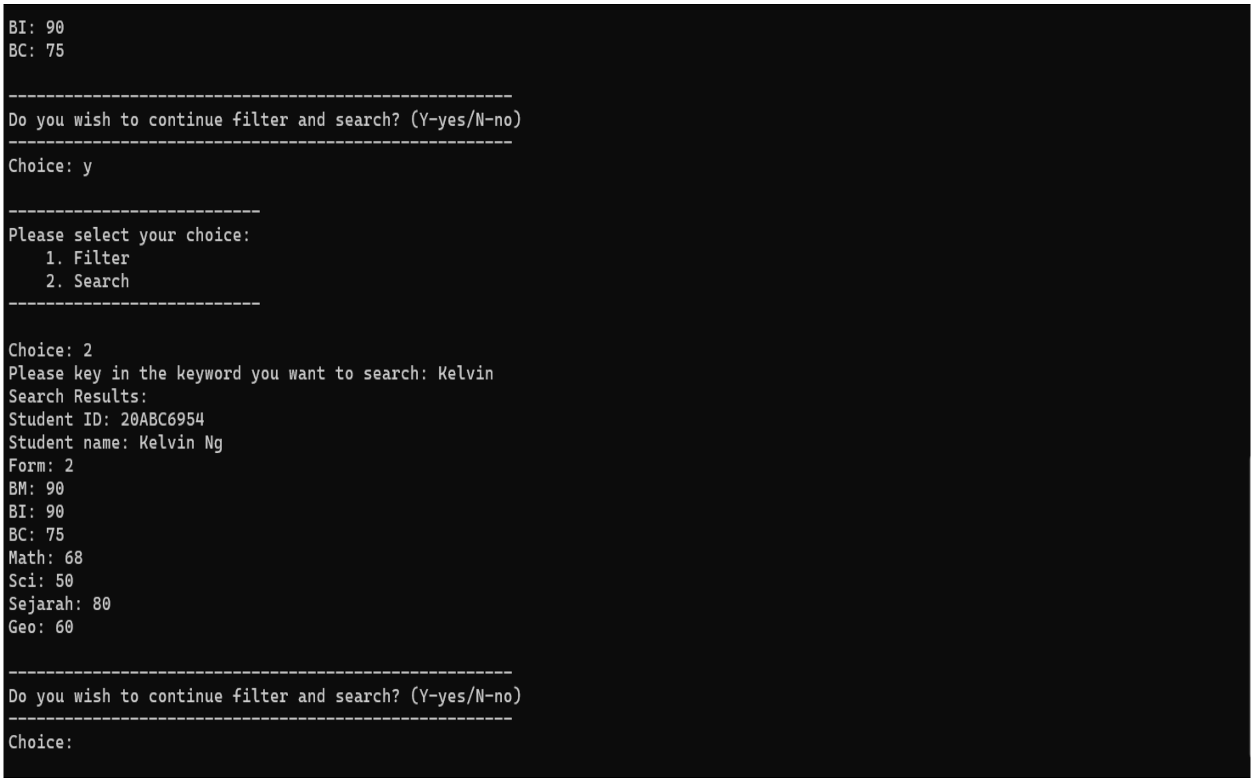
```
************************
*       Welcome        *
*         to           *
*        JKJT          *
************************
Main Menu Page:
Please select your choice:
1. Insert
2. Filter/Search
3. Edit/Delete
4. View Results
5. Exit
Choice: 1
-------------------------------------------
Please enter student information and marks.
-------------------------------------------
Student ID: 20ABC6954
Student Name: Kelvin Ng
Form: 2
BM: 90
BI: 90
BC: 75
Math: 68
Sci: 50
Sejarah: 80
Geo: 60
Do you want to key in another record? (Y/N):
```

2. Filter Form 2 and the subjects BM, BI, and BC.



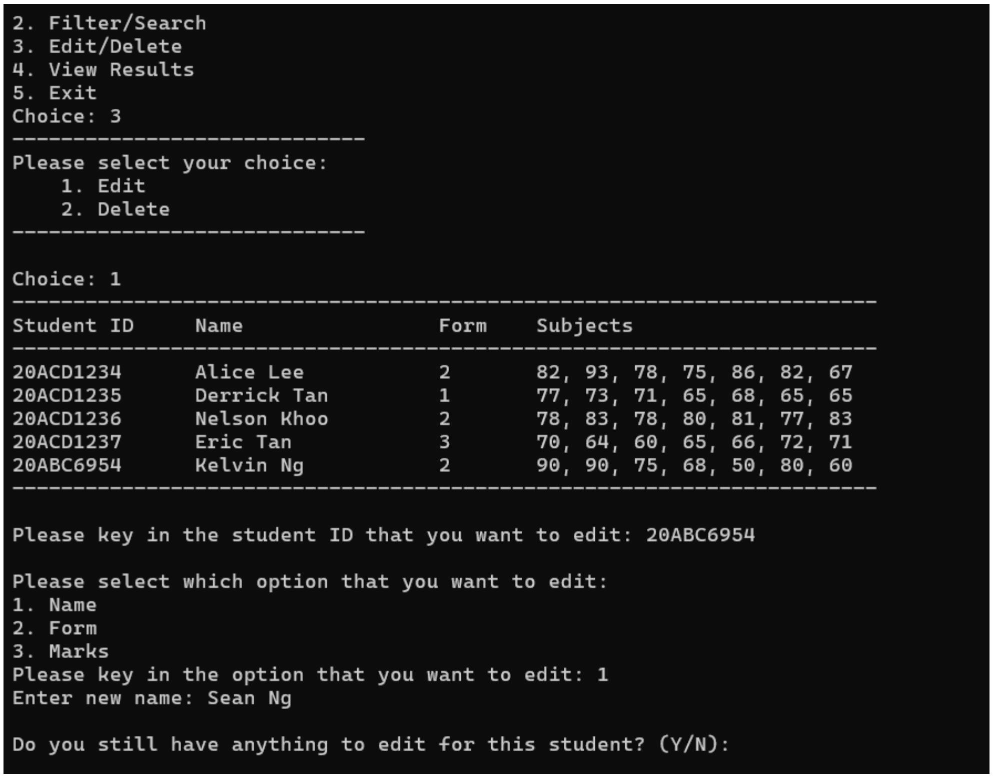
```
Choice: 2
---------------------------
Please select your choice:
    1. Filter
    2. Search
---------------------------

Choice: 1
Please key in the information that you want to filter.
Form: 2
Subject(s): BM BI BC
==============
Filtered list:
==============
Student ID: 20ACD1234
Student name: Alice Lee
Form: 2
BM: 82
BI: 93
BC: 78

Student ID: 20ACD1236
Student name: Nelson Khoo
Form: 2
BM: 78
BI: 83
BC: 78

Student ID: 20ABC6954
Student name: Kelvin Ng
Form: 2
BM: 90
```

