

UNIVERSITI TUNKU ABDUL RAHMAN

ACADEMIC YEAR 2021/2022



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

UCCD 1004 PROGRAMMING CONCEPTS AND PRACTICES

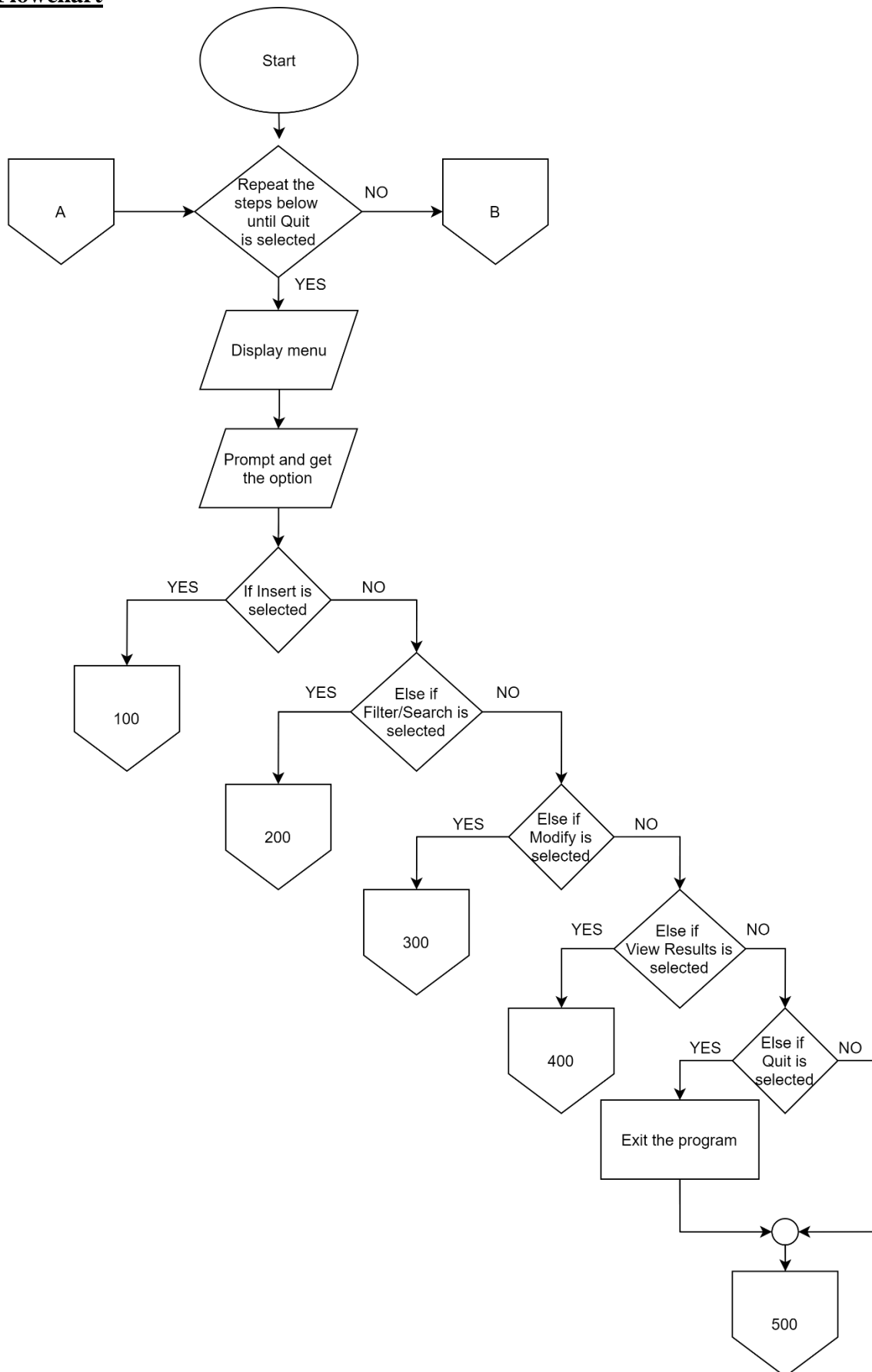
UFCB 1004 PROGRAMMING CONCEPTS AND PRACTICES

ASSIGNMENT 2

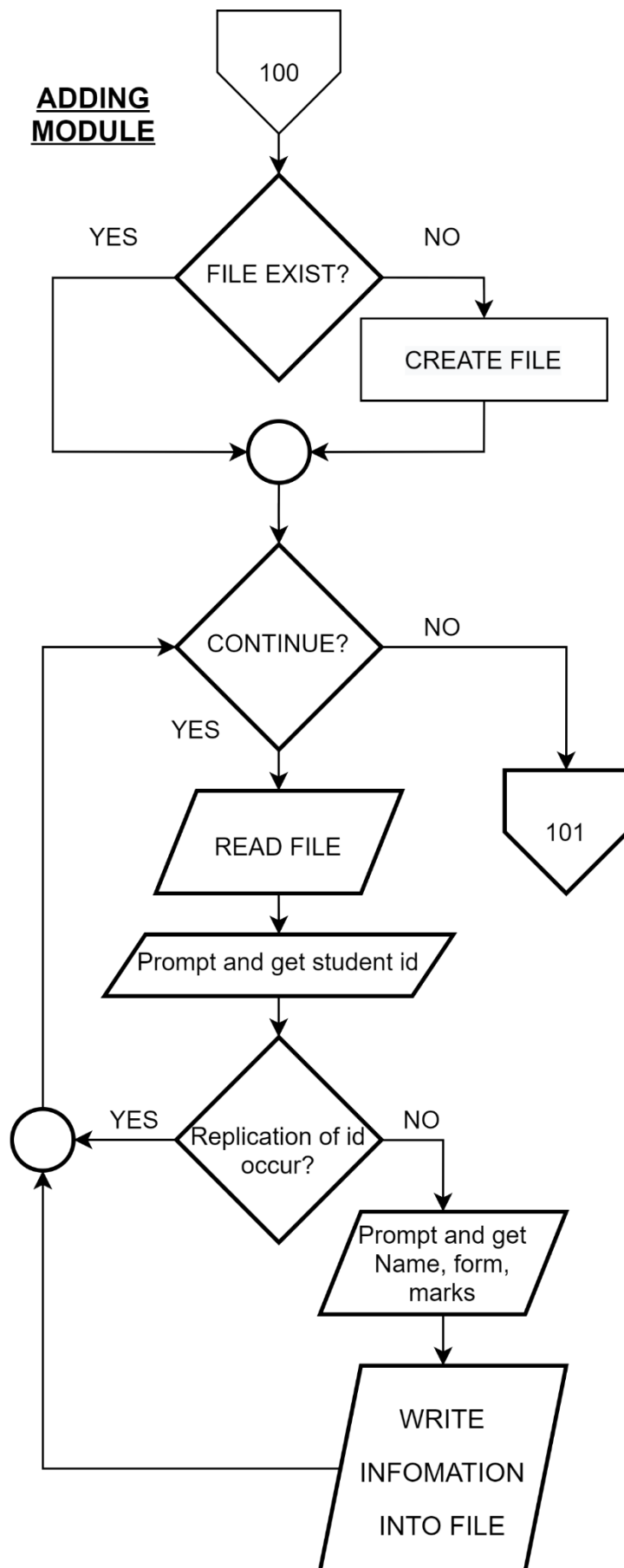
Group Number: 32

Group Members	Name	Student ID	Programme
Member 1	Seah Eu Jin	20ACB01553	CT
Member 2	Khoo Zi Yi	20ACB03614	CS
Member 3	Foo Jia Syuen	20ACB01451	CS

