HO CHI MINH UNIVERSITY OF SCIENCE
FACULTY OF INFORMATION TECHNOLOGY
SOFTWARE ENGINEERING DEPARTMENT
ADVANCED PROGRAM IN COMPUTER SCIENCE

COURSE: **INTRODUCTION TO PROGRAMMING AND PROBLEM-SOLVING**

LECTURER: Dr. ĐINH BÁ TIẾN

Struct & File

- TRƯƠNG PHƯỚC LỘC
- HỒ TUẤN THANH

HCMC, 2015

Contents

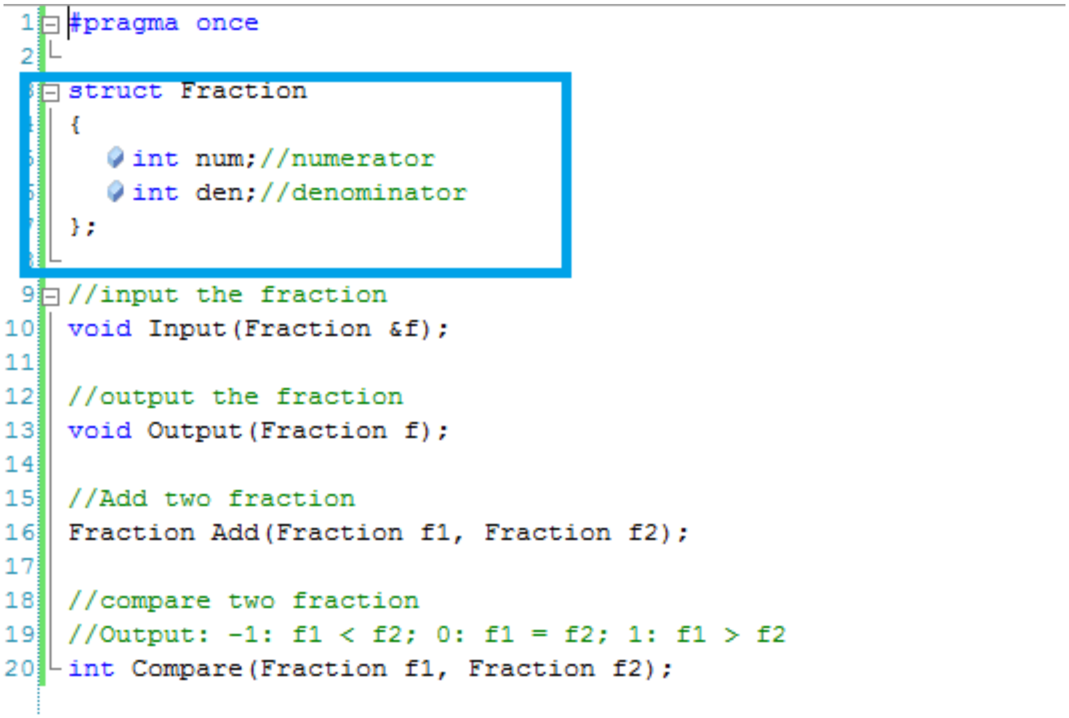
1 Objectives

- Be able to use the **struct** in the program.
- Be able to use file to save data of the program.

