

5/14/2018

# Lập trình hướng đối tượng

Bài tập thực hành – Phần 2

Hoang Van Tuan

# BÀI TẬP THỰC HÀNH

## LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG – TH03008

### PHẦN 2: LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Để tránh dài dòng, mất thời gian. Tôi tạo 1 khung chương trình chung cho tất cả các bài lập trình trên máy chủ Linux như sau:

```
//Ho ten:  
//MaSV:  
//Lop:  
//De:  
#include<iostream>  
  
using namespace std;  
  
//===chuong trinh chinh===  
int main()  
{  
  
    cout<<endl;  
    return 0;  
}  
  
//===đinh nghĩa ham===
```

Bài 13: (lthdtbai13.cpp): Tính tổng 2 phân số, đưa các phân số ra màn hình ở dạng chưa rút gọn và đã rút gọn.

Yêu cầu:

1. Dùng toán tử nhập >> để nhập vào phân số.
2. Dùng toán tử xuất << để hiện đúng dạng phân số.
3. Dùng toán tử + để tính tổng 2 phân số.

```
#include<iostream>
```

```
#include<math.h>
```

```
using namespace std;
```

```
//khai bao lop
```

```
class PhanSo
```

```
{
```

```
    private:
```

```
        int a,b;
```

```
        int USCLN(int a,int b);
```

```
    public:
```

```
        friend istream& operator>>(istream &cin,PhanSo &ps);
```

```
        friend ostream& operator<<(ostream &cout,PhanSo &ps);
```

```
        void RutGon();
```

```
        PhanSo operator+(const PhanSo &ps2);
```

```
};
```

```
//===chuong trinh chinh===
```

```
int main()
```

```
{
```

```
    PhanSo ps1,ps2,tong;
```

```
//nhap phan so
cout<<"Nhap vao phan so 1,2:\n";
cin>>ps1;
cout<<"Nhap vao phan so 2:\n";
cin>>ps2;

//cong phan so
tong = ps1 + ps2;

//phan so chua rut gon
cout<<"\n\nCac phan so chua rut gon:\n";
cout<<"Phan so 1: ";cout<<ps1;
cout<<"\nPhan so 2: ";cout<<ps2;
cout<<"\nPhan so tong: ";cout<<tong;

//rut gon cac phan so
ps1.RutGon();
ps2.RutGon();
tong.RutGon();

//phan so da rut gon
cout<<"\n\nCac phan so da rut gon:\n";
cout<<"Phan so 1: ";cout<<ps1;
cout<<"\nPhan so 2: ";cout<<ps2;
cout<<"\nPhan so tong: ";cout<<tong;

cout<<endl;
```

```
        return 0;
    }
    //===định nghĩa hàm===
    //Hàm thông thường, ngoại đối tượng
    istream& operator>>(istream &cin, PhanSo &ps)
    {
        do
        {
            cout<<"Nhập tử số a = ";
            cin>>ps.a;
            cout<<"Nhập mẫu số b = ";
            cin>>ps.b;
            if(ps.b==0)
                cout<<"Mẫu số b phải khác 0. Mời bạn nhập lại phân số!\n";
        }
        while(ps.b==0);

        return cin;
    }
    //-----
    ostream& operator<<(ostream &cout, PhanSo &ps)
    {
        if(ps.a*ps.b<0)
            cout<<"-"<<fabs(ps.a)<<"/"<<fabs(ps.b);
        else
            cout<<fabs(ps.a)<<"/"<<fabs(ps.b);
    }
```

```
        return cout;
    }
//-----
void PhanSo::RutGon()
{
    int x=USCLN(a,b);
    a/=x;
    b/=x;
}
//-----
int PhanSo::USCLN(int a,int b)
{
    int r=a%b;

    while(r)
    {
        a=b;
        b=r;
        r=a%b;
    }

    return b;
}
//-----
PhanSo PhanSo::operator+(const PhanSo &ps2)
{
    PhanSo Tg;
```

```
Tg.a = a*ps2.b + b*ps2.a;  
Tg.b = b*ps2.b;  
  
return Tg;  
}  
//-----
```

Bài 14(lthdtbai14.cpp): Cho điểm A(x1,y1) và điểm B(x2,y2). Tính khoảng cách AB.

