BÁO CÁO BÀI TẬP 1

**Môn học: Lập trình hướng đối tượng**

**Tên chủ đề: báo cáo bài tập 1**

*GVHD: Nguyễn Hữu Quyền*

*Ngày báo cáo: 22/03/2023*

**Nhóm: 02**

1. **THÔNG TIN CHUNG:**

*(Liệt kê tất cả các thành viên trong nhóm)*

Lớp: IT002.N28.2

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Họ và tên** | **MSSV** | **Email** |
| 1 | Nguyễn Hải Phong | 22521088 | 22521088@gm.uit.edu.vn |
| 2 | Hồ Trung Kiên | 22520704 | 22520704@gm.uit.edu.vn |

1. **NỘI DUNG THỰC HIỆN:**

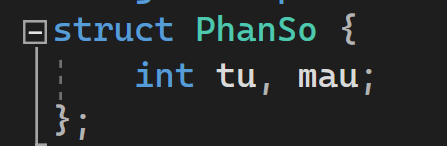
|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Công việc** | **Kết quả tự đánh giá** | **Người đóng góp** |
| 1 | Câu hỏi 01 | 100% | Nguyễn Hải Phong |
| 2 | Câu hỏi 02 | 100% | Nguyễn Hải Phong |
| 3 | Câu hỏi 03 | 100% | Nguyễn Hải Phong |
| 4 | Câu hỏi 04 | 100% | Nguyễn Hải Phong |
| 5 | Câu hỏi 05 | 100% | Nguyễn Hải Phong |
| 6 | Câu hỏi 06 | 100% | Hồ Trung Kiên |
| 7 | Câu hỏi 07 | 100% | Hồ Trung Kiên |
| 8 | Câu hỏi 08 | 100% | Hồ Trung Kiên |

**Phần bên dưới của báo cáo này là tài liệu báo cáo chi tiết của nhóm thực hiện.**

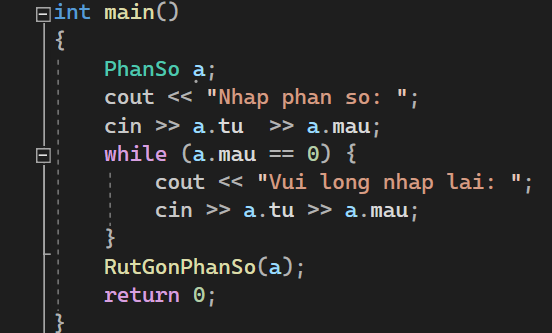
BÁO CÁO CHI TIẾT BÀI THỰC HÀNH 1

## Câu hỏi 01

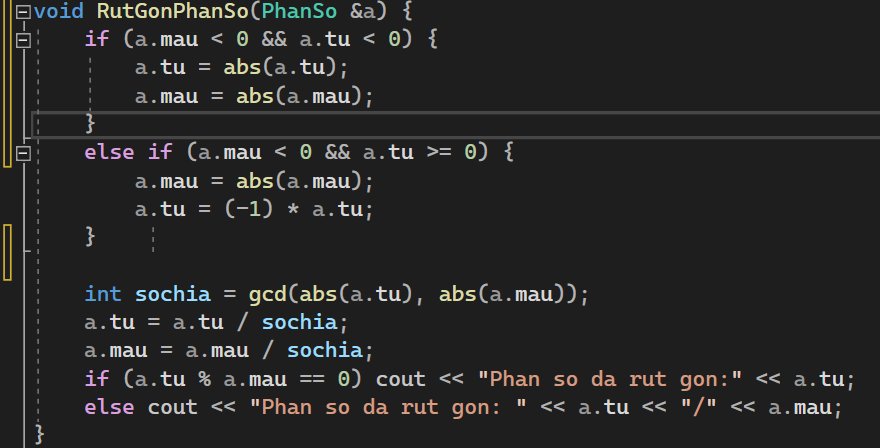
* Mô tả/mục tiêu: Viết chương trình nhập vào một phân số, rút gọn phân số và xuất kết quả.
* Các bước thực hiện/Phương pháp thực hiện:
* Bước 1: Tạo một struct có tên PhanSo bao gồm 2 thành phần là tử số (tu) và mẫu số (mau). Tử số và mẫu số đều có kiểu dữ liệu là integer



* Bước 2: Trong hàm main tạo một biến a có kiểu dữ liệu là struct PhanSo. Sau đó thực hiện bước nhập tử và mẫu của phân số. Đồng thời viết theo câu lệnh kiểm tra xem phân số nhập vào có hợp lệ không bằng cách kiểm tra mẫu số đã khác 0 chưa, nếu chưa thì yêu cầu nhập lại. Nếu đã hợp lệ thì sẽ sử dụng hàm đã được viết trước là hàm RutGonPhanSo để tiến hành rút gọn phân số và xuất ra kết quả.



* Bước 3: Tạo 1 hàm kiểu dữ liệu void dùng để rút gọn phân số được nhập vào và xuất ra kết quả, đặt tên là RutGonPhanSo, trong đó tham số truyền vào là một biến a có kiểu dữ liệu là PhanSo.Trong hàm này, tiến hành đổi dấu cho tử và mẫu nếu như mẫu số là số âm và tử là số dương. Còn nếu tử và mẫu cùng âm thì cho tử và mẫu trở thành số dương. Sau đó sử dụng hàm gcd trong thư viện numeric để tìm ước chung lớn nhất cho tử số và mẫu số và gán cho biến sochia. Sau đó tiến hành chia tử và mẫu số cho biến sochia và tiến hành xuất ra phân số đã được rút gọn. Nếu kết quả có tử chia hết cho mẫu thì chỉ xuất ra tử số của phân số đó

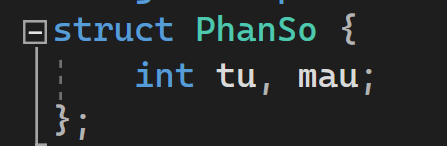


## Câu hỏi 02

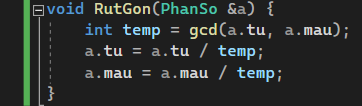
## Mô tả/mục tiêu: Viết chương trình nhập vào hai phân số, tìm phân số lớn nhất và xuất kết quả.

