Assignment

Value **100%** of Coursework Resit

Individual work

Background:

An elevator is a vertical transportation device typically found in buildings with multiple floors, designed to carry passengers or goods between different levels. The FCFS (First-Come, First-Served) principle is a very crude method for elevator scheduling.

The FCFS (First-Come, First-Served) principle for elevator scheduling operates on a simple rule:

- The elevator serves requests in the order they are received.
- When a passenger requests an elevator, it is dispatched to their floor.
- If multiple requests are made while the elevator is in transit, they are queued up and serviced in the order they were received.
- This method does not consider factors such as destination floors, traffic patterns, or passenger distribution, making it a straightforward but often inefficient scheduling approach.

General requirements

This coursework is to write a program to simulate the FCFS scheduling method based on the start codes provided.

When you have completed all tasks, you should be able to generate two separate executable projects, otherwise you will lose all marks.

The coursework submitted should be composed of three files：

- *a report as a Microsoft Word* document containing a description and explanation of the encoding work.
  - filename format: *Student ID_CHC5028_CW_Resit_Report.docx*

- a *.zip file* containing the first project:
  - all the project's source files (*.cpp, .h, and CMakeLists.txt*), including those provided.
  - filename format: *Student ID_CHC5028_ CW_Resit_Project1.zip*

- a *.zip file* containing the second project:
  - all the project's source files (*.cpp, .h, and CMakeLists.txt*), including those provided.
  - filename format: *Student ID_CHC5028_ CW_Resit_Project2.zip*

If you do not submit the files according to the requirements, you may lose 10 marks for the coursework.

Requirements

Task 1

Please create a project in your IDE, containing all the provided start codes. You may modify the CMakeLists.txt file so that the project can be executable upon completion.

5marks

Task 2

The class ***Passenger*** is a class used to represent passengers who take the elevator. The attribute 'name' represents the passenger's name, the attribute 'from' represents the initial floor where the passenger is located, and the attribute 'to' represents the floor the passenger wants to reach by taking the elevator.

You must write the corresponding code in the Passenger. cpp file based on the declarations provided in the Passenger.h file.

5 marks

Explain the encoding work in the report rationally and explicitly.

5 marks

Task 3

The class ***Node*** is used to generate nodes that make up a linked list. Each node contains a pointer to a passenger object and a pointer to the next node.

You must write the corresponding code in the Node.cpp file based on the declarations provided in the Node.h file.

5 marks

Explain the encoding work in the report rationally and explicitly.

5 marks

Task 4

The class ***ListOfPassenger*** is used to generate a singly linked list, where each node of the list is an object of the class ***Node***. The attribute ***head*** represents a pointer to the head node of the list. You must write the corresponding code in the ListOfPassenger.cpp file based on the declarations provided in the ListOfPassenger.h file.

5 marks

Explain the encoding work in the report rationally and explicitly.

5 marks

Task 5

The Elevator.cpp file contains the main program of the project, in which there are already provided examples for utilizing class ***Passenger*** and class ***Node*** to form one list.

You must create more passengers according to the following information, the order in which these passengers are created represents the order in which they request to take the elevator.:

passengers	name	from	to
	A	4	7
	B	8	11
	C	7	8
	D	9	22
	E	10	7
	F	6	3
	Your name +	the last two digits of your student ID	the first two digits of your student ID

5 marks

Explain the encoding work in the report rationally and explicitly.

5 marks

Task 6

You must create the corresponding nodes in the main program, then add the nodes created into the object of the class ***ListOfPassenger*** provided.

3 marks

Explain the encoding work in the report rationally and explicitly.

2 marks

Task 7

There is one function ***displayRequest()*** in the Elevator.cpp file, this function is used to display all information of the passengers that are contained in the node of the list. The information displayed includes the request order which represents

the order in which the boarding request is, the passenger's name, the start floor, the destination floor, and the direction that the passenger requested.

You must write the codes to complete the ***displayRequest()*** function and display the information in the format shown following:

```
request order     passenger      from     to    Direct
   1                  A           4       7     up
   2                  B           8       10     up
```

5 marks

Explain the encoding work in the report rationally and explicitly.

5 marks

Task 8

There is one function ***FCFS()*** in the Elevator.cpp file, this function is used to apply the FCFS principle to operate the elevator and display the information about the moving process of the elevator.

You must write the codes to complete the ***FCFS()*** function and display the information in the format shown following:

```
passenger    from     end     moved
  Empty       1       4        3
    A         4       7        3
  Empty       7       8        1
    B         8       11       3
```

The column ***passenger*** represents who is taking the elevator. If there is no passenger in the elevator, display "Empty". The column ***moved*** represents how many floors passed during the process.

```
request order     passenger      from     to    Direct
   1                  A           4       7     up
   2                  B           8       10     up
```

5 marks

Explain the encoding work in the report rationally and explicitly.

10 marks

Task 9

You must add a compile guard to each .h file to avoid issues such as excessive compilation.

5 marks

Now project 1 should be completed after finishing tasks 1 to 9, please execute the project and show the result as evidence in your report.

The project 1 should be submitted following the instructions mentioned in the General Requirements.

Task 10

You may have noticed that the ***Node*** class provided in the start codes is quite redundant. Please merge the ***Node*** class with the ***Passenger*** class to create a new class called ***PassengerNode*** to replace these two classes. Modify the rest of the program so that it can become a new executable project and still meet the requirements mentioned in tasks 5 to 9.

The new project is project 2 that should be submitted following the instructions mentioned in the General Requirements too.

10 marks

Explain the encoding work in the report rationally and explicitly.

10 marks