Yêu cầu:

1. Dùng toán tử nhập >> để nhập vào tọa độ điểm.
2. Dùng toán tử xuất << để hiện thị đúng dạng tọa độ điểm.
3. Dùng toán tử - để tính khoảng cách giữa 2 điểm.
4. Có thể khởi tạo mặc định tọa độ điểm là (0, 0).
5. Có thể khởi tạo mặc định do người dùng nhập vào.

```
#include<iostream>
```

```
#include<math.h>
```

```
using namespace std;
```

```
//khai bao lop
```

```
class Diem
```

```
{
```

```
private:
```

```
float x,y;
```

```
public:
```

```
Diem();
```

```
Diem(float x1, float y1);
```

```
friend istream& operator>>(istream &cin, Diem &D);
friend ostream& operator<<(ostream &cout, Diem &D);

float operator-(const Diem &B);
};

//===chuong trinh chinh===
int main()
{
    Diem A,B;

    cout<<"Nhap vao toa do diem A:\n";
    cin>>A;
    cout<<"Nhap vao toa do diem B:\n";
    cin>>B;

    cout<<"\n\nToa do diem A vua nhap la: "; cout<<A;
    cout<<"\n\nToa do diem B vua nhap la: "; cout<<B;

    cout<<"\n\nKhoang cach giua 2 diem A va B la: "; cout<<A-B;

    cout<<endl;
    return 0;
}

//===dinh nghia ham===
istream& operator>>(istream &cin, Diem &D)
```



```
{  
    cout<<"\tNhap vao hoành do: "; cin>>D.x;  
    cout<<"\tNhap vao tung do: "; cin>>D.y;  
  
    return cin;  
}  
//-----  
ostream& operator<<(ostream &cout, Diem &D)  
{  
    cout<<"("<<D.x<<" , "<<D.y<<")";  
  
    return cout;  
}  
//-----  
float Diem::operator-(const Diem &B)  
{  
    return sqrt((B.x-x)*(B.x-x)+(B.y-y)*(B.y-y));  
}  
//-----  
Diem::Diem():x(0),y(0)  
{  
  
}  
//-----  
Diem::Diem(float x1, float y1):x(x1),y(y1)  
{
```

```
}
```

Bài 15 (lthdtbai15.cpp): Cho ma trận nguyên Amxn. Tính tổng và trung bình cộng các phần tử của ma trận. Đưa ra ma trận chuyển vị của A.

```
#include<stdio.h>
```

```
#include<iostream>
```

```
#include<fstream>
```

```
using namespace std;
```

```
//khai bao lop
```

```
class MaTran
```

```
{
```

```
private:
```

```
    enum{SIZE=30};
```

```
    int a[SIZE][SIZE];
```

```
    int m,n;
```

```
public:
```

```
    friend istream& operator>>(istream &cin,MaTran &mt);
```

```
    friend ostream& operator<<(ostream &cout,MaTran &mt);
```

```
    friend ifstream& operator>>(ifstream &fin,MaTran &mt);
```

```
    MaTran();
```

```
    MaTran(int x,int y);
```

```
    void ChuyenVi();
```

```
    int TinhTong();
```

```
    float TinhTBC();
```

```
};
```

```
//===chuong trinh chinh===  
int main()  
{  
    MaTran a;  
  
    ifstream fin("matran.txt");  
  
    fin>>a;  
  
    cout<<"Ma tran doc vao tu tep la:\n";  
    cout<<a;  
  
    cout<<"\nMa Tran chuyen vi la:\n";  
    a.ChuyenVi();  
  
    cout<<"\nTong cac phan tu la: "<<a.TinhTong();  
    printf("\nTrung binh cong la: %0.2f",a.TinhTBC());  
  
    cout<<endl;  
    return 0;  
}  
  
//===dinh nghĩa hàm===  
istream& operator>>(istream &cin, MaTran &mt)  
{  
    for(int i=0; i<mt.m; i++)  
        for(int j=0; j<mt.n; j++)
```

```
{
    printf("a[%d][%d] = ",i+1,j+1);
    cin>>mt.a[i][j];
}
return cin;
}
//-----

ostream& operator<<(ostream &cout, MaTran &mt)
{
    for(int i=0; i<mt.m; i++)
    {
        for(int j=0; j<mt.n; j++)
            printf("%-5d", mt.a[i][j]);
        printf("\n");
    }
    return cout;
}
//-----

ifstream& operator>>(ifstream &fin, MaTran &mt)
{
    fin>>mt.m>>mt.n;
    for(int i=0; i<mt.m; i++)
        for(int j=0; j<mt.n; j++)
            fin>>mt.a[i][j];
    return fin;
}
//-----
```

```
MaTran::MaTran():m(5),n(5)
```

```
{
```

```
}
```

```
//-----
```

```
MaTran::MaTran(int x,int y)
```

```
{
```

```
    if(x>SIZE || y>SIZE)
```

```
    {
```

```
        cout<<"So hang va so cot phai <= "<<SIZE;
```

```
        cout<<"\nSo hang va so cot dua ve mac dinh!";
```

```
        m=5;
```

```
        n=5;
```

```
    }
```

```
    else
```

```
    {
```

```
        m=x;
```

```
        n=y;
```

```
    }
```

```
}
```

```
//-----
```

```
void MaTran::ChuyenVi()
```

```
{
```

```
    for(int j=0;j<n;j++)
```

```
    {
```

```
        for(int i=0;i<m;i++)
```

```
            printf("%-5d",a[i][j]);
```

```
        printf("\n");
    }
}
//-----

int MaTran::TinhTong()
{
    int s=0;
    for(int i=0;i<m;i++)
        for(int j=0;j<n;j++)
            s+=a[i][j];
    return s;
}
//-----

float MaTran::TinhTBC()
{
    return (float)TinhTong()/m/n;
}
//-----
```

