1. New Judge

You need to modify your code in Course 7 with the following requirements:

#No old "Judge" class today.

1) Add a new class "JudgeWindow" to replace "Judge" class
2) The JudgeWIndow is create as "QT 設計師表單類別"
3) In "JudgeWindow" constructer:
   I. Drop and create database FINAL
      Create table to store testdata with follow format

Field	Type	Description
ID	CHAR(8)	(N floor)-(Mth testdata) EX:: 01-00001
Floor	Int	
Question	TEXT	
Answer	TEXT	

   II. Create table to store Initial condition just like Course6

Field	Type	Description
Id	char(8)	Id is primary key and the format of id is (nth initial condition)-(nth floor). For example, 00001-10 means the tenth floor of the first initial condition.
Nowfloor	int	
Destination	int	
Number	int	This "Number" means there are N people in corresponding floor.

   III. Load two csv file into two tables.
4) Finish follow function in **Judgewindow.h**

```
string getData(int floor,int b);//input 0-26
bool submitData(string ans);
void setSeed(int seed){srand(seed);}
qint64 getSpendTime(){return costtime;}
int getConditionNum();//return 1-300
int getDistance(){return distance;}
void scheduleEnd();
```

a) `getData()` Just like old "Judge" but get data from SQL not from TXT file, **b=0 for leave elevator**, **b=1 to enter elevator**, and calculate move distance.
b) `setSeed()` set the random seed to "Judge", today pass 0 to judge.

c) `getConditionNum()` get initial condition(people list) , and return which condition to get

d) `getDistance()` return how far the elevator move(Demo in Problem 2)

e) `scheduleEnd()` after schedule end, output the **floor**,**cost time, num of correct answer, num of question** for each problem to "time.txt".(Demo in Problem 2)