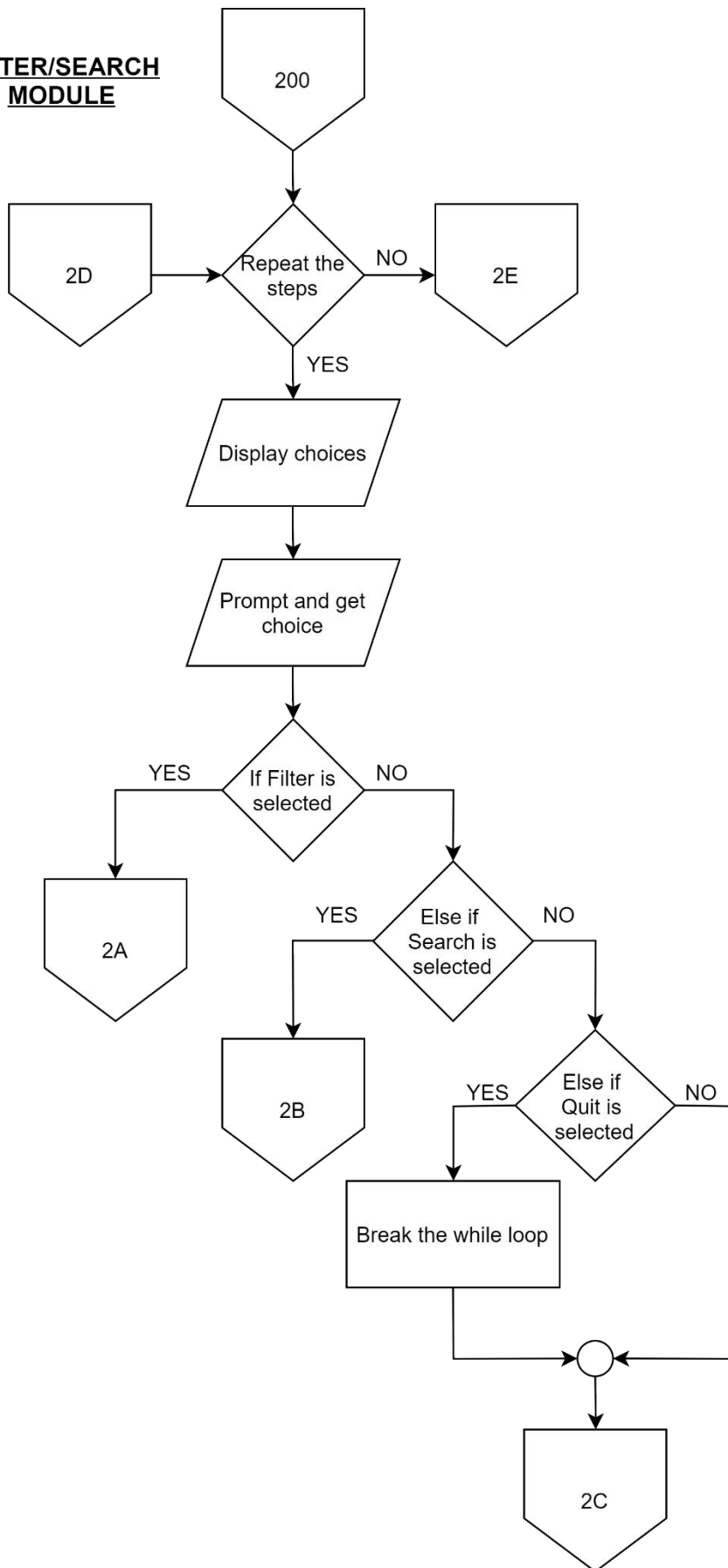
Flowchart

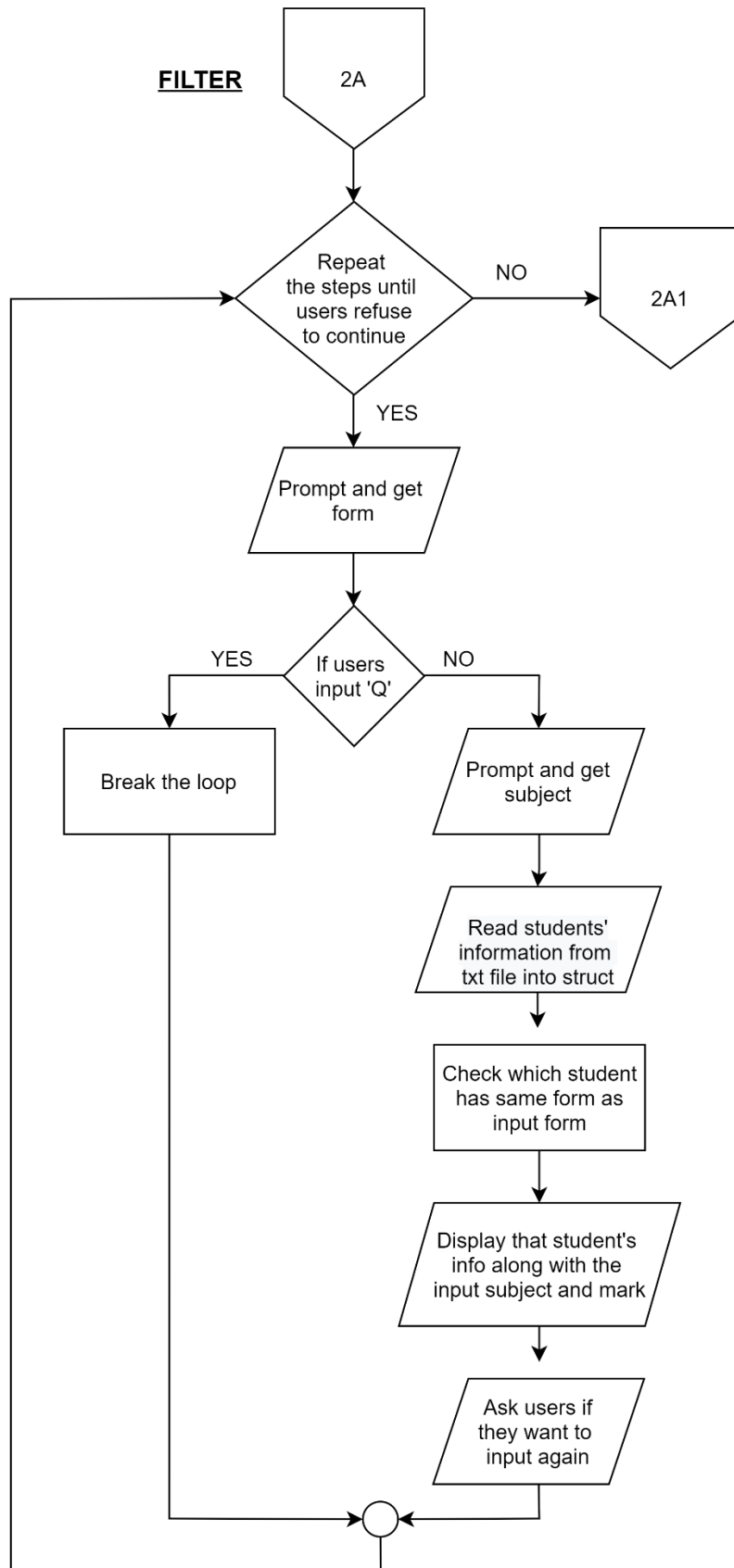


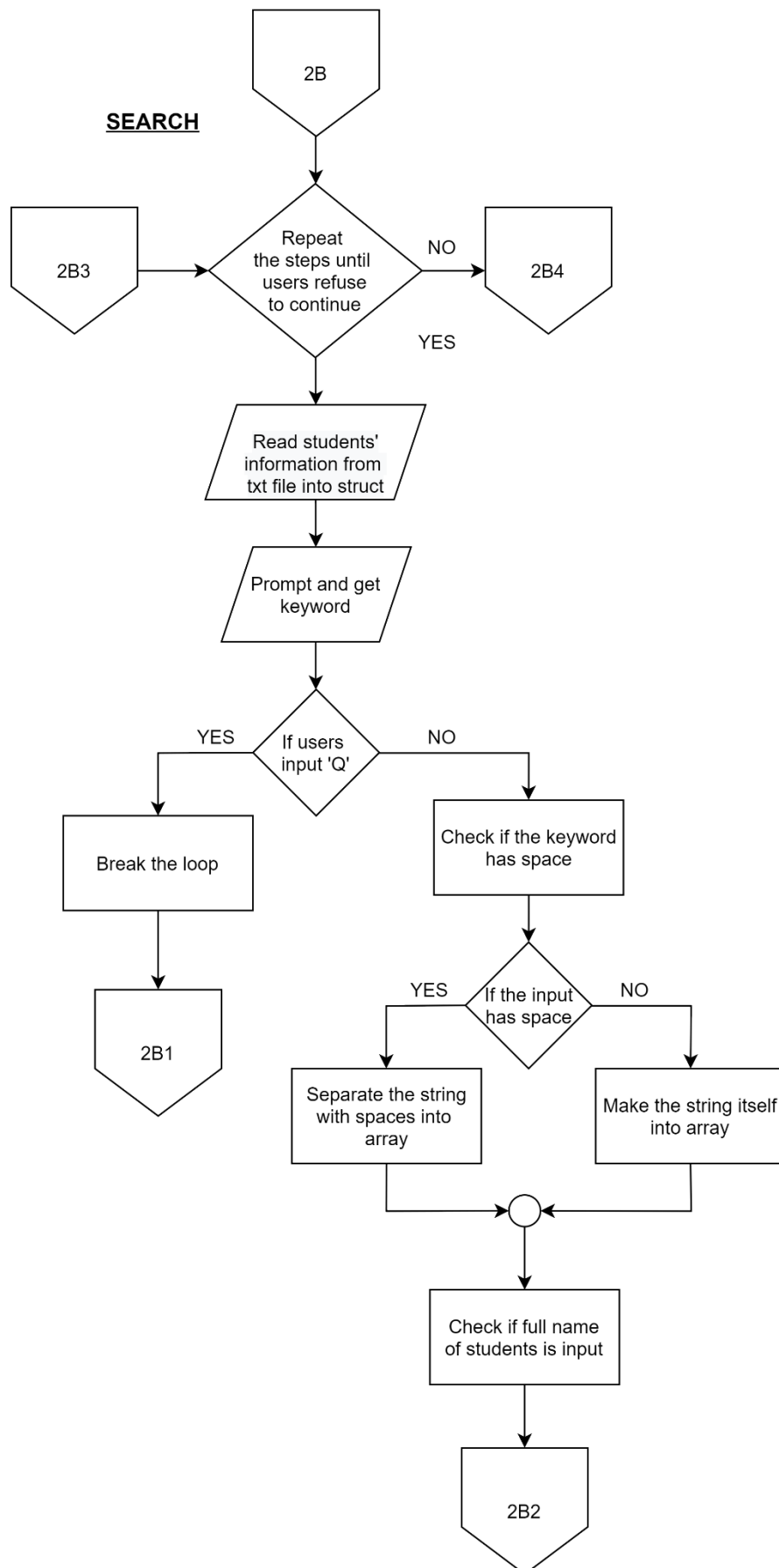
**ADDING
MODULE**

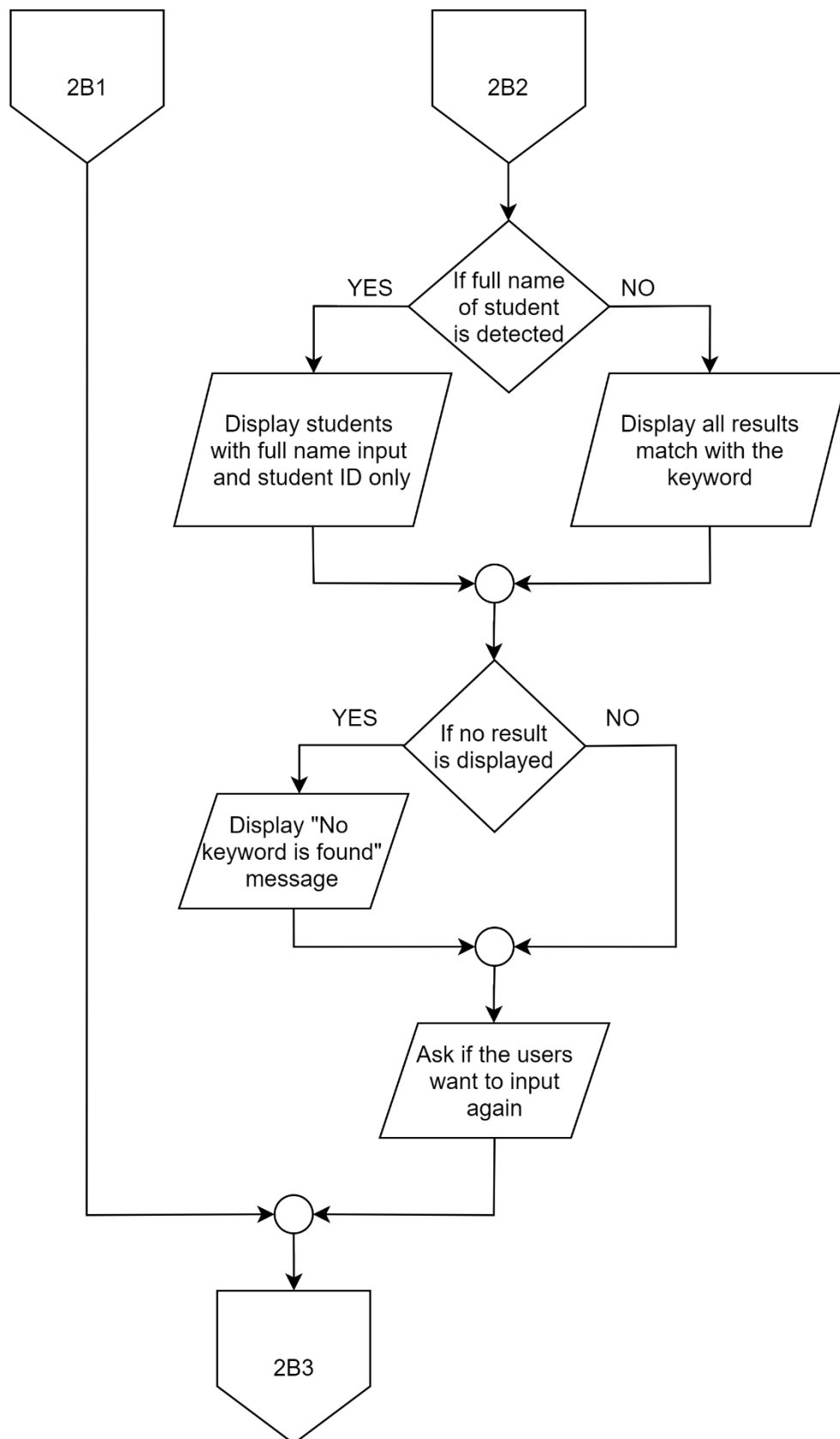


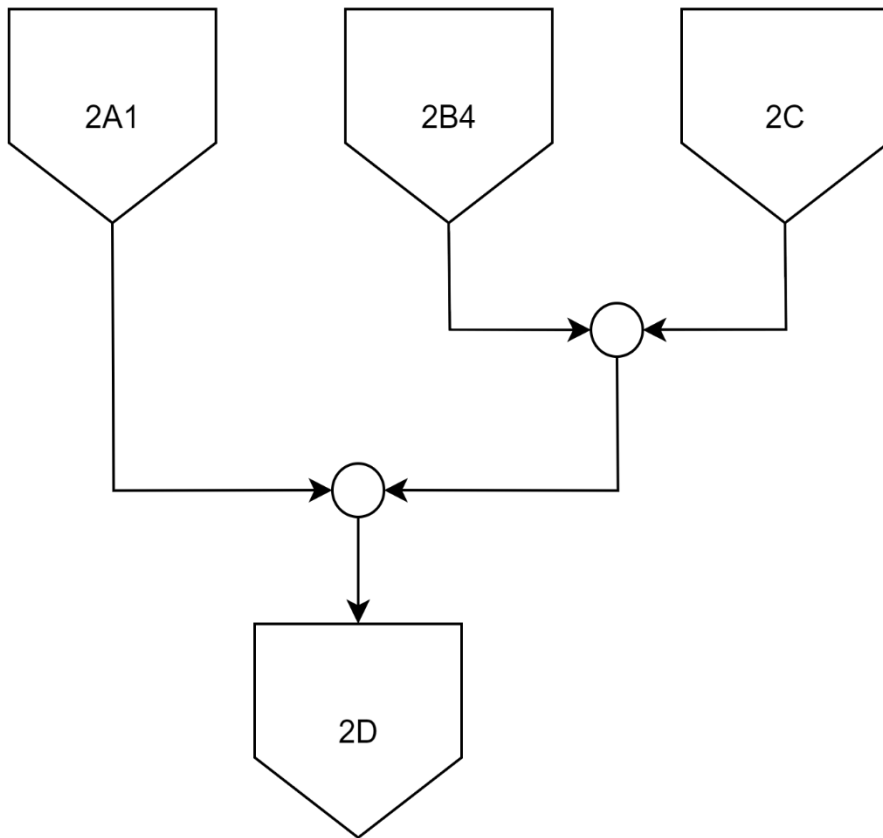
**FILTER/SEARCH
MODULE**

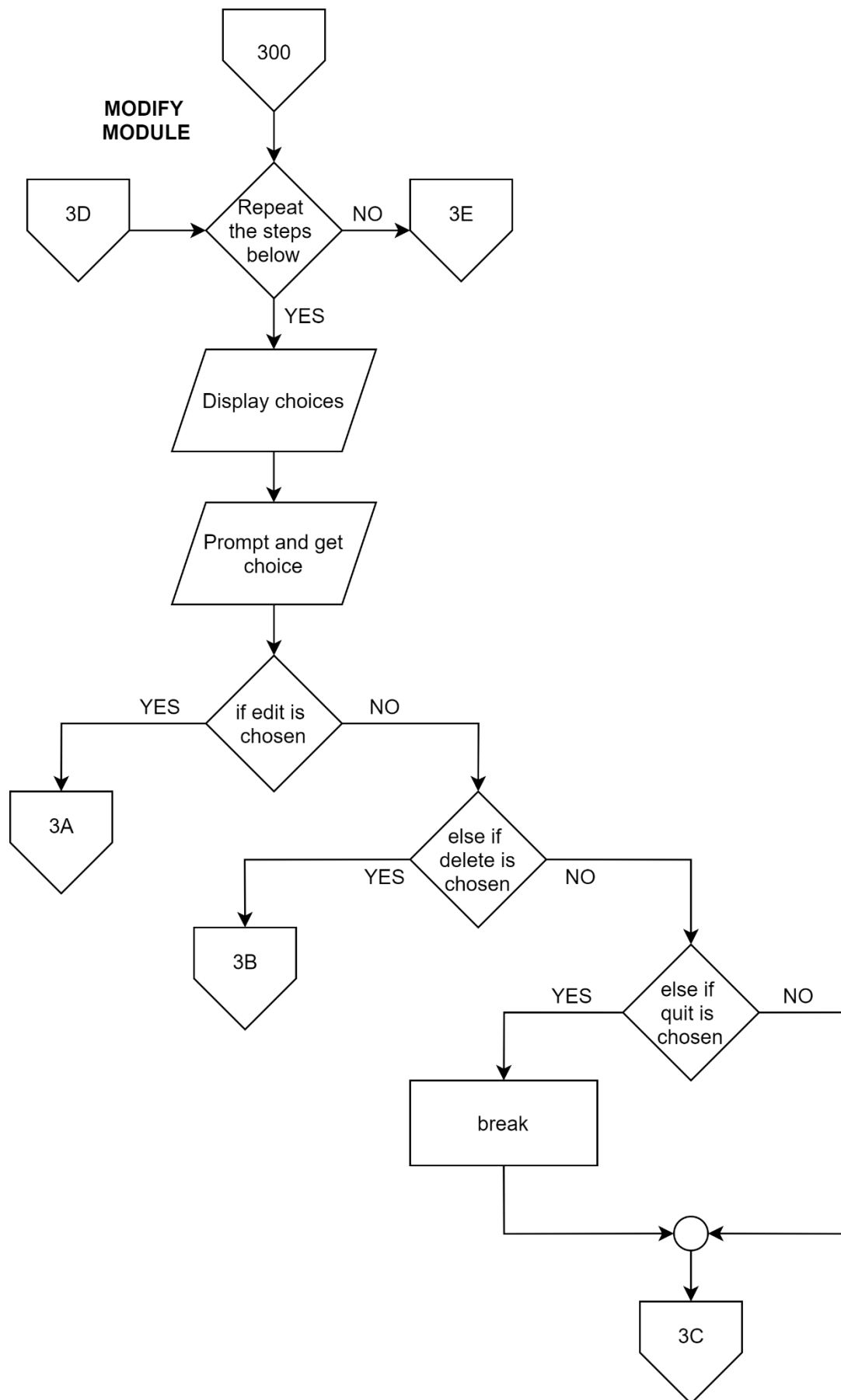


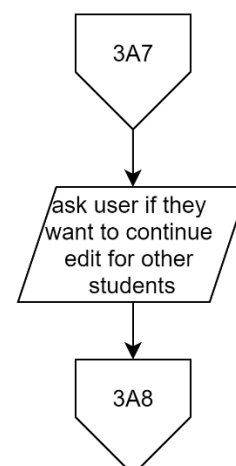
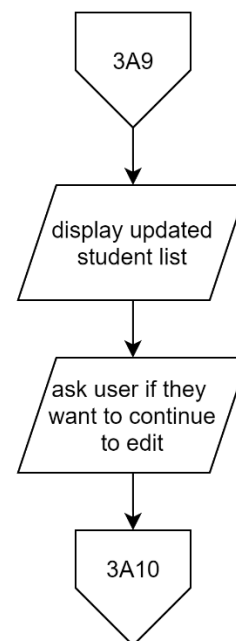
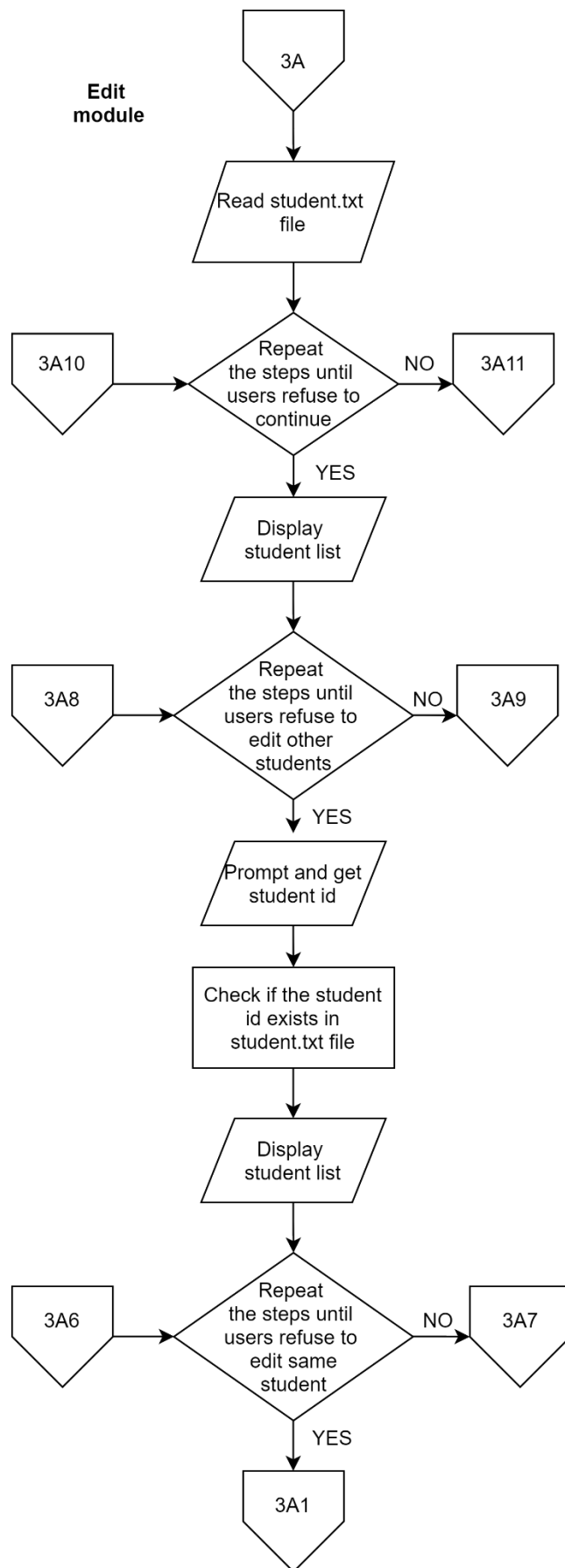