Bài 16 (lthdtbai16.cpp): Nhập vào một danh sách n mặt hàng; mỗi mặt hàng có tên hàng, số lượng, đơn giá.

Tính tổng tiền của n mặt hàng.

(Viết lại chương trình khi sử dụng cấp phát mảng đối tượng động).

```
#include<stdio.h>
```

```
#include<iostream>
```

```
using namespace std;
```

```
//khai bao lop
class MatHang
{
private:
    char Ten[31];
    int SoLuong;
    float DonGia;

public:
    void Nhap();
    void Hien();
    float TinhTien();
};

//===chuong trinh chinh===
int main()
{
    MatHang dsMH[10];
    int n,i;
    float TongTien=0;

    cout<<"Nhap vao so luong mat hang: "; cin>>n;

    cout<<"Nhap thong tin cho tung mat hang:\n";
    for(i=0;i<n;i++)
    {
        printf("Nhap mat hang thu %d:\n",i+1);
```

```
        dsMH[i].Nhap();
    }

    cout<<"\n\nThong tin cua cac mat hang da nhap la:\n";
    for(i=0;i<n;i++)
    {
        printf("Mat hang thu %d:\n",i+1);
        dsMH[i].Hien();
        printf("\n");
    }

    for(i=0;i<n;i++)
        TongTien+=dsMH[i].TinhTien();
    cout<<"\n\nTinh tong tien cua "<<n<<" mat hang da nhap la: "<<TongTien;

    cout<<endl;
    return 0;
}

//===định nghĩa hàm===
void MatHang::Nhap()
{
    cout<<"\tNhap vao mat hang: "; scanf(" "); cin.get(Ten,sizeof(Ten));
    cout<<"\tNhap vao so luong: "; cin>>SoLuong;
    cout<<"\tNhap vao don gia: "; cin>>DonGia;
}

//-----

void MatHang::Hien()
```



```
{  
    cout<<"\tTen mat hang: "<<Ten;  
    cout<<"\tSo luong: "<<SoLuong;  
    cout<<"\tDon gia: "<<DonGia;  
    cout<<"\t=>Thanh tien: "<<TinhTien();  
}  
//-----  
float MatHang::TinhTien()  
{  
    return SoLuong*DonGia;  
}  
//-----
```

Bài 17 (lthdtbai17.cpp): Nhập vào chuỗi ký tự số nhị phân có tối đa 16 bit.

Tính và đưa ra giá trị của số nhị phân đó. Đưa ra màn hình số hex tương ứng với số nhị phân.

Bài 18 (lthdtbai18.cpp): Viết chương trình nhập vào một thời gian có giờ và phút.