## Các bước thực hiện/ Phương pháp thực hiện:

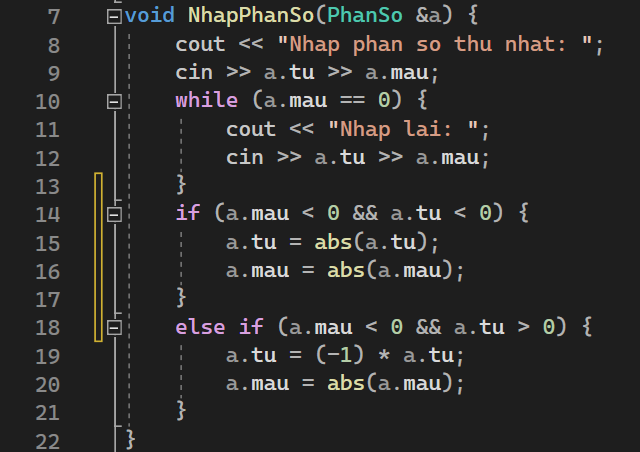
* Bước 1: Tạo một struct có tên PhanSo bao gồm 2 thành phần là tử số (tu) và mẫu số (mau). Tử số và mẫu số đều có kiểu dữ liệu là integer



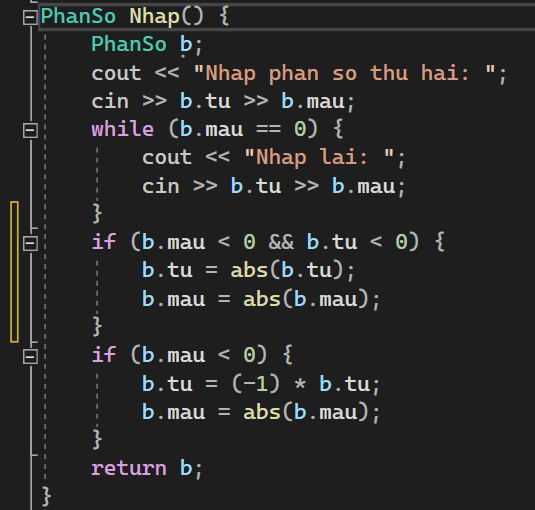
* Bước 2: Tạo ra 1 hàm dùng để rút gọn 2 phân số vừa nhập vào (nếu rút gọn được), là hàm RutGon không có kiểu trả về (void), có tham số truyền vào là tham chiếu của biến a có kiểu dữ liệu struct PhanSo. Trong hàm tạo biến kiểu int temp lưu trữ giá trị của phép tính ước chung lớn nhất của tử và mẫu (hàm gcd có sẵn trong thư viện numeric) và lần lượt chia cả tử và mẫu cho temp.



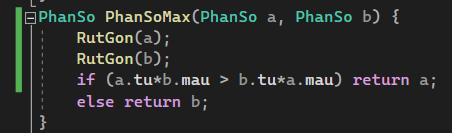
* Bước 3: Tạo ra một hàm dùng để nhập phân số thứ nhất có tên NhapPhanSo, kiểu dữ liệu hàm trả về là void và tham số truyền vào là biến a có kiểu dữ liệu là struct PhanSo. Trong hàm tiến hành việc nhập phân số và kiểm tra xem phân số đó có hợp lệ thôi bằng cách cho một lệnh while để kiểm tra mẫu số có bằng 0. Nếu có thì nhập lại cho đến khi nhập vào một mẫu số khác 0. Sau đó kiểm tra dấu của tử số và mẫu số. Nếu cả tử và mẫu đều là số âm thì đổi dấu cả hai, còn nếu chỉ có mẫu số là số âm thì cho tử là số âm còn mẫu số là số dương.



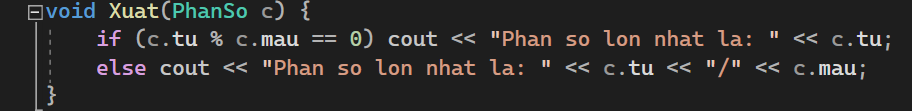
* Bước 3: Tạo ra hàm dùng để nhập vào phân số thứ hai, kiểu dữ liệu trả về là struct PhanSo và không có tham số truyền vào. Trong hàm, tạo một biến b có kiểu dữ liệu struct PhanSo và có chức năng tương tự với hàm NhapPhanSo ở bước 2.



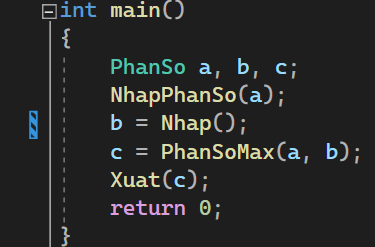
* Bước 4: Tạo tiếp một hàm khác dùng để so sánh hai phân số vừa nhập vào tên PhanSoMax có kiểu dữ liệu trả về là struct PhanSo. Truyền vào hàm là hai biến a và b có kiểu dữ liệu là struct PhanSo. Trong hàm gọi đến hàm RutGon để rút gọn hai phân số nhập vào và thực hiện bước quy đồng với tử số của hai phân số và tiến hành so sánh. Nếu tử số a lớn hơn tử số b thì trả về phân số a và ngược lại.



* Bước 5: Tạo thêm một hàm để xuất ra phân số lớn nhất, đặt tên là Xuat, kiểu dự liệu là void, tham số truyền vào là kết quả của phép so sánh hai phân số có kiểu dữ liệu là struct PhanSo. Trong hàm, nếu tử chia hết mẫu thì xuất ra tử của phân số, ngược lại thì xuất cả tử lẫn mẫu của phân số.