2 Contents

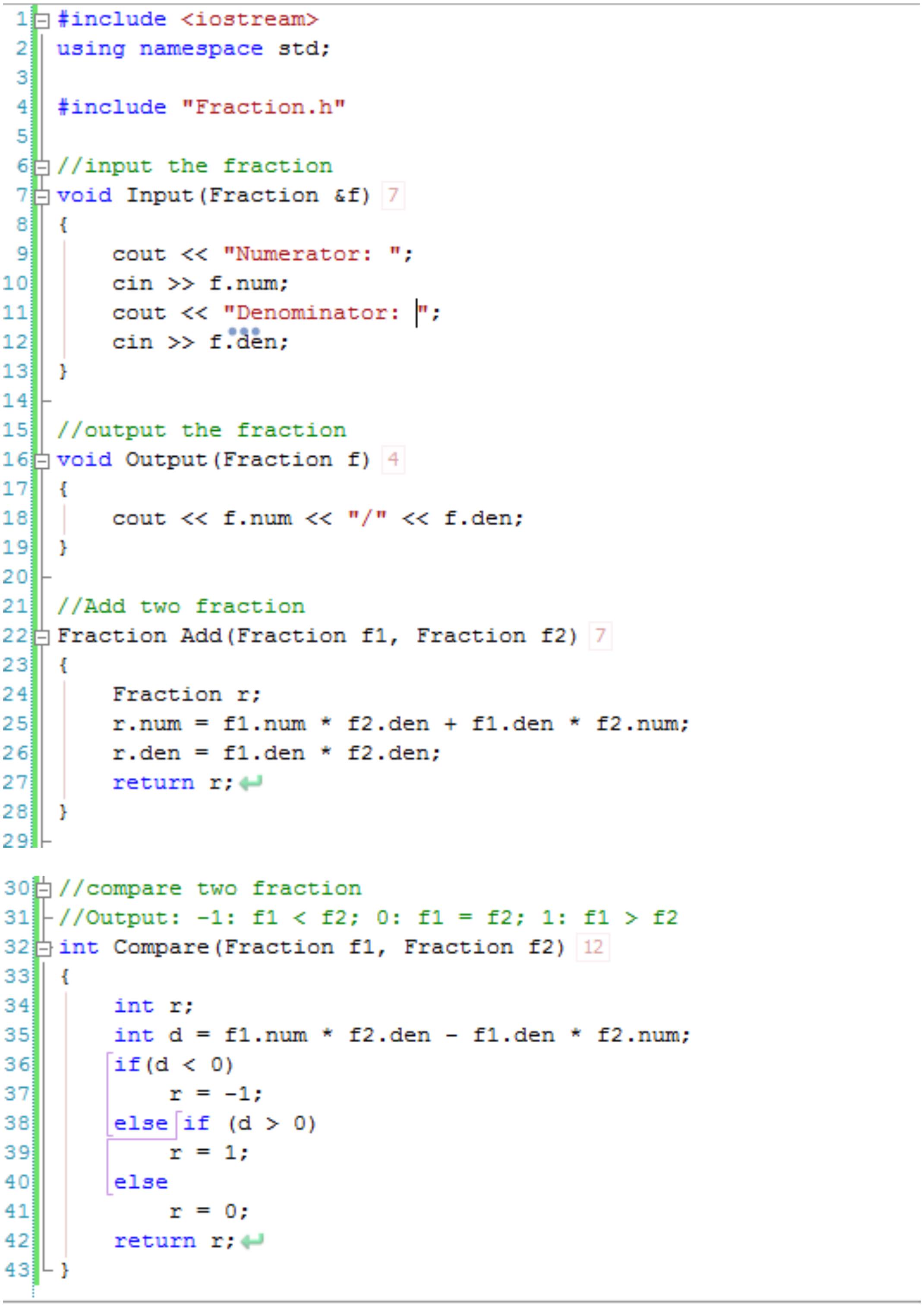
2.1 Struct

- Fraction have numerator and denominator ➔ using a variable ➔ define the new data type ➔ using struct.
- Write the program to input the two fraction and output sum of them.
- Declare struct and the functions into .h file. Example: fraction struct ➔ **Fraction.h file**



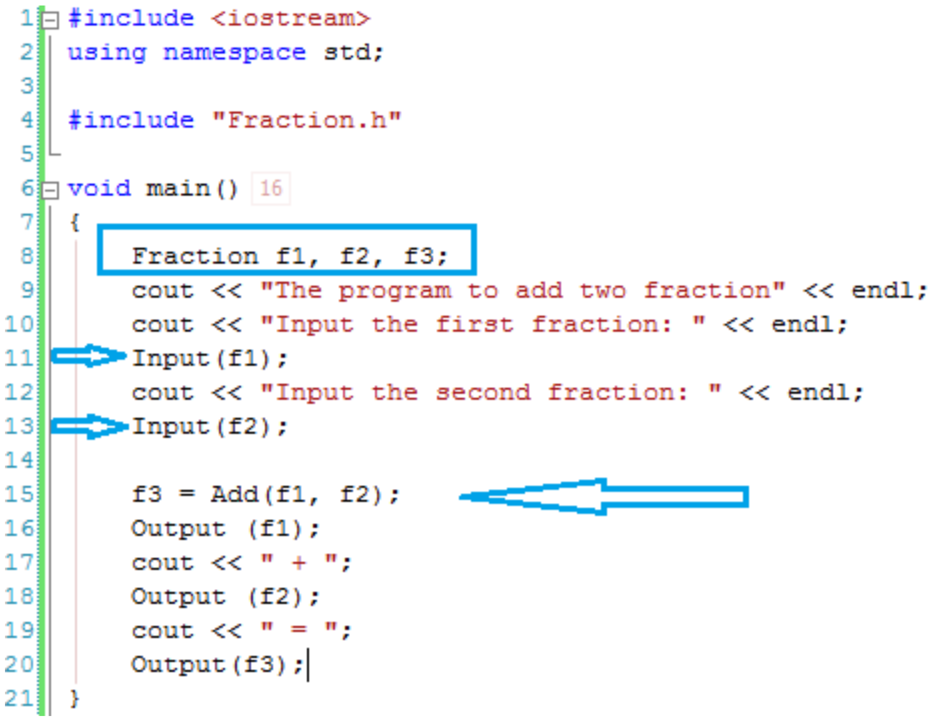
```
#pragma once

struct Fraction
{
    int num;//numerator
    int den;//denominator
};

//input the fraction
void Input(Fraction &f);

//output the fraction
void Output(Fraction f);

//Add two fraction
Fraction Add(Fraction f1, Fraction f2);

//compare two fraction
//Output: -1: f1 < f2; 0: f1 = f2; 1: f1 > f2
int Compare(Fraction f1, Fraction f2);
```