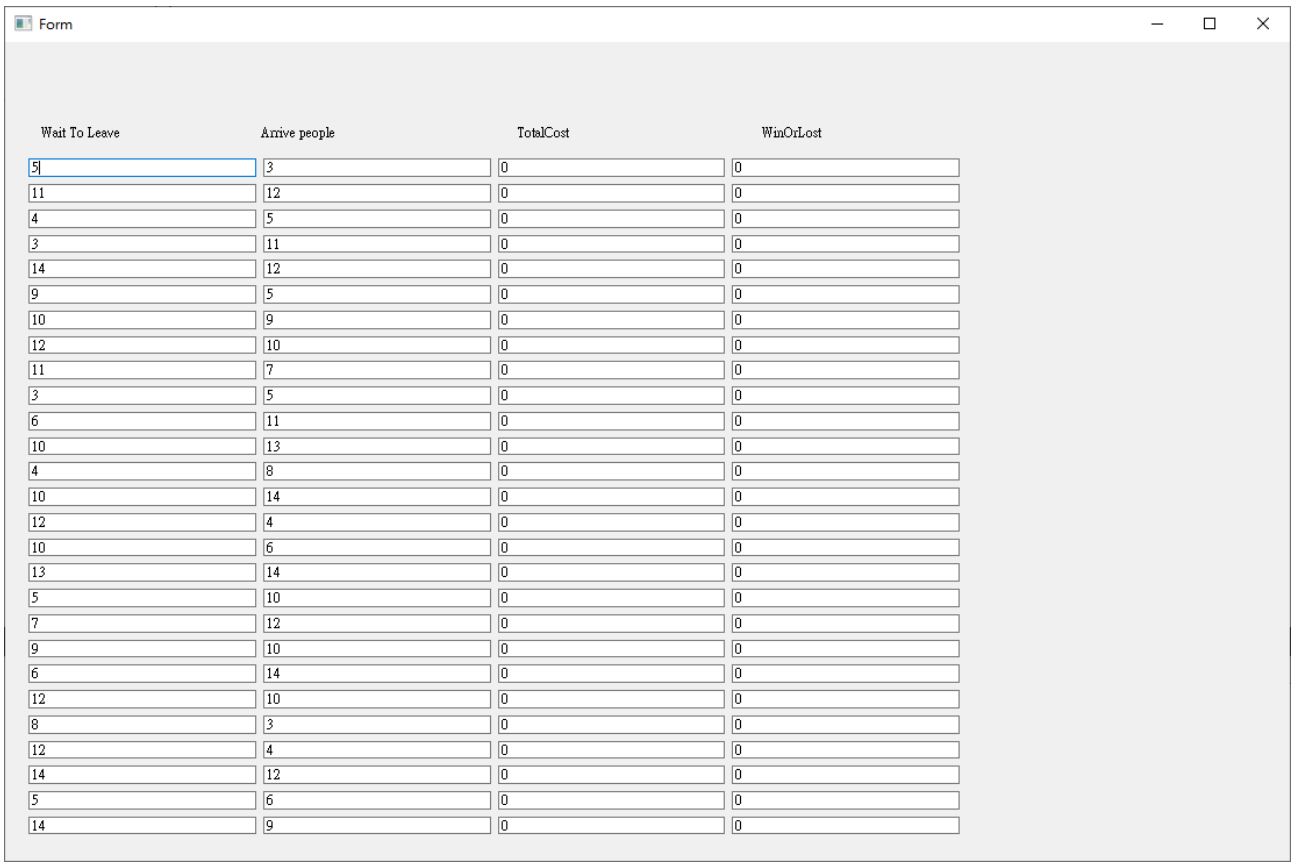
```
0,123456,7,7
1,23456789,8,8
……
```

5) In "JUDGEWINDOW" widget, add 27 linedit to display people wait to leave,27 linedit to display how many people haven't arrived yet,27 linedit to display total cost time and (27 linedit to display win or not)(We don't do these today)

In building constructer

```
judge.show();
judge.setSeed(0);
int n=judge.getConditionNum();
//get People data according variable n
```

Sample Output



2. Scheduler

Finish the “Scheduler” and display the result on JUDGEWINDOW GUI

1) In “Building ” constructer get which initial condition to run from Judge
2) Also load initial condition in building constructer(Course 6), and then finish the scheduler path
3) In building update function, run building.run() N times if there are N people want to leave or enter elevator
4) Modify some code in building

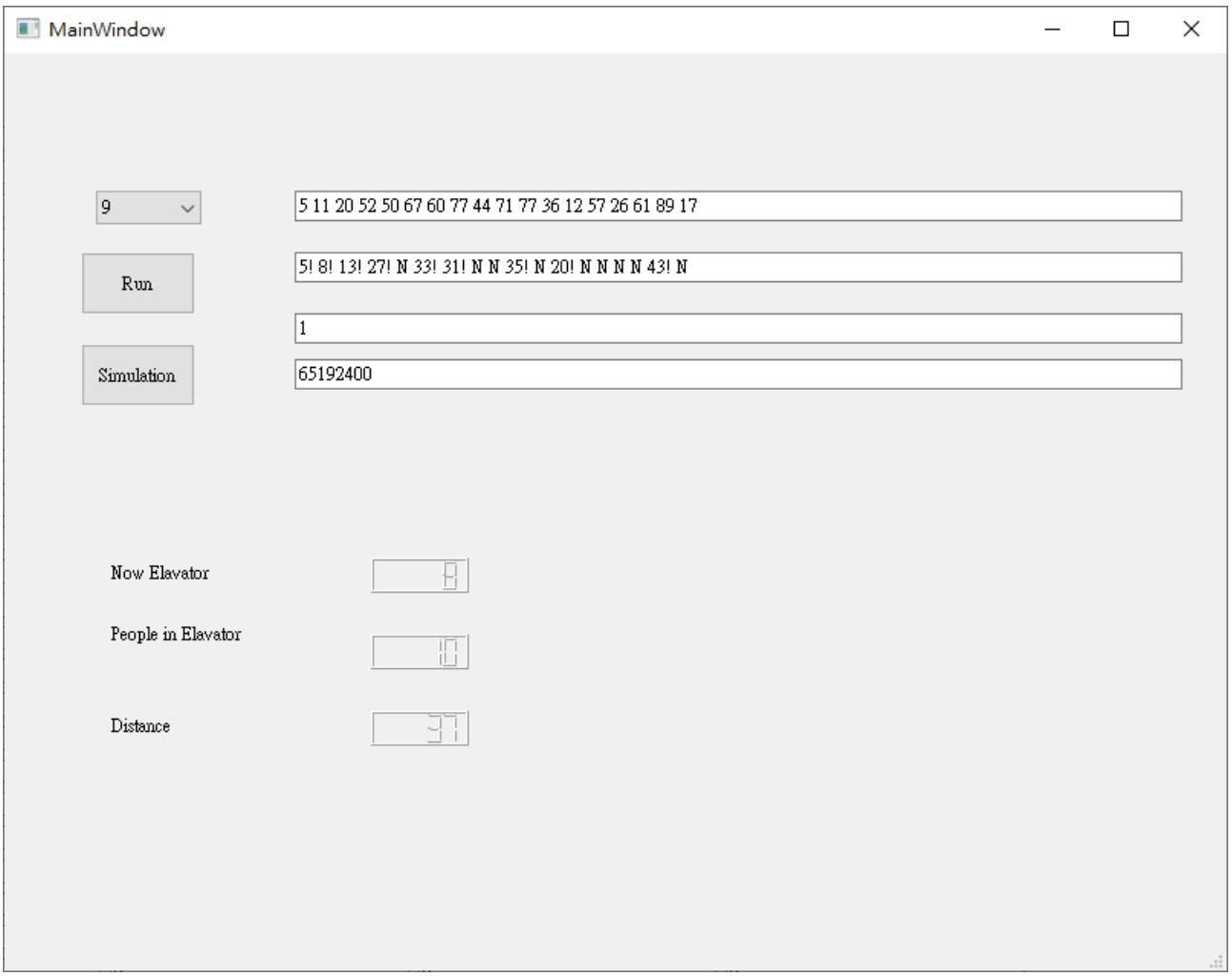
```
void Building::startSimulation(){
timer.start(100);
timer.setSingleShot(true);
}
void Building::update(){
    //…
    timer.start(100);


    emit updateGUI();
}
```

