



Course: programming language (CSE115)

Sheet No.:4

Date: 18/12/2018

Due:-/12/2018

Time:2 Hours

Problems:

- 1- Write a C++ program to add two fractions and display the result fraction. Your program will prompt the user to input fraction 1 and fraction 2. The numerator and denominator of each fraction are input separately by space. See the example output below. You will need to use a C++ structure to define a fraction. The structure has two members: numerator and denominator.

Enter fraction 1(numerator denominator): 1 2

Enter fraction 2(numerator denominator): 2 5

Result: 9/10

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

struct Fract{
    int num;
    int den;
};

// prototype of function
Fract sum(Fract, Fract);
```

```

int main(){
    Fract f1, f2, s;

    cout << "Enter fraction1 as follows: 1 2 ";
    cin >> f1.num >> f1.den;

    cout << "Enter fraction2 as follows: 3 4";
    cin >> f2.num >> f2.den;

    s = sum(f1, f2);

    // Display result
    cout << "sum = " << s.num << "/" << s.den << endl;

    return 0;
}
Fract sum(Fract x, Fract y){
    // define a variable which is the output of function
    Fract result;

    result.num = x.num * y.den + x.den * y.num;
    result.den = x.den * y.den;
    // result = {x.num * y.den + x.den * y.num, x.den * y.den}
    return result;
}

```

- 2- Write a C++ program uses structures to implement the employees sort problem according to their birth date. The program must read the number of employees, names, birth date (day/month/year) and salary of each employee.

```
#include <iostream>
#include <string>
using namespace std;
struct SEmployee{
    string name;
    int birth_day;
    int birth_month;
    int birth_year;
    int salary;
};

int main(){
    int n;
    cout << "Please, Enter the number of employees: ";
    cin >> n;
    SEmployee * X = new SEmployee[n];
    SEmployee temp;
    // To read each employee
    for (int i = 0; i < n; i++){
        cout << "Enter the name of employee: ";

        cin >> ws;
        getline(cin, X[i].name);

        cout << "Date of birth as follows (12 11 1995): ";
        cin >> X[i].birth_day >> X[i].birth_month >> X[i].birth_year;

        cout << "Salary: ";
        cin >> X[i].salary;
    }
}
```

```

for (int i = 0; i < n-1; i++){
    for (int j = i + 1; j < n; j++){
        if (X[i].birth_year > X[j].birth_year ||
            ((X[i].birth_year == X[j].birth_year) && (X[i].birth_month > X[j].birth_month)) ||
            ((X[i].birth_year == X[j].birth_year) && (X[i].birth_month == X[j].birth_month) && (X[i].birth_day == X[j].birth_day))){
            temp = X[i];
            X[i] = X[j];
            X[j] = temp;
        }
    }
}

for (int i = 0; i < n; i++){
    cout << i << ": ";
    cout << X[i].name << " " << X[i].birth_day << "/" << X[i].birth_month << "/" << X[i].birth_year << " " << X[i].salary << endl;
}
return 0;
}

```

- 3- Write a C++ program uses nested data structures to implement the employees sort problem according to their birth date. The program must read the number of employees, names, birth date (day/month/year), graduation date (day/month/year) and salary of each employee.

```

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

struct SDate{
    int day;
    int month;
    int year;
};

struct SEmployee{
    string name;

    SDate birth_date;
    SDate graduation_date;

    int salary;
};

int main(){
    int n;
    cout << "Please, Enter the number of employees: ";
    cin >> n;
    SEmployee * X = new SEmployee[n];
    SEmployee temp;
    // To read each employee
    for (int i = 0; i < n; i++){
        cout << "Enter the name of employee: ";

        cin >> ws;
        getline(cin, X[i].name);

        cout << "Date of birth as follows (12 11 1995): ";
        cin >> X[i].birth_date.day >> X[i].birth_date.month >> X[i].birth_date.year;
    }
}

```

```

cout << "Graduation Date as follows (12 11 2005): ";
cin >> X[i].graduation_date.day >> X[i].graduation_date.month >> X[i].graduation_date.year;

cout << "Salary: ";
cin >> X[i].salary;
}

for (int i = 0; i < n - 1; i++){
    for (int j = i + 1; j < n; j++){
        if (X[i].birth_date.year > X[j].birth_date.year ||
            ((X[i].birth_date.year == X[j].birth_date.year) && (X[i].birth_date.month > X[j].birth_date.month)) ||
            ((X[i].birth_date.year == X[j].birth_date.year) && (X[i].birth_date.month == X[j].birth_date.month) && (X[i].birth_date.day == X[j].birth_date.day))){
            temp = X[i];
            X[i] = X[j];
            X[j] = temp;
        }
    }
}

for (int i = 0; i < n; i++){
    cout << "Name: " << X[i].name;
    cout << " " << X[i].birth_date.day << "/" << X[i].birth_date.month << "/" << X[i].birth_date.year;
    cout << " " << X[i].graduation_date.day << "/" << X[i].graduation_date.month << "/" << X[i].graduation_date.year;
    cout << " " << X[i].salary;
    cout << endl;
}
return 0;
}

```

- 4- Write a C++ program uses nested data structures and functions to read two different dates (X,Y) and then comparing two dates. The output f should be:

$$f = \begin{cases} -1 & X > Y \\ 0 & X = Y \\ 1 & X < Y \end{cases}$$

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

struct SDate{
    int day;
    int month;
    int year;
};

struct SDate ReadDate(string dataname){
    struct SDate date;

    cout << "Enter " << dataname << "as follows (12 3 1883): ";
    cin >> date.day >> date.month >> date.year;

    return date;
}

int CompareDates(struct SDate A, struct SDate B){
    if (A.year > B.year || (A.year == B.year && A.month > B.month) || (A.year == B.year && A.month == B.month && A.day > B.day))
        return 1;
    else if (A.year == B.year && A.month == B.month && A.day == B.day)
        return 0;
    else
        return -1;
}
```

```

int main(){
    struct SDate X, Y;

    X = ReadDate("Enter X date ");
    Y = ReadDate("Enter Y date ");

    switch (CompareDates(X, Y)){
    case -1:
        cout << "X date is less than Y date" << endl;
        break;
    case 1:
        cout << "X date is greater than Y date" << endl;
        break;
    case 0:
        cout << "X date is equal to Y date" << endl;
        break;
    }
}

```

[Best Wishes 😊]