* Bước 6: Trong hàm main ta tạo ba biến a, b, c có cùng kiểu dữ liệu struct PhanSo. Trong đó a sẽ được gắn giá trị của phân số thứ nhất nhập vào, b được gắn giá trị của phân số thứ hai nhập vào và c sẽ được gắn giá trị của phân số lớn nhất trong hai phân số trên. Sau đó sử dụng hàm Xuat để xuất giá trị của phân số kết quả.

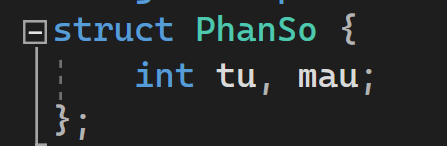


## Câu hỏi 03:

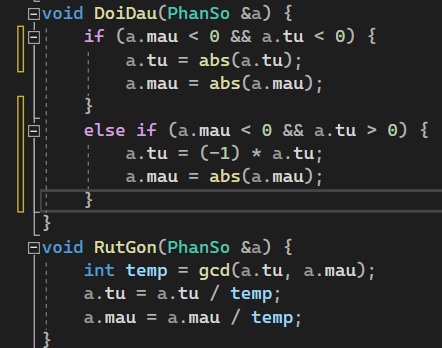
## Mô tả/mục tiêu: Viết chương trình nhập vào hai phân số. Tính tổng, hiệu, tích, thương giữa chúng và xuất kết quả.

## Các bước thực hiện/ Phương pháp thực hiện:

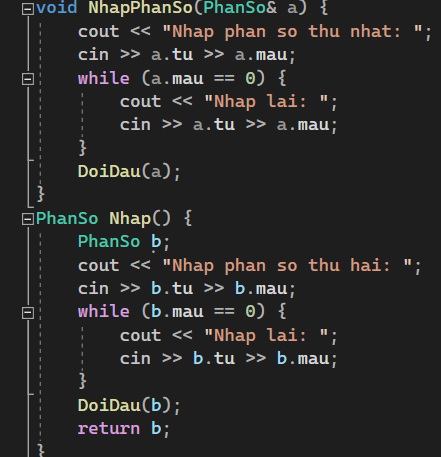
* Bước 1: Tạo một struct có tên PhanSo bao gồm 2 thành phần là tử số (tu) và mẫu số (mau). Tử số và mẫu số đều có kiểu dữ liệu là integer



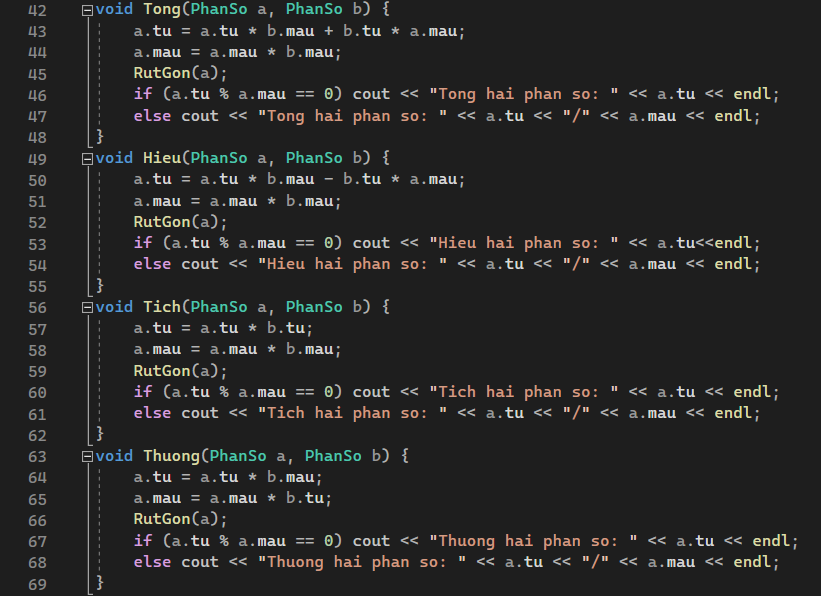
* Bước 2: Tạo hai hàm dùng để chỉnh sửa hai phân số nhập vào. Một hàm tên RutGon không có kiểu trả về (void), có tham số truyền vào là tham chiếu của biến a có dạng struct PhanSo. Bên trong hàm, tạo một biến kiểu số nguyên temp và gắn giá trị của phép ước chung lớn nhất (hàm gcd của thư viện numeric) của tử và mẫu tham số truyền vào. Rồi tiến hành chia cả tử và mẫu cho temp để rút gọn phân số. Hàm còn lại là DoiDau cũng không có kiểu trả về, cũng có tham số truyền vào là tham chiếu của biến a có dạng stuct PhanSo. Trong hàm sẽ kiểm tra xem phân số nhập vào có cả tử và mẫu có ở số âm không. Nếu có thì tiến hành đổi dấu, còn không thì nếu như chỉ có mẫu ở số âm thì cho tử là số âm, còn mẫu số là số dương.



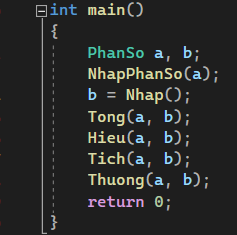
* Bước 3: Tạo hai hàm dùng để lần lượt nhập vào phân số thứ nhất và phân số thứ hai. Trong đó một hàm tên NhapPhanSo không có kiểu dữ liệu trả về (void), có tham số truyền vào là tham chiếu của biến a có dạng struct PhanSo. Trong hàm cho phép nhập tử số và mẫu số và chỉ tiến hành bước nhập lại nếu như có mẫu số nhập vào bằng 0. Sau đó gọi hàm DoiDau ở đổi dấu cho cả tử và mẫu (nếu có số âm). Hàm thứ hai là Nhap có kiểu trả về là struct PhanSo, không cần tham số truyền vào và có chức năng tương tự như hàm NhapPhanSo.