3. Search for the name Kelvin Ng.



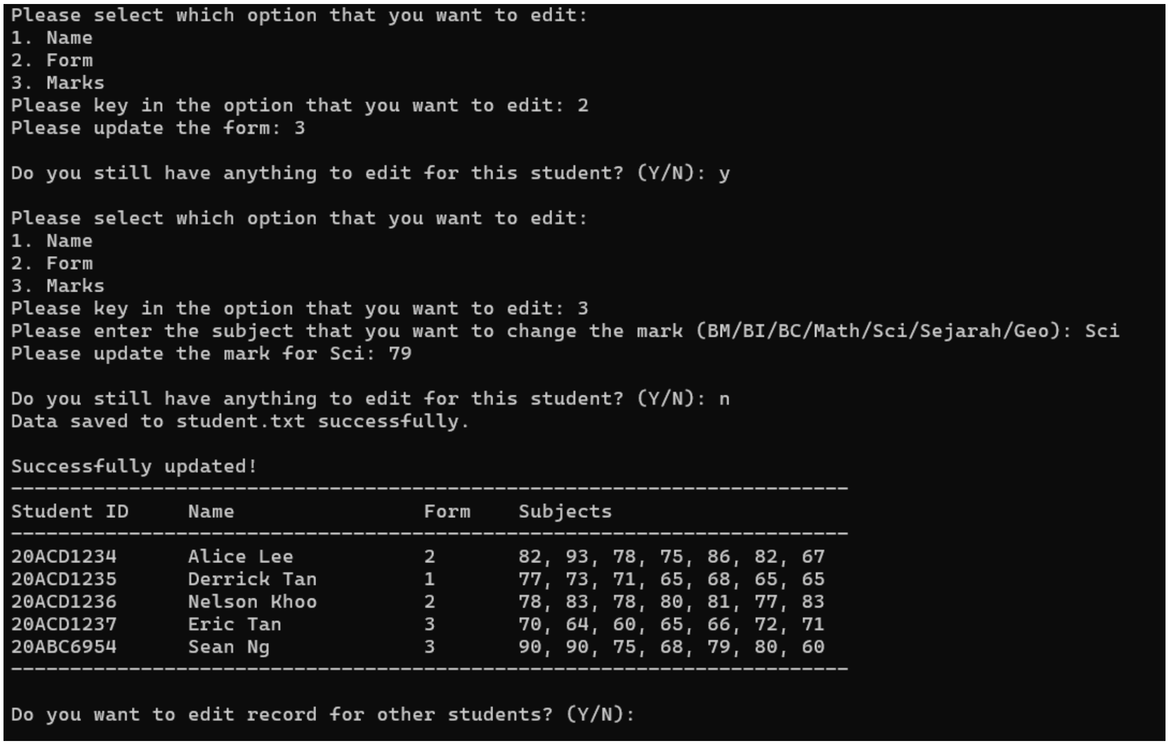
```
BI: 90
BC: 75

----------------------------------------------------
Do you wish to continue filter and search? (Y-yes/N-no)
----------------------------------------------------
Choice: y

--------------------------
Please select your choice:
    1. Filter
    2. Search
--------------------------

Choice: 2
Please key in the keyword you want to search: Kelvin
Search Results:
Student ID: 20ABC6954
Student name: Kelvin Ng
Form: 2
BM: 90
BI: 90
BC: 75
Math: 68
Sci: 50
Sejarah: 80
Geo: 60

----------------------------------------------------
Do you wish to continue filter and search? (Y-yes/N-no)
----------------------------------------------------
Choice:
```

4. Edit the student ID 20ABC6954 name from Kelvin Ng to Sean Ng.



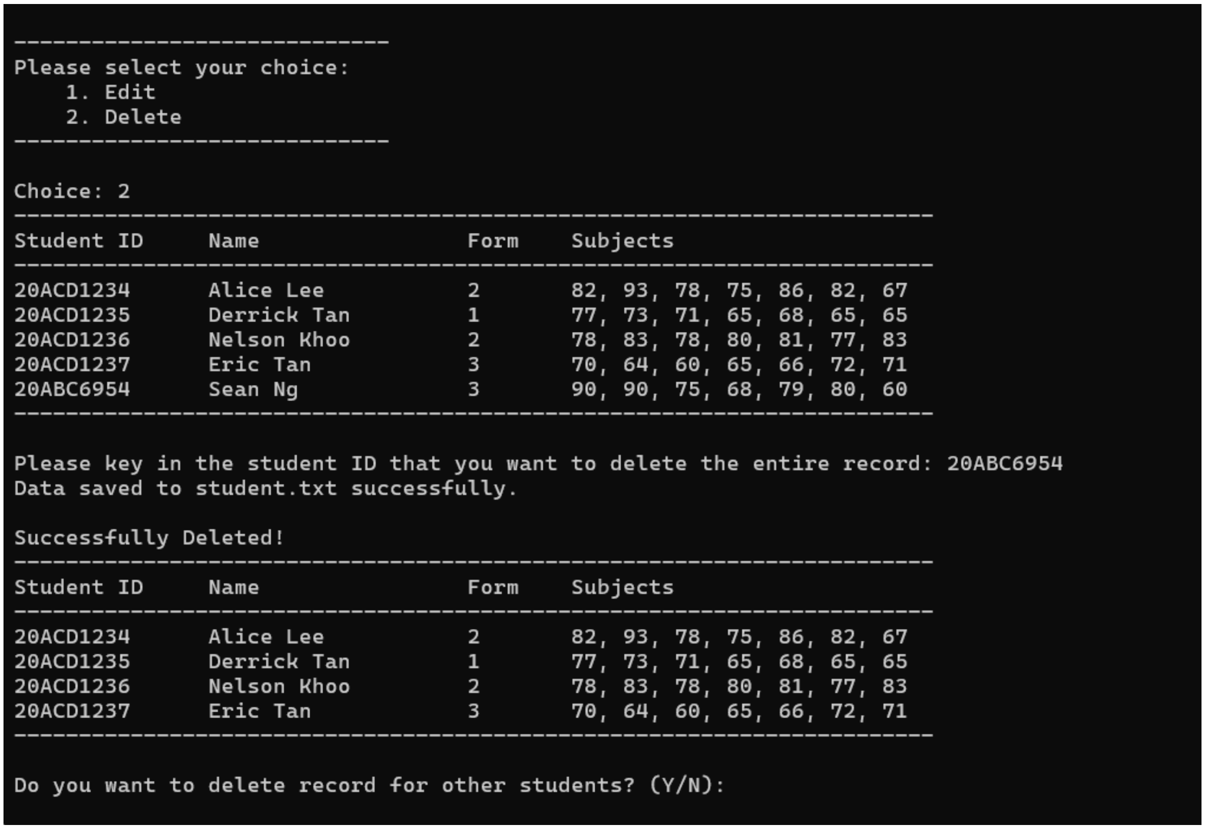
```
2. Filter/Search
3. Edit/Delete
4. View Results
5. Exit
Choice: 3
-----------------------------
Please select your choice:
    1. Edit
    2. Delete
-----------------------------

Choice: 1
-----------------------------------------------------------------------
Student ID     Name                Form    Subjects
-----------------------------------------------------------------------
20ACD1234      Alice Lee           2       82, 93, 78, 75, 86, 82, 67
20ACD1235      Derrick Tan         1       77, 73, 71, 65, 68, 65, 65
20ACD1236      Nelson Khoo         2       78, 83, 78, 80, 81, 77, 83
20ACD1237      Eric Tan            3       70, 64, 60, 65, 66, 72, 71
20ABC6954      Kelvin Ng           2       90, 90, 75, 68, 50, 80, 60
-----------------------------------------------------------------------

Please key in the student ID that you want to edit: 20ABC6954

Please select which option that you want to edit:
1. Name
2. Form
3. Marks
Please key in the option that you want to edit: 1
Enter new name: Sean Ng

Do you still have anything to edit for this student? (Y/N):
```