- **Fraction.cpp file** ➔ define the functions:



```
#include <iostream>
using namespace std;

#include "Fraction.h"

//input the fraction
void Input(Fraction &f)
{
    cout << "Numerator: ";
    cin >> f.num;
    cout << "Denominator: ";
    cin >> f.den;
}

//output the fraction
void Output(Fraction f)
{
    cout << f.num << "/" << f.den;
}

//Add two fraction
Fraction Add(Fraction f1, Fraction f2)
{
    Fraction r;
    r.num = f1.num * f2.den + f1.den * f2.num;
    r.den = f1.den * f2.den;
    return r;
}

//compare two fraction
//Output: -1: f1 < f2; 0: f1 = f2; 1: f1 > f2
int Compare(Fraction f1, Fraction f2)
{
    int r;
    int d = f1.num * f2.den - f1.den * f2.num;
    if(d < 0)
        r = -1;
    else if (d > 0)
        r = 1;
    else
        r = 0;
    return r;
}
```

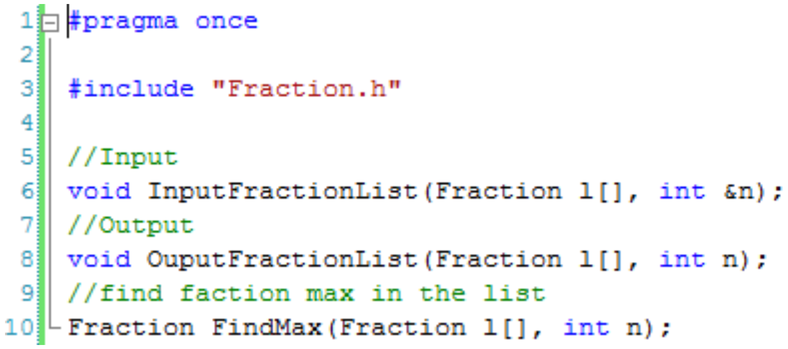
- **main.cpp file** ➔ using the truct and the functions



```
#include <iostream>
using namespace std;

#include "Fraction.h"

void main()
{
    Fraction f1, f2, f3;
    cout << "The program to add two fraction" << endl;
    cout << "Input the first fraction: " << endl;
    Input(f1);
    cout << "Input the second fraction: " << endl;
    Input(f2);

    f3 = Add(f1, f2);
    Output (f1);
    cout << " + ";
    Output (f2);
    cout << " = ";
    Output(f3);
}
```

