



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. ĐÌNH BÁ TIẾN

## **Struct & File**

TRƯỜNG PHƯỚC LỘC  
HỒ TUẤN THANH

HCMC, 2015

## Contents

1	Objectives .....	3
2	Contents .....	3
2.1	Struct .....	3
2.2	Array .....	5
2.3	File.....	7
3	Assignment 09 .....	11
3.1	Assignment 09 - 01 .....	11
3.2	Assignment 09 – 02.....	11
3.3	Assignemnt 09 – 03.....	11
3.4	Assignemnt 09 – 04.....	11
3.5	Assignemnt 09 – 05.....	11
3.6	Assignemnt 09 – 06.....	11

## 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**

```
1 #pragma once
2
3 struct Fraction
4 {
5     int num;//numerator
6     int den;//denominator
7 };
8
9 //input the fraction
10 void Input(Fraction &f);
11
12 //output the fraction
13 void Output(Fraction f);
14
15 //Add two fraction
16 Fraction Add(Fraction f1, Fraction f2);
17
18 //compare two fraction
19 //Output: -1: f1 < f2; 0: f1 = f2; 1: f1 > f2
20 int Compare(Fraction f1, Fraction f2);
```

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

```

1  #include <iostream>
2  using namespace std;
3
4  #include "Fraction.h"
5
6  //input the fraction
7  void Input(Fraction &f) 7
8  {
9      cout << "Numerator: ";
10     cin >> f.num;
11     cout << "Denominator: ";
12     cin >> f.den;
13 }
14
15 //output the fraction
16 void Output(Fraction f) 4
17 {
18     cout << f.num << "/" << f.den;
19 }
20
21 //Add two fraction
22 Fraction Add(Fraction f1, Fraction f2) 7
23 {
24     Fraction r;
25     r.num = f1.num * f2.den + f1.den * f2.num;
26     r.den = f1.den * f2.den;
27     return r;
28 }
29
30 //compare two fraction
31 //Output: -1: f1 < f2; 0: f1 = f2; 1: f1 > f2
32 int Compare(Fraction f1, Fraction f2) 12
33 {
34     int r;
35     int d = f1.num * f2.den - f1.den * f2.num;
36     if(d < 0)
37         r = -1;
38     else if (d > 0)
39         r = 1;
40     else
41         r = 0;
42     return r;
43 }

```

- **main.cpp file** → using the struct and the functions

```

1 #include <iostream>
2 using namespace std;
3
4 #include "Fraction.h"
5
6 void main() {
7     Fraction f1, f2, f3;
8     cout << "The program to add two fraction" << endl;
9     cout << "Input the first fraction: " << endl;
10    Input(f1);
11    cout << "Input the second fraction: " << endl;
12    Input(f2);
13
14    f3 = Add(f1, f2);
15    Output(f1);
16    cout << " + ";
17    Output(f2);
18    cout << " = ";
19    Output(f3);
20 }

```

## 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:

```

1 #pragma once
2
3 #include "Fraction.h"
4
5 //Input
6 void InputFractionList(Fraction l[], int &n);
7 //Output
8 void OutputFractionList(Fraction l[], int n);
9 //find fraction max in the list
10 Fraction FindMax(Fraction l[], int n);

```

- FractionList.cpp file:

```

1  #pragma once
2  #include <iostream>
3  using namespace std;
4  #include "Fraction.h"
5
6  //Input
7  void InputFractionList(Fraction l[], int &n) 10
8  {
9      cout << "The number of the list: ";
10     cin >> n;
11     for (int i = 0; i < n; i++)
12     {
13         cout << "The element " << i << ": " << endl;|
14         Input(l[i]);
15     }
16 }
17 //Output
18 void OuputFractionList(Fraction l[], int n) 9
19 {
20     for (int i = 0; i < n; i++)
21     {
22         cout << "The element " << i << ": ";
23         Output(l[i]);
24         cout << endl;
25     }
26 }
27 //find faction max in the list
28 Fraction FindMax(Fraction l[], int n) 10
29 {
30     Fraction max = l[0];
31     for (int i = 1; i < n; i++)
32     {
33         if(Compare(max, l[i]) == -1)
34             max = l[i];
35     }
36     return max; ↵
37 }

```

- Main.cpp file:

```
1 #include <iostream>
2 using namespace std;
3
4 #include "Fraction.h"
5 #include "FractionList.h"
6
7 void main()
8 {
9     Fraction list[100];
10    int n;
11
12    cout << "Find max in the fraction list" << endl;
13    InputFractionList(list, n);
14
15    cout << "The fraction list" << endl;
16    OutputFractionList(list, n);
17
18    Fraction max = FindMax(list, n);
19    cout << "The max: ";
20    Output(max);
21 }
```