5. Edit Sean Ng form to 3 and the Sci mark to 79.



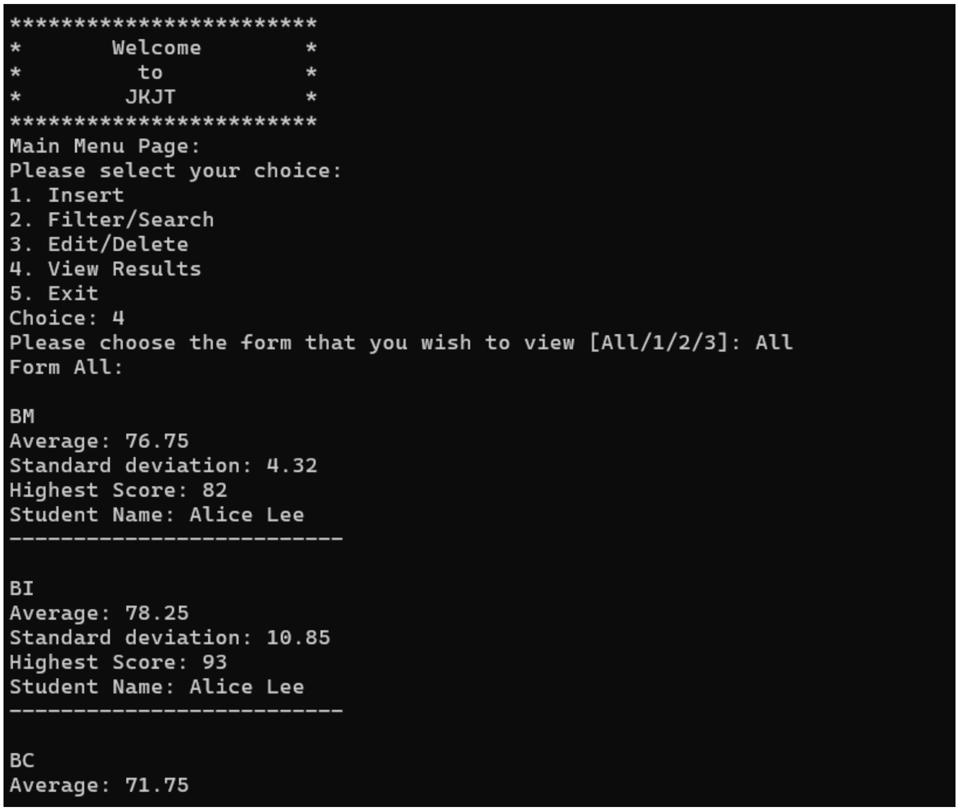
```
Please select which option that you want to edit:
1. Name
2. Form
3. Marks
Please key in the option that you want to edit: 2
Please update the form: 3

Do you still have anything to edit for this student? (Y/N): y

Please select which option that you want to edit:
1. Name
2. Form
3. Marks
Please key in the option that you want to edit: 3
Please enter the subject that you want to change the mark (BM/BI/BC/Math/Sci/Sejarah/Geo): Sci
Please update the mark for Sci: 79

Do you still have anything to edit for this student? (Y/N): n
Data saved to student.txt successfully.

Successfully updated!
-----------------------------------------------------------------------
Student ID     Name                Form    Subjects
-----------------------------------------------------------------------
20ACD1234      Alice Lee           2       82, 93, 78, 75, 86, 82, 67
20ACD1235      Derrick Tan         1       77, 73, 71, 65, 68, 65, 65
20ACD1236      Nelson Khoo         2       78, 83, 78, 80, 81, 77, 83
20ACD1237      Eric Tan            3       70, 64, 60, 65, 66, 72, 71
20ABC6954      Sean Ng             3       90, 90, 75, 68, 79, 80, 60
-----------------------------------------------------------------------

Do you want to edit record for other students? (Y/N):
```

6. Delete the new student information.

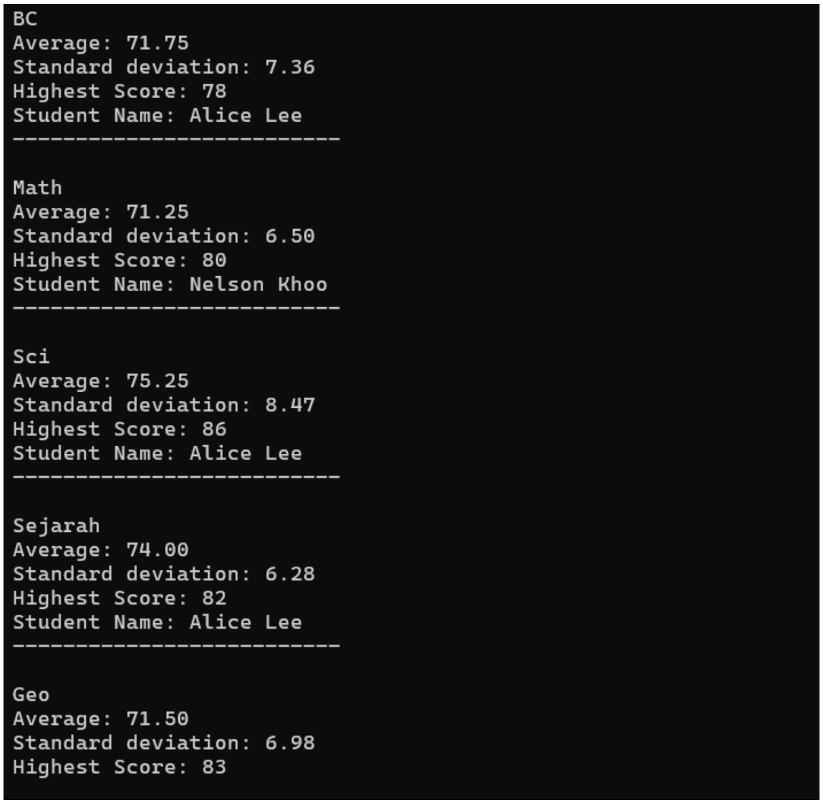


```

-----------------------------
Please select your choice:
    1. Edit
    2. Delete
-----------------------------

Choice: 2
-----------------------------------------------------------------------
Student ID     Name                Form    Subjects
-----------------------------------------------------------------------
20ACD1234      Alice Lee           2       82, 93, 78, 75, 86, 82, 67
20ACD1235      Derrick Tan         1       77, 73, 71, 65, 68, 65, 65
20ACD1236      Nelson Khoo         2       78, 83, 78, 80, 81, 77, 83
20ACD1237      Eric Tan            3       70, 64, 60, 65, 66, 72, 71
20ABC6954      Sean Ng             3       90, 90, 75, 68, 79, 80, 60
-----------------------------------------------------------------------

Please key in the student ID that you want to delete the entire record: 20ABC6954
Data saved to student.txt successfully.

Successfully Deleted!
-----------------------------------------------------------------------
Student ID     Name                Form    Subjects
-----------------------------------------------------------------------
20ACD1234      Alice Lee           2       82, 93, 78, 75, 86, 82, 67
20ACD1235      Derrick Tan         1       77, 73, 71, 65, 68, 65, 65
20ACD1236      Nelson Khoo         2       78, 83, 78, 80, 81, 77, 83
20ACD1237      Eric Tan            3       70, 64, 60, 65, 66, 72, 71
-----------------------------------------------------------------------

Do you want to delete record for other students? (Y/N):
```