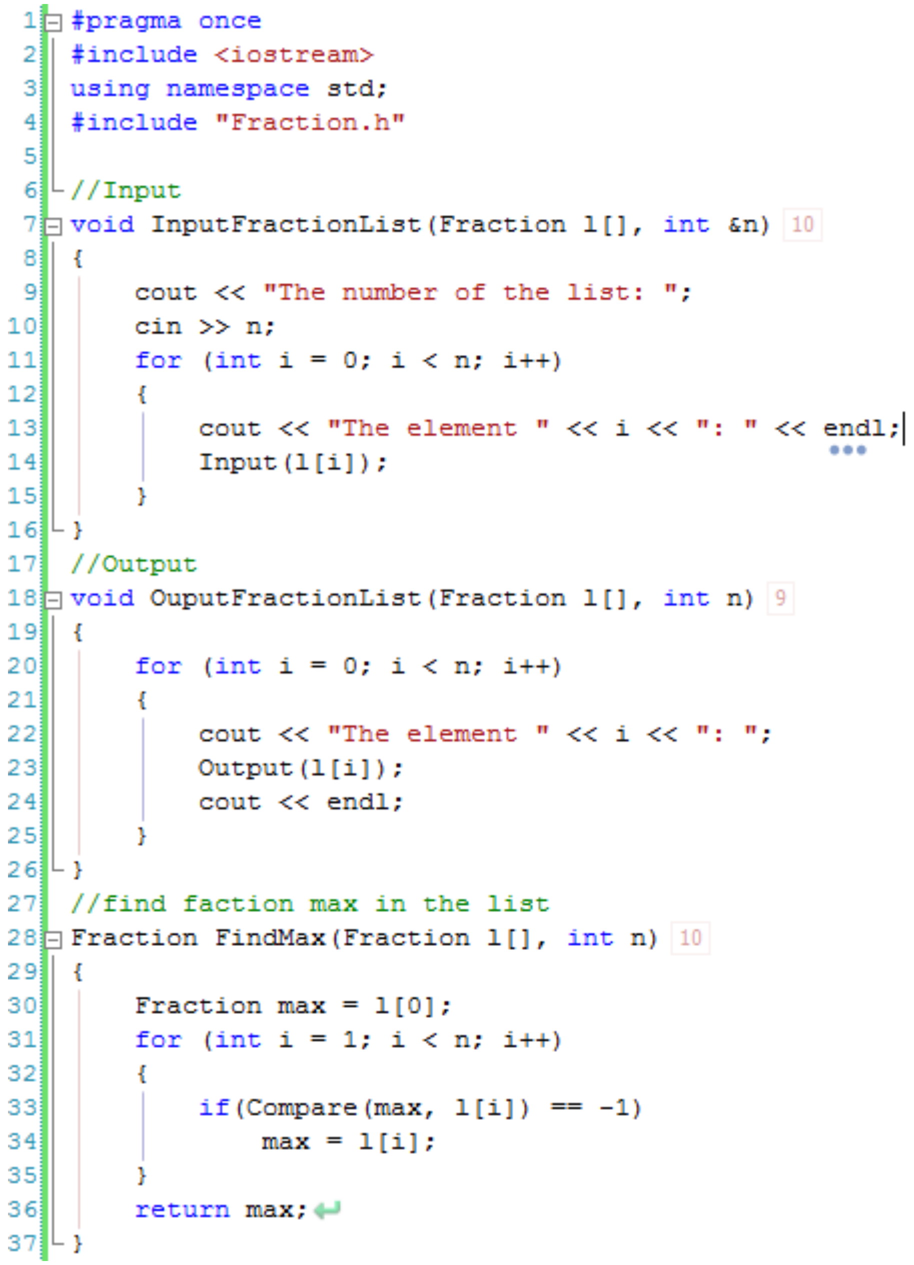
2.2 Array

- Write the program to input the fraction list and output the maximum element (use the fraction.h and fraction.cpp file into the Struct)
- FractionList.h file:



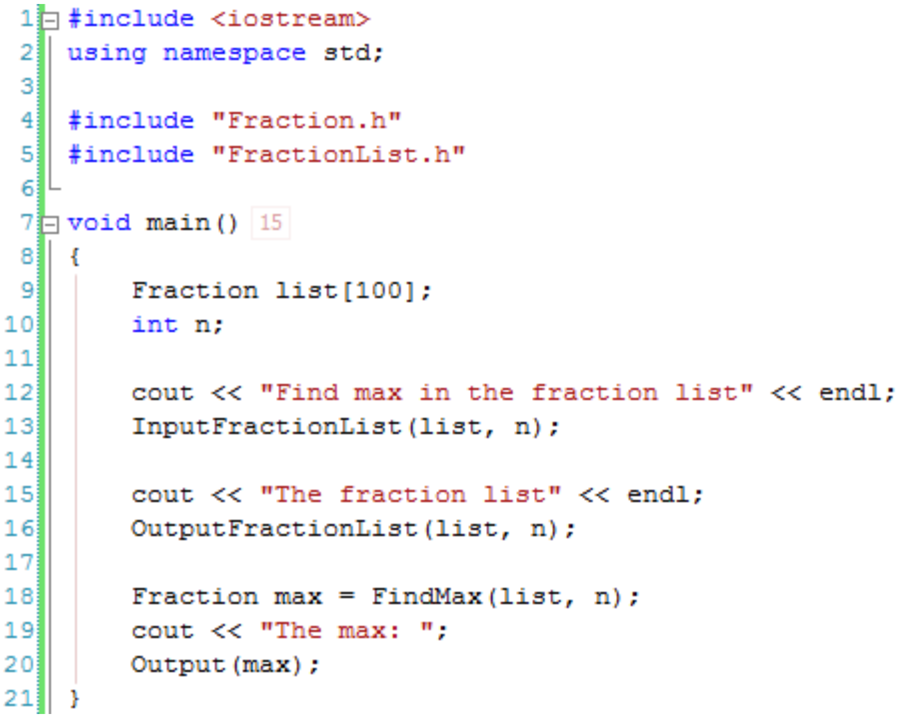
```
#pragma once

#include "Fraction.h"

//Input
void InputFractionList(Fraction l[], int &n);
//Output
void OuputFractionList(Fraction l[], int n);
//find faction max in the list
Fraction FindMax(Fraction l[], int n);
```