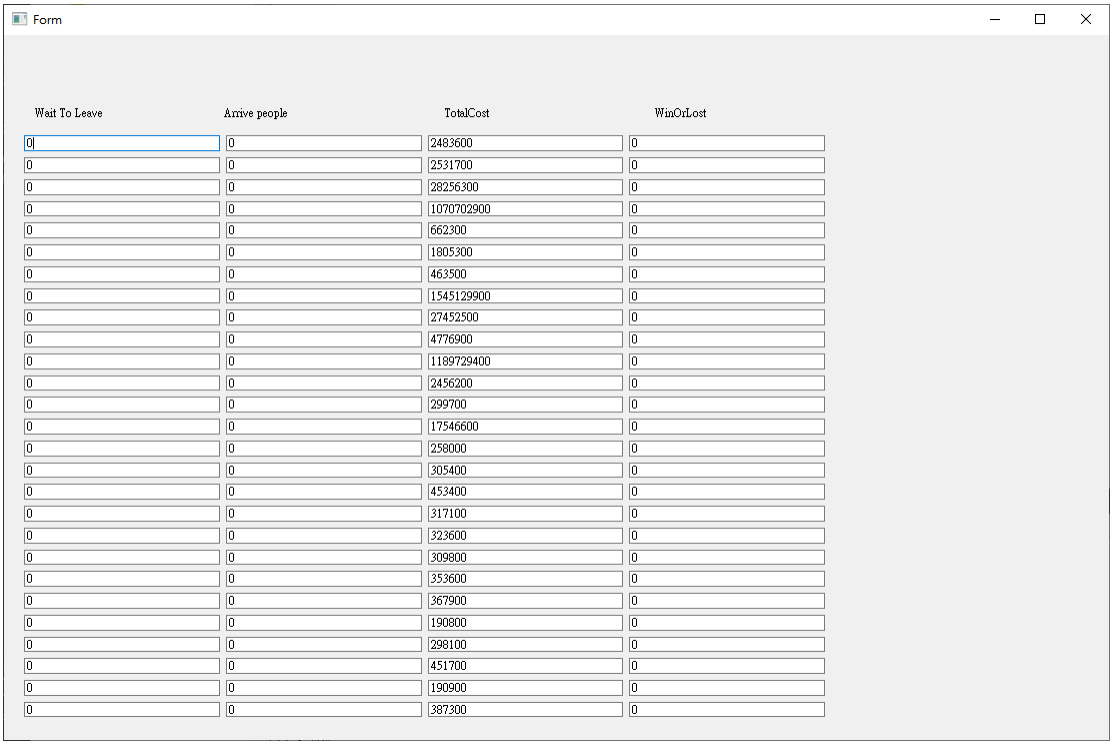
5) Start simulation and run until all people arrive their destination
6) Update 2 widget while run simulation.
7) Output your path to the “Schedule.txt” file with follow format

```
Floor(),(I/O),Number
Ex:
0,I,5
1,I,5
5,O,5
……
```