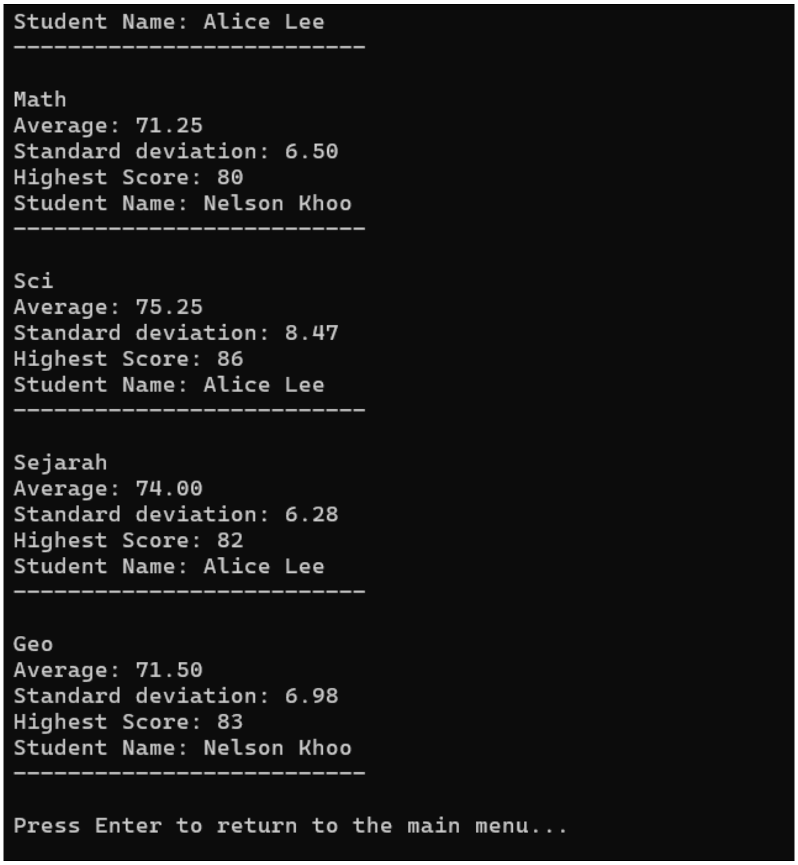
7. View Results for All.



```
************************
*       Welcome        *
*         to           *
*        JKJT          *
************************
Main Menu Page:
Please select your choice:
1. Insert
2. Filter/Search
3. Edit/Delete
4. View Results
5. Exit
Choice: 4
Please choose the form that you wish to view [All/1/2/3]: All
Form All:

BM
Average: 76.75
Standard deviation: 4.32
Highest Score: 82
Student Name: Alice Lee
--------------------------

BI
Average: 78.25
Standard deviation: 10.85
Highest Score: 93
Student Name: Alice Lee
--------------------------

BC
Average: 71.75
```

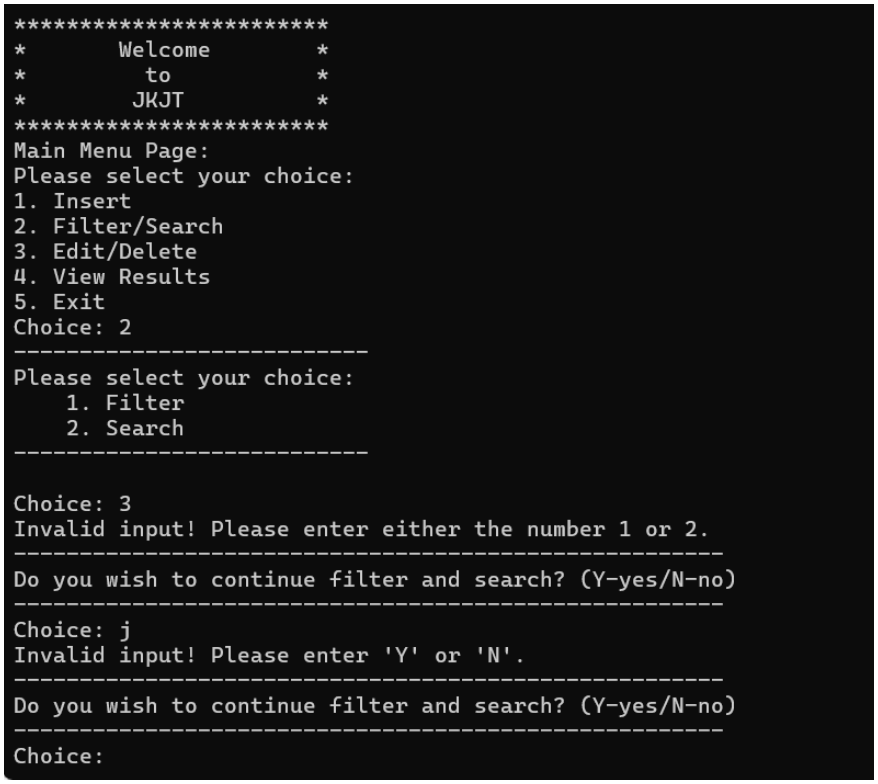


```
BC
Average: 71.75
Standard deviation: 7.36
Highest Score: 78
Student Name: Alice Lee
--------------------------

Math
Average: 71.25
Standard deviation: 6.50
Highest Score: 80
Student Name: Nelson Khoo
--------------------------

Sci
Average: 75.25
Standard deviation: 8.47
Highest Score: 86
Student Name: Alice Lee
--------------------------

Sejarah
Average: 74.00
Standard deviation: 6.28
Highest Score: 82
Student Name: Alice Lee
--------------------------

Geo
Average: 71.50
Standard deviation: 6.98
Highest Score: 83
```



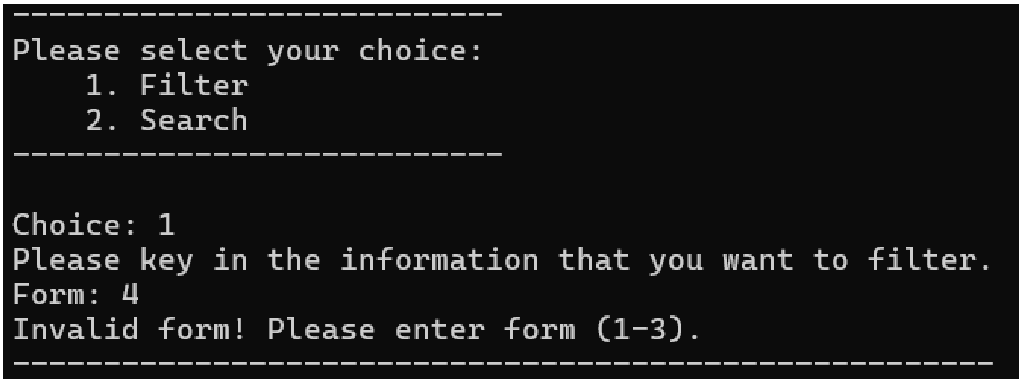
```
Student Name: Alice Lee
--------------------------

Math
Average: 71.25
Standard deviation: 6.50
Highest Score: 80
Student Name: Nelson Khoo
--------------------------

Sci
Average: 75.25
Standard deviation: 8.47
Highest Score: 86
Student Name: Alice Lee
--------------------------

Sejarah
Average: 74.00
Standard deviation: 6.28
Highest Score: 82
Student Name: Alice Lee
--------------------------

Geo
Average: 71.50
Standard deviation: 6.98
Highest Score: 83
Student Name: Nelson Khoo
--------------------------

Press Enter to return to the main menu...
```