- FractionList.cpp file:



```
#pragma once
#include <iostream>
using namespace std;
#include "Fraction.h"

//Input
void InputFractionList(Fraction l[], int &n)
{
	cout << "The number of the list: ";
	cin >> n;
	for (int i = 0; i < n; i++)
	{
		cout << "The element " << i << ": " << endl;
		Input(l[i]);
	}
}
//Output
void OuputFractionList(Fraction l[], int n)
{
	for (int i = 0; i < n; i++)
	{
		cout << "The element " << i << ": ";
		Output(l[i]);
		cout << endl;
	}
}
//find faction max in the list
Fraction FindMax(Fraction l[], int n)
{
	Fraction max = l[0];
	for (int i = 1; i < n; i++)
	{
		if(Compare(max, l[i]) == -1)
			max = l[i];
	}
	return max;
}
```

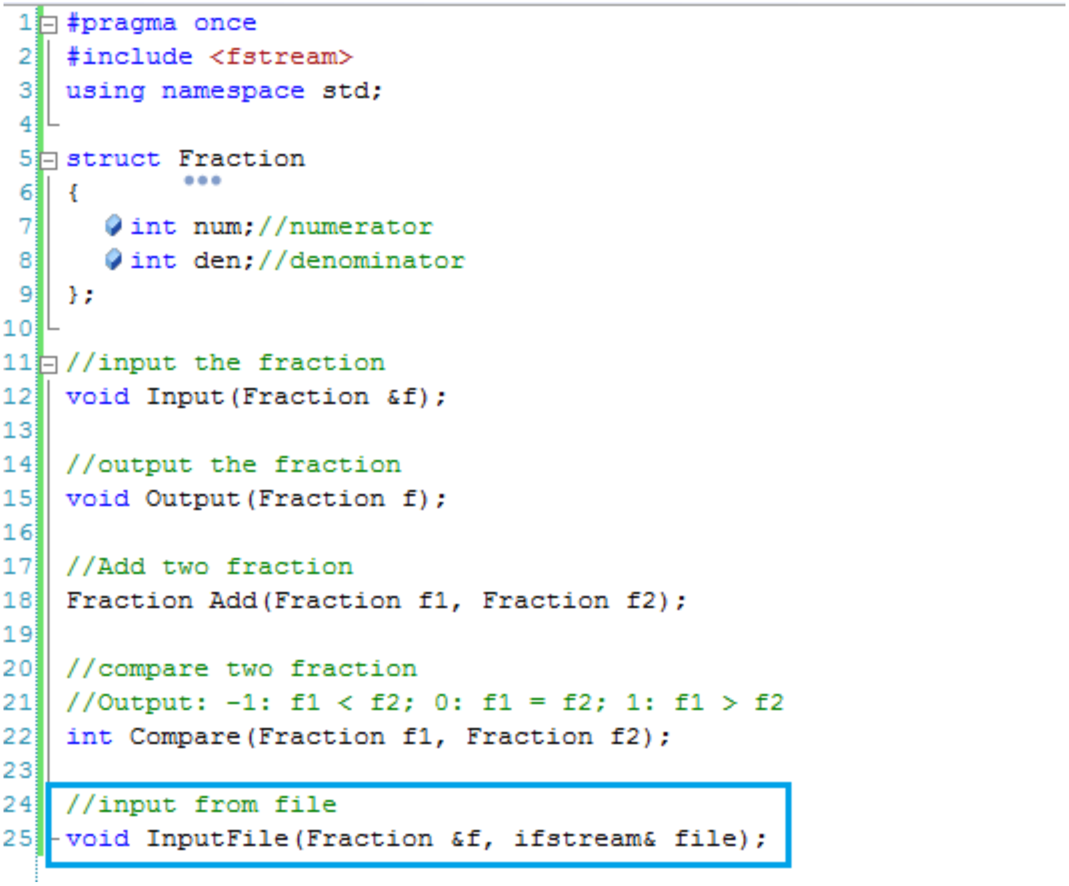
- Main.cpp file:



```
#include <iostream>
using namespace std;

#include "Fraction.h"
#include "FractionList.h"

void main()
{
    Fraction list[100];
    int n;

    cout << "Find max in the fraction list" << endl;
    InputFractionList(list, n);

    cout << "The fraction list" << endl;
    OutputFractionList(list, n);

    Fraction max = FindMax(list, n);
    cout << "The max: ";
    Output(max);
}
```

