

## 1. Porject

You need to modify your code in **Course 6** with the following requirements:

- 1) Add a new button "Start Simulation", a new label "Finish!!", a new label "Score", and a new lineEdit for "score".
- 2) Add a new timer, which is for simulation, and a new slot "update" in the class "Building", and add this line into Building's constructor:  
`connect(timer,SIGNAL(timeout()),this, SLOT(update()));`
- 3) After clicking the button "Start Simulation", set the label "Finish!!" invisible and call this function: `building.startSimulation`. In "startSimulation, you should start the timer in "Building" with 1000ms.
- 4) Add a new class "Scheduler" with the following definition:

```
class Scheduler
{
public:
    Scheduler();
    int getNowFloor();
private:
    vector<int> nowFloor; // temp
    int index;           // temp
};
```

a) **Scheduler::Scheduler()**

Initialize the vector "nowFloor" with numbers {1, 3, 4, 6, 9, 10, 8, 7, 5, 2, 0}, each number means the floor where now the elevator stops at, and 0 means finishing the schedule. Set count to zero.

b) **int Scheduler::getNowFloor ()**

Return the number in the vector "nowFloor" one by one.

- 5) In class "Data", add a data member "score" with data type int, and "nowfloor" with data type int, which means your score and the floor where now the elevator stops at, correspondingly.
- 6) In class "Judge", add a new data member "score" with data type int. If your answer is correct, add 100 to "score".
- 7) In Building's function "run": get score from "judge" and assign to "score" in "Data".
- 8) In Building's slot "update":
  - a) You should get "nowfloor" from "scheduler".

- b) If “nowfloor” is not zero, you should call the function “run” to solve the corresponding problem in Floor.
- c) If “nowfloor equals zero, you should stop the timer in “Building”.
- d) At the end of this slot, you should add this line:

`emit this->updateGUI();`

This line means when you finish this function, you should send a signal to update the information in MainWindow’s.

- 9) You should add this line into MainWindow’s constructor:

`connect(&building,SIGNAL(updateGUI()),this, SLOT(slot_update_data()));`

- 10) In MainWindow’s slot “slot\_update\_data”:

- a) You should get “data” from “building”.
- b) If “nowfloor” in data is not zero, update the information in MainWindow’s, such as testdata, submitdata, spend time, correct or not, score, and nowfloor.
- c) If “nowfloor” equals zero, set the label “Finish!!” visible, which means finish the simulation.

## Sample Output

The screenshot shows a Qt-style window titled "MainWindow". On the left, there is a button labeled "Start Simulation" and a label "Finish!!" which is currently visible. On the right, there are several input fields and buttons:

- Testdata:** A text box containing the string "6 18395.83 13773.86 13423.69 20854.89 42340.85 42140.98 19342.07".
- Submitdata:** A text box containing the value "43891".
- Spend time:** A text box containing the value "6098900".
- Correct or not:** A text box containing the value "0".
- Score:** A text box containing the value "900".
- Nowfloor:** A dropdown menu currently showing the value "2".
- Run:** A button next to the Nowfloor dropdown.
- People Information:** A section header above two more input fields:
  - People Num:** An empty text box.
  - Destination:** An empty text box.

## 2. The Easy City 1 (MyDatabase) (Easy)

Please implement the C++ code using SQL command in QT, and follow the steps to output the final result: (You have to establish a connection with MySQL Server first.)

- 1) Drop the schema "CITYDATABASE" if it exists in MySQL server.
- 2) Create a new schema "CITYDATABASE" in MySQL server.
- 3) Set "CITYDATABASE" as default database.
- 4) Drop the table "CITYTABLE" if it exists in "CITYDATABASE".
- 5) Create a new table "CITYTABLE" described as follows:

Field	Type
ID	Number (Primary Key)
COUNTRY	VARCHAR (50)
CITY	VARCHAR (60)
LAT	NUMBER
LON	NUMBER

Note: LAT is the latitude and LON is the longitude.