* Bước 4: Lần lượt tạo ra các hàm để thực hiện các phép toán cộng, trừ, nhân, chia cho hai phân số vừa nhập vào. Cả 4 hàm để cùng có kiểu dữ liệu trả về là struct PhanSo, cùng cần hai tham số truyền vào là a và b có cùng kiểu dữ liệu struct PhanSo và đều in ra kết quả của phép tính đó. Trong hàm cộng và trừ, tạo một biến kiểu struct PhanSo tên result và sử dụng quy đồng để thực hiện các phép tính, sau đó lần lượt gán giá trị phép tính cho biến result và gọi đến hàm RutGon để rút gọn kết quả. Còn trong hàm nhân và chia, kết quả sẽ là tích của tử hai phân số chia cho tích của mẫu hai phân số, đối với chia là phép nhân nghịch đảo đối với phân số b và cũng đều gọi đến hàm RutGon để rút gọn kết quả về phân số tối giản.

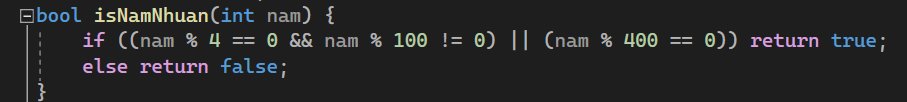


* Bước 5: Tạo hàm main và khai báo hai biến a và b có cùng kiểu dữ liệu là struct PhanSo. Gọi đến hàm NhapPhanSo và hàm Nhap để nhập giá trị phân số a và b. Cuối cùng gọi đến các hàm cộng trừ nhân chia để thực hiện phép tính và đồng thời xuất ra kết quả.

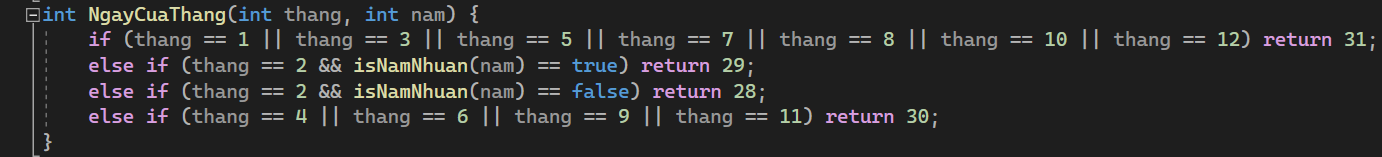


## Câu hỏi 04:

* Mô tả/Mục tiêu: Viết chương trình nhập vào một ngày. Tìm ngày kế tiếp và xuất kết quả
* Các bước thực hiện/ Phương pháp thực hiện:
* Bước 1: Tạo ra 1 hàm để kiểm tra xem năm được nhập vào có phải là năm nhuận hay không. Đó là hàm isNamNhuan có kiểu trả về là bool, có tham số truyền vào là giá trị của năm nhập vào. Nếu như năm đó chia hết cho 4 và không chia hết cho 100 hoặc là năm đó chia hết cho 400 thì trả về là true (đúng), còn nếu không là false (sai). Thuật toán kiểm tra năm nhuận có là do Trái Đất mất 365 ngày 5 giờ 59 phút và 16 giây để xoay quay Mặt Trời, và để khiến cho lịch chúng ta đang xài không bị lệch so với thời gian Trái Đất quay quanh Mặt Trời thì cứ đúng 4 năm Trái Đất sẽ lệch khoảng 23 giờ 57 phút và 4 giây so với lịch của chúng ta, nên phải bù thêm 1 ngày vào tháng 2 để tránh sự chệnh lệch ấy.



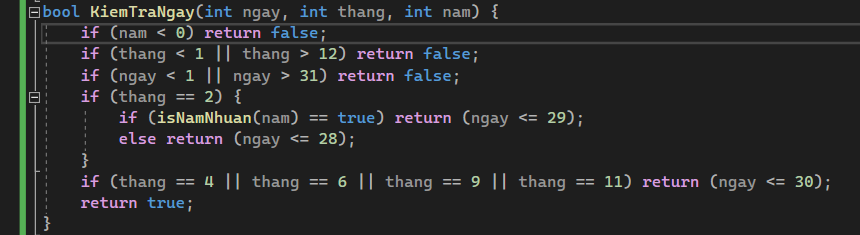
* Bước 2: Tạo ra một hàm để kiểm tra tháng nhập vào có tối đa bao nhiêu ngày, là hàm NgayCuaThang, có kiểu dữ liệu trả về là số nguyên (int), có tham số truyền vào là giá trị của tháng nhập vào và năm nhập vào và sẽ trả về số ngày của tháng đó. Trong hàm sẽ kiểm tra nếu tháng nhập vào có phải là tháng 31 ngày (tháng 1, 3, 5, 7, 8, 10, 12) hay là tháng có 30 ngày (tháng 4, 6, 9, 11). Còn riêng tháng 2 thì ta cần kiểm tra xem năm nhập vào có là năm nhuận không bằng gọi hàm isNamNhuan ở bước 1. Nếu đúng thì tháng có 29 ngày, ngược lại thì là 28 ngày.



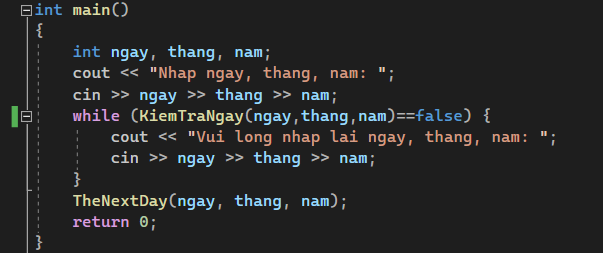
* Bước 3: Tạo ra một hàm để xuất ra được ngày kế tiếp của ngày được nhập vào, là hàm TheNextDay, không có kiểu trả về (void), cần truyền vào tham số là giá trị của ngày tháng năm được nhập vào. Trong hàm so sánh nếu ngày đó chưa phải là ngày cuối tháng thì cộng ngày đó thêm 1. Còn nếu ngày đó mà là ngày cuối tháng nhưng tháng nhập vào chưa phải là cuối năm (tháng 12) thì reset ngày về ngày 1 và tháng đó được cộng thêm 1. Còn nếu như mà là ngày 31 tháng 12 thì thực hiện cộng năm nhập vào thêm 1 và reset ngày và tháng về 1. Riêng đối với tháng 2 thì nếu là nhập vào là 29/2 và là năm nhuận thì ngày tiếp theo sẽ reset về 1 và tháng được cộng thêm 1. Và nếu như nhập vào là 28/2 của năm không nhuận thì cũng sẽ reset ngày về 1 và tháng cộng thêm 1. Cuối cùng xuất ra ngày tháng năm kế tiếp.