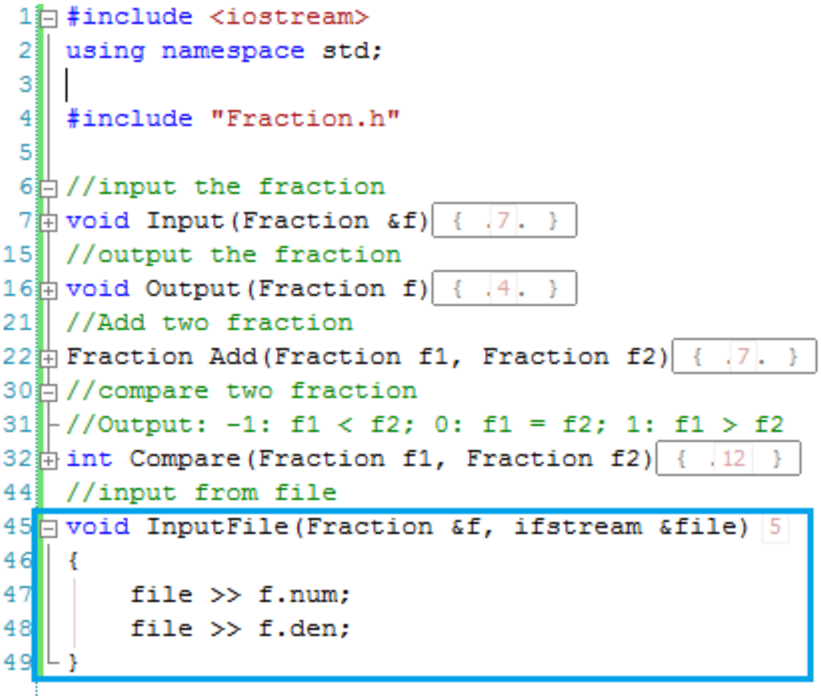
2.3 File

- Write the program to read the fraction list from file and output the maximum element (use the fraction.h and fraction.cpp file in the Struct, fractionlist.h and fractionlist.cpp in the Array, but add the `InputFile, InputFileFractionList`)
- Fraction.h



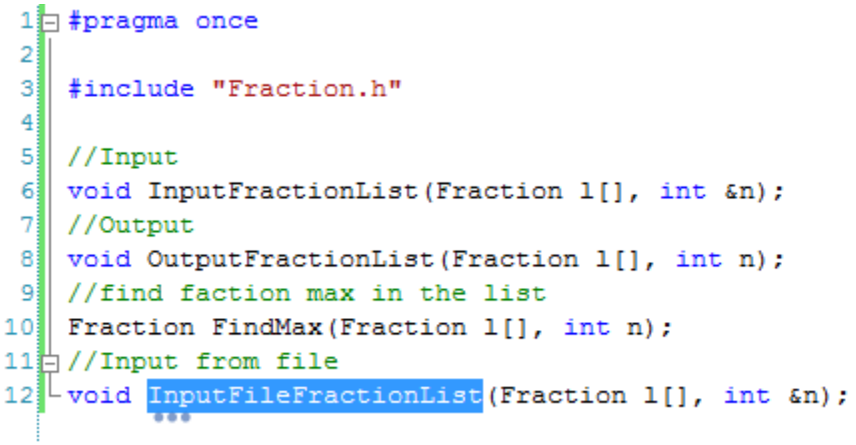
```
#pragma once
#include <fstream>
using namespace std;

struct Fraction
{
    int num;//numerator
    int den;//denominator
};

//input the fraction
void Input(Fraction &f);

//output the fraction
void Output(Fraction f);

//Add two fraction
Fraction Add(Fraction f1, Fraction f2);

//compare two fraction
//Output: -1: f1 < f2; 0: f1 = f2; 1: f1 > f2
int Compare(Fraction f1, Fraction f2);

//input from file
void InputFile(Fraction &f, ifstream& file);
```