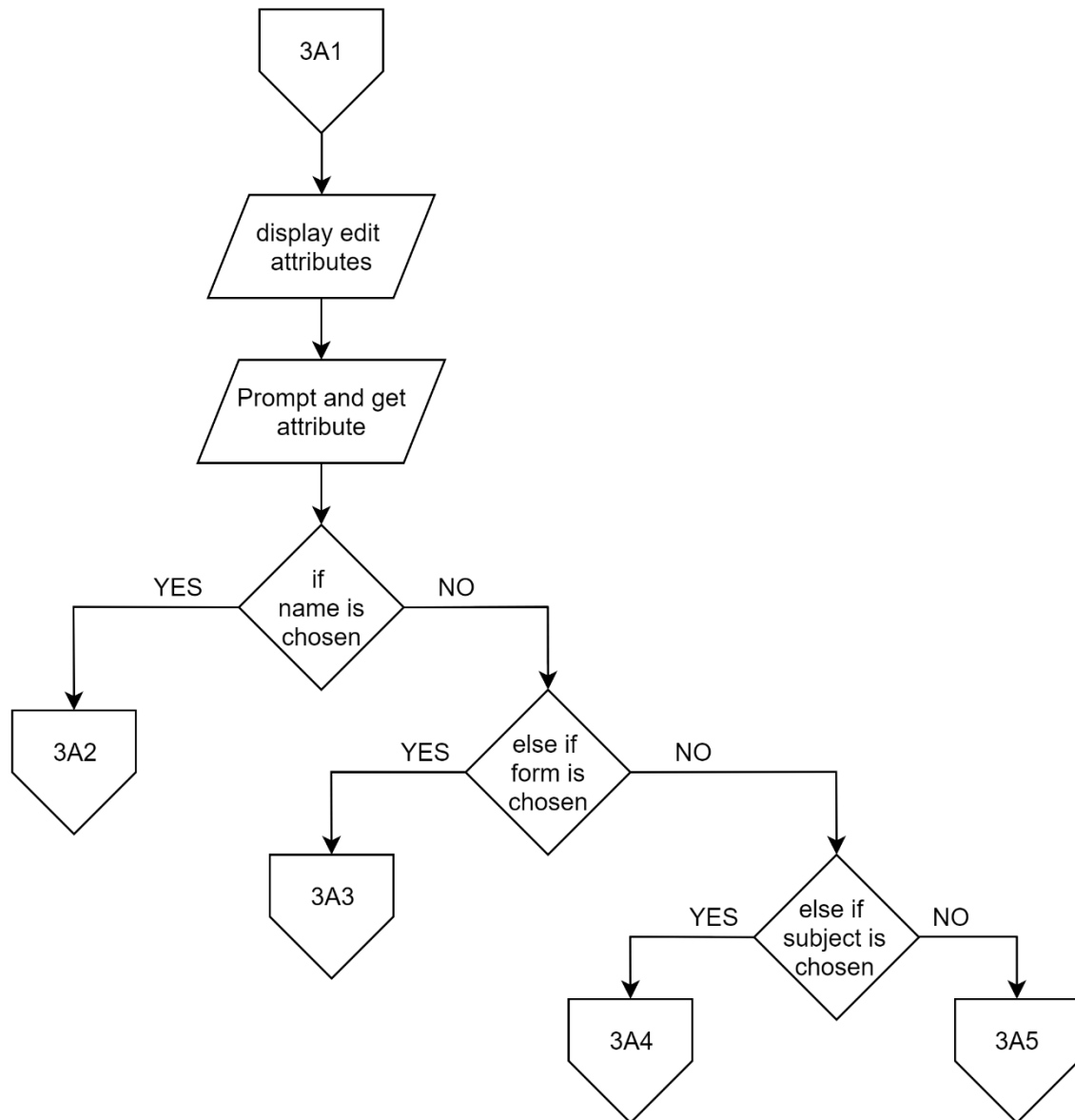


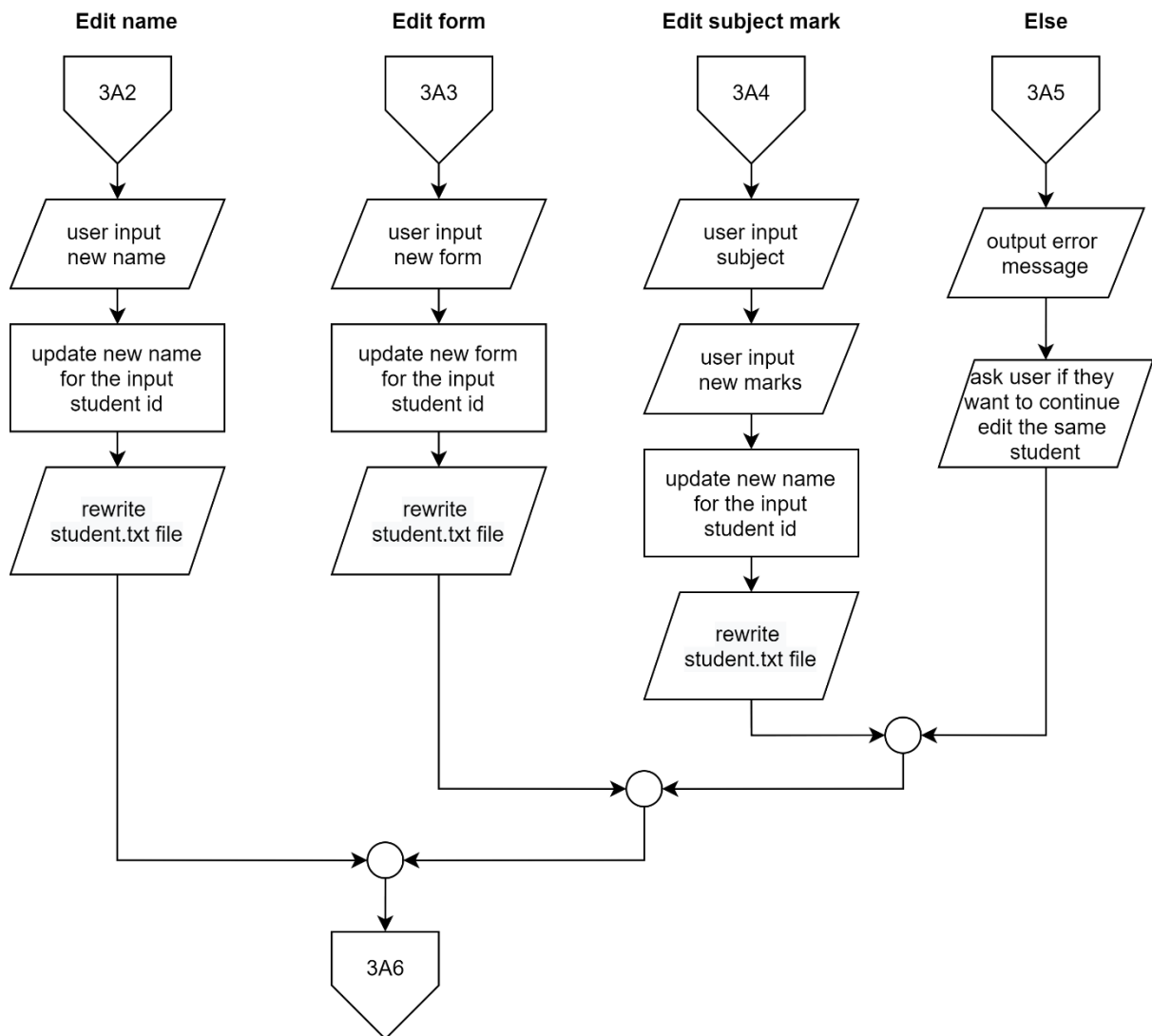


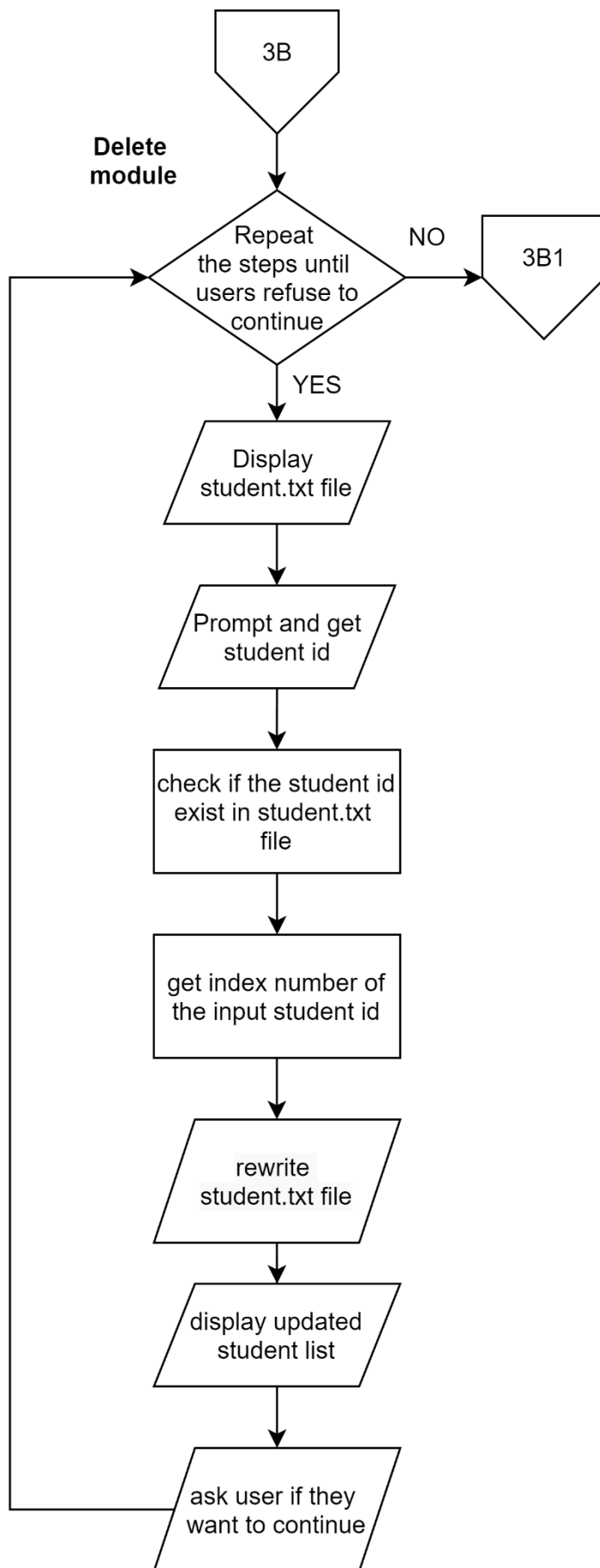


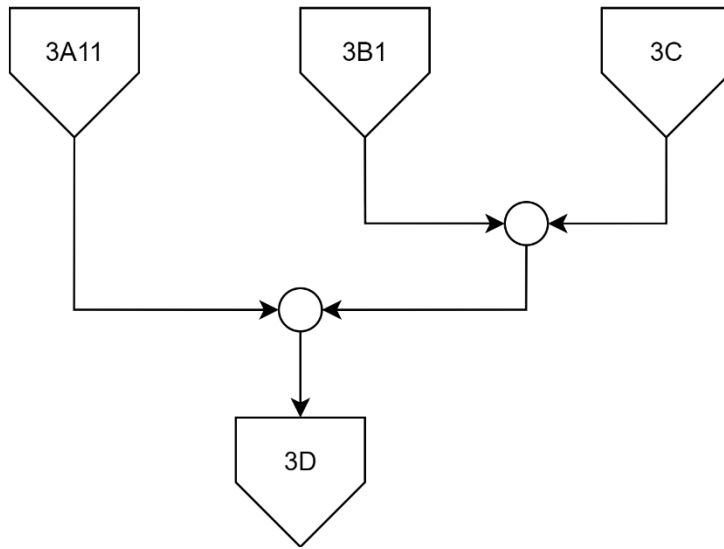


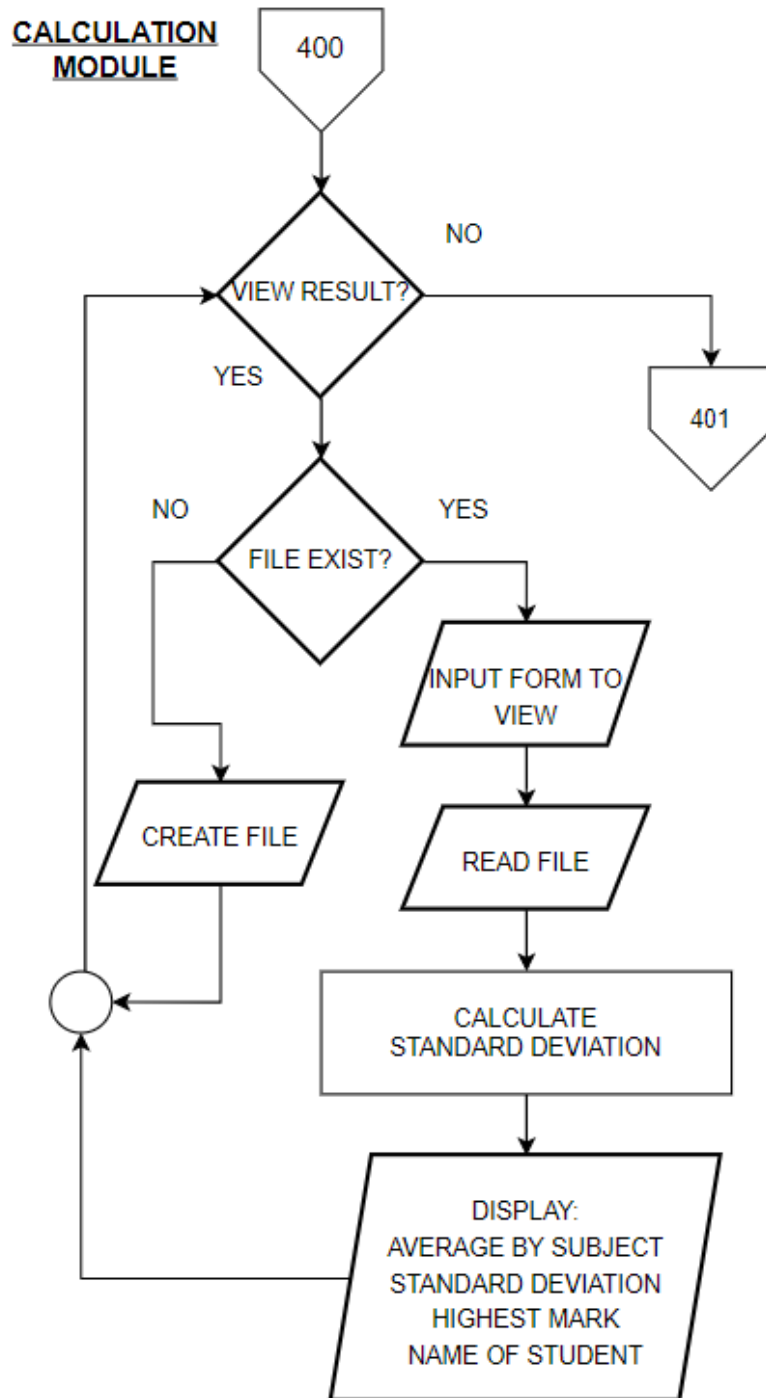


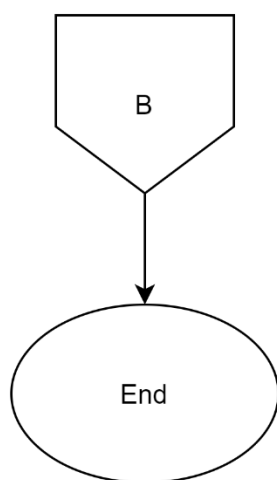
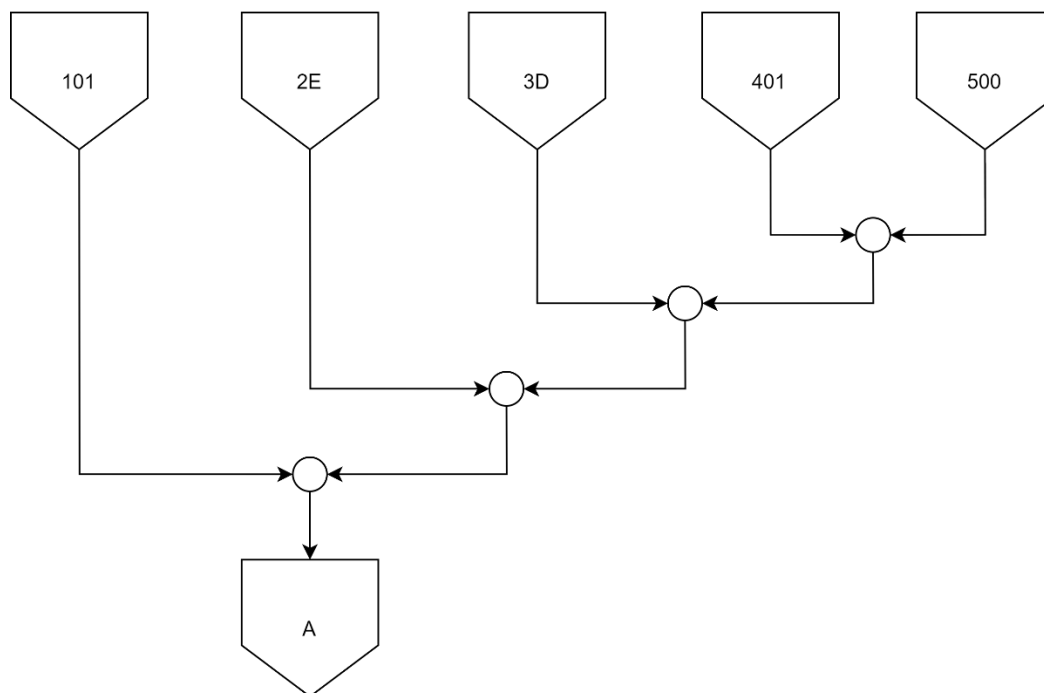












Pseudocode

Repeat the following steps until Quit is selected:

 Display the main menu.

 If Insert is selected,

 Check if "student.txt" exists, if not, create file

 Repeat the following steps until users refuse to continue:

 Read all student's information from txt file into struct.

 Prompt and get new student ID

 If User input id with correct format:

 Check if student ID is repeated

 If student ID is not repeated:

 Prompt and get name, form, and student marks

 Open "student.txt" to append

 If "student.txt" is not empty:

 Rewrite the file with arranged sequence of students

 Else

 Write the file with new students' information

 Close "student.txt"

 Else

 Please make sure no replication of ID

 Ask if the users want to continue

 Else if Filter/Search is selected, repeat the steps below:

 Display the choices for the users.

 Prompt and get choices.

 If Filter is selected, repeat the steps below until users refuse to continue:

 Prompt and get form

 Prompt and get subject

 Read all student's information from txt file into struct.

 Check which student has same form as input form.

 Display that student's info along with the input subject.

 Ask if the users want to input again.

 Else if Search is selected, repeat the steps below until users refuse to continue:

 Prompt and get keyword.

 Check if the keyword has space.

 If it has space, separate the string with spaces into array.

 Else, make the string itself into array.

 Check if full name of students is input.

 If yes, display students with full name input and student ID only

 Else, display all results match with the keyword.

 If no result is displayed, output error message.

 Ask if the users want to input again.

 Else if Quit is selected, break the while loop.

Else if Modify is selected, repeat the steps below:

Ask user to input choice to edit or delete

If choice is 1 (edit module), read student.txt file into struct

Repeat the steps below until users refuse to continue to edit:

Display student list using struct

Repeat the steps below until users refuse to edit for other students:

Prompt and get student ID

Check if the student ID is in correct format and exists in file

Display students list using struct

Repeat the steps below until user refuse to edit for the same student:

Prompt and get attribute to edit

If name is chosen,

Ask users to input new name

Update name for the student id and rewrite the student.txt file

If Form is chosen

Ask users to input new form

Update form for the student id and rewrite the student.txt file

If Subject is chosen

Ask users to input new subject and mark

Update the mark(subject) for the student id and rewrite the student.txt file

Else output error message

Ask users if they want to continue to edit same student