* Bước 4: Tạo thêm 1 hàm dùng để kiểm tra ngày tháng năm nhập vào có hợp lệ hay chưa, là hàm KiemTraNgay có kiểu trả về là bool, tham số truyền vào là các biến số nguyên ngày, tháng, năm nhập vào. Lần lượt kiểm tra năm đã hợp lệ chưa ( nếu năm <0 thì trả về false), với ngày (ngày < 1 hoặc > 31 thì trả về false), tháng (tháng < 1 hoặc > 12 thì trả về false). Còn nếu nhập vào là tháng 2 năm nhuận thì so sánh thêm ngày đó <=29 ngày không, năm không nhuận thì kiểm tra ngày đó <=28 ngày chưa. Đồng thời cũng kiểm tra các tháng 30 ngày (gồm tháng 4, 6, 9, 11) thì ngày nhập vào đã <=30 ngày chưa.

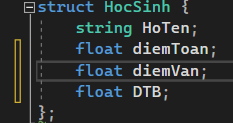


* Bước 5: Khởi tạo hàm main và tạo ra ba biến ngay, thang, nam cùng kiểu dữ liệu số nguyên (int) để nhập vào ngày tháng năm. Và đồng thời gọi đến hàm KiemTraNgay kiểm tra xem ngày tháng năm được nhập vào có là ngày hợp lệ không. Nếu không thì sẽ yêu cầu nhập lại đến khi nhập đúng thì thôi. Sau đó gọi đến hàm TheNextDay để in ra ngày tháng năm kế tiếp.

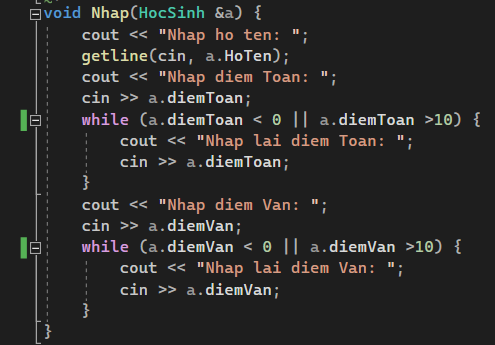


## Câu hỏi 05:

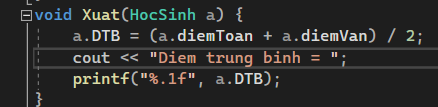
* Mô tả/ Mục tiêu: Viết chương trình nhập họ tên, điểm toán, điểm văn của một học sinh. Tính điểm trung bình và xuất kết quả
* Các bước thực hiện/ Phương pháp thực hiện:
* Bước 1: Tạo ra một struct HocSinh bao gồm các thuộc tính họ tên có kiểu dữ liệu là string, điểm toán và điểm văn đều có kiểu dữ liệu là float (vì điểm có thể là số thập phân)



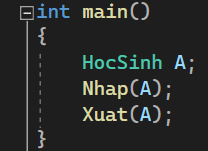
* Bước 2: Tạo ra một hàm dùng để nhập dữ liệu của học sinh bao gồm họ tên, điểm toán và điểm văn, có tên là Nhap, không có kiểu dữ liệu trả về (void) và có tham số truyền vào là tham chiếu của một biến a có kiểu dữ liệu là struct HocSinh. Đồng thởi khi nhập họ tên thì sử dụng hàm getline trong thư viện string (để có thể nhập họ tên có khoảng trắng ở giữa) và cũng như kiểm tra điều kiện nhập của điểm toán và văn. Nếu điểm toán, văn < 0 hoặc > 10 thì tiến hành nhập lại.



* Bước 3: Tạo ra một hàm dùng để tính điểm trung bình và đồng thời xuất ra kết quả, hàm đó là hàm Xuat không có kiểu dữ liệu trả về (void), có tham số truyền vào là giá trị của một biến a có kiểu struct HocSinh. Công thức của điểm trung bình là (Toán + Văn)/2. Sau đó tiến hành xuất ra điểm trung bình, sử dụng lệnh printf(“%.1f”) để in ra kết quả làm tròn đến 1 chữ số thập phân.



* Bước 4: Tạo hàm main và trong đó, khởi tạo một biến A có kiểu dữ liệu struct HocSinh. Gọi đến hàm Nhap để nhập vào họ tên, điểm toán và điểm văn của một học sinh. Sau đó gọi đến hàm Xuat để lần lượt tính điểm trung bình và xuất ra kết quả.



## Câu hỏi 06

## Tài nguyên: lớp PhanSo có dữ liệu tuSo,mauSo và các hàm thành phần cộng, trừ, nhân, chia, xuất, nhập, định giá trị cho phân số.

## Mô tả/mục tiêu: Viết chương trình cho phép nhập vào hai phân số, in ra kết quả các phép toán cộng, trừ, nhân, chia hai phân số kể trên.

## Các bước thực hiện/ Phương pháp thực hiện:

Bước 1: Khai báo thư viện và namespace

Khởi tạo những thư viện cần thiết như iostream và algorithm, thư viện <algorithm> là cần thiết để tính ước chung lớn nhất (greatest common divisor viết tắt là gcd) hoặc có thể tự viết hàm tính gcd.

Text

Description automatically generated

(thuật toán tính gcd bằng đệ quy và giải thuật Euclid)

Tiếp theo có thể khai báo namespace standard (using namespace std) để có thể sử dụng những câu lệnh như std::cin,std::cout,… trở nên ngắn gọn hơn, nhưng nó có thể làm chậm tốc độ chạy của chương trình hay không quá cần thiết thì ta nên không dùng để tối ưu tốc độ.

