# LAB 1113 AOOP

1. **Project**

   You need to modify your code in Course 3 with the following requirement:

   1) All the functions in Base Class are pure virtual function.
   2) You have to declare a new Class "Floor" which has a pointer of ProblemSet.
   3) In "Building" class, we have 30 "Floor" objects in an array and each floor represents a different "ProblemSet". (In this Course, you only need 5 floors.)
   4) Use pointer of ProblemSet to call the "solve" function.

      ```
      floor[0]=new Floor(new Add1() );
      string s2=floor[n-1]->p->solve(s);
      ```

   5) Please modify GUI:

      Use combo box to select a question instead of showing all of them in a window.

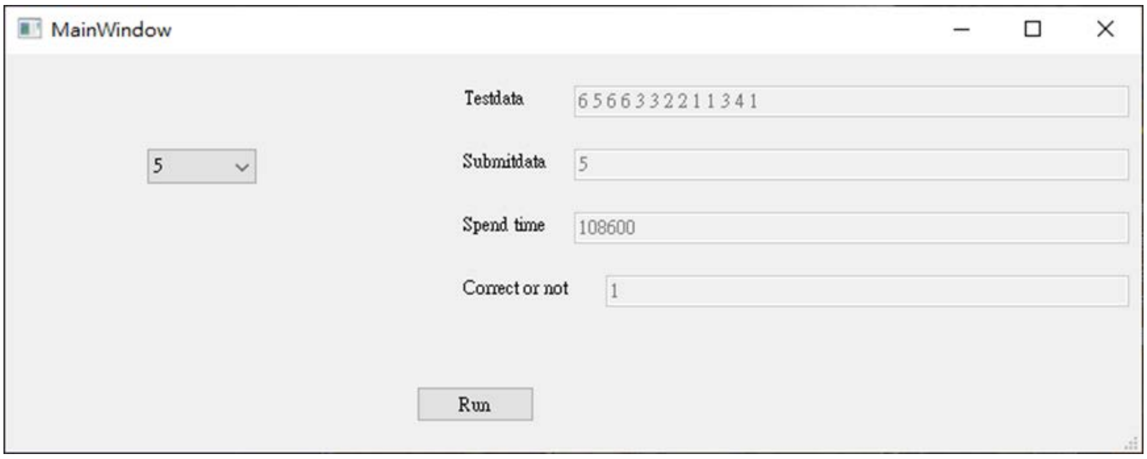
   **Hint:**

   Floor.h

   ```
   class Floor
   {
   public:
      Floor();
      Floor(ProblemSet *problem){this->p=problem;}
      void setProblem(ProblemSet *problem){this->p=problem;}
      ProblemSet *p;
   private:
      int fn;
   };

   #endif // FLOOR_H
   ```

**Sample Output**


MainWindow

5

Testdata 6 5 6 6 3 3 2 2 1 1 3 4 1

Submitdata 5

Spend time 108600

Correct or not 1

## 2. Closest Pair (MyMath)

In this problem, you need to find the shortest distance between two three-dimensional points.

Requirement:

1) Input is a string which contains N sets of numbers with "double" data type (2<N<10000). Each set contains three values: x, y and z.
2) Output number should contain two digits of precision of floating-point number.

**Hint**

You may need to include "iomanip" and "sstream" libraries in this problem.

**Sample Input**

1.00 1.00 1.00 1.05 1.00 1.00 3.05 4.87 5.78 3.87 4.12 8.24 7.19 1000000000 9.57

**Sample Output**

0.05

## 3. Minesweeper(踩地雷)(MyOther)

The goal of the Minesweeper is to find where are all the mines within a M × N field. Your goal is to find the original map that shows a number in a square which tells you how many mines there are adjacent to that square.

**Example**

Suppose the following 4 × 4 field with 2 mines (which are represented by an '*' character):

```
*...
....
.*..
....
```

If we would represent the same field placing the hint numbers described above, we would end up with:

```
*100
2210
1*10
1110
```

As you may have already noticed, each square may have at most 8 adjacent squares.

The following sample input let you easy to understand the meaning of the question. They are not the real **format** of testdata.

**Sample Input**

```
4 4
*...
....
.*..
....
3 5
**...
.....
.*...
0 0
```

**Sample Output**

```
*100
2210
1*10
1110

**100
33200
1*100
```

**The real input and output format will be in the file on new E3,** each line is a test data and the next line is the answer.

**Ex:**

```
4 4 *... .... .*.. ....       //testdata 1
*100 2210 1*10 1110           //ans 1
3 5 **... ..... .*...         //testdata 2
**100 33200 1*100             //ans 2
```

Hint: use “stringstream” to achieve the function of cin cout.

4. **Get signature(MyOther)**

You have a blank paper. There are many celebrities in the plaza. You can only ask one celebrity to sign on the paper. Then the celebrity each pick one other person they know to sign for you. (exactly one, no less, no more and never themselves). Which celebrity should you ask to maximize the number of signatures that you get?

Requirement:

1) Each case starts with a line containing an integer $N$ ($2 \le N \le 50000$) denoting the number of celebrities in the plaza. Each of the next N lines contains two integers: u v ($1 \le u, v \le N, u \ne v$) meaning that celebrity u asks celebrity v to sign for you.
2) Print out the celebrity m that you should ask first to get maximize the number of signatures. If there is more than one correct answer, output the smallest number.

**Sample Input**

```
3
1 2
2 3
3 1
4
1 2
2 1
4 3
3 2
```

**Sample Output**

```
1
4
```

**The real input and output format will be in the file on new E3,** each line is a test data and the next line is the answer.

```
3 1 2 2 3 3 1              //testdata 1
1                          //ans 1
4 1 2 2 1 4 3 3 2          //testdata 2
4                          //ans 2
```