- Fraction.cpp



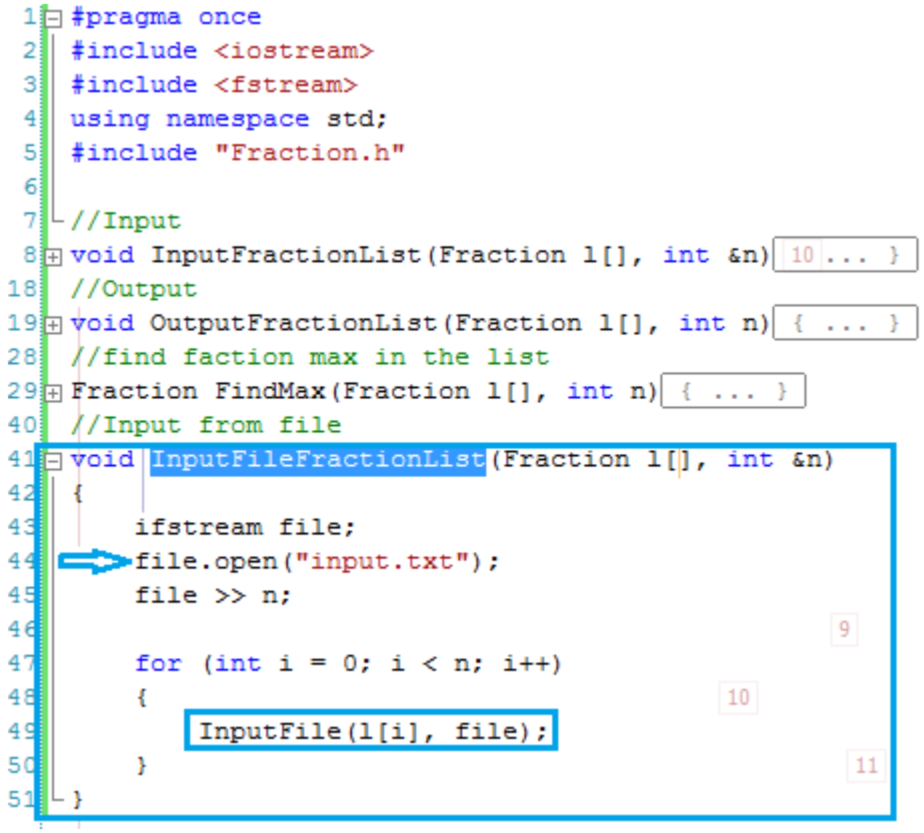
```
#include <iostream>
using namespace std;

#include "Fraction.h"

//input the fraction
void Input(Fraction &f) { .7. }
//output the fraction
void Output(Fraction f) { .4. }
//Add two fraction
Fraction Add(Fraction f1, Fraction f2) { .7. }
//compare two fraction
//Output: -1: f1 < f2; 0: f1 = f2; 1: f1 > f2
int Compare(Fraction f1, Fraction f2) { .12 }
//input from file
void InputFile(Fraction &f, ifstream &file) 5
{
    file >> f.num;
    file >> f.den;
}
```

- FractionList.h



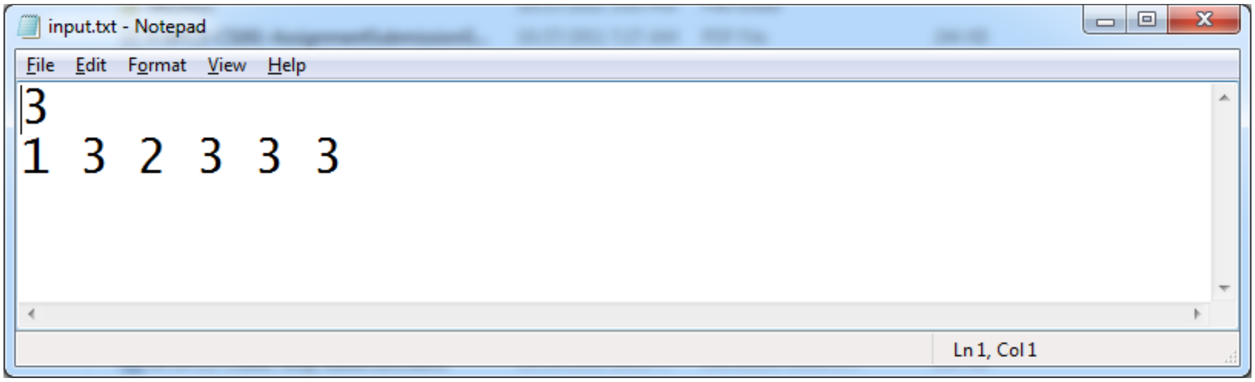
```
#pragma once

#include "Fraction.h"

//Input
void InputFractionList(Fraction l[], int &n);
//Output
void OutputFractionList(Fraction l[], int n);
//find faction max in the list
Fraction FindMax(Fraction l[], int n);
//Input from file
void InputFileFractionList(Fraction l[], int &n);
```