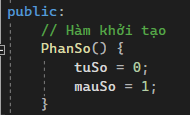
Bước 2: Khởi tạo lớp PhanSo có dữ liệu và các hàm cần thiết

Khởi tạo lớp PhanSo với các thuộc tính và hàm ở những phạm vi truy cập phù hợp. tuSo và mauSo ta sẽ để ở private và các hàm cần thiết thì ta sẽ để ở public

Text

Description automatically generated

Ở hàm khởi tạo (constructor) ta sẽ khai báo biến tuSo và mauSo, mặc định sẽ là 0 và 1 vì đây là một phân số và khi ta khai báo biến lớp PhanSo thì mặc định của phân số đó nên bằng 0, từ đó tử số bằng 0 và mẫu số phải khác 0 để có thể là một phân số hợp lệ (ta có thể đặt mauSo là một số nào đó tuỳ ý nhưng vì đã khai báo biến mauSo là int nên việc đặt mauSo quá cao là không cần thiết và có thể gây interger overflow) và vì tuSo và mauSo là những số có kiểu dữ liệu int nên có thể gây interger overflow nếu nhập biến quá lớn.



Hoặc ta có thể định nghĩa biến tuSo và mauSo ở phần private và không cần khai báo constructor như trên (ví dụ minh hoạ)

Graphical user interface, text, application, chat or text message, website

Description automatically generated

Tiếp đến là một hàm khởi tạo khác (constructor overloading) để truyền đối số, thuận tiện hơn cho việc khởi tạo một biến lớp PhanSo nhanh và thuận tiện hơn cho việc khởi tạo lớp với dữ liệu có sẵn. Ta sẽ không cho phép trường hợp mauSo=0 nên nếu mauSo=0 thì mặc định tuSo=0 và mauSo=1

Text

Description automatically generated

Sau đó là hàm nhập hay có thể gọi là hàm cập nhật. Hàm này sẽ cho ta thay đổi giá trị mặc định của các biến tuSo và mauSo. Trong hàm này ta sẽ chỉ cho phép nhập những giá trị tuSo nguyên và những giá trị mauSo nguyên khác 0. Vì vậy nên nếu mauSo = 0 ta sẽ tiến hành nhập lại phân số.

Text

Description automatically generated

Hàm rút gọn phân số sẽ cần thiết để có thể rút gọn phân số. Ta thực hiện bằng cách lấy gcd của tuSo và mauSo gán cho biến temp và chia lại cho temp để được một phân số rút gọn. Hàm này có thể không có kiểu trả về (void) và truyền tham chiếu (hoặc khởi tạo theo những cách khác với kiểu trả về khác)

Graphical user interface, text, application, chat or text message

Description automatically generated

ví dụ: tuSo=-4, mauSo=-6 thì temp = gcd(-4,-6)=2

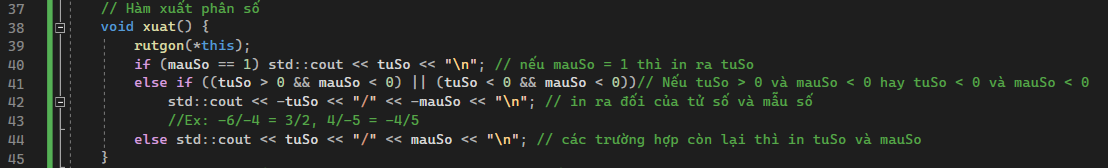
* tuSo=tuSo/2=-4/2=-2
* mauSo=mauSo/2=-6/2=-3

Text

Description automatically generated

(một cách viết hàm rút gọn khác với kiểu trả về PhanSo)

Với hàm xuất phân số ta sẽ tiến hành rút gọn phân số bằng con trỏ this (\*this) trước rồi in ra theo nguyên tắc sau.



Cuối cùng là khai báo những hàm toán tử +, - , \*, / có kiểu trả về là PhanSo theo nguyên tắc cộng trừ nhân chia phân số thông thường, sau đó rút gọn và trả về.

Text

Description automatically generated

Bước 3: Khai báo hàm main

Đầu tiên ta khai báo và nhập hai biến lớp PhanSo là ps1 và ps2 theo 2 cách:

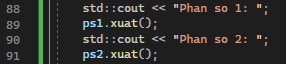
Cách 1: dùng hàm nhập của lớp PhanSo ps1.nhap()

Cách 2: dùng hàm khởi tạo có đối số với hai biến số nguyên a và b có sẵn và đã có giá trị sau khi nhập.

Text

Description automatically generated

Tiếp đến ta sẽ xuất ps1 và ps2



Cuối cùng ta tính toán những phép toán cộng, trừ, nhân, chia hai phân số ps1 và ps2 và xuất ra kết quả các phép toán.

Text

Description automatically generatedText

Description automatically generated(ví dụ demo)

## Câu hỏi 07

## Tài nguyên: Xây dựng lớp biểu diễn khái niệm số phức với hai thành phần dữ liệu thuc, ao và các hàm thành phần xuất, nhập, định giá trị cho số phức, cộng, trừ, nhân, chia hai số phức.

## Mô tả/mục tiêu: Viết chương trình cho phép nhập vào hai số phức, in ra kết quả các phép toán cộng, trừ, nhân, chia hai số phức kể trên.

## Các bước thực hiện/ Phương pháp thực hiện:

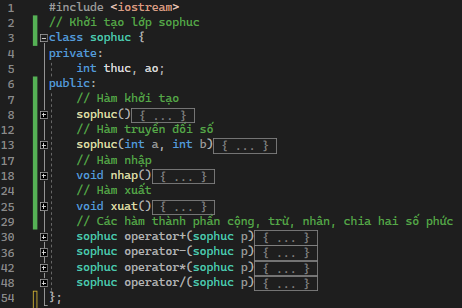
Vì bài này khá tương tự với bài trên nên cũng sẽ gồm 3 bước chính.