8. Inputting choice other than 1 or 2 and choice other than y/n.



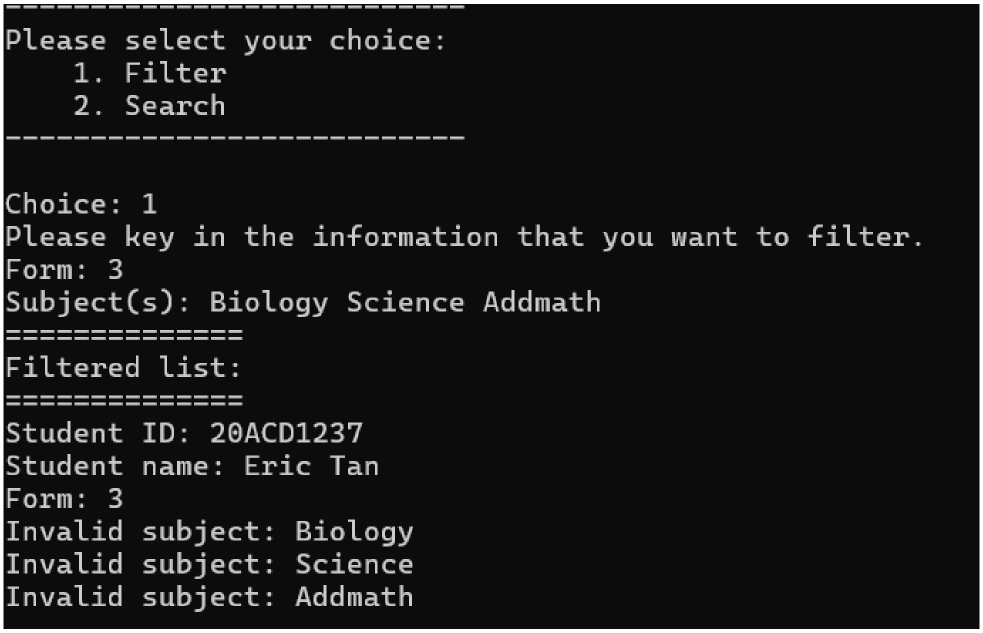
```
************************
*       Welcome        *
*         to           *
*        JKJT          *
************************
Main Menu Page:
Please select your choice:
1. Insert
2. Filter/Search
3. Edit/Delete
4. View Results
5. Exit
Choice: 2
---------------------------
Please select your choice:
    1. Filter
    2. Search
---------------------------

Choice: 3
Invalid input! Please enter either the number 1 or 2.
------------------------------------------------------
Do you wish to continue filter and search? (Y-yes/N-no)
------------------------------------------------------
Choice: j
Invalid input! Please enter 'Y' or 'N'.
------------------------------------------------------
Do you wish to continue filter and search? (Y-yes/N-no)
------------------------------------------------------
Choice:
```

9. Input other form that is not form 1 to 3.



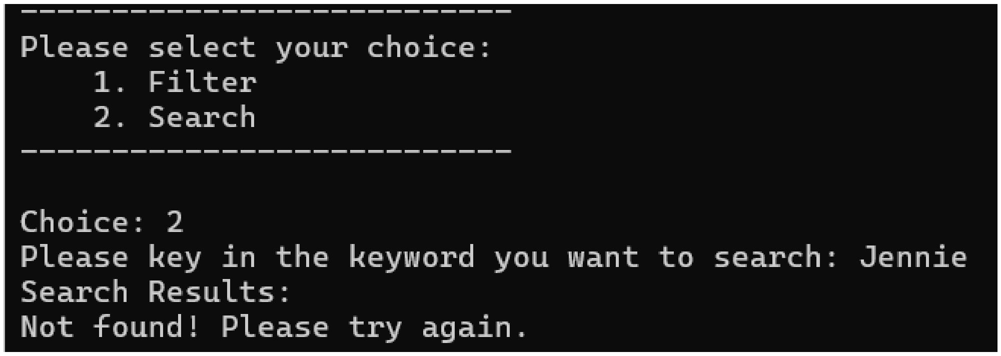
```
---------------------------
Please select your choice:
    1. Filter
    2. Search
---------------------------

Choice: 1
Please key in the information that you want to filter.
Form: 4
Invalid form! Please enter form (1-3).
------------------------------------------------------
```

10. Filter invalid subjects.



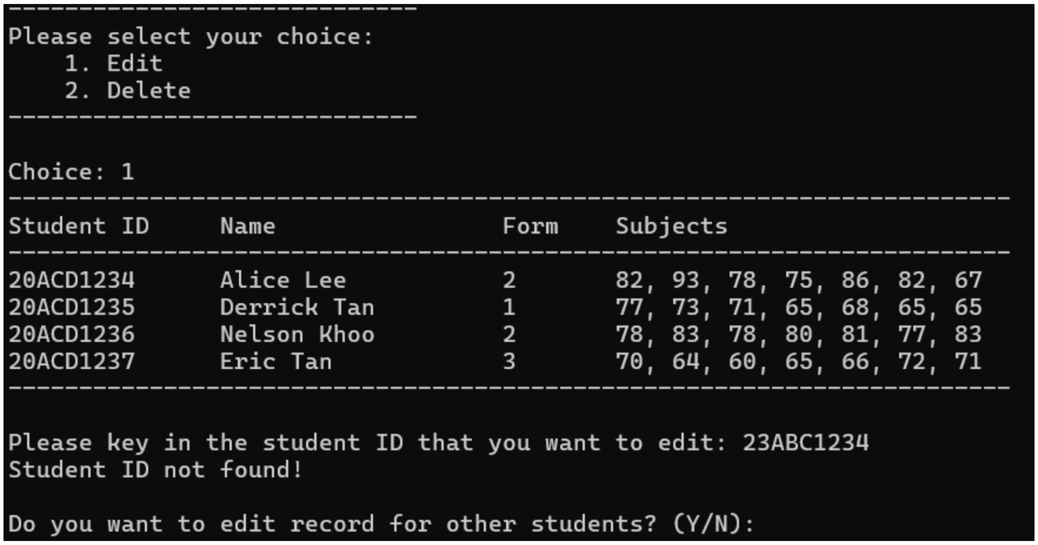
```
---------------------------
Please select your choice:
    1. Filter
    2. Search
---------------------------

Choice: 1
Please key in the information that you want to filter.
Form: 3
Subject(s): Biology Science Addmath
==============
Filtered list:
==============
Student ID: 20ACD1237
Student name: Eric Tan
Form: 3
Invalid subject: Biology
Invalid subject: Science
Invalid subject: Addmath
```