- FractionList.cpp



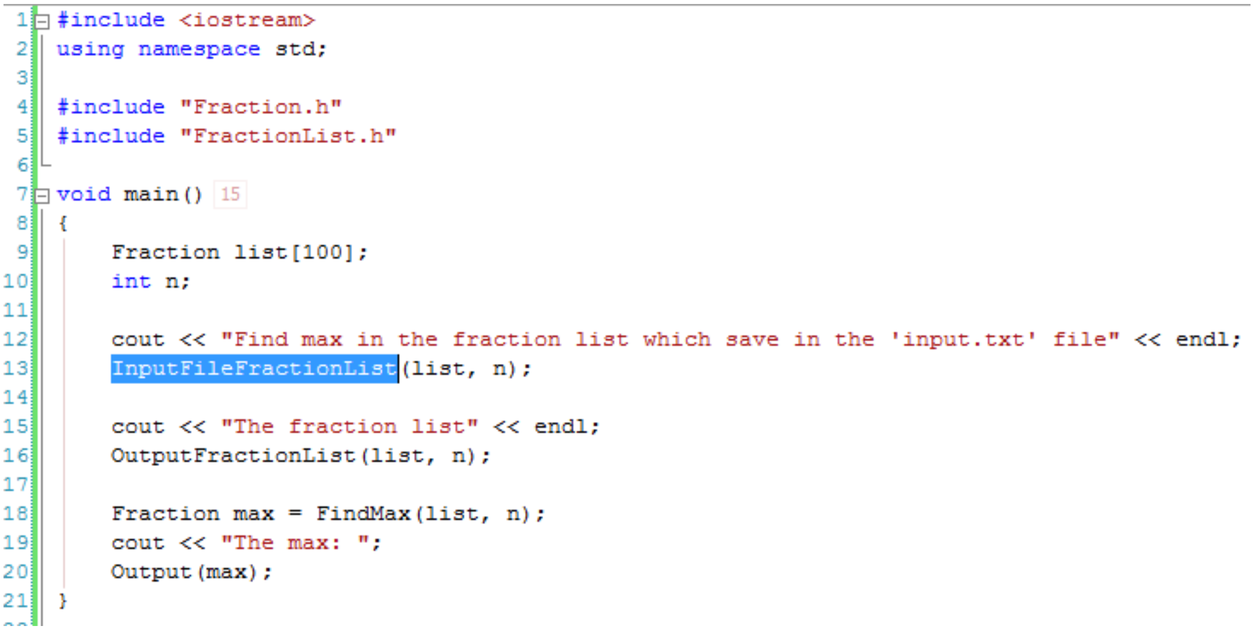
```
#pragma once
#include <iostream>
#include <fstream>
using namespace std;
#include "Fraction.h"

//Input
void InputFractionList(Fraction l[], int &n) 10 ... }
//Output
void OutputFractionList(Fraction l[], int n) { ... }
//find faction max in the list
Fraction FindMax(Fraction l[], int n) { ... }
//Input from file
void InputFileFractionList(Fraction l[], int &n)
{
	ifstream file;
	file.open("input.txt");
	file >> n;

	for (int i = 0; i < n; i++)
	{
		InputFile(l[i], file);
	}
}
```

- Input.txt



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

- Main.cpp



```cpp
#include <iostream>
using namespace std;

#include "Fraction.h"
#include "FractionList.h"

void main()
{
    Fraction list[100];
    int n;

    cout << "Find max in the fraction list which save in the 'input.txt' file" << endl;
    InputFileFractionList(list, n);

    cout << "The fraction list" << endl;
    OutputFractionList(list, n);

    Fraction max = FindMax(list, n);
    cout << "The max: ";
    Output(max);
}
```

3 Assignment 09

3.1 Assignment 09 - 01

Write the program to input the fraction list and output max fraction.

3.2 Assignment 09 – 02

Write the program to input the fraction list from file and output (screen and file) max fraction.

3.3 Assignemnt 09 – 03

Write the program to input the list of triangles from file and output (screen and file) focus of the triangle which have max perimeter.

3.4 Assignemnt 09 – 04

Write the program to input the list of circles from file and output (screen and file) focus of the circle which have max perimeter.

3.5 Assignemnt 09 – 05

Write the program to input the list of products (weight, price) from file and output (screen and file) focus of the product which have max weight & max price.

3.6 Assignemnt 09 – 06

Write the program to read information of the student list in file and output (screen and file) the information and number of days (left or over) of the student whose birthday is nearest with current day.

The information of a student:

- Full name (more than 10 character)
- Birthday (greater than or equal 18 years old, valid date. Ex: 13/01/1990)
- Email (must be have "@apcs.vn")
- Address (less than 128 character).