Bước 1: khai báo thư viện

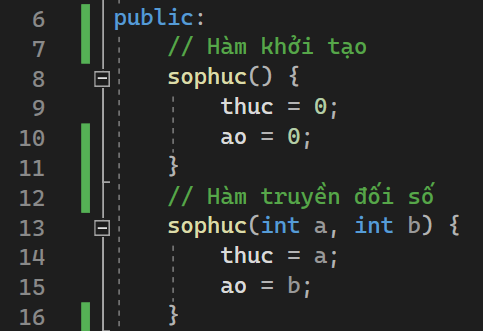


Bước 2: Khai báo lớp sophuc có dữ liệu và các hàm cần thiết

Khởi tạo lớp sophuc với các thuộc tính và hàm ở những phạm vi truy cập phù hợp. Thuốc tính thuc và ao ta sẽ để ở private và các hàm cần thiết thì để ở public.



Đầu tiên là 2 hàm khởi tạo (Constructor), hàm khởi tạo mặc định và hàm truyền đối số với cấu trúc và các giá trị mặc định khá giống với bài trên. Nhưng với giá trị mặc định của số phức thì thực và ảo không cần phải khác 0 nhưng bắt buộc phải là số nguyên nên có thể gán giá trị mặc định của thuc và ao là 0, cùng với đó hàm truyền đối số cũng không cấm các trường hợp = 0 nên không loại trừ trường hợp thuc hay ao = 0.



Tiếp đến là hàm nhập, hàm này sẽ cập nhật lại giá trị của thuc và ao và có thể tuỳ chỉnh cho mục đích sử dụng.

Graphical user interface, text

Description automatically generated

Và tiếp theo là hàm xuất. Ta sẽ xuất thuc và ao theo đúng định dạng số phức

Text

Description automatically generated

Ex: thuc=0, ao=-2 => 0–2i

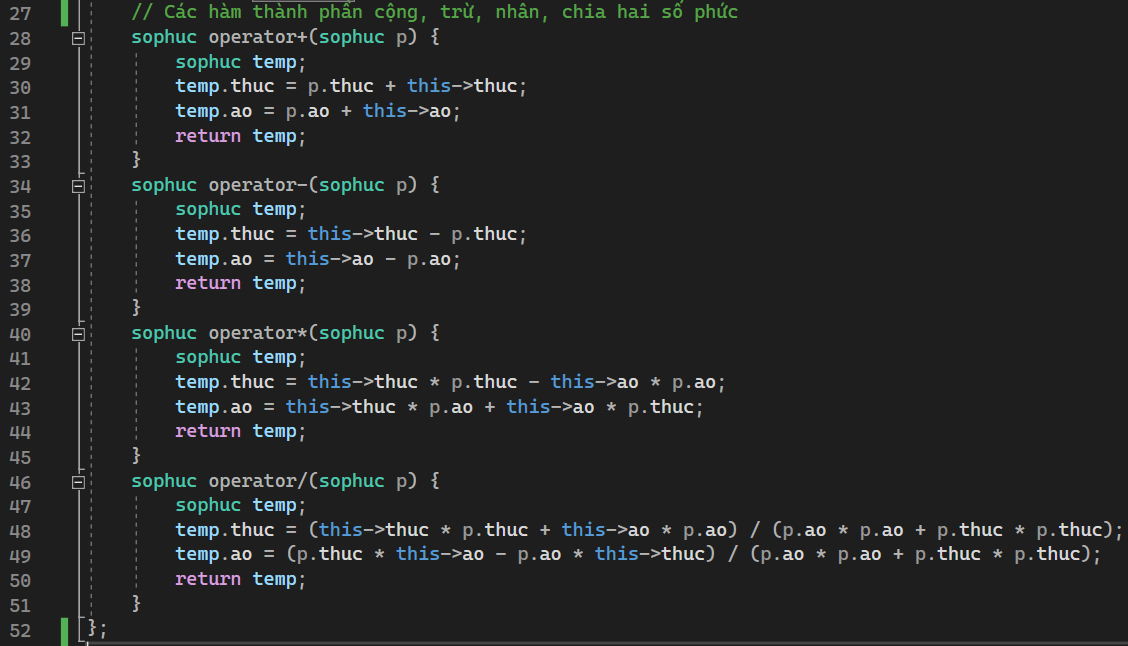
thuc =-2, ao=4 => -2+4i

thuc=2, ao=0 => 2

Text, letter

Description automatically generatedCuối cùng là các hàm toán tử +, -, \*, / hai số phức với các công thức tính như sau:

Với cách tính số phức như trên ta thực hiện các phép toán bằng cách khởi tạo các hàm với kiểu trả về là sophuc và khởi tạo sophuc temp, tính toán các phép toán sau đó gán cho temp và trả về temp. Ta có thể sử dụng con trỏ this để dễ phân biệt hơn.

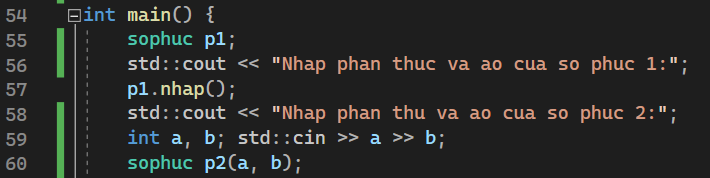


Bước 3: Khai báo hàm main

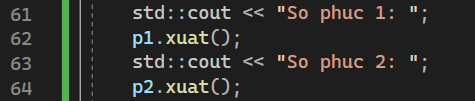
Đầu tiên ta khai báo và nhập hai biến lớp sophuc là p1 và p2 theo 2 cách:

Cách 1: dùng hàm nhập của lớp sophuc p1.nhap()

Cách 2: dùng hàm khởi tạo có đối số với hai biến số nguyên a và b có sẵn và đã có giá trị sau khi nhập.



Tiếp đến ta xuất p1 và p2



Cuối cùng ta tính toán những phép toán cộng, trừ, nhân, chia hai phân số p1 và p2 và xuất ra kết quả các phép toán.