Ask user if they want to continue to edit for other students

Display updated student list.

Ask users if they want to continue to edit

Else if choice is 2 (delete module), repeat the steps below until users refuse to continue:

Read student.txt file into struct

Display student list using struct

Prompt and get student ID

Check if the student ID exists in the txt file

Get the index number of the input student id

Skip the info of the input student id when rewrite student.txt file

Read student.txt file into struct

Display updated student list using struct

Ask users if they want to continue to delete

Else if choice is 3, quit the modify function

Else if View Results is selected,

Open "student.txt" file

Repeat following steps until users refuse to continue:

If file is failed to be opened

```
        create the file and end the session
    Else if file exists:
        If file has content:
            Prompt and get form
            Read the file "student.txt"
            Calculate information inside
            Display the result
        Else
            Output "student.txt" has no record!
    Ask if the users want to continue

Else if Quit is selected, exit the program.
```

Test cases

ADDING MODULE:

TEST CASE 1a: Error student ID test

C:\Users\Totroseah\Desktop\UTAR\UnderGd\Y1S1\UCCD1004 Programming Concepts and Practices

```
Please enter student information and marks [Q to exit].
Student ID   : 11abc11111
Error        : Please enter the ID in correct format (01AAA1234)!
               Or type [Q] to exit
Student ID   :
```

TEST CASE 1b: Form Input Error Test

C:\Users\Totroseah\Desktop\UTAR\UnderGd\Y1S1\UCCD1004 Programming Concepts and Practices

```
Please enter student information and marks [Q to exit].
Student ID   : 11abc11111
Error        : Please enter the ID in correct format (01AAA1234)!
               Or type [Q] to exit
Student ID   : 11abc11111
Name         : Seah Eu Jin
Form         : 4
Error        : Only '1','2','3' is allowed to be input
Form         :
```

TEST CASE 1c: Marks Input Error Test

C:\Users\Totroseah\Desktop\UTAR\UnderGd\Y1S1\UCCD1004 Programming Concepts and Practices

```
Please enter student information and marks.
Student ID   : 11ABC1111
Name         : Seah Eu Jin
Form         : 2
_____ Marks _____
BM mark      :
BI mark      :
BC mark      :
MATH mark    :
SCI mark     :
SEJARAHAH mark :
GEO mark     :
-----
BM mark : 333
Scanning input...
Error! Input must be an integer(within 0 to 100)!
Press any key to continue . . .
```

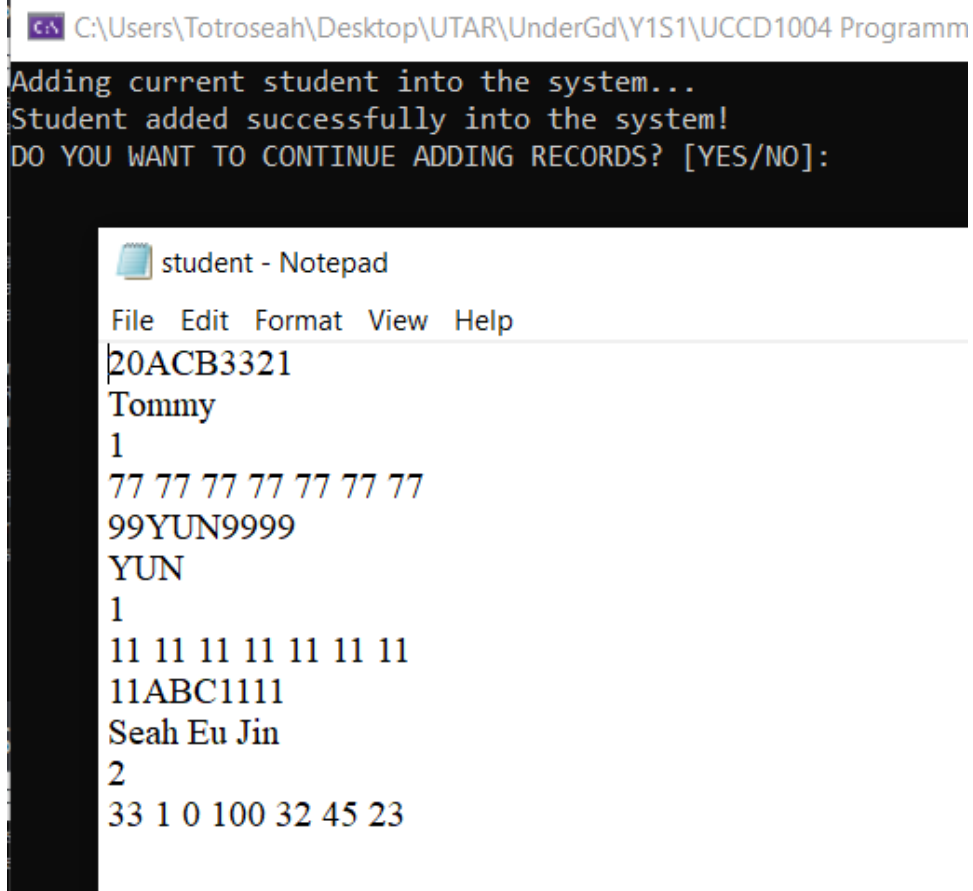
TEST CASE 1d: Marks Input Error Test 2

```
C:\Users\Totroseah\Desktop\UTAR\UnderGd\Y1S1\UCCD1004 Programming Concepts and Practices\
Please enter student information and marks.
Student ID   : 11ABC1111
Name        : Seah Eu Jin
Form        : 2
            Marks
-----
BM mark      :
BI mark      :
BC mark      :
MATH mark    :
SCI mark     :
SEJARAH mark :
GEO mark     :
-----
BM mark : -56
Scanning input...
Error! Input must be an integer(within 0 to 100)!
Press any key to continue . . .
```