11. Search for student that is not in the list.



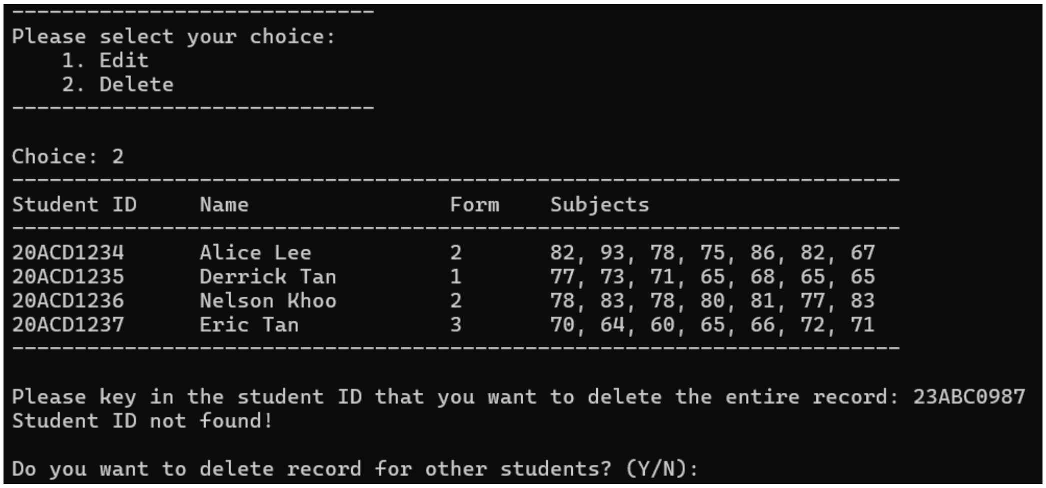
```
---------------------------
Please select your choice:
    1. Filter
    2. Search
---------------------------

Choice: 2
Please key in the keyword you want to search: Jennie
Search Results:
Not found! Please try again.
```

12. When edit a student ID that is not in the list.



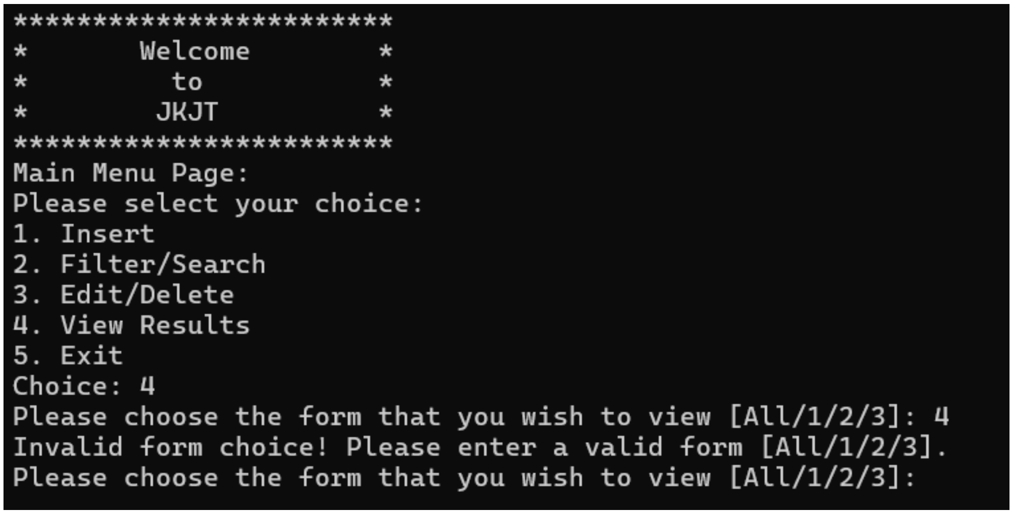
```
-----------------------------
Please select your choice:
    1. Edit
    2. Delete
-----------------------------

Choice: 1
------------------------------------------------------------------------
Student ID     Name                Form    Subjects
------------------------------------------------------------------------
20ACD1234      Alice Lee           2       82, 93, 78, 75, 86, 82, 67
20ACD1235      Derrick Tan         1       77, 73, 71, 65, 68, 65, 65
20ACD1236      Nelson Khoo         2       78, 83, 78, 80, 81, 77, 83
20ACD1237      Eric Tan            3       70, 64, 60, 65, 66, 72, 71
------------------------------------------------------------------------

Please key in the student ID that you want to edit: 23ABC1234
Student ID not found!

Do you want to edit record for other students? (Y/N):
```

13. When delete a student ID that is not in the list.



```
-----------------------------
Please select your choice:
    1. Edit
    2. Delete
-----------------------------

Choice: 2
------------------------------------------------------------------------
Student ID     Name                Form    Subjects
------------------------------------------------------------------------
20ACD1234      Alice Lee           2       82, 93, 78, 75, 86, 82, 67
20ACD1235      Derrick Tan         1       77, 73, 71, 65, 68, 65, 65
20ACD1236      Nelson Khoo         2       78, 83, 78, 80, 81, 77, 83
20ACD1237      Eric Tan            3       70, 64, 60, 65, 66, 72, 71
------------------------------------------------------------------------

Please key in the student ID that you want to delete the entire record: 23ABC0987
Student ID not found!

Do you want to delete record for other students? (Y/N):
```

14. Input 4 or something else other than All/1/2/3.



```
************************
*       Welcome        *
*         to           *
*        JKJT          *
************************
Main Menu Page:
Please select your choice:
1. Insert
2. Filter/Search
3. Edit/Delete
4. View Results
5. Exit
Choice: 4
Please choose the form that you wish to view [All/1/2/3]: 4
Invalid form choice! Please enter a valid form [All/1/2/3].
Please choose the form that you wish to view [All/1/2/3]:
```

Source code (Appendix)

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

struct Student {
    string studentId;
    string name;
    int form=0;
    int marks[7];
};

const int MAX_STUDENTS = 100;
const string subjects[] = { "BM", "BI", "BC", "Math", "Sci", "Sejarah", "Geo" };
Student students[MAX_STUDENTS];
int studentCount = 0;

// Function prototypes
void saveToFile();
void readFile();
int getValidScore(const string& subject);
void addRecords();
double calculateAverage(int scores[], int n);
double calculateStandardDeviation(int scores[], int n, double average);
void displayStatistics(int form);
void displayResults();
void filter(int form, const string filterSubjects[], int subjectCount);
void search(const string& keyword);
void filterAndSearch();
void showMenu();
void displayStudentList();
void editStudent();
void deleteStudent();
void editAndDelete();

int main() {
    readFile();
    showMenu();
    return 0;
}

// Function to save student data to a file
void saveToFile() {
    ofstream myfile("student.txt");

    if (!myfile) {
        cout << "Error opening file for writing.\n";
        return;
    }

    for (int i = 0; i < studentCount; i++) {
        myfile << students[i].studentId << "\n";
        myfile << students[i].name << "\n";
        myfile << students[i].form << "\n";
        for (int j = 0; j < 7; j++) {
            myfile << students[i].marks[j] << " ";
        }
        myfile << "\n";
    }

    myfile.close();
    cout << "Data saved to student.txt successfully.\n";
}

// Function to read student data from a file
void readFile() {
    ifstream file("student.txt");

    if (file.fail()) {
        cout << "Error in opening file" << endl;
    }
    else {
        while (file >> students[studentCount].studentId) {
            file.ignore();
            getline(file, students[studentCount].name);
            file >> students[studentCount].form;
            for (int i = 0; i < 7; i++) {
                file >> students[studentCount].marks[i];
            }
            studentCount++;
        }
    }
    file.close();
}