Text

Description automatically generatedText

Description automatically generated (ví dụ demo)

## Câu hỏi 08

## Tài nguyên: Xây dựng lớp Candidate (Thí sinh) gồm các thuộc tính: mã, tên, ngày tháng năm sinh, điểm thi Toán, Văn, Anh và các phương thức cần thiết.

## Mô tả/mục tiêu: Xây dựng lớp TestCandidate để kiểm tra lớp trên:

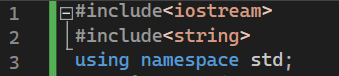
## Nhập vào n thí sinh (n do người dùng nhập).

## In ra thông tin về các thí sinh có tổng điểm lớn hơn 15.

## Các bước thực hiện/ Phương pháp thực hiện:

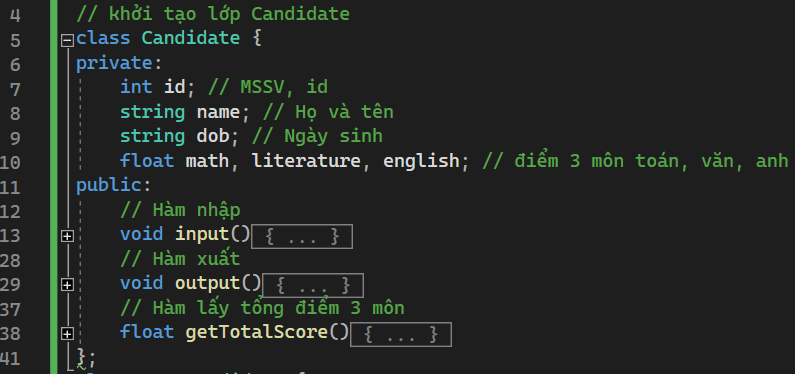
Bước 1: khai báo thư viện và namespace

Khởi tạo những thư viện cần thiết như <iostream> và <string>, thư viện <string> được thêm vào để có thể lưu trữ các biến có nhiều ký tự . Tiếp theo có thể khai báo namespace standard (using namespace std) để có thể sử dụng những câu lệnh như std::cin,std::cout,… trở nên ngắn gọn hơn.



Bước 2: Khai báo lớp Candidate cùng các thuộc tính và các phương thức cần thiết.

Các thuộc tính của Candidate gồm MSSV (int id), Họ tên (string name), ngày sinh (string dob) và điểm 3 môn toán, văn, anh (float math, literature, english) sẽ được để ở private. Còn các phương thức sẽ được định vị ở public bao gồm hàm nhập (input()), hàm xuất (output()) và hàm lấy tổng điểm 3 môn toán, văn và anh (getTotalScore()).



Về hàm nhập ta sẽ hướng dẫn cách nhập từng thuộc tính và tiến hành cập nhật các thuộc tính của lớp. Đương nhiên ta có thể thêm những thuật toán để kiểm tra tính đúng đắn của dữ liệu nhập vào như điểm phải bé hơn hoặc bằng 10 và lớn hơn hoặc bằng 0, ngày sinh có hợp lệ hay không?...

Text

Description automatically generated

Về hàm xuất ta sẽ in lần lượt các thuộc tính của lớp Candidate

Text

Description automatically generated

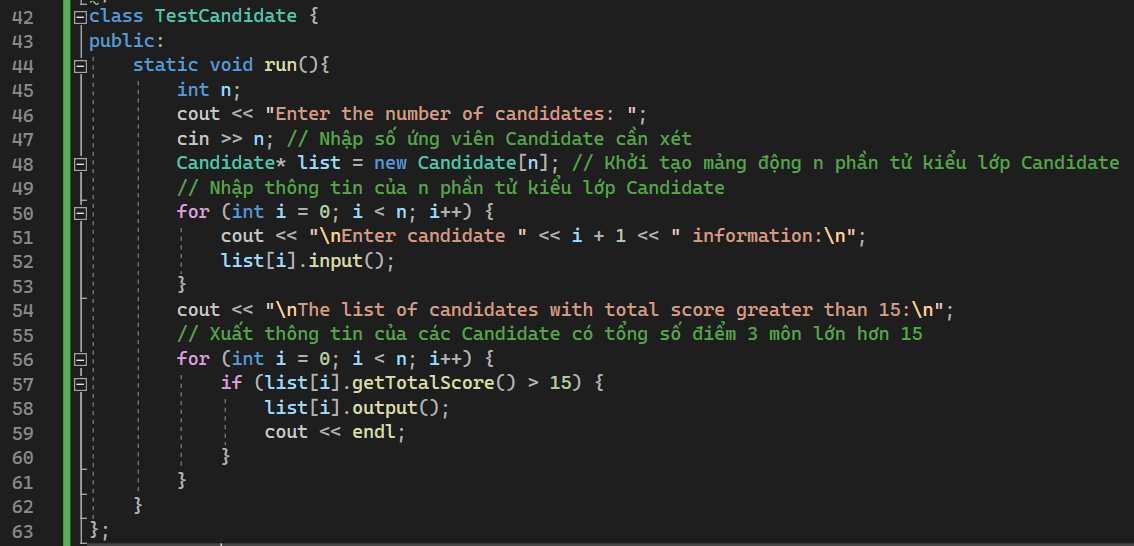
Cuối cùng là hàm lấy tổng 3 điểm toán, văn, anh với kiểu trả về float

Graphical user interface, text, application

Description automatically generated

Bước 3: Khởi tạo lớp TestCandidate

Ta sẽ khởi tạo lớp TestCandidate với một hàm duy nhất là hàm static void run() có vai trò nhập n số thí sinh và kiểm tra tổng số điểm 3 môn của n số thí sinh đó, nếu lớn hơn 15 ta sẽ in ra thông tin của thí sinh.



Bước 4: Khai báo hàm main

Trong hàm main ta sẽ gọi tới hàm run() của lớp TestCandidate để có thể khởi động hàm run() của lớp TestCandidate và kết thúc chương trình sau khi đã chạy chương trình hoàn tất.

Graphical user interface, text

Description automatically generated

Text

Description automatically generated(ví dụ demo)