Tính và đưa ra màn hình thời gian sau n phút nhập vào từ bàn phím.

```
#include<stdio.h>  
#include<iostream>  
  
using namespace std;
```

```
//khai bao lop  
class ThoiGian  
{  
private:
```

```
int Gio,Phut;

public:
    void Nhap();
    void Hien();
    void TinhTG(int x);
};

//===chuong trinh chinh===
int main()
{
    ThoiGian T;
    int x;

    cout<<"Nhap vao thoi gian ban dau:\n";
    T.Nhap();

    cout<<"Nhap vao so phut luc sau: "; cin>>x;

    T.TinhTG(x);

    cout<<"Dua ra thoi gian sau do: ";
    T.Hien();

    cout<<endl;
    return 0;
}
```

```
//===định nghĩa ham===
```

```
void ThoiGian::Nhap()
```

```
{
```

```
do
```

```
{
```

```
    cout<<"\tNhap vao gio: "; cin>>Gio;
```

```
    if(Gio<0 || Gio>=24)
```

```
        cout<<"Nhap sai gio. Hay nhap lai!";
```

```
    }
```

```
while(Gio<0 || Gio>=24);
```

```
do
```

```
{
```

```
    cout<<"\tNhap vao phut: "; cin>>Phut;
```

```
    if(Phut<0 || Phut>=60)
```

```
        cout<<"Nhap sai phut. Hay nhap lai!";
```

```
    }
```

```
while(Phut<0 || Phut>=60);
```

```
}
```

```
//-----
```

```
void ThoiGian::Hien()
```

```
{
```

```
    cout<<"Gio-phut: "<<Gio<<": "<<Phut;
```

```
}
```

```
//-----
```

```
void ThoiGian::TinhTG(int x)
```

```
{
```

```
    Phut+=x;
```

```
    if(Phut>=60)
```

```
    {
```

```
        Gio+=Phut/60;
```

```
        Phut%=60;
```

```
    }
```

```
    while(Gio>=24)
```

```
        Gio-=24;
```

```
}
```

```
//-----
```

Bài 19(lthdtbai19.cpp): Viết chương trình nhập vào n số phức. Đưa các số phức đã nhập ra màn hình.

Yêu cầu trong chương trình phải tạo đối tượng động.

```
#include<stdio.h>
```

```
#include<iostream>
```

```
using namespace std;
```

```
//khai bao lop
```

```
class SoPhuc
```

```
{
```

```
private:
```

```
    float a,b;
```

public:

void Nhap();

void Hien();

};

//===chương trình chính===

int main()

{

SoPhuc \*p;

int n,i;

cout<<"Nhập số lượng số phức: "; cin>>n;

p=new SoPhuc[n];

cout<<"Nhập vào danh sách số phức:\n";

for(i=0;i<n;i++)

{

printf("Số phức thứ %d:\n",i+1);

p[i].Nhap();

}

cout<<"Các số phức đã nhập là:\n";

for(i=0;i<n;i++)

{

p[i].Hien();

```
        printf("\n");
    }

    delete [] p;

    cout<<endl;
    return 0;
}

//===định nghĩa hàm===
void SoPhuc::Nhap()
{
    cout<<"\tNhap vao phan thuc: "; cin>>a;
    cout<<"\tNhap vao phan ao: "; cin>>b;
}

//-----

void SoPhuc::Hien()
{
    if(b<0)
        cout<<a<<"-"<<b<<"i";
    else
        cout<<a<<"+"<<b<<"i";
}

//-----
```

Bài 20(lthdtbai20.cpp): Viết chương trình nhập vào danh sách n sinh viên, mỗi sinh viên có thông tin về mã sinh viên, tên và điểm tbc.

Mã SV là các số nguyên được lấy tự động có giá trị từ 11 trở đi.

Đưa ra màn hình số lượng (không dùng n) và danh sách sinh viên đã nhập.