// Function to get a valid score between 0 to 100
int getValidScore(const string& subject) {
    int score;
    while (true) {
        cout << subject << ": ";
        cin >> score;
        if (score >= 0 && score <= 100) {
            break;
        }
        else {
            cout << "Invalid number. Please enter a score between 0 and 100.\n";
        }
    }
    return score;
}

// Function to add student records
void addRecords() {
    char addMore;
    do {
        if (studentCount >= MAX_STUDENTS) {
            cout << "Cannot add more students. Maximum capacity reached.\n";
            return;
        }

        Student s;
        cout << "----------------------------------------" << endl;
        cout << "Please enter student information and marks." << endl;
        cout << "----------------------------------------" << endl;
        cout << "Student ID: ";
        cin >> s.studentId;
        cout << "Student Name: ";
        cin.ignore();
        getline(cin, s.name);
        cout << "Form: ";
        cin >> s.form;

        // Get valid scores for each subject
        for (int i = 0; i < 7; i++) {
            s.marks[i] = getValidScore(subjects[i]);
        }

        students[studentCount] = s;
        studentCount++;

        cout << "Do you want to key in another record? (Y/N): ";
        cin >> addMore;

    } while (addMore == 'Y' || addMore == 'y');

    saveToFile();
}

// Function to calculate average score
double calculateAverage(int scores[], int n) {
    double sum = 0;
    for (int i = 0; i < n; i++) {
        sum += scores[i];
    }
    return sum / n;
}

// Function to calculate standard deviation
double calculateStandardDeviation(int scores[], int n, double average) {
    double sumOfSquares = 0;
    for (int i = 0; i < n; i++) {
        double diff = scores[i] - average;
        sumOfSquares += diff * diff;
    }
    double variance = sumOfSquares / n;
    return sqrt(variance);
}

// Function to display statistics for a specific form
void displayStatistics(int form) {
    const int SUBJECT_COUNT = 7;
    int scores[MAX_STUDENTS];

    for (int subj = 0; subj < SUBJECT_COUNT; subj++) {
        int count = 0;
        int highestScore = -1;
        string topStudent;

        // Display results for the selected form or all forms
        for (int i = 0; i < studentCount; i++) {
            if (students[i].form == form || form == 0) {
                int score = students[i].marks[subj];
                scores[count++] = score;
                if (score > highestScore) {
                    highestScore = score;
                    topStudent = students[i].name;
                }
            }
        }

        if (count > 0) {
            double average = calculateAverage(scores, count);
            double stdDev = calculateStandardDeviation(scores, count, average);

            cout << fixed << setprecision(2) << endl;
            cout << subjects[subj] << "\n";
            cout << "Average: " << average << "\n";
            cout << "Standard deviation: " << stdDev << "\n";
            cout << "Highest Score: " << highestScore << "\n";
            cout << "Student Name: " << topStudent << "\n";
            cout << "-------------------------\n";
        }
    }
}

// Function to display results for a selected form
void displayResults() {
    if (studentCount == 0) {
        cout << "No records available.\n";
        return;
    }

    string choice;
    int formChoice = -1;

    // Validation loop to ensure valid form choice
    while (true) {
        cout << "Please choose the form that you wish to view [All/1/2/3]: ";
        cin >> choice;

        if (choice == "All" || choice == "all") {
            formChoice = 0;  // 0 will represent "All"
            break;
        }
        else if (choice == "1" || choice == "2" || choice == "3") {
            formChoice = stoi(choice);
            break;
        }
        else {
            cout << "Invalid form choice! Please enter a valid form [All/1/2/3].\n";
        }
    }

    // Display the results for the selected form
    cout << "Form " << (formChoice == 0 ? "All" : to_string(formChoice)) << ":\n";
    displayStatistics(formChoice);

    cout << "\nPress Enter to return to the main menu...";
    cin.ignore();
    cin.get();
}

//Function of filter for selected record
void filter(int form, const string filterSubjects[], int subjectCount) {
    for (int i = 0; i < studentCount; i++) {
        if (students[i].form == form) {
            cout << "Student ID: " << students[i].studentId << endl;
            cout << "Student name: " << students[i].name << endl;
            cout << "Form: " << students[i].form << endl;

            for (int j = 0; j < subjectCount; j++) {
                bool subjectFound = false; //Check the subject is valid or not
                for (int k = 0; k < 7; k++) {
                    if (subjects[k] == filterSubjects[j]) {
                        cout << subjects[k] << ": " << students[i].marks[k] << endl;
                        subjectFound = true;
                        break;
                    }
                } if (!subjectFound) {
                    cout << "Invalid subject: " << filterSubjects[j] << endl;
                }
            }
            cout << "\n";
        }
    }
}

//Fucntion of search for selected record
void search(const string& keyword) {
    bool found = false; //To check if result was found

    for (int i = 0; i < studentCount; i++) {
        if (students[i].name.find(keyword) != string::npos) {
            //.find(keyword) function search for the name that user input
            //string::npos represent the value when the keyword was not found

            found = true;
            cout << "Search Results:" << endl;
            cout << "Student ID: " << students[i].studentId << endl;
            cout << "Student name: " << students[i].name << endl;
            cout << "Form: " << students[i].form << endl;

            for (int j = 0; j < 7; j++) {
                cout << subjects[j] << ": " << students[i].marks[j] << endl;
            }
            cout << "\n";
        }
    }

    if (!found) {
        //If the keyword name was not found from the file
        cout << "Search Results: " << endl;
        cout << "Not found! Please try again." << endl;
    }
}

//Function of filter and search for selected record in menu
void filterAndSearch() {
    int choices; //For user to enter 1 or 2
    char cont; //For user to enter (Y-Yes) to continue or (N-No) to exit
    string input;

    do {
        cout << "--------------------------" << endl;
        cout << "Please select your choice: " << endl;
        cout << "    1. Filter" << endl;
        cout << "    2. Search" << endl;
        cout << "--------------------------" << endl;
        cout << "\nChoice: ";

        if (cin >> choices && (choices == 1 || choices == 2)) {
            cin.ignore();

            if (choices == 1) {
                int form;
        cout << "Please key in the information that you want to filter." << endl;
        cout << "Form: ";

        if (cin >> form && (form >= 1 && form <= 3)) {
            cin.ignore();

            cout << "Subject(s): ";
            getline(cin, input);

            string filterSubjects[7];
            int subjectCount = 0;
            string subject;

            for (int i = 0; i < input.size(); i++) {
                if (input[i] != ' ') { //If its not space, the character will store into the 'subject'
                    subject += input[i];
                }
                else {
                    filterSubjects[subjectCount++] = subject; //If its have space, it will ends the
subject name and
                    subject.clear();                          //store into 'subject' then into 'filterSubject' for
later.
                }
            }
            filterSubjects[subjectCount++] = subject; //Final subject
            cout << "=============" << endl;
            cout << "Filtered list:" << endl;
            cout << "=============" << endl;
            filter(form, filterSubjects, subjectCount);
        }
        else {
            cin.clear(); //Clear the error flag in case the form is not between 1 and 3 inclusive
            cin.ignore();
            cout << "Invalid form! Please enter form (1-3)." << endl;
        }

