

1. Project

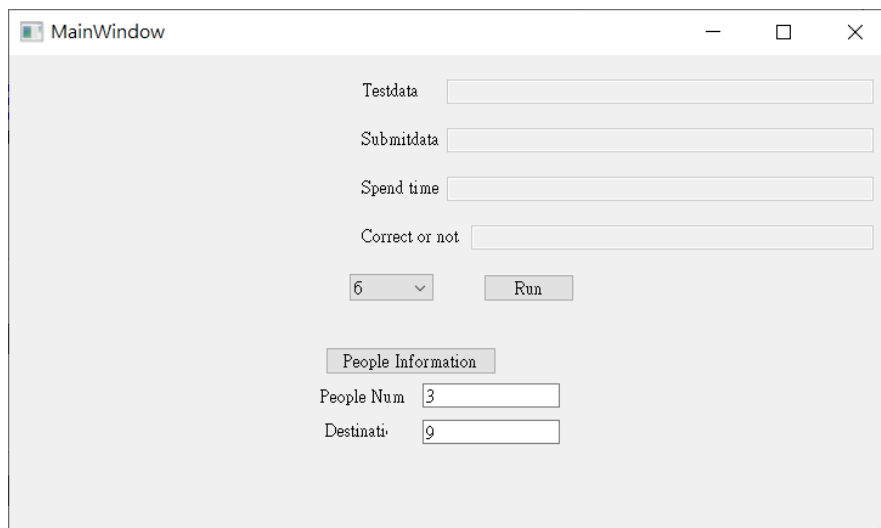
You need to modify your code in **Course 4** with the following requirements:
(Do following steps in QT.)

- 1) Establish a connection with MySQL server. (**Set database name is not needed.**)
- 2) Drop the schema "Course6" if it exists in MySQL server.
- 3) Create a new schema "Course6" in MySQL server.
- 4) Use the schema "Course6".
- 5) Drop the table "poepelist" if it exists in "Course6".
- 6) Create a new table "poepelist" with the following format:

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.

- 7) Load data from **csv** file by **using Qt**, not workbench. (**After loading data in QT, please check if your database successfully build in WorkBench.**)
- 8) You have to declare a new Class "People", and add a "People" object as a data member in "Floor" or "Building" class. (Today we have 1-10 floors)
- 9) Randomly get an initial condition from the database by using SQL command "**where like**" and initialize all data members in people. (Today we have 30 initial conditions)
- 10) Please modify GUI:
Add a button and two lineEdit to show the number of people on the nth floor and their destination.

Sample Output



2. Distance Between Prime (MyMath)

Given a positive integer k , k lies between two prime numbers p and $p+n$, i.e., $p \leq k \leq p+n$. Your goal is to find the distance between two prime numbers p and $p+n$.

Example

Take $k=10$, for example. k lies between two prime numbers 7 and 11, i.e., $7 \leq 10 \leq 11$, so the answer is 4.

Sample Input

10 11 27 2

Sample Output

4 0 6 0

3. Mario Ojisan (MyMath)

Mario likes coins. He travels to a country where there are **N** different types of coins. He wants to collect **as many different types of coins as** he can. Mario can withdraw **any** amount of money from the bank. He should maximize the number of different coins in a single withdrawal.

Requirement:

If we want to withdraw **X** amount of money from the bank, the bank will use the following algorithm.

```
withdraw(X){  
    if( X == 0) return;  
        //Let Y be the highest valued coin that does not exceed X.  
        //Give the customer Y valued coin.  
    withdraw(X-Y);  
}
```

- 1) Each test case starts with **N** different types of coin.
- 2) Following N integers represent value of each coin type.
- 3) Find the **maximum** number of different coins **in a single withdraw**.
- 4) Each test case **must** have a coin type that the **value is 1**.

Sample input:

```
6 1 2 4 8 16 32 6 1 3 6 8 15 20  
↑(N)           ↑(N)
```

Sample output:

```
6 4
```

4. Escape! If You Want to Survive (MyOther)

Your name is Joe a janitor in the maze. One day, your ex set the fire in the maze, RUN JOE RUN for your life.

Given Joe's location in the maze and which squares of the maze are on fire, you must determine whether you can exit the maze before the fire reaches you, and find the fastest way you can escape.

Joe and the fire each move one square per minute, vertically or horizontally (not diagonally). The fire spreads all four directions from each square that is on fire. Joe may exit the maze from any square that borders the edge of the maze. Neither Joe nor the fire may enter a square that is occupied by a wall.

Requirement:

- 1) The first two integers are R and C, with $1 \leq R, C \leq 1000$.
- 2) The following R string, separated by space, of the test case each contain one row of the maze. Each of these strings contains exactly C characters.
- 3) #=walls, .=place can pass, J=Joe, F=place on fire.
- 4) Output the fastest time Joe can safely exit the maze in minutes.
- 5) If there is not possible way to escape, please print "N".

Example

4 4

####	####	####	
#JF#	#FF#	#FF#	
#..# =>	#JF# =>	#FF# =>	Escaped! =>3 min
#..#	#..#	#JF#	

Sample input:

4 4 #### #JF# #..# #..#

Sample output:

3