OOP&DS Midterm (2018/4/23)

Q1 (20%)

Submit: Q1.cpp (include main function)

Given two positive integers which are less than 2^31-1, please find the Greatest Common Divisor.

Input :

# of testing data

[positive integer1] [positive integer2]

(2 positive integers are seperated by space.)

Output :

[Greatest Common Divisor]

Example:

Input	3 8 16 36 24 17 51
Output	8 12 17

Q2 (20%)

Submit: Q2.cpp (exclude main function)

Please complete the cpp file. In addition, you are not allowed to modify Coordinate.h.

In Coordinate.h:

(a)Coordinate() is already completed.

(b) Coordinate(float x, float y): When a pair of floating number is passed to constructor, the private data xValue, yValue should be assigned to x, y and the constructor should print "constructor\n" as well.

(c) void setNewValue(float x, float y): Set new value to this point.

(d) void getMidPointWith(Coordinate c): Get midpoint with c and store in original point.

(e) float getLengthWith(Coordinate c): Get the length with c.($\sqrt{(\Delta x)^2 + (\Delta y)^2}$)

(f) float getSlopeWith(Coordinate c): Get the slope (斜率) with c.

(g) float showXValue(): return xValue;

(h) float showyValue(): return yValue;

Please make sure your code can run with this main function.(Do not write any main function in your hand in code)

```cpp
#include <iostream>
using namespace std;
#include "Coordinate.h"
int main()
{
    Coordinate c1(1.0,2.0);
    Coordinate c2(3.5,4.5);
    Coordinate c3;
    cout << c1.getLengthWith(c2) <<endl;
    cout << "(" << c1.showXValue() << "," << c1.showYValue() << ")" <<endl;
    c1.getMidPointWith(c2);
    cout << "(" << c1.showXValue() << "," << c1.showYValue() << ")" <<endl;
    cout << c1.getLengthWith(c2) <<endl;
    cout << c1.getSlopeWith(c3) <<endl;
    c1.setNewValue(5.1,6.2);
    cout << "(" << c1.showXValue() << "," << c1.showYValue() << ")" <<endl;
    return 0;
}
```

Output:

constructor

constructor

constructor

3.53553

(1,2)

(2.25,3.25)

1.76777

1.44444

(5.1,6.2)

Q3 (20%)

Submit: Q3.cpp (include main function)

Please complete cpp file

Use class template to implement 2X2 matrix

Function set(int i, int j, M value)：

The first parameter means ith row

The second parameter means jth column

The third parameter means value

You should use these operator below:

Operator () means calculate matrix determinant

Ex: $(\begin{bmatrix} a & b \\ c & d \end{bmatrix})$=ad-bc

Operator ^ means calculate matrix power

Ex: $(\begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix})^{(2)} = \begin{bmatrix} 2 & 3 \\ 3 & 5 \end{bmatrix}$

Operator<< to print out your answer

Input :

The input consists of a number of cases. Each case starts with the integer n.

If n is 1, then the next 2 lines will consists of 4 integers, the element (a,b,c,d) of matrix $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$, and 1 integer, the number of power.

If n is 2, then the next 2 lines will consists of 4 *floating point,* the element (a,b,c,d) of matrix $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$, and 1 integer, the number of power.

The end of input is specified by a line in which n = 0.

Output :

Each group of output consists 2 lines.

The first line print the determinant

The second line print the matrix after calculated power

Example：

Input	1 2 1 6 3 2 2 1 2.3 1 1.5 2 0
Output	0 10 5 30 15 -0.8 3.3 5.75 2.5 4.55

Q4 (20%)

Submit: Q4.cpp (include main function)

You have to set a class which have the information of fruit, include name and cost.

We will give you 5 kinds of fruit and its cost as below

name	cost
apple	10
banana	5
orange	20
watermelon	50
lemon	15

You have to set a class which have the information of guests, include name and how much money they spend

We will type 4 guests and what they buy

Ex.

Alan (guests name)

2 (how many kind of fruit he buy)

lemon 2 (what kind of fruit and how many)

orange 3

Output :

Print the guest name and much money they spend

#The information of fruit should come from the class you set above

Example：

Input	Alan 2 lemon 2 orange 3 Alice 3 apple 2 banana 3 orange 1 Lisa 1 watermelon 2

	Lucy 2 apple 4 banana 1
Output	Alan 90 Alice 55 Lisa 100 Lucy 45

Q5 (20%)

Submit: Q5.cpp (include main function)

Please make an adder that can add two integers and can also concatenate two strings. Write a function called "add" that uses function template. The function has two parameters and return the result of adding.

(function "add" can add two integers, and concatenate two strings.)

Input

The input consists of a number of cases. Each case starts with the integer n.

If n is 1, then the next line will consist of two integers x and y. (0≤x≤1000, 0≤y≤1000)

If n is 2, then the next 2 lines will consist of 2 strings str1 and str2.

The end of input is specified by a line in which n = 0.

Output

When the input number n = 1, you should print the result of (x + y).

When the input number n = 2, you should print "str1str2".

Example:

Input	1 10 27 1 33 78 2 OOP DS 0
Output	37

	111 OOPDS

Example:

```
int num1= 1;
int num2 = 2;
cout << add(num1,num2);
//it should output "3" on the screen.

string string1 = "ABC";
string string2 = "DEF";
cout << add(string1,string2);
//it should output "ABCDEF" on the screen
```

WARNING: YOU MUST USE FUNCTION TEMPLATE OTHERWISE YOU WILL GET 0 POINT.

Q6 (20%)

Submit: Q6.h (exclude main function)

Your boss want you to extend more function on stl vector function which type is integer. So you must create your own class myVector that inherit Vector<int> and extend 4 functions:

(1) sum : retrun sum of the whole vector which type is integer

(2) concat : concat another integer vector

(3) sort : sort the whole vector

(4) show : show the whole vector value

However, you're in the bad mood. You want to play a trick. You overwrite a size function that always return zero:

(5) size : return zero always

Example:

```
myVector myvector1,myvector2;
cout<< myvector1.size() <<endl;
for (int i=0; i<5; i++) myvector1.push_back(i);
for (int i=10; i>5; i--) myvector2.push_back(i);
cout<< myvector1.size() <<endl;
cout<< "sum :" << myvector1.sum() <<endl;
myvector1.concat(myvector2);
cout<< "sum :" << myvector1.sum() <<endl;
myvector1.show();
myvector1.sort();
myvector1.show();
```

Output:

0

0

sum :10

sum :50

0 1 2 3 4 10 9 8 7 6

0 1 2 3 4 6 7 8 9 10

Implement your own .h Class file

Submission:

1. **Submit your file to e3, and name it with "Q[question number].cpp or .h". EX: Q1.cpp for question 1, Q2.cpp for question 2.**
2. **Submit only one cpp or header file for each question.**
3. **Upload the corresponding file of each question to e3.**
4. **Do not compress it.**