    }
    else if (choices == 2) {
        cout << "Please key in the keyword you want to search: ";
        getline(cin, input);
        search(input);
    }
}
else {
    cin.clear(); //Clear the error flag in case the input is not 1 or 2
    cin.ignore();
    cout << "Invalid input! Please enter either the number 1 or 2." << endl;
}

do {
    cout << "-------------------------------------------------------" << endl;
    cout << "Do you wish to continue filter and search? (Y-yes/N-no)" << endl;
    cout << "-------------------------------------------------------" << endl;
    cout << "Choice: ";
    cin >> cont;
    cin.ignore();
    cont = tolower(cont);

    if (cont != 'y' && cont != 'n') {
        cout << "Invalid input! Please enter 'Y' or 'N'." << endl;
    }
} while (cont != 'y' && cont != 'n');
```

```
        cout << "\n";
    } while (cont == 'y'); //The program will continue when 'Y' or 'y' was entered

}

// Function to display the current list of students
void displayStudentList() {
    cout << "----------------------------------------------------------------------\n";
    cout << left << setw(15) << "Student ID" << setw(20) << "Name" << setw(8) << "Form" <<
"Subjects\n";
    cout << "----------------------------------------------------------------------\n";

    for (int i = 0; i < studentCount; i++) {
        cout << left << setw(15) << students[i].studentId
            << setw(20) << students[i].name
            << setw(8) << students[i].form;

        // Display marks for each subject
        for (int j = 0; j < 7; j++) {
            cout << students[i].marks[j];
            if (j < 6)
                cout << ", ";
        }
        cout << "\n";
    }
    cout << "----------------------------------------------------------------------\n";
}

// Function to modify existing student records
void editStudent() {
```

```
displayStudentList();


string studentId;
cout << "\nPlease key in the student ID that you want to edit: ";
cin >> studentId;


bool found = false;
for (int i = 0; i < studentCount; i++) {
    if (students[i].studentId == studentId) {
        found = true;
        char cont;
        do {
            int editChoice;
            cout << "\nPlease select which option that you want to edit:\n";
            cout << "1. Name\n"; //Edit name
            cout << "2. Form\n"; //Edit form
            cout << "3. Marks\n"; //Edit marks
            cout << "Please key in the option that you want to edit: ";
            cin >> editChoice;


            if (editChoice == 1) {
                cout << "Enter new name: ";
                cin.ignore();
                getline(cin, students[i].name);
            }
            else if (editChoice == 2) {
                cout << "Please update the form: ";
                cin >> students[i].form;
            }
            else if (editChoice == 3) {
                string subject;
                cout << "Please enter the subject that you want to change the mark (BM/BI/BC/Math/Sci/Sejarah/Geo): ";
                cin >> subject;

                bool validSubject = false;
                for (int j = 0; j < 7; j++) {
                    if (subjects[j] == subject) {
                        cout << "Please update the mark for " << subjects[j] << ": ";
                        cin >> students[i].marks[j];
                        validSubject = true;
                        break;
                    }
                }
                if (!validSubject) {
                    cout << "Invalid subject entered!\n";
                }
            }
            else {
                cout << "Invalid choice! Try again.\n";
            }

            cout << "\nDo you still have anything to edit for this student? (Y/N): ";
            cin >> cont;
            cont = tolower(cont);

        } while (cont == 'y');

        // After finishing edits for this student, save changes
        saveToFile();
        cout << "\nSuccessfully updated!\n";
```

```
            // Show updated list
            displayStudentList();

            break;
        }
    }

    if (!found) {
        cout << "Student ID not found!\n";
    }

    // Ask if the user wants to edit another student
    char editMore;
    cout << "\nDo you want to edit record for other students? (Y/N): ";
    cin >> editMore;

    if (tolower(editMore) == 'y') {
        editStudent();
    }
}

// Function to delete a student record
void deleteStudent() {
    displayStudentList();

    string studentId;

    cout << "\nPlease key in the student ID that you want to delete the entire record: ";
    cin >> studentId;

    bool found = false;
    for (int i = 0; i < studentCount; i++) {
        if (students[i].studentId == studentId) {
            found = true;

            // Shift remaining students up
            for (int j = i; j < studentCount - 1; j++) {
                students[j] = students[j + 1];
            }
            studentCount--;
            saveToFile();

            cout << "\nSuccessfully Deleted!\n";

            displayStudentList();
            break;
        }
    }

    if (!found) {
        cout << "Student ID not found!\n";
    }

    // Ask if the user wants to delete another student
    char deleteMore;
    cout << "\nDo you want to delete record for other students? (Y/N): ";
    cin >> deleteMore;

    if (tolower(deleteMore) == 'y') {
        deleteStudent();
    }
}
```

```
//Function to edit and delete for main menu
void editAndDelete() {
    int modifyChoice;
    char cont;

    do {
        cout << "-----------------------------" << endl;
        cout << "Please select your choice: " << endl;
        cout << "    1. Edit" << endl;
        cout << "    2. Delete" << endl;
        cout << "-----------------------------" << endl;
        cout << "\nChoice: ";

        if (cin >> modifyChoice && (modifyChoice == 1 || modifyChoice == 2)) {
            cin.ignore();

            if (modifyChoice == 1) {
                editStudent();
            }
            else if (modifyChoice == 2) {
                deleteStudent();
            }
        }
        else {
            cin.clear();
            cin.ignore();
            cout << "Invalid choice! Please try again." << endl;
        }

        do {
            cout << "--------------------------------------------------" << endl;
            cout << "Do you wish to continue edit and delete? (Y-yes/N-no)" << endl;
            cout << "--------------------------------------------------" << endl;
            cout << "Choice: ";
            cin >> cont;
            cin.ignore();
            cont = tolower(cont);

            if (cont != 'y' && cont != 'n') {
                cout << "Invalid input! Please enter 'Y' or 'N'." << endl;
            }
        } while (cont != 'y' && cont != 'n');

        cout << "\n";
    } while (cont == 'y');
}

// Function to display the main menu and handle user input
void showMenu() {
    int choice;
    do {
        system("cls");
        cout << "************************" << endl;
        cout << "*       Welcome        *" << endl;
        cout << "*         to           *" << endl;
        cout << "*        JKJT          *" << endl;
        cout << "************************" << endl;
        cout << "Main Menu Page:\n";
        cout << "Please select your choice:\n";
        cout << "1. Insert\n";
        cout << "2. Filter/Search\n";
        cout << "3. Edit/Delete\n";
        cout << "4. View Results\n";
        cout << "5. Exit\n";
        cout << "Choice: ";

        if (!(cin >> choice)) {
            cin.clear();
            cin.ignore();
            cout << "Invalid input! Please enter a number between 1 to 5.\n";
            continue;
        }

        switch (choice) {
        case 1:
            addRecords();
            break;
        case 2:
            filterAndSearch();
            break;
        case 3:
            editAndDelete();
            break;
        case 4:
            displayResults();
            break;
        case 5:
            cout << "Exiting...\n";
            break;
        default:
            cout << "Invalid choice! Please try again.\n";
            continue;
        }
    } while (choice != 5);
}
```