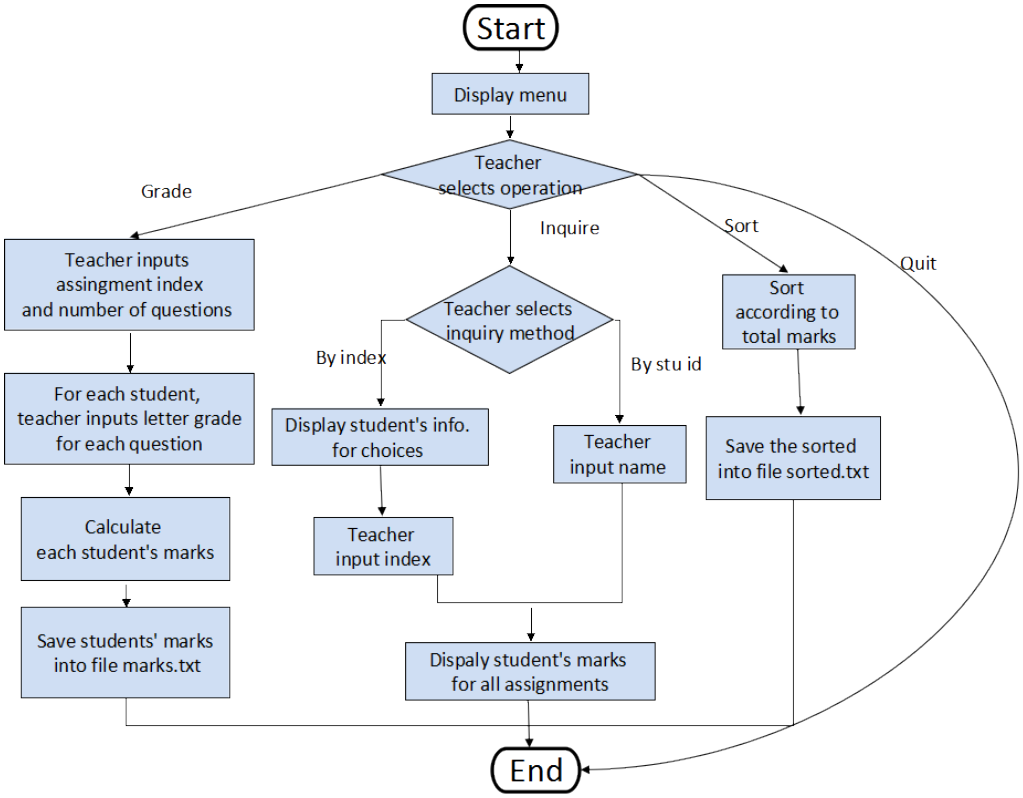
Project - 2017

In this project, we are going to develop a system to help teacher grade assignments. The expected system flow is like follows.



1. The system will display the main menu at the beginning. The menu includes the following four items:

```
1. Grade
2. Inquire
3. Sort
4. Quit
```

2. Teacher will choose one operation from the menu.

a) If teacher chooses “Grade”, then he will start grading students’ assignment. Information should be prompted for teacher to input the index of this assignment (for example: 5 to indicate “assignment 5”) and number of questions in this assignment.

Then for each student, the teacher will input the grade for each question. After teacher finishes the inputs for all the questions in this assignment for a student, system will calculate the final marks of this assignment for this student. After one student is finished, the teacher will go to the next student.

After all the students’ marks have been calculated, the results will be saved in a file ***`marks.txt`***.

b) If teacher chooses “Inquire”, a sub-menu is displayed. The sub-menu includes the following items:

```
1. By index
2. By student id
3. Quit
```

i. If the teacher chooses “By index”, a list of students’ id and name with index is displayed for

teacher to choose from.

ii. If the teacher chooses “By student id”, he is then required to input student id.

iii. If the teacher chooses “Quit”, the system will go back to the main menu.

After a choice is made, the system will display the marks for the selected student.

c) If the teacher chooses “Sort”, the system will sort the students’ list according to the total marks for all the assignments. The sorted results are saved in a file *sorted.txt*. At the same time, the sorted results should also be outputted to the screen.

d) If the teacher chooses “Quit”, then the system finishes work and quit.

Assumptions:

1. A file named *students.txt* already exists. The format of the information in this file is as follows:

```
Name   ID
Tony   1234
Andrew 1235
Amy    2323
……
```

2. The format in the file *marks.txt* is as follows:

```
Name   ID     Assignment1 Assignment2 (if any)  ......
Tony   1234   3.5         ...
Andrew 1235   3.3         ...
Amy    2323   4           …
……..
```

3. The file *sorted.txt* is as follows:

```
Name   ID     Total
Andrew 1235   15
Amy    2323   12
Tony   1234   11.3        ...
……..                     ...
```

4. There are no more than 20 students in total.
5. The student id has four digits and the first digit is not 0.
6. The number of total assignments does not exceed 5.
7. The number of questions in each assignment does not exceed 10.
8. Each question takes same percentage in one assignment. For example, if there are 2 questions, each will take 50%. If 4, then 25% for each.
9. Letter grades: A(4 pts), B(3 pts), C(2 pts), D(1 pts), F(0 pts).
10. Marks for an assignment are calculated according to points for each grade. For example, if there are two questions with grades A and C respectively, the marks for this assignment will be 4 * 50% + 2 * 50% = 3.
11. The users always give the valid inputs.

Requirements:

1. The program should be well-structured with main function and some sub-functions. They are stored

in several files so that team members can work independently.

2. The system should be user-friendly. Users (teacher) should understand what should be inputted and they can quit back to the main menu when they do not want to go on the current operation.
3. In the "Sort" part, the system is required to save the sorted results in the file `sorted.txt` first. Then the system reads the contents from the file and puts them in a linked list, and then print out the data in the linked list to screen.
4. No marks will be given to a project which cannot be run due to grammar errors.

Submission:

1. All *hpp*, *cpp* files.
2. A *txt* file includes each member's work

Deadline: 11:55 pm 27 December 2017