### 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

```

1 #pragma once
2 #include <fstream>
3 using namespace std;
4
5 struct Fraction
6 {
7     int num;//numerator
8     int den;//denominator
9 };
10
11 //input the fraction
12 void Input(Fraction &f);
13
14 //output the fraction
15 void Output(Fraction f);
16
17 //Add two fraction
18 Fraction Add(Fraction f1, Fraction f2);
19
20 //compare two fraction
21 //Output: -1: f1 < f2; 0: f1 = f2; 1: f1 > f2
22 int Compare(Fraction f1, Fraction f2);
23
24 //input from file
25 void InputFile(Fraction &f, ifstream& file);

```

- Fraction.cpp

```

1 #include <iostream>
2 using namespace std;
3
4 #include "Fraction.h"
5
6 //input the fraction
7 void Input(Fraction &f) { .7. }
15 //output the fraction
16 void Output(Fraction f) { .4. }
21 //Add two fraction
22 Fraction Add(Fraction f1, Fraction f2) { .7. }
30 //compare two fraction
31 //Output: -1: f1 < f2; 0: f1 = f2; 1: f1 > f2
32 int Compare(Fraction f1, Fraction f2) { .12. }
44 //input from file
45 void InputFile(Fraction &f, ifstream &file) { 5
46 {
47     file >> f.num;
48     file >> f.den;
49 }

```



- FractionList.h

```

1 #pragma once
2
3 #include "Fraction.h"
4
5 //Input
6 void InputFractionList(Fraction l[], int &n);
7 //Output
8 void OutputFractionList(Fraction l[], int n);
9 //find fraction max in the list
10 Fraction FindMax(Fraction l[], int n);
11 //Input from file
12 void InputFileFractionList(Fraction l[], int &n);

```

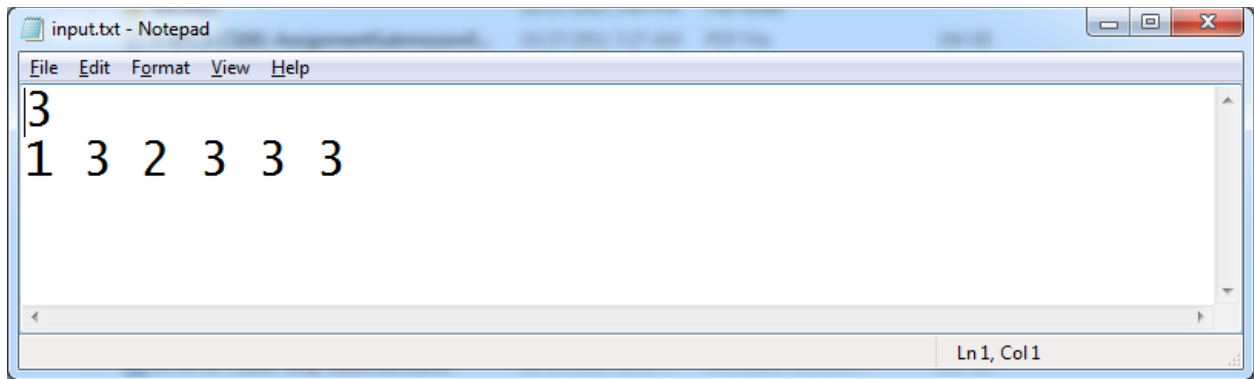
- FractionList.cpp

```

1 #pragma once
2 #include <iostream>
3 #include <fstream>
4 using namespace std;
5 #include "Fraction.h"
6
7 //Input
8 void InputFractionList(Fraction l[], int &n) { 10 ... }
9 //Output
10 void OutputFractionList(Fraction l[], int n) { ... }
11 //find fraction max in the list
12 Fraction FindMax(Fraction l[], int n) { ... }
13 //Input from file
14 void InputFileFractionList(Fraction l[], int &n)
15 {
16     ifstream file;
17     file.open("input.txt");
18     file >> n;
19
20     for (int i = 0; i < n; i++)
21     {
22         InputFile(l[i], file);
23     }
24 }

```

- Input.txt



- Main.cpp

```
1 #include <iostream>
2 using namespace std;
3
4 #include "Fraction.h"
5 #include "FractionList.h"
6
7 void main() {
8     {
9         Fraction list[100];
10        int n;
11
12        cout << "Find max in the fraction list which save in the 'input.txt' file" << endl;
13        InputFileFractionList(list, n);
14
15        cout << "The fraction list" << endl;
16        OutputFractionList(list, n);
17
18        Fraction max = FindMax(list, n);
19        cout << "The max: ";
20        Output(max);
21    }
22 }
```

### 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).