- 6) Load data from "city.csv" into table "CITYTABLE". The data format in "city.csv" is as following example:

ID,Country,City,Latitude,Longitude

100,"Taipei","Zhongzheng",121.5198839,25.03240487

130,"Taipei","Datong",121.5130417,25.06342433

104,"Taipei","Zhongshan",121.5381597,25.06969917

150,"Taipei","Songshan",121.5575876,25.05999101

106,"Taipei","Da'an",121.5434446,25.02677012

108,"Taipei","Wanhua",121.4979858,25.02858990

110,"Taipei","Xinyi",121.5716697,25.03062083

111,"Taipei","Shilin",121.5508473,25.12546704

112,"Taipei","Beitou",121.5177992,25.14806820

114,"Taipei","Neihu",121.5923828,25.08370623

115,"Taipei","Nangang",121.6097573,25.03600934

116,"Taipei","Wenshan",121.5736082,24.98857934

200,"Keelung","Ren'ai",121.7434205,25.11945421

201,"Keelung","Xinyi",121.772646,25.125765790

202,"Keelung","Zhongzheng",121.7783549,25.14365754

203,"Keelung","Zhongshan",121.7308913,25.14986365

204,"Keelung","Anle",121.7078325,25.14139521

205,"Keelung","Nuannuan",121.7447344,25.08097003  
206,"Keelung","Qidu",121.683628,25.109620280  
556,"Nantou","Xinyi",121.0212867,23.6554647

Please use only one query statement to solve each requirement.

- 7) Query the difference (**dcc**) between the total number of city and the number of distinct city with the specified data requirements:
  - a) The names of city start as specified characters (**ch**) and end with no vowels (i.e., a, e, i, o, and u). The specified starting character can be one of multiple characters in a specified range.
  - b) The latitude is greater than **la** and the longitude is smaller than **lo**.
- 8) Update the values of latitudes to **p** times and the values of longitude to **q** times if the id number is even/odd (**ev/od**).
- 9) Query the sum of the values in latitude (**sla**), the sum of the values in longitude (**slo**), and the difference between maximum length of city name and minimum length of city name (**dmm**). The query data limitations are as follows:
  - a) The first **m even/odd** ID numbers with ascending/descending (**asc/des**) length of city name and reverse alphabetical order only.
  - b) Both summations of latitude and longitude must be rounded to a scale of **n (n<=7)** decimal places.

### Input Format

**ch la lo p q ev/od m even/odd asc/des n**

### Output Format

**dcc sla slo dmm**

### Sample Input

nz 121.5 25.3 2 5 od 5 even des 3

### Sample Output

2 1823.360 374.701 6

### 3. Manhattan Distance (MyDatabase) (Medium)

Please implement the C++ code using SQL command in QT, and follow the steps to output the final result: (You have to establish a connection with MySQL Server first.)

- 1) Drop the schema "CITYDATABASE" if it exists in MySQL server.
- 2) Create a new schema "CITYDATABASE" in MySQL server.
- 3) Set "CITYDATABASE" as default database.
- 4) Drop the table "CITYTABLE" if it exists in "CITYDATABASE".
- 5) Create a new table "CITYTABLE" described as follows:

Field	Type
ID	Number (Primary Key)
COUNTRY	VARCHAR (50)
CITY	VARCHAR (60)
LAT	NUMBER
LON	NUMBER

Note: LAT is the latitude and LON is the longitude.

- 6) Load data from "city.csv" into table "CITYTABLE". The data format in "city.csv" is as follows:

ID,Country,City,Latitude,Longitude

100,"Taipei","Zhongzheng",121.5198839,25.03240487

130,"Taipei","Datong",121.5130417,25.06342433

104,"Taipei","Zhongshan",121.5381597,25.06969917

150,"Taipei","Songshan",121.5575876,25.05999101

106,"Taipei","Da'an",121.5434446,25.02677012

108,"Taipei","Wanhua",121.4979858,25.02858990

110,"Taipei","Xinyi",121.5716697,25.03062083

111,"Taipei","Shilin",121.5508473,25.12546704

112,"Taipei","Beitou",121.5177992,25.14806820

114,"Taipei","Neihu",121.5923828,25.08370623

115,"Taipei","Nangang",121.6097573,25.03600934

116,"Taipei","Wenshan",121.5736082,24.98857934

200,"Keelung","Ren'ai",121.7434205,25.11945421

201,"Keelung","Xinyi",121.772646,25.125765790

202,"Keelung","Zhongzheng",121.7783549,25.14365754

203,"Keelung","Zhongshan",121.7308913,25.14986365

204,"Keelung","Anle",121.7078325,25.14139521

205,"Keelung","Nuannuan",121.7447344,25.08097003  
206,"Keelung","Qidu",121.683628,25.109620280  
556,"Nantou","Xinyi",121.0212867,23.6554647

Please use only one query statement to solve the requirement.