8) In “Data” class, add 2 data **distance** and **elevatorpeople**.
9) Modify GUI to display **elevator position**, how many **people in elevator** and **distance**.(you can use LCDNumber or Label)

Sample Output:

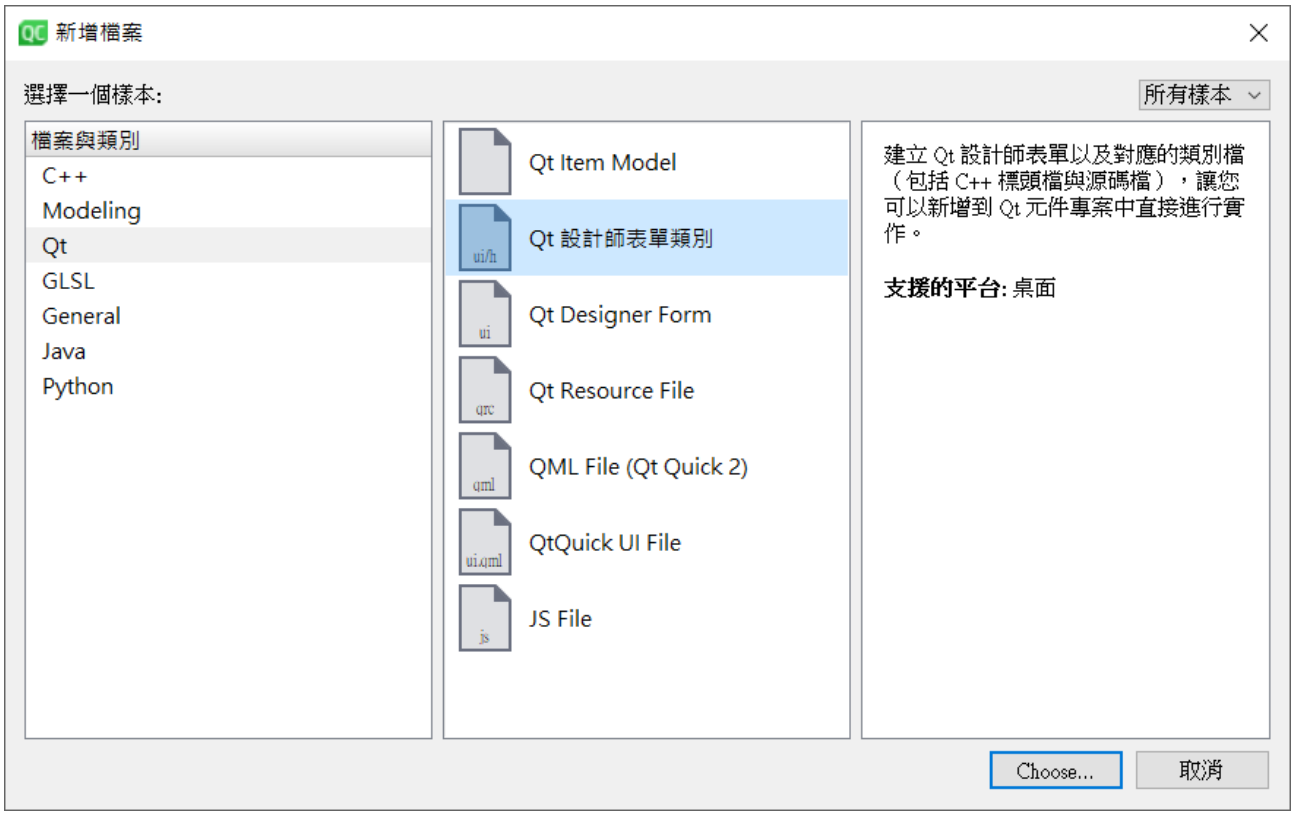




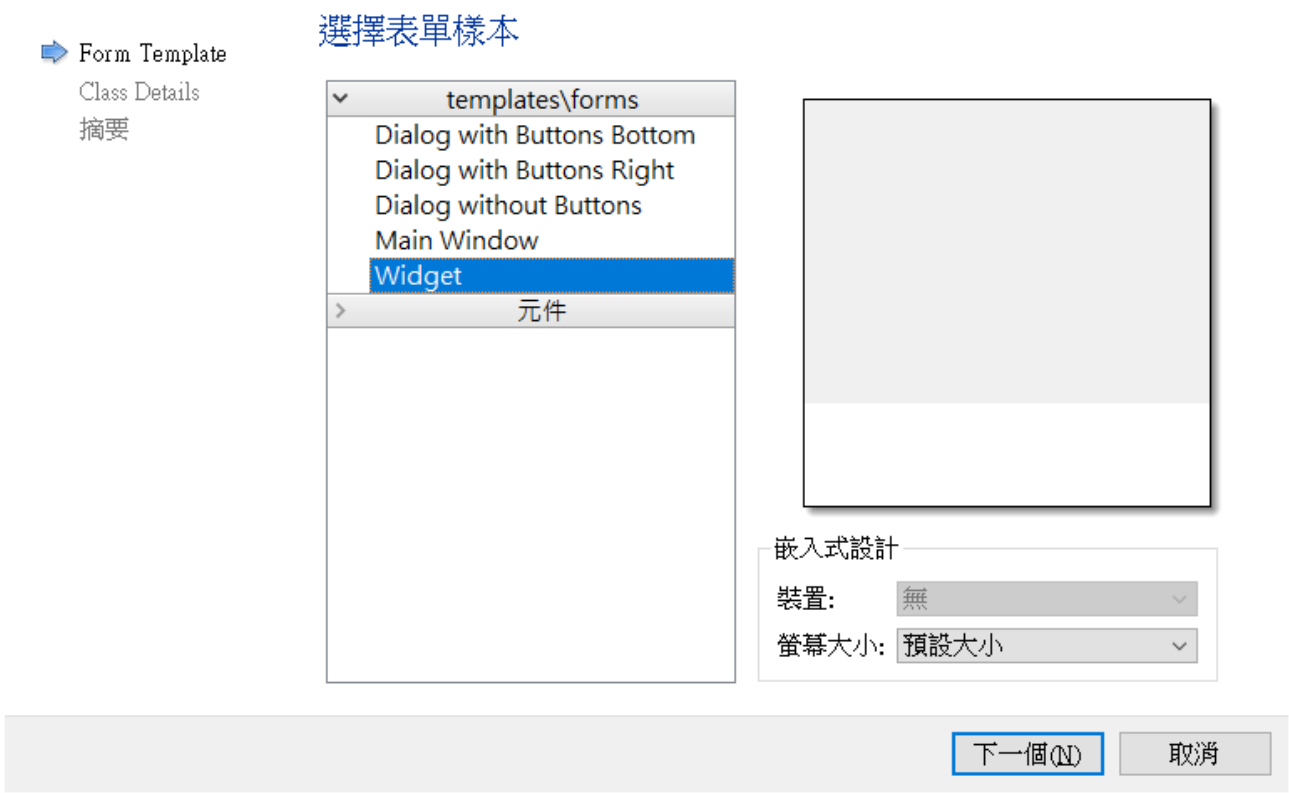
Wait To Leave	Arrive people	TotalCost	WinOrLost
0	0	2483600	0
0	0	2531700	0
0	0	28256300	0
0	0	1070702900	0
0	0	662300	0
0	0	1805300	0
0	0	463500	0
0	0	1545129900	0
0	0	27452500	0
0	0	4776900	0
0	0	1189729400	0
0	0	2456200	0
0	0	299700	0
0	0	17546600	0
0	0	258000	0
0	0	305400	0
0	0	453400	0
0	0	317100	0
0	0	323600	0
0	0	309800	0
0	0	353600	0
0	0	367900	0
0	0	190800	0
0	0	298100	0
0	0	451700	0
0	0	190900	0
0	0	387300	0

HINT:

JudgeWindow

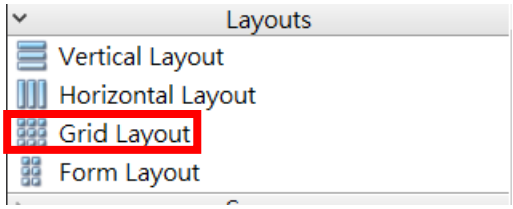


Qt 設計器表單類別



Add lots of item on widget

Add grid layout to widget



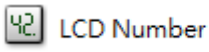
In .h file

```
QLineEdit showline[27][4];
```

In constructer

```
for(int i=0;i<27;i++){
    for(int j=0;j<4;j++){
        ui->gridLayout->addWidget(&showline[i][j],i,j);

    }
}
```

LCDNUMBER




```
ui->lcdNumber->display(windata.nowfloor);
```