TEST CASE 1e: Marks Input Error Test 3

```
C:\Users\Totroseah\Desktop\UTAR\UnderGd\Y1S1\UCCD1004 Programming Concepts and Practices\
Please enter student information and marks.
Student ID   : 11ABC1111
Name        : Seah Eu Jin
Form        : 2
            Marks
-----
BM mark      :
BI mark      :
BC mark      :
MATH mark    :
SCI mark     :
SEJARAH mark :
GEO mark     :
-----
BM mark : a
Scanning input...
Error! Input must be an integer(within 0 to 100)!
Press any key to continue . . .
```

TEST CASE 1f: Check file Input



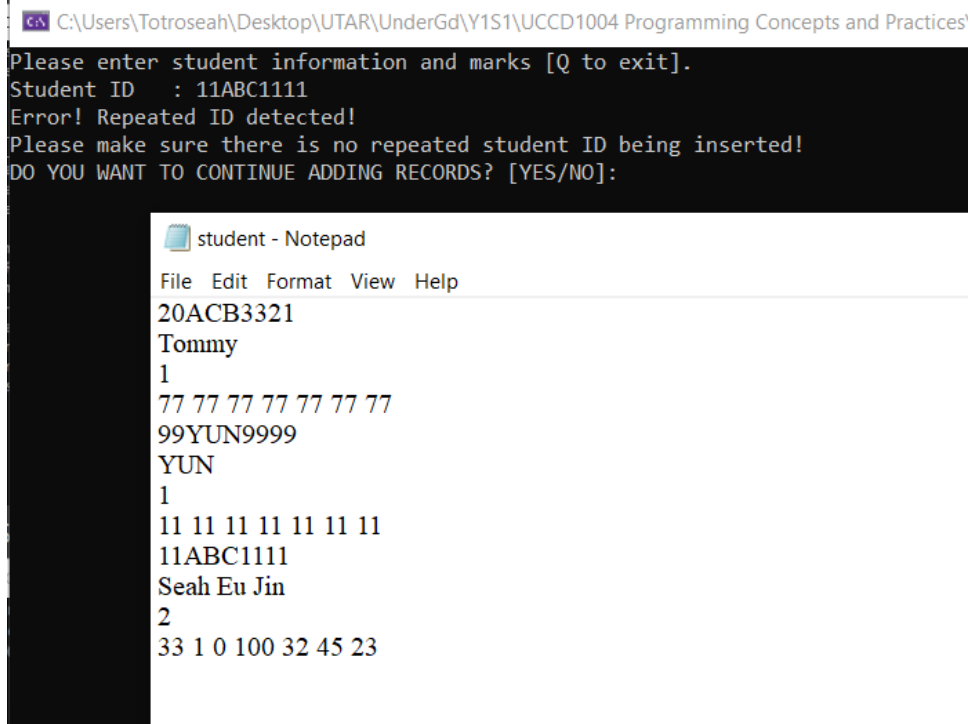
The screenshot shows a Windows command prompt window with the title bar 'C:\Users\Totroseah\Desktop\UTAR\UnderGd\Y1S1\UCCD1004 Programm'. The command prompt displays the following text:

```
Adding current student into the system...
Student added successfully into the system!
DO YOU WANT TO CONTINUE ADDING RECORDS? [YES/NO]:
```

Below the command prompt is a Notepad window titled 'student - Notepad'. The Notepad window contains the following text:

```
File Edit Format View Help
20ACB3321
Tommy
1
77 77 77 77 77 77 77
99YUN9999
YUN
1
11 11 11 11 11 11 11
11ABC1111
Seah Eu Jin
2
33 1 0 100 32 45 23
```

TEST CASE 1g: Replication Student ID Input Error



The screenshot shows a Windows command prompt window with the title bar 'C:\Users\Totroseah\Desktop\UTAR\UnderGd\Y1S1\UCCD1004 Programming Concepts and Practices'. The command prompt displays the following text:

```
Please enter student information and marks [Q to exit].
Student ID : 11ABC1111
Error! Repeated ID detected!
Please make sure there is no repeated student ID being inserted!
DO YOU WANT TO CONTINUE ADDING RECORDS? [YES/NO]:
```

Below the command prompt is a Notepad window titled 'student - Notepad'. The Notepad window contains the following text:


```
File Edit Format View Help
20ACB3321
Tommy
1
77 77 77 77 77 77 77
99YUN9999
YUN
1
11 11 11 11 11 11 11
11ABC1111
Seah Eu Jin
2
33 1 0 100 32 45 23
```

TEST CASE 1h: Continue Response Error Test

```
C:\Users\Totroseah\Desktop\UTAR\UnderGd\Y1S1\UCCD1004 Programming Concepts and Practices\
Please enter student information and marks [Q to exit].
Student ID   : 11ABC1111
Error! Repeated ID detected!
Please make sure there is no repeated student ID being inserted!
DO YOU WANT TO CONTINUE ADDING RECORDS? [YES/NO]: Maybe
Error: Enter only [YES] or [NO]
DO YOU WANT TO CONTINUE ADDING RECORDS? [YES/NO]:
```

FILTER/SEARCH Module:


TEST CASE 2a: Filter Form Input Validation

 C:\Users\Acer\Desktop\Programming Practical\Assignment 2\Group 32\Debug\Group 32.exe

```
=====
>>      Please key in the information that you want to filter.      <<
>>      You can filter multiple forms at a same time.              <<
>>      Press Q to exit.                                           <<
=====

Form      : asdsdd
Error     : Please key in 1/2/3 only
Form      : 456576
Error     : Please key in 1/2/3 only
Form      : a b c
Error     : Please key in 1/2/3 only
Form      : 0 2 4
Error     : Please key in 1/2/3 only
Form      : 1 2
```


TEST CASE 2b: Filter Subject Input Validation

 C:\Users\Acer\Desktop\Programming Practical\Assignment 2\Group 32\De

```
=====
>> Lists of subject: BM BI BC Math Sci Sejarah Geo <<
=====

Form      : 1 2
Subjects: asdasa
Error     : Please input subject correctly.
Subjects: 4234234
Error     : Please input subject correctly.
Subjects: bm sds bc
Error     : Please input subject correctly.
Subjects: asda bm bi
Error     : Please input subject correctly.
Subjects: bm bi
```


TEST CASE 2c: Filter Result and Continue Validation

 C:\Users\Acer\Desktop\Programming Practical\Assignment 2\Group 32\

```
=====
Form    : 1 2
Subject: bm bi

=====
Featured list:
Student ID:20ACD1234
Student Name: Alice Lim
Form: 1
BM: 11
BI: 11


Student ID:20ACD1235
Student Name: Derrick Tan
Form: 1
BM: 22
BI: 22

Student ID:20ACD1236
Student Name: Nelson Khoo
Form: 2
BM: 33
BI: 33

Student ID:20ACD0953
Student Name: Ricka Tan
Form: 2
BM: 44
BI: 44

DO YOU WANT TO CONTINUE TO FILTER (YES/NO): asdasd
Error: Enter only [YES] or [NO]
DO YOU WANT TO CONTINUE TO FILTER (YES/NO): 1232
Error: Enter only [YES] or [NO]
DO YOU WANT TO CONTINUE TO FILTER (YES/NO): yes_
```

TEST CASE 2d: Search Keyword Input Validation

 C:\Users\Acer\Desktop\Programming Practical\Assignment 2\Group 32\Debug\Gr

```
=====
Searched Keyword: 20ACD1234 E T
=====

Search Results:
Student ID:20ACD1234
Student Name: Alice Lim
Form: 1
BM: 11
BI: 11
BC: 11
Math: 11
Sci: 11
Sejarah: 11
Geo: 11

Student ID:20ACD1237
Student Name: Eric Tan
Form: 3
BM: 55
BI: 55
BC: 55
Math: 55
Sci: 55
Sejarah: 55
Geo: 55

DO YOU WANT TO CONTINUE TO SEARCH (YES/NO):
```

C:\Users\Acer\Desktop\Programming Practical\Assignment 2\Group 32\Debug'

```
=====
Searched Keyword: Nelson Khoo Alice
=====

Search Results:
Student ID:20ACD1236
Student Name: Nelson Khoo
Form: 2
BM: 33
BI: 33
BC: 33
Math: 33
Sci: 33
Sejarah: 33
Geo: 33

DO YOU WANT TO CONTINUE TO SEARCH (YES/NO): _
```

```
=====
Searched Keyword: 20ACD1234 Khoo
=====
```

Search Results:

Student ID:20ACD1234

Student Name: Alice Lim

Form: 1

BM: 11

BI: 11

BC: 11

Math: 11

Sci: 11

Sejarah: 11

Geo: 11

Student ID:20ACD1236

Student Name: Nelson Khoo

Form: 2

BM: 33

BI: 33

BC: 33

Math: 33

Sci: 33

Sejarah: 33

Geo: 33

Student ID:20ACD3614

Student Name: Windy Khoo

Form: 3

BM: 12

BI: 23

BC: 34


Math: 45

Sci: 56

Sejarah: 90

Geo: 78

DO YOU WANT TO CONTINUE TO SEARCH (YES/NO): ☐

 C:\Users\Acer\Desktop\Programming Practical\Assignment 2\Group 32\Debug

```
=====
Searched Keyword: Nelson Alice
=====
```

```
Search Results:
Student ID:20ACD1234
Student Name: Alice Lim
Form: 1
BM: 11
BI: 11
BC: 11
Math: 11
Sci: 11
Sejarah: 11
Geo: 11
```

```
Student ID:20ACD1236
Student Name: Nelson Khoo
Form: 2
BM: 33
BI: 33
BC: 33
Math: 33
Sci: 33
Sejarah: 33
Geo: 33
```

```
DO YOU WANT TO CONTINUE TO SEARCH (YES/NO): _
```

C:\Users\Acer\Desktop\Programming Practical\Assignment 2\Group 32\Debug

```
=====
Searched Keyword: 20acd1234 oo
=====
```

Search Results:

Student ID:20ACD1236

Student Name: Nelson Khoo

Form: 2

BM: 33

BI: 33

BC: 33

Math: 33

Sci: 33

Sejarah: 33

Geo: 33

Student ID:20ACD3614

Student Name: Windy Khoo

Form: 3

BM: 12

BI: 23

BC: 34

Math: 45

Sci: 56

Sejarah: 90

Geo: 78

DO YOU WANT TO CONTINUE TO SEARCH (YES/NO): ☐

MODIFY

EDIT MODULE:

Test case 3a: Invalid student id error test

```
Student ID: 20ACD1236
Student Name: Nelson Khoo
Form: 2
BM: 78
BI: 83
BC: 78
Math: 80
Sci: 81
Sejarah: 77
Geo: 83

>> Please enter the student id that you want to edit [Q to exit] <<
Student ID   : 20acc12csadc
Error        : Please enter the ID in correct format (01AAA1234)!
               Or type [Q] to exit
Student ID   :
```

Test case 3b: Valid student id but does not exist in student.txt file

```
Student ID: 20ACD1236
Student Name: Nelson Khoo
Form: 2
BM: 78
BI: 83
BC: 78
Math: 80
Sci: 81
Sejarah: 77
Geo: 83

>> Please enter the student id that you want to edit [Q to exit] <<
Student ID   : 20acc1236
Error        : The student ID does not exist!

DO YOU WANT TO EDIT RECORD FOR OTHER STUDENTS? (YES/NO): yes
>> Please enter the student id that you want to edit [Q to exit] <<
Student ID   :
```

Test case 3c: Edit name

```
Student ID: 20ACD1237
Student Name: Eric Tan
Form: 3
BM: 70
BI: 64
BC: 60
Math: 65
Sci: 66
Sejarah: 72
Geo: 71

Please update the name for student ID 20ACD1237: Michael Tan
DO YOU STILL HAVE ANYTHING TO EDIT FOR THIS STUDENTS? (YES/NO): no
DO YOU WANT TO EDIT RECORD FOR OTHER STUDENTS? (YES/NO): no_
```

```
Student ID: 20ACD1237
Student Name: Michael Tan
Form: 3
BM: 70
BI: 64
BC: 60
Math: 65
Sci: 66
Sejarah: 72
Geo: 71

DO YOU WANT TO CONTINUE TO EDIT? (YES/NO):_
```

Test case 3d: Edit form

```
Student ID: 20ACD1236
Student Name: donnie yip
Form: 2
BM: 78
BI: 83
BC: 78
Math: 80
Sci: 81
Sejarah: 77
Geo: 83

Please update the form (1/2/3) for student ID20ACD1236: 3
DO YOU STILL HAVE ANYTHING TO EDIT FOR THIS STUDENTS? (YES/NO): no
DO YOU WANT TO EDIT RECORD FOR OTHER STUDENTS? (YES/NO): no_
```

```
Student ID: 20ACD1236
Student Name: donnie yip
Form: 3
BM: 78
BI: 83
BC: 78
Math: 80
Sci: 81
Sejarah: 77
Geo: 83

DO YOU WANT TO CONTINUE TO EDIT? (YES/NO):
```

Test case 3e: form error test

```
Student ID: 20ACD1236
Student Name: donnie yip
Form: 3
BM: 78
BI: 83
BC: 78
Math: 80
Sci: 81
Sejarah: 77
Geo: 83

Please update the form (1/2/3) for student ID 20ACD1236: 4
Error          : Only '1','2','3' is allowed to be input
Please update the form (1/2/3) for student ID 20ACD1236: a
Error          : Only '1','2','3' is allowed to be input
Please update the form (1/2/3) for student ID 20ACD1236: 123
Error          : Only '1','2','3' is allowed to be input
Please update the form (1/2/3) for student ID 20ACD1236: _
```