7) If  $x_1$  and  $x_2$  are the smallest and the largest longitudes for the  $s^{\text{st}}$  and  $t^{\text{th}}$  largest latitudes which is less than  $v_1$ ,  $y_1$  and  $y_2$  are the smallest and the largest latitudes for the  $s^{\text{st}}$  and  $t^{\text{th}}$  smallest longitude which is larger than  $v_2$ , please query the Manhattan Distance ( $md$ ) between points  $(x_1, y_1)$  and  $(x_2, y_2)$ . The answer should be rounded to a scale of 4 decimal places. The formula of Manhattan Distance is  $|x_1 - x_2| + |y_1 - y_2|$ . You can use `ABS()` function in MySQL.

### Input Format

$s$   $t$   $v_1$   $v_2$

### Output Format

$md$

### Sample Input

4 9 121.7 25

### Sample Output

0.3686

#### 4. The Median (MyDatabase) (Hard)

Please implement the C++ code using SQL command in QT, and follow the steps to output the final result: (You have to establish a connection with MySQL Server first.)

- 1) Drop the schema "CITYDATABASE" if it exists in MySQL server.
- 2) Create a new schema "CITYDATABASE" in MySQL server.
- 3) Set "CITYDATABASE" as default database.
- 4) Drop the table "CITYTABLE" if it exists in "CITYDATABASE".
- 5) Create a new table "CITYTABLE" described as follows:

Field	Type
ID	Number (Primary Key)
COUNTRY	VARCHAR (50)
CITY	VARCHAR (60)
LAT	NUMBER
LON	NUMBER

Note: LAT is the latitude and LON is the longitude.

- 6) Load data from "city.csv" into table "CITYTABLE". The data format in "city.csv" is as follows:

ID,Country,City,Latitude,Longitude

100,"Taipei","Zhongzheng",121.5198839,25.03240487

130,"Taipei","Datong",121.5130417,25.06342433

104,"Taipei","Zhongshan",121.5381597,25.06969917

150,"Taipei","Songshan",121.5575876,25.05999101

106,"Taipei","Da'an",121.5434446,25.02677012

108,"Taipei","Wanhua",121.4979858,25.02858990

110,"Taipei","Xinyi",121.5716697,25.03062083

111,"Taipei","Shilin",121.5508473,25.12546704

112,"Taipei","Beitou",121.5177992,25.14806820

114,"Taipei","Neihu",121.5923828,25.08370623

115,"Taipei","Nangang",121.6097573,25.03600934

116,"Taipei","Wenshan",121.5736082,24.98857934

200,"Keelung","Ren'ai",121.7434205,25.11945421

201,"Keelung","Xinyi",121.772646,25.125765790

202,"Keelung","Zhongzheng",121.7783549,25.14365754

203,"Keelung","Zhongshan",121.7308913,25.14986365

204,"Keelung","Anle",121.7078325,25.14139521

205,"Keelung","Nuannuan",121.7447344,25.08097003  
206,"Keelung","Qidu",121.683628,25.109620280  
556,"Nantou","Xinyi",121.0212867,23.6554647

Please use only one query statement to solve the requirement.

7) The median is the value separating the larger half from the smaller half of a data sample. Please query the median (**med**) of the range of the **a<sup>th</sup>** to the **b<sup>th</sup>** latitudes/longitudes (**lat/lon**) with the following dataset: the last number of id of the corresponding cities is **m** in ascending/descending (**asc/des**) order. The answer should be rounded to a scale of 4 decimal places.

### Input Format

**a b lat/lon m asc/des**

### Output Format

**med**

### Sample Input

1 4 lon 6 des

### Sample Output

25.0077