Yêu cầu trong chương trình có sử dụng biến tĩnh và hàm tĩnh; sử dụng mảng đối tượng động.

```
#include<stdio.h>
```

```
#include<iostream>
```

```
using namespace std;
```

```
//khai bao lop
```

```
class SinhVien
```

```
{
```

```
private:
```

```
    char Ten[31];
```

```
    int MaSV;
```

```
    float DiemTBC;
```

```
    static int STT;
```

```
public:
```

```
    void Nhap();
```

```
    void Hien();
```

```
    static int LaySl(); //tra ve so luong sinh vien da nhap
```

```
};
```

```
//định nghĩa biến tĩnh
```

```
int SinhVien::STT=10;
```

```
//===chương trình chính===
```

```
int main()
```

```
{  
    SinhVien *p;  
    int n,i=-1;  
    char traloi;  
  
    p=new SinhVien[20];  
  
    do  
    {  
        p[++i].Nhap();  
        cout<<"Co nhap tiep nua hay khong (c/k): ";  
        cin>>traloi;  
    }  
    while(traloi=='c' || traloi=='C');  
  
    cout<<"Danh sach sinh vien da nhap:\n";  
    n=SinhVien::LaySl(); //goi ham thanh vien tinh  
    for(i=0;i<n;i++)  
    {  
        p[i].Hien();  
        printf("\n");  
    }  
  
    delete [] p;  
  
    cout<<endl;  
    return 0;
```



```
}  
  
//===định nghĩa ham===  
void SinhVien::Nhap()  
{  
    //tao masv tu dong  
    MaSV=++STT;  
  
    cout<<"\tNhap vao ten sinh vien: "; scanf(" "); cin.get(Ten,sizeof(Ten));  
    cout<<"\tNhap vao diem TBC: "; cin>>DiemTBC;  
}  
  
//-----  
void SinhVien::Hien()  
{  
    cout<<"\tMaSV: "<<MaSV;  
    cout<<"\n\tTen: "<<Ten;  
    cout<<"\n\tDiem TBC: "<<DiemTBC;  
}  
  
//-----  
int SinhVien::LayStt()  
{  
    return STT-10;  
}  
  
//-----
```

Bài 21 (lthdtbai21.cpp): Nhập vào một số nguyên dương, đưa ra số nhị phân tương ứng.

Y/c trong chương trình có sử dụng ngăn xếp để chuyển từ số thập phân sang nhị phân.

Review Stack:

1. Phép toán 1: Bỏ sung 1 phần tử vào Stack:
  - Vào: Phần tử  $x$ , ngăn xếp  $(S, T)$ .
  - Ra: Không có.

{ Thủ tục này bỏ sung phần tử dữ liệu  $x$  vào ngăn xếp được lưu trữ bởi vector  $S$  có kích thước là  $n$ , có chỉ số đỉnh là  $T$  }

**Procedure Push( $S, T, x$ )****1. Kiểm tra ngăn xếp đầy?****If  $T=n$  then****Begin****Write('Ngăn xếp đầy!');****Return;****End;****2. Tăng  $T$  lên 1 đơn vị.** **$T:=T+1;$** **3. Bỏ sung phần tử mới vào ngăn xếp.** **$S[T]:=x;$** **Return**

2. Phép toán 2: Loại bỏ 1 phần tử ra khỏi Stack:
  - Vào: Ngăn xếp  $(S, T)$ .
  - Ra: Giá trị của phần tử loại bỏ (đỉnh).

{ Hàm này thực hiện việc loại bỏ phần tử ở đỉnh của ngăn xếp  $(S, T)$  và trả về phần tử này }

**Function Pop( $S, T$ )****1. Kiểm tra ngăn xếp rỗng.****If  $T=0$  then****Begin****Write('Ngăn xếp rỗng!');****Return;****End;****2. Giữ lại phần tử đỉnh.** **$Tg:=S[T];$** **3. Giảm  $T$  xuống 1 đơn vị.** **$T:=T-1;$** **4. Đưa phần tử ra ngoài.** **$Pop:=Tg;$** **Return**

## 3. Phép toán 3: Hàm kiểm tra ngăn xếp rỗng:

- Vào: Ngăn xếp (S, T).
- Ra: Trả về FALSE nếu ngăn xếp không rỗng. Trả về TRUE nếu ngăn xếp rỗng.  
{Hàm này thực hiện kiểm tra ngăn xếp đã rỗng hay chưa. Nếu ngăn xếp rỗng thì trả về TRUE. Ngược lại, trả về FALSE}

**Function Empty(S, T)**

**If T=n then Empty:=TRUE**

**Else Empty:=FALSE;**

**Return**

## 4. Ứng dụng:

- Vào: n.
- Ra: Số nhị phân.