Test case 3f: Edit subject marks

```
Student ID: 20ACD1236
Student Name: donnie yip
Form: 3
BM: 78
BI: 83
BC: 78
Math: 80
Sci: 81
Sejarah: 77
Geo: 83

Please enter the subject that you want to change the mark (BM/BI/BC/Math/Sci/Sejarah/Geo):bm
Please update the mark for BM in range 0 - 100: 80
DO YOU STILL HAVE ANYTHING TO EDIT FOR THIS STUDENTS? (YES/NO): no
DO YOU WANT TO EDIT RECORD FOR OTHER STUDENTS? (YES/NO): no_
```

```
Student ID: 20ACD1236
Student Name: donnie yip
Form: 3
BM: 80
BI: 83
BC: 78
Math: 80
Sci: 81
Sejarah: 77
Geo: 83

DO YOU WANT TO CONTINUE TO EDIT? (YES/NO):
```

Test case 3g: subject error test

```
Student ID: 20ACD1236
Student Name: donnie yip
Form: 3
BM: 80
BI: 83
BC: 78
Math: 80
Sci: 81
Sejarah: 77
Geo: 83

Please enter the subject that you want to change the mark (BM/BI/BC/Math/Sci/Sejarah/Geo):c
Please enter the subject that you want to change the mark (BM/BI/BC/Math/Sci/Sejarah/Geo):gee
Please enter the subject that you want to change the mark (BM/BI/BC/Math/Sci/Sejarah/Geo):123
Please enter the subject that you want to change the mark (BM/BI/BC/Math/Sci/Sejarah/Geo):1
Please enter the subject that you want to change the mark (BM/BI/BC/Math/Sci/Sejarah/Geo):_
```

Test case 3h: mark error test

```
Student ID: 20ACD1236
Student Name: donnie yip
Form: 3
BM: 80
BI: 83
BC: 78
Math: 80
Sci: 81
Sejarah: 77
Geo: 83

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 in range 0 - 100: a
Please update the mark for SCI in range 0 - 100: 101
Please update the mark for SCI in range 0 - 100: -1
Please update the mark for SCI in range 0 - 100: avd
Please update the mark for SCI in range 0 - 100:
```

Test case 3i: Delete a student id

```
Student ID: 20ACD1235
Student Name: Derrick Tan
Form: 1
BM: 77
BI: 73
BC: 71
Math: 65
Sci: 68
Sejarah: 65
Geo: 65
```

```
Student ID: 20ACD1236
Student Name: Sean Tan
Form: 1
BM: 80
BI: 83
BC: 78
Math: 80
Sci: 90
Sejarah: 77
Geo: 83
```

```
>> Please enter the student id that you want to delete [Q to exit] <<
Student ID : 20acd1236
```

```
Student ID: 20ACD1235
Student Name: Derrick Tan
Form: 1
BM: 77
BI: 73
BC: 71
Math: 65
Sci: 68
Sejarah: 65
Geo: 65
```

```
DO YOU WANT TO CONTINUE TO DELETE? (YES/NO):
```

CALCULATION MODULE:

TEST CASE 4a: Invalid Form Input Error Test

```
C:\Users\Totroseah\Desktop\UTAR\UnderGd\Y1S1\UCCD1004 Programming Concepts and Practices\
Detecting relevant files...
Locating "student.txt"...
Please choose the form that you wish to view [All,1,2,3]: 4
Please choose the form that you wish to view [All,1,2,3]:
```

TEST CASE 4b: View Null Information Error Test

```
C:\Users\Totroseah\Desktop\UTAR\UnderGd\Y1S1\UCCD1004 Programming Concepts and Practices\
Form 3:
BM
Average          :-nan(ind)
Standard deviation :-nan(ind)
Highest Score     :0
Highest score student :
BI
Average          :-nan(ind)
Standard deviation :-nan(ind)
Highest Score     :0
Highest score student :
BC
Average          :-nan(ind)
Standard deviation :-nan(ind)
Highest Score     :0
Highest score student :
MATH
Average          :-nan(ind)
Standard deviation :-nan(ind)
Highest Score     :0
Highest score student :
SCI
Average          :-nan(ind)
Standard deviation :-nan(ind)
Highest Score     :0
Highest score student :
SEJARAH
Average          :-nan(ind)
Standard deviation :-nan(ind)
Highest Score     :0
Highest score student :
GEO
Average          :-nan(ind)
Standard deviation :-nan(ind)
Highest Score     :0
Highest score student :
DO YOU WANT TO CONTINUE TO VIEW OTHER RESULT (YES/NO):
```

student - Notepad
File Edit Format View Help
20ACB3321
Tommy
1
77 77 77 77 77 77 77
99YUN9999
YUN
1
11 11 11 11 11 11 11
11ABC1111
Seah Eu Jin
2
33 1 0 100 32 45 23

TEST CASE 4c: View result with correct Calculation

```
C:\Users\Totroseah\Desktop\UTAR\UnderGd\Y1S1\UCCD1004 Programming Concepts and Practices\Assignment 2\Assi
Please choose the form that you wish to view [All,1,2,3]: 1

Form 1:
BM
Average           :44.00
Standard deviation :33.00
Highest Score     :77
Highest score student :Tommy

BI
Average           :44.00
Standard deviation :33.00
Highest Score     :77
Highest score student :Tommy

BC
Average           :44.00
Standard deviation :33.00
Highest Score     :77
Highest score student :Tommy

MATH
Average           :44.00
Standard deviation :33.00
Highest Score     :77
Highest score student :Tommy

SCI
Average           :44.00
Standard deviation :33.00
Highest Score     :77
Highest score student :Tommy

SEJARAH
Average           :44.00
Standard deviation :33.00
Highest Score     :77
Highest score student :Tommy

GEO
Average           :44.00
Standard deviation :33.00
Highest Score     :77
Highest score student :Tommy

DO YOU WANT TO CONTINUE TO VIEW OTHER RESULT (YES/NO):
```

```
student - Notepad
File Edit Format View Help
20ACB3321
Tommy
1
77 77 77 77 77 77 77
99YUN9999
YUN
1
11 11 11 11 11 11 11
11ABC1111
Seah Eu Jin
2
33 1 0 100 32 45 23
Ln 1, Col 1
```

TEST CASE 4d: View result with correct Calculation 2

C:\Users\Totroseah\Desktop\UTAR\UnderGd\Y1S1\UCCD1004 Programming Concepts and Practices\Assignment 2\Assignn

Detecting relevant files...

Locating "student.txt"...

Please choose the form that you wish to view [All,1,2,3]: 1

Form 1:

BM

Average :52.50

Standard deviation :27.29

Highest Score :77

Highest score student :Ash, Tommy

BI

Average :44.25

Standard deviation :33.25

Highest Score :78

Highest score student :Kong Kong Kong

BC

Average :44.50

Standard deviation :23.34

Highest Score :77

Highest score student :Tommy

MATH

Average :54.50

Standard deviation :25.59

Highest Score :77

Highest score student :Tommy

SCI

Average :49.50

Standard deviation :25.20

Highest Score :77

Highest score student :Tommy

SEJARAH

Average :55.75

Standard deviation :27.36

Highest Score :79

Highest score student :Kong Kong Kong

GEO

Average :49.50

Standard deviation :28.20

Highest Score :77

Highest score student :Tommy

DO YOU WANT TO CONTINUE TO VIEW OTHER RESULT (YES/NO):

student - Notepad

File Edit Format View Help

23RYU1234

Ash

1

77 11 45 65 44 56 34

45TYU0998

Kong Kong Kong

1

45 78 45 65 66 79 76

20ACB3321

Tommy

1

77 77 77 77 77 77 77

99YUN9999

YUN

1

11 11 11 11 11 11 11

11ABC1111

Seah Eu Jin

2

33 1 0 100 32 45 23

Ln 1, Col 1

Source Code

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

struct student {
    string ID;
    string name;
    char FORM;
    int mark[7];
}; student number[100];

const string SUBJECT[7]{ "BM","BI","BC","MATH","SCI","SEJARAH","GEO" };
const string displaySubject[7]{ "BM","BI","BC","Math","Sci","Sejarah","Geo" };
string pass{  };
ofstream myfile;

//Lists of function prototype
string string_toupper(string&);                //uppercase letter in a string
int check_id(string&);                          //validate the id input to pre-set format
string get_form(string&);                      //restrict the form input to 1,2,3
string confirm_marks(string&);                 //read marks input from the user
bool politely_ask(string);                    //if input not "YES" or "NO", will ask again
for the input
void student_mark(string[], string, string, string); //prompt to get student marks from
user (marks, id, name, form)
void splitStringWithSpaceIntoArray(string, string[], int&);
void read(int&);
void displayResult(int);
void Valid_the_Form(string&);                  //Validate the form result to view
void Calculation(string);                      // (Form) calculate average, standard
deviation, highest score, name
void display(int&);
string subject_toupper(string&);
bool exist_id(string&);
int getindex(int&, string&, int&);
void dlt(int&, int&);
void edit(int&, int&, string&);
bool subjectv(string&);
void writeFile(int, int, int);
bool validate_marks(string&);
bool validate_form(string&);

int main() {
```

```

string option;
do {
    system("cls");
    cout << "=====" << endl;
    cout << " STUDENT MANAGEMENT SYSTEM " << endl;
    cout << "=====" << endl << endl;
    cout << "Main Menu:" << endl;
    cout << "1. Insert" << endl;
    cout << "2. Filter/Search" << endl;
    cout << "3. Modify" << endl;
    cout << "4. View Results" << endl;
    cout << "5. Quit" << endl << endl;
    cout << "Option(1-5): ";
    cin >> option;
    cin.ignore();
    system("CLS");

    if (option == "1") {
        bool result;
        ifstream check;
        myfile.open("student.txt", ios::app); //make sure file exists
        cout << "Opening \"student.txt\"..." << endl;
        Sleep(500);
        cout << "Checking files content..." << endl;
        Sleep(500);
        myfile.close();

        do {
            int count, numForm = 0;
            read(count);
            system("CLS");
            bool replication = false;
            string name, id, form, pass, question;
            string marks[7] = { "" };
            cout << "Please enter student information and marks [Q to exit]." << endl;
            check_id(id); //obtain new student ID

            //check replication of id
            if (id.length() == 9) {
                for (int i = 0; i < count; i++) {
                    if (id == number[i].ID) {
                        replication = true; //Only true if replication occur
                        cout << "Error! Repeated ID detected!" << endl;
                    }
                }
            }

            if (replication == false) { //no replication
                cout << "Name" << left << setw(11) << internal << ": ";
            }
        }
    }
}

```



```

getline(cin, name); //Name to be record
get_form(form);    //form of the student
//BM BI BC Math Sci Sejarah Geo DISPLAY
student_mark(marks, id, name, form);
system("CLS");

/// write into FILE
myfile.open("student.txt");//ready to write into the file
cout << "Adding current student into the system\r" << endl;
Sleep(200);
system("CLS");
cout << "Adding current student into the system.\r" << endl;
Sleep(300);
system("CLS");
cout << "Adding current student into the system..\r" << endl;
Sleep(300);
system("CLS");
cout << "Adding current student into the system...\r" << endl;
Sleep(300);

if (number[0].name != "") { //check 1st variable empty or not
    //Write information into the file
    for (int i = 0; i < count; i++) {
        if (number[i].FORM < form[0])
            numForm++; //Increase numForm if students' form is smaller than
input form
        else if (number[i].FORM == form[0] &&
number[i].name.compare(name) == -1)
            numForm++; //Increase numForm if students' name is smaller than
input name
    }
    //numForm will be the index of the input student in struct

    for (int i = 0; i < count + 1; i++) {
        //When the index is less than numForm, write struct info into txt file
        if (i < numForm)
            writeFile(i, i, count + 1);

        //When the index is equal to numForm, write the input info into txt file
        if (i == numForm) {
            myfile << id << endl;
            myfile << name << endl;
            myfile << form << endl;
            for (int j = 0; j < 6; j++)
                myfile << marks[j] << " ";
            myfile << marks[6];
            if (i < count)
                myfile << endl;
        }
    }
}

```

```

        //When the index is greater than numForm, continue writing the struct
info into txt file
        if (i > numForm)
            writeFile(i - 1, i, count + 1);
        }
    }
    else { //if 1st variable empty
        myfile << id << endl;
        myfile << name << endl;
        myfile << form << endl;
        for (int j = 0; j < 7; j++) {
            if (j < 6)
                myfile << marks[j] << " ";
            else
                myfile << marks[j];
        }
        cout << "Student added successfully into the system!" << endl;
        myfile.close();
    }
    else {
        cout << "Please make sure there is no repeated student ID being inserted!" <<
endl;
    }
}
question = "DO YOU WANT TO CONTINUE ADDING RECORDS? [YES/NO]:
";
result = politely_ask(question);
} while (result);
}

else if (option == "2") {
    string choice;
    do {
        system("cls");
        cout << "=====" << endl;
        cout << " STUDENT MANAGEMENT SYSTEM " << endl;
        cout << "=====" << endl << endl;
        cout << "Please select your choice:\n1. Filter [FORM/SUBJECT]\n2. Search
[ID/NAME]\n3. Quit" << endl << endl;
        cout << "Choice: ";
        getline(cin, choice);
        system("cls");

        //if Filter is selected
        if (choice == "1") {
            bool again, error, gotSpace, display;
            string form, subject_chosen, subject[7];
            int k, size, index[7], count;

```

```

        do {
            again = false;
            cout <<
"=====
===== " << endl;
            cout << ">>    Please key in the information that you want to filter.    <<"
<< endl;
            cout << ">>    You can filter multiple forms at a same time.    <<" <<
endl;
            cout << ">>                Press Q to exit.                <<" << endl;
            cout <<
"=====
===== " << endl << endl;

        do {
            //Ask for input of form
            error = false;
            cout << "Form    : ";
            getline(cin, form);

            //Allow users to break the loop if Q is input
            if (toupper(form[0]) == 'Q')
                break;

            //Check if alphabets exist in the input or input is not 1,2,3
            for (int i = 0; i < form.length(); i++) {
                if (isalpha(form[i]) || form[i] > '3' || form[i] == '0')
                    error = true;
            }

            if (error)
                cout << "Error    : Please key in 1/2/3 only" << endl;

        } while (error); //If error exists, keep looping.