**Procedure ChuyenDoi(n)**

**+ While n ≠ 0 do**

**Begin**

- **r:=n mod 2;**
- **Call Push(S, T, r);**
- **n:=n div 2;**

**End;**

**+ while S ≠ NULL do**

**begin**

- **r:=Pop(S, T);**
- **write(r);**

**end;**

**Return**

```
#include<stdio.h>
```

```
#include<iostream>
```

```
using namespace std;
```

```
//khai bao lop
```

```
class Stack
{
private:
    enum{SIZE=32};
    int S[SIZE];
    int T;

public:
    Stack();
    void Push(int x);
    int Pop();
    int IsEmpty();
};

//===chuong trinh chinh===
int main()
{
    int thuong,n;
    Stack S;

    cout<<"Nhap vao so nguyen duong n = ";
    cin>>n;

    thuong=n;
    while(thuong)
    {
        S.Push(thuong%2);
```

```
        thuong/=2;
    }

    cout<<"So nhi phan tuong ung "<<n<<" la: ";
    while(!S.IsEmpty())
        cout<<S.Pop();

    cout<<endl;
    return 0;
}

//===định nghĩa ham===
Stack::Stack():T(-1)
{

}

//-----
void Stack::Push(int x)
{
    //buoc 1: kiem tra ngan xep day
    if(T==SIZE-1)
    {
        cout<<"Ngan xep da day!";
        return;
    }

    //buoc 2: bo sung phan tu du lieu x vao dinh cua ngan xep
    S[++T]=x;
```

```
}  
//-----  
int Stack::Pop()  
{  
    //buoc 1:kiem tra ngan xep rong  
    if(T==-1)  
    {  
        cout<<"Ngan xep rong!";  
        return -1;  
    }  
  
    //buoc 2: tra ve phan tu dinh va thay doi T  
    return S[T--];  
}  
//-----  
int Stack::IsEmpty()  
{  
    if(T==-1)  
        return 1;  
    return 0;  
}  
//-----
```

Bài 22 (lthdtbai22.cpp): Nhập vào một số nguyên dương, đưa ra số hex tương ứng.

Y/c trong chương trình có sử dụng ngăn xếp để chuyển từ số thập phân sang hex.

Ngăn xếp sử dụng cấu trúc lưu trữ phân tán.

Review Stack:

1. Phép toán 1: Thủ tục bổ sung 1 phần tử vào Stack.
  - Vào: T, x.
  - Ra: không có.

### Procedure Push(T, x)

1. Kiểm tra ngăn sắp đầy?
  - +  $N \leq AVAIL$ ;
  - +  $Infor(N) := x$ ;
  - +  $Link(N) := \emptyset$ ;
2. Bổ sung nút mới vào ngăn xếp
  - +  $Link(N) := T$ ;
  - +  $T := N$ ;

### Return

2. Phép toán 2: Hàm loại bỏ phần tử khỏi Stack.
  - Vào: T.
  - Ra: Giá trị phần tử loại bỏ.

{Hàm này thực hiện việc loại bỏ phần tử ở đỉnh ngăn xếp lưu trữ bằng danh sách liên kết đơn T và trả về phần tử này}

### Function Pop(T)

1. Kiểm tra ngăn xếp rỗng?
  - If  $T = \emptyset$  then
  - Begin
  - Write('Ngăn xếp rỗng!');
  - Return;
  - End;
2. Giữ lại phần tử đỉnh.
  - $Tg := Infor(T)$ ;
3. Chuyển con trỏ và hủy bỏ nút lấy ra.
  - +  $P := T$ ;
  - +  $T := Link(T)$ ;
  - +  $P \Rightarrow AVAIL$ ;
4. Trả về phần tử nút loại bỏ.
  - $Pop := Tg$ ;