        //Allow users to exit Filter
        if (toupper(form[0]) == 'Q')
            continue;

        else {
            system("cls");
            cout <<
"=====
===== " << endl;
            cout << ">> Lists of subject: BM BI BC Math Sci Sejarah Geo <<" <<
endl;
            cout <<
"=====
===== " << endl;
            cout << "Form    : " << form << endl;

            do {

```

```

//Ask users to input the subject they want to filter
error = false;
gotSpace = false;
cout << "Subjects: ";
getline(cin, subject_chosen);

//Check if the input has space
for (int i = 0; i < subject_chosen.length(); i++) {
    if (isspace(subject_chosen[i]))
        gotSpace = true;
}

//Make the input with/without spaces into array
size = 0;
if (gotSpace == true)
    splitStringWithSpaceIntoArray(subject_chosen, subject, size);
else {
    subject[0] = subject_chosen;
    size = 1;
}

k = 0;
for (int i = 0; i < size; i++) {
    //To make all subjects' alphabets into uppercase letter
    subject[i] = string_toupper(subject[i]);

    //Store the matching index of each subject into index array
    for (int j = 0; j < 7; j++) {
        if (subject[i] == SUBJECT[j]) {
            index[k] = j;
            k++; //count the size of index
        }
    }
}

//Output error message if size of index != size of subject input
if (k != size) {
    cout << "Error : Please input subject correctly." << endl;
    error = true;
}
} while (error);

//To display the filter result
system("cls");
count = 0;
read(count); //Read the students' info into struct
cout <<
"===== " << endl;
cout << "Form : " << form << endl;
cout << "Subject: " << subject_chosen << endl;

```

```

        cout <<
"===== " << endl << endl;
        cout << "Featured list: " << endl;

        for (int i = 0; i < count; i++) {
            display = false;
            //Check which student has the same form as input form
            for (int j = 0; j < form.length(); j++) {
                if (number[i].FORM == form[j]) {
                    display = true; //If same form, display = true
                }
            }

            //Display the student's info
            if (display) {
                cout << "Student ID:" << number[i].ID << endl;
                cout << "Student Name: " << number[i].name << endl;
                cout << "Form: " << number[i].FORM << endl;
                for (int j = 0; j < size; j++) {
                    cout << displaySubject[index[j]] << ": ";
                    cout << number[i].mark[index[j]] << endl;
                }
                cout << endl;
            }
        }

        //Ask users if they want to input again
        string question = "DO YOU WANT TO CONTINUE TO FILTER (YES/NO):
";

        again = politely_ask(question);
        system("cls");
    } while (again);
}

else if (choice == "2") {
    //If search is selected
    bool again, display, gotSpace, store;
    int count, size, numDisplay, match, index[10], num, total;
    string keyword, keywordArray[10];
    do {
        numDisplay = 0;
        again = false;
        size = 0;
        read(count); //Read the students' info into struct

        //Get input from users
        cout <<
"=====
===== " << endl;

```

```

        cout << ">> You can key in multiple NAME and FULL ID separated by
space [Q to exit] <<" << endl;
        cout << ">> If you type the FULL NAME of students, you will only get
particular students <<" << endl;
        cout << ">> (IMPORTANT: THE INPUT IS CASE SENSITIVE!)
<<" << endl;
        cout <<
"=====
===== " << endl << endl;
        cout << "Keyword: ";
        getline(cin, keyword);

//If user input Q, allow them to exit
if (toupper(keyword[0]) == 'Q')
    continue;

else { //Check if it has space
    gotSpace = false;
    for (int i = 0; i < keyword.length(); i++) {
        if (isspace(keyword[i]))
            gotSpace = true;
    }

    //Make string into array
    if (gotSpace)
        splitStringWithSpaceIntoArray(keyword, keywordArray, size);
    else {
        keywordArray[0] = keyword;
        size = 1;
    }

    system("CLS");
    cout <<
"===== " << endl;
    cout << "Searched Keyword: " << keyword << endl;
    cout <<
"===== " << endl
<< endl;

    cout << "Search Results: " << endl;

    //To check if full name of students is input by users
    num = 0;
    total = 0;
    for (int i = 0; i < count; i++) {
        match = 0;
        store = false;
        display = false;
        //Check how many words match with name of each student
        for (int j = 0; j < size; j++) {

```

```

        if (number[i].name.find(keywordArray[j]) != -1)
            match++; //store the numbers of matching words

        //Check if keyword and student ID is the same
        if (number[i].ID == keywordArray[j]) {
            store = true;
        }
    }

    //If match full name or match ID, store the index into array
    if (match >= 2) {
        index[num] = i;
        num++;
        total++;
    }

    else if (store) {
        index[num] = i;
        num++;
    }
}

//Output students' info
if (total > 0) { //Display particular students with more than 2 matching
words of name
    for (int i = 0; i < num; i++) {
        displayResult(index[i]);
        numDisplay++;
    }
}

//If full name of student is not input
else {
    //Check if keyword match with students' info
    for (int i = 0; i < count; i++) {
        display = false;
        for (int j = 0; j < size; j++) {
            //Check if keyword and student ID is the same
            if (number[i].ID == keywordArray[j]) {
                display = true;
            }
            //Check if keyword and student name is the same
            if ((number[i].name.find(keywordArray[j]) != -1))
                display = true;
        }
    }

    //Display the results
    if (display) {
        displayResult(i);
        numDisplay++;
    }
}

```

```

        }
    }
}

//If no result is displayed, tell users
if (numDisplay == 0)
    cout << "Keyword is not found in the data." << endl << endl;
}

//Ask if they want to input again
string question = "DO YOU WANT TO CONTINUE TO SEARCH
(YES/NO): ";
again = politely_ask(question);
system("CLS");
} while (again);
}

else if (choice == "3")
    break;

} while (true);
}

else if (option == "3") {
{
    int count, index;
    const int noStu = 500;
    string input_id, pass, choice;
    bool boolean, result;

    do {
        system("CLS");
        cout << "===== " << endl;
        cout << " STUDENT MANAGEMENT SYSTEM " << endl;
        cout << "===== " << endl << endl;
        cout << "Please select your choice" << endl;
        cout << "1. Edit" << endl;
        cout << "2. Delete" << endl;
        cout << "3. Quit" << endl << endl;
        cout << "Choice: ";
        cin >> choice;
        cin.ignore();

        if (choice == "1") { //edit module
            read(count);
            do {
                system("CLS");

```



```

cout << "===== " << endl;
cout << " CURRENT LIST " << endl;
cout << "===== " << endl << endl;
display(count);
do {
    cout << ">> Please enter the student id that you want to edit [Q to exit]
<< " << endl;
    check_id(input_id); //ask user key in student id and check student id
format
    if (input_id.length() == 9) { //execute if user not input q

        boolean = exist_id(input_id);
        if (boolean == true) {
            getindex(count, input_id, index); //determine the index number of
the student id key in
            system("CLS");
            cout << "Current list" << endl;
            cout << "-----" << endl;
            display(count);
            edit(count, index, input_id);
        }
        else
            cout << "Error      : The student ID does not exist!" << endl <<
endl;
    }

    string question = "DO YOU WANT TO EDIT RECORD FOR OTHER
STUDENTS? (YES/NO): ";
    result = politely_ask(question); //to ask user input yes and no only

    } while (result);

    system("CLS");
    cout << "STATUS: Successfully updated!!" << endl << endl;
    cout << "===== " << endl;
    cout << " UPDATED LIST " << endl;
    cout << "===== " << endl << endl;
    display(count);

    string question = "DO YOU WANT TO CONTINUE TO EDIT?
(YES/NO):";
    result = politely_ask(question);
    } while (result);
}

else if (choice == "2") { //delete module
    do {
        system("CLS");

```

```

        read(count);
        cout << "===== " << endl;
        cout << " CURRENT LIST " << endl;
        cout << "===== " << endl << endl;
        display(count);
        cout << endl;
        cout << ">> Please enter the student id that you want to delete [Q to exit]
<<" << endl;

        check_id(input_id);
        if (input_id.length() == 9) { //execute if user not input q
            boolean = exist_id(input_id);
            if (boolean == true) {
                getindex(count, input_id, index);
                dlt(count, index);

                system("CLS");
                cout << "STATUS: Successfully deleted!!" << endl << endl;
                cout << "===== " << endl;
                cout << " UPDATED LIST " << endl;
                cout << "===== " << endl << endl;
                read(count);
                display(count);
            }
            else
                cout << "Error      : The student ID does not exist!" << endl;
        }
        string question = "DO YOU WANT TO CONTINUE TO DELETE?
(YES/NO):";
        result = politely_ask(question);
    } while (result);
}

    else if (choice == "3") {
        break;
    }

    } while (true);
}

else if (option == "4") {
    bool result;
    string form, pass;
    ifstream myReadfile;
    myReadfile.open("student.txt");
    cout << "Detecting relevant files..." << endl;
    Sleep(1000);
    cout << "Locating \"student.txt\"..." << endl;
    Sleep(1000);

```

```

do {

    if (myReadfile.fail()) {
        cout << "Error! \"student.txt\" not found! The file is now generating, please
wait..." << endl;
        ofstream setupfile;
        setupfile.open("student.txt");
        setupfile.close();
        Sleep(1000);
        cout << "The file \"student.txt\" is being generated. " << endl << "Please
proceed.";
    }
    else {
        myReadfile >> pass;
        myReadfile.close(); //need to reset position to the 1st char
        if (pass != "") {
            //determine input
            cout << "Please choose the form that you wish to view [All,1,2,3]: ";
            getline(cin, form);
            Valid_the_Form(form);
        }
        else {
            cout << "\"student.txt\" have no record!" << endl;
        }
    }
    string question = "DO YOU WANT TO CONTINUE TO VIEW OTHER RESULT
(YES/NO): ";
    result = politely_ask(question);
    system("CLS");
} while (result);
}

else if (option == "5")
    exit;

} while (option != "5");
return 0;
}

string string_toupper(string& a) { //uppercase the letter in string
    for (int i = 0; i < int(a.length()); i++) {
        char b = a[i];
        a[i] = toupper(b);
    }
    return a;
}

int check_id(string& a) { //until getting the correct id format, else repeat to prompt
    int i, counter;
    bool checking = true, blank_logic, alphabet;

```

```

while (checking == true) {
    cout << "Student ID" << left << setw(5) << internal << ": ";
    getline(cin, a); //get input id from user
    string_toupper(a);
    if (a.length() != 9 && a.length() != 1) {
        cout << "Error      : Please enter the ID in correct format (01AAA1234)!" << endl;
        cout << "          Or type [Q] to exit" << endl;
    }
    else if (a.length() == 9) {
        counter = 0;
        alphabet = true, blank_logic = true;
        for (i = 0; i < 9; i++) { //check if there is ' ' in input
            if (isspace(a[i])) {
                cout << "Error      : Space ' ' is an invalid input" << endl;
                blank_logic = false;
            }
            if (i >= 2 && i <= 4) { //check if alphabet is in correct place
                if (isalpha(a[i]) != 1)
                    alphabet = false;
            }
            if (i < 2 || i > 4) { //check if number is in correct place
                int b = a[i];
                for (int j = 0; j < 10; j++) {
                    if (b == (j + 48))
                        counter += 1; //counting how many numbers
                }
            }
        }
        // determine the loop to be continue or not
        if (counter == 6 && alphabet == true && blank_logic == true)
            checking = false;
        else
            cout << "Error      : Please enter the ID in correct format (01AAA1235)!" << endl;
    }
    else {
        if (a == "Q")
            checking = false;
    }
}
return 0;
}

```

```

bool politely_ask(string a) { //if a is not answered "YES" or "NO", it will repeat asking
    bool logic = true;
    string answer;
    while (logic == true) {
        cout << a;
        getline(cin, answer);
        string_toupper(answer);
        if (answer != "YES" && answer != "NO")

```

```

        cout << "Error: Enter only [YES] or [NO]" << endl;
    else if (answer == "YES" || answer == "NO")
        logic = false;
    }
    if (answer.compare("YES") == 0)
        return true;
    else
        return false;
}

string get_form(string& a) { //get only form1, 2, 3 from user
    bool logic = true;
    while (logic) {
        cout << "Form      : ";
        getline(cin, a);
        if (a.length() == 1) {
            char b = a[0];
            char c[3] = { '1','2','3' };
            if (b == c[0] || b == c[1] || b == c[2]) {
                logic = false;
            }
            else
                cout << "Error      : Only '1','2','3' is allowed to be input" << endl;
        }
        else
            cout << "Error      : Only '1','2','3' is allowed to be input" << endl;
    }
    return a;
}

string confirm_marks(string& a) {
    bool outcome = true;
    if (0 < a.length() && a.length() < 4) { //ensure it is 3 digit input
        for (int i = 0; i < int(a.length()); i++) { //ensure it is numeric number
            if (!isdigit(a[i])) {
                outcome = false;
            }
            if (outcome) {
                if (stoi(a) > 100 || stoi(a) < 0) { //ensure numeric value not larger than 100, or
smaller than 0
                    outcome = false;
                }
            }
        }
    }
    else {
        outcome = false;
    }
    if (outcome == false) {

```

```

    a = "";
}
return a;
}

void student_mark(string a[], string b, string c, string d) {
    bool result, result2;
    string blank_marks;
    do {
        system("CLS");
        result = 0; //reset the sentinel
        cout << "Please enter student information and marks." << endl;
        cout << "Student ID" << left << setw(5) << internal << ": " << b << endl;
        cout << "Name" << left << setw(11) << internal << ": " << c << endl;
        cout << "Form      : " << d << endl;
        cout << "_____Marks_____" << endl;
        cout << "_____ " << endl;
        for (int i = 0; i < 7; i++) { //display the subjects and marks
            cout << SUBJECT[i] << " mark";
            for (int j = 0; j < (14 - (SUBJECT[i].length())); j++) {
                cout << " ";
            }
            cout << ": " << a[i] << endl;
            if (a[i] == "") {
                result = true;
            }
        }
    }
    if (result) { // obtain marks value from user
        cout << "-----" << endl;
        result2 = true;
        int test;
        for (test = 0; result2 == true; test++) {
            if (a[test] == "") {
                cout << SUBJECT[test] << " mark : ";
                getline(cin, blank_marks);
                a[test] = confirm_marks(blank_marks);
                result2 = false;
            }
        }
        cout << "Scanning input\r";
        Sleep(100);
        cout << "Scanning input.\r";
        Sleep(100);
        cout << "Scanning input..\r";
        cout << "Scanning input..." << endl;
        if (a[test - 1] == "") {
            cout << "Error! Input must be an integer(within 0 to 100)!" << endl;
            system("PAUSE");
        }
        else
    }
}

```

```

        cout << "Input recorded!" << endl;
    }
    else {
        result = false;
    }
} while (result);
}

void splitStringWithSpaceIntoArray(string line, string strarray[], int& size) {
    int index[100];
    int j = 0, k = 0, pointerOne, pointerTwo;
    bool firstTime = true;

    for (int i = 0; i < line.length(); i++) {
        if (isspace(line[i])) {
            index[j] = i;
            j++;
        }

        if (firstTime && isspace(line[i])) {
            strarray[k] = line.substr(0, i);
            k++;
            firstTime = false;
        }

        else if (firstTime == false && isspace(line[i])) {
            pointerOne = index[j - 2] + 1;
            strarray[k] = line.substr(pointerOne, i - pointerOne);
            k++;
        }

        else if (i == (line.length() - 1)) {
            pointerTwo = index[j - 1] + 1;
            strarray[k] = line.substr(pointerTwo, (i + 1) - pointerTwo);
        }
    }
    size = j + 1;
}

void read(int& count) {
    ifstream infile("student.txt");
    if (infile.fail())
        cout << "Unable to read the file";

    else {
        count = 0;
        while (!infile.eof()) {
            getline(infile, number[count].ID);
            getline(infile, number[count].name);
            infile >> number[count].FORM;
        }
    }
}

```

```

        infile.ignore();
        for (int i = 0; i < 7; i++) {
            infile >> number[count].mark[i];
        }
        infile.ignore();
        count++;
    }
    infile.close();

}

}

void displayResult(int i) {
    cout << "Student ID:" << number[i].ID << endl;
    cout << "Student Name: " << number[i].name << endl;
    cout << "Form: " << number[i].FORM << endl;
    for (int j = 0; j < 7; j++) {
        cout << displaySubject[j] << ": ";
        cout << number[i].mark[j] << endl;
    }
    cout << endl;
}

void Calculation(string Form) {
    cout << "_____ " << endl << endl;
    cout << "Form " << Form << ":" << endl;
    for (int i = 0; i < 7; i++) { //indicates 7 subject to calculate
        cout << SUBJECT[i] << endl;
        ifstream Read;
        string Name_true, Name_hold, studentForm, Mark_hold;
        int count = 0, Mark_true = 0;
        double average, SD, sum = 0.0;
        Read.open("student.txt");
        while (!Read.eof()) {
            getline(Read, pass);
            getline(Read, Name_hold);
            getline(Read, studentForm);
            if (studentForm == Form) { //determine the form is same as desire
                for (int j = 0; j < i; j++) {
                    Read >> pass; //Skip unwanted score
                }
                Read >> Mark_hold; //hold wanted score
                sum += stoi(Mark_hold); //add to sum
                count += 1; //count number of value added
                if (Mark_true < stoi(Mark_hold)) {
                    Mark_true = stoi(Mark_hold);
                    Name_true = Name_hold; //New highest score
                }
                else if (Mark_true == stoi(Mark_hold)) {

```



```

        Name_true = Name_true + ", " + Name_hold; //if there is more than 1 student
being best
    }
    for (int k = (6 - i); k > 0; k--) { //recalibration to ID line
        Read >> pass;
    }
    Read.ignore();
}
else {
    getline(Read, pass); //recalibration to ID line
}
}
Read.close();

//Get average
average = static_cast<double>(sum) / count;

sum = 0; //reset sum to be used in Standard deviation

//Get Standard deviation
Read.open("student.txt");
while (!Read.eof()) {
    getline(Read, pass);
    getline(Read, Name_hold);
    getline(Read, studentForm);
    if (studentForm == Form) { //determine the form is same as desire
        for (int j = 0; j < i; j++) {
            Read >> pass; //Skip unwanted score
        }
        Read >> Mark_hold; //hold wanted score
        for (int k = (6 - i); k > 0; k--) { //recalibration to ID line
            Read >> pass;
        }
        Read.ignore();
        sum += pow((stoi(Mark_hold) - average), 2);
    }
    else {
        getline(Read, pass);
    }
}
Read.close();
SD = sqrt(static_cast<double>(sum) / count);
cout << fixed << setprecision(2);
cout << "Average          :" << average << endl;
cout << "Standard deviation  :" << SD << endl;
cout << fixed << setprecision(0);
cout << "Highest Score      :" << Mark_true << endl;
cout << "Highest score student :" << Name_true << endl << endl;
}
}

```

```

void Valid_the_Form(string& form) {
    if (form == "1" || form == "2" || form == "3") {
        Calculation(form);
    }
    else if (string_toupper(form) == "ALL") {
        Calculation("1");
        Calculation("2");
        Calculation("3");
    }
    else {
        cout << "Please choose the form that you wish to view [All,1,2,3]: ";
        getline(cin, form);
        Valid_the_Form(form);
    }
}

```

```

void display(int& count) {
    int i = 0;
    for (i = 0; i < count; i++) {
        cout << "Student ID: " << number[i].ID << endl;
        cout << "Student Name: " << number[i].name << endl;
        cout << "Form: " << number[i].FORM << endl;
        for (int j = 0; j < 7; j++) {
            cout << displaySubject[j] << ": ";
            cout << number[i].mark[j] << endl;
        }
        cout << endl;
    }
}

```

```

bool exist_id(string& input_id) {
    string pass;
    bool replication = false;
    ifstream infile("student.txt");
    getline(infile, pass); //pass=first line student id
    if (pass != "" && pass == input_id) {
        cout << "Valid" << endl;
        infile.close();
        replication = true;
    }
}

```

```

else if (pass != "") {
    while (!infile.eof()) {
        for (int i = 0; i < 4; i++) {
            getline(infile, pass);
        }
        if (pass == input_id) {
            cout << "Valid" << endl;
            infile.close();
        }
    }
}

```

```

        replication = true;
    }
}
else
    replication = false;
return(replication);
}

int getIndex(int& count, string& input_id, int& index) {
    index = 0;
    for (int i = 0; i < count; i++) {
        if (input_id.compare(number[i].ID) != 0)
            index += 1; //if input student id not same with the student id[i] in .txt file, indec +1
        else
            break;
    }
    return index;
}

void dlt(int& count, int& index) {
    myfile.open("student.txt");
    int newline = 0;
    for (int i = 0; i < count; i++) {
        if (i == index) {
            continue;
        }
        else {
            if (newline == 0) { //if the file is empty
                myfile << number[i].ID << endl;
                myfile << number[i].name << endl;
                myfile << number[i].FORM << endl;
                for (int j = 0; j < 6; j++) {
                    myfile << number[i].mark[j] << " ";
                }
                myfile << number[i].mark[6];
                newline = 1;
            }
            else { // the file is not empty
                myfile << endl;
                myfile << number[i].ID << endl;
                myfile << number[i].name << endl;
                myfile << number[i].FORM << endl;
                for (int j = 0; j < 6; j++) {
                    myfile << number[i].mark[j] << " ";
                }
                myfile << number[i].mark[6];
            }
        }
    }
}

```

```

    }
}
myfile.close();
}

```

```

void edit(int& count, int& index, string& input_id)
{
    string attribute;
    string subject;
    bool result;
    string marks;

    do {
        cout << endl;
        system("CLS");
        cout << "===== " << endl;
        cout << " STUDENT MANAGEMENT SYSTEM " << endl;
        cout << "===== " << endl << endl;
        cout << ">> Please select the attribute that you want to edit <<" << endl;
        cout << "1." << setw(4) << "Name" << endl;
        cout << "2." << setw(4) << "Form" << endl;
        cout << "3." << setw(4) << "Subjects" << endl;
        cout << "Option: ";
        getline(cin, attribute);

        int newline = 0, i, num;

        if (attribute == "1") { //edit name
            system("CLS");
            getindex(count, input_id, index);
            cout << "===== " << endl;
            cout << " CURRENT LIST " << endl;
            cout << "===== " << endl << endl;
            display(count);
            for (i = 0; i < count; i++) {
                if (i == index) {
                    cout << "Please update the name for student ID " << input_id << ": ";
                    getline(cin, number[index].name); //user input new name
                }
            }

            //Get the new position of student after the name is updated.
            num = 0;
            for (int i = 0; i < count; i++) {
                if (number[i].FORM < number[index].FORM)
                    num++; //Increase num if students' form is smaller than input form
                else if (number[i].FORM == number[index].FORM &&
                    number[i].name.compare(number[index].name) == -1)
                    num++; //Increase num if students' name is smaller than input name
            }

```

```

//If the index is last position or second last position, special case to write file
bool special = false;
if (index == count - 1 || index == count - 2)
    special = true;

myfile.open("student.txt");
//If the new position moves forward, write file
if (num < index) {
    for (i = 0; i < count + 1; i++) {
        if (i == index + 1)
            continue;

        if (i < num)
            writeFile(i, i, count);

        else if (i == num)
            writeFile(index, i, count);

        else if (i > num)
            if (special)
                writeFile(i - 1, i, count);
            else
                writeFile(i - 1, i, count + 1);
    }
}

//If the position does not change, write file
else if (index == num) {
    for (i = 0; i < count; i++)
        writeFile(i, i, count);
}

//If the position moves backward, write file
else if (num > index) {
    num++;
    for (i = 0; i < count + 1; i++) {
        if (i == index)
            continue;

        if (i < num)
            writeFile(i, i, count + 1);

        else if (i == num)
            writeFile(index, i, count + 1);

        else if (i > num)
            writeFile(i - 1, i, count + 1);
    }
}

```

```

myfile.close();
read(count);
}

else if (attribute == "2") { //edit form
    system("CLS");
    getindex(count, input_id, index);
    cout << "=====" << endl;
    cout << " CURRENT LIST " << endl;
    cout << "=====" << endl << endl;
    display(count);
    bool valid;
    string form;
    for (i = 0; i < count; i++) {
        if (i == index) {
            do {
                cout << "Please update the form (1/2/3) for student ID " << input_id << ": ";
                cin >> form;
                cin.ignore();
                valid = validate_form(form); //form validation only allow '1','2','3'
            } while (!valid); //If error exists, keep looping.

            number[index].FORM = form[0];
        }
    }

    //Get the new position of updated student
    num = 0;
    for (int i = 0; i < count; i++) {
        if (number[i].FORM < number[index].FORM)
            num++; //Increase numForm if students' form is smaller than input form
        else if (number[i].FORM == number[index].FORM &&
number[i].name.compare(number[index].name) == -1)
            num++; //Increase numForm if students' name is smaller than input name
    }

    //If the index is last position or second last position, special case to write file
    bool special = false;
    if (index == count - 1 || index == count - 2)
        special = true;

    myfile.open("student.txt");
    //If position moves backward or equal to old position, write file
    if (index <= num) {
        num++;
        for (i = 0; i < count + 1; i++) {
            if (i == index)
                continue;

```

```

        if (i < num)
            writeFile(i, i, count + 1);

        else if (i == num)
            writeFile(index, i, count + 1);

        else if (i > num)
            writeFile(i - 1, i, count + 1);

    }
}

//If position moves forward, write file
else if (index > num) {
    for (i = 0; i < count + 1; i++) {
        if (i == index + 1)
            continue;

        if (i < num)
            writeFile(i, i, count + 1);

        else if (i == num)
            writeFile(index, i, count + 1);

        else if (i > num)
            if (special)
                writeFile(i - 1, i, count);
            else
                writeFile(i - 1, i, count + 1);

    }
}
myfile.close();
read(count);
}

else if (attribute == "3") { //edit subject's mark
    system("CLS");
    getindex(count, input_id, index);

    cout << count << " " << index << endl;

    cout << "===== " << endl;
    cout << " CURRENT LIST " << endl;
    cout << "===== " << endl << endl;
    display(count);
    string line;
    bool valid, subvalid;
    for (i = 0; i < count; i++) {

```

```

        if (i == index) {
            do {
                cout << "Please enter the subject that you want to change the mark
(BM/BI/BC/Math/Sci/Sejarah/Geo):";
                cin >> subject;
                cin.ignore();
                subvalid = subjectv(subject); //subject validation
            } while (subvalid != true); //if false, loop again

            do {
                cout << "Please update the mark for " << subject << " in range 0 - 100: ";
                cin >> marks;
                cin.ignore();
                valid = validate_marks(marks); //mark validation
            } while (valid != true);
            int x = stoi(marks); //assing string marks into integer type
            for (int i = 0; i < 7; i++) {
                if (SUBJECT[i] == subject)
                    number[index].mark[i] = x;
            }
        }
    }

    //Write all updated student info into text
    myfile.open("student.txt");
    for (int i = 0; i < count; i++)
        writeFile(i, i, count);

    myfile.close();
    read(count);
}
else
    cout << "Invalid input" << endl;

    string question = "DO YOU STILL HAVE ANYTHING TO EDIT FOR THIS
STUDENTS? (YES/NO): ";
    result = politely_ask(question);

} while (result);
}

bool subjectv(string& subject) {
    bool sub_in_range;
    if (subject_toupper(subject) == "BM" || subject_toupper(subject) == "BI" ||
subject_toupper(subject) == "BC"
        || subject_toupper(subject) == "MATH" || subject_toupper(subject) == "SCI" ||
subject_toupper(subject) == "SEJARAH"
        || subject_toupper(subject) == "GEO")
        sub_in_range = true;
}

```



```

    else
        sub_in_range = false;

    return(sub_in_range);
}

string subject_toupper(string& subject) { //uppercase the character of subject
    for (int i = 0; i<int(subject.length()); i++) {
        char c = subject[i];
        subject[i] = toupper(c);
    }
    return subject;
}

void writeFile(int index, int i, int count) {
    myfile << number[index].ID << endl;
    myfile << number[index].name << endl;
    myfile << number[index].FORM << endl;
    for (int j = 0; j < 6; j++) {
        myfile << number[index].mark[j] << " ";
    }
    myfile << number[index].mark[6];
    if (i < count - 1)
        myfile << endl;
}

bool validate_marks(string& a) {
    bool outcome = true;
    if (0 < a.length() && a.length() < 4) { //ensure it is 3 digit input
        for (int i = 0; i < int(a.length()); i++) { //ensure it is numeric number
            if (!isdigit(a[i])) {
                outcome = false;
            }
            if (outcome) {
                if (stoi(a) > 100 || stoi(a) < 0) { //ensure numeric value not larger than 100, or
smaller than 0
                    outcome = false;
                }
            }
        }
    }
    else {
        outcome = false;
    }
    if (outcome == false) {
        a = "";
    }
    return (outcome);
}

```

```

bool validate_form(string& a) { //get only form1, 2, 3 from user
    bool logic = true;
    if (a.length() == 1) {
        char b = a[0];
        char c[3] = { '1','2','3' };
        if (b == c[0] || b == c[1] || b == c[2]) {
            logic = false;
        }
        else
            cout << "Error      : Only '1','2','3' is allowed to be input" << endl;
    }
    else
        cout << "Error      : Only '1','2','3' is allowed to be input" << endl;
    return (logic);
}

```