### Return

```
#include<stdio.h>
```

```
#include<iostream>
```

```
using namespace std;
```

```
//khai bao lop
```

```
class Stack
```

```
{
```

```
private:
```

```
    struct Node
```

```
    {
```

```
        int Infor;
```

```
        Node *Link;
```

```
    } *T;
```

```
public:
```

```
    Stack();
```

```
    ~Stack();
```

```
    Stack(const Stack &S);
```

```
    void Push(int x);
```

```
    int Pop();
```

```
    int IsEmpty();
```

```
    Stack operator=(const Stack &s2);
```

```
};
```

```
//===chuong trinh chinh===
```

```
int main()
```

```
{
```

```
    Stack S; //Tao 1 doi tuong ngan xep
```

```
    int n,Thuong,Du;
```



```
cout<<"Nhập vào 1 số nguyên dương: ";
cin>>n;

//Chuyển sang nhị phân
Thuong=n;
while(Thuong)
{
    S.Push(Thuong%16);
    Thuong/=16;
}

//Dưa ra
cout<<"Số nhị phân của "<<n<<" là: ";
while(!S.IsEmpty())
{
    Du=S.Pop();
    if(Du<10)
        cout<<Du;
    else
        printf("%c",Du+55);
}

cout<<endl;
return 0;
}
```

```
//===định nghĩa ham===
```

```
Stack::Stack():T(NULL)
```

```
{
```

```
}
```

```
//-----
```

```
Stack::~~Stack()
```

```
{
```

```
    Node *P;
```

```
    while(T)
```

```
    {
```

```
        P=T;
```

```
        T=T->Link;
```

```
        delete P;
```

```
    }
```

```
}
```

```
//-----
```

```
Stack::Stack(const Stack &S):T(NULL)
```

```
{
```

```
    Node *P=S.T; //dung de duyet qua tat ca cac nut cua ngan xep da co
```

```
    Node *N,*Last; //tao cac nut trong ngan xep moi
```

```
    while(P)
```

```
    {
```

```
        //tao nut moi cho ngan xep moi va sap chep phan tu du lieu sang
```

```
        N=new Node;
```

```
N->Infor=P->Infor;
```

```
N->Link=NULL;
```

```
//noi nut moi vao ngan xep moi
```

```
if(T==NULL)
```

```
{
```

```
    T=N;
```

```
    Last=N;
```

```
}
```

```
else
```

```
{
```

```
    Last->Link=N;
```

```
    Last=N;
```

```
}
```

```
//xet cac nut tiep theo
```

```
P=P->Link;
```

```
}
```

```
}
```

```
//-----
```

```
void Stack::Push(int x)
```

```
{
```

```
    //buoc 1: tao nut moi
```

```
    Node *N=new Node;
```

```
    N->Infor=x;
```

```
    N->Link=NULL;
```

```
//buoc 2: noi nut moi vao tren nut T
```

```
N->Link=T;
```

```
//buoc 3: cho T tro sang nut moi
```

```
T=N;
```

```
}
```

```
//-----
```

```
int Stack::Pop()
```

```
{
```

```
//buoc 1:kiem tra ngan xep rong
```

```
if(T==NULL)
```

```
{
```

```
    cout<<"Ngan xep rong!";
```

```
    return -1;
```

```
}
```

```
//buoc 2: giu lai phan tu dinh
```

```
int Tg=T->Infor;
```

```
//buoc 3: chuyen con tro va huy bo nut lay ra
```

```
Node *P=T;
```

```
T=T->Link;
```

```
delete P;
```

```
//buoc 4: tra ve phan tu nut loai bo
```

```
return Tg;
```

```
}
```

```
//-----
```

```
int Stack::IsEmpty()
```

```
{  
    if(T==NULL)  
        return 1;  
    return 0;  
}
```

```
//-----
```

```
Stack Stack::operator=(const Stack &s2)
```

```
{  
    //Huy ngan xep ben trai  
    Node *P;  
  
    while(T)  
    {  
        P=T;  
        T=T->Link;  
        delete P; //Huy nut  
    }
```

```
//Tao ngan xep ben trai giong ngan xep ben phai
```

```
Node *N,*Last; //Tao cac nut trong ngan xep moi
```

```
P=s2.T; //Dung de duyett qua cac nut cua ngan xep da co
```

```
while(P)
```

```
{
```

```
    //Tao nut moi cho NXep moi va copy phan tu du lieu sang
```

```
N = new Node;
N->Infor = P->Infor;
N->Link = NULL;

//Noi nut moi vao NXep moi
if(T==NULL)
{
    T=N;
    Last=N;
}
else
{
    Last->Link=N;
    Last=N;
}

//Xet nut tiep theo
P=P->Link;
}
}
//-----
```

Bài 23(lthdtbai22.cpp): Nhập vào một dãy số nguyên. Lưu dãy số trong danh sách liên kết đơn.

Đưa dãy số đã nhập ra màn hình theo thứ tự đã nhập. Tìm xem trong dãy số có phần tử nào bằng x không. Xóa một phần tử có giá trị bằng x.

```
#include<iostream>
```

```
#include<stdio.h>
```

```
using namespace std;
```

```
//khai bao lop
```

```
class DSLKD
```

```
{
```

```
private:
```

```
    struct Node
```

```
    {
```

```
        int Infor;
```

```
        Node *Link;
```

```
    } *F;
```

```
public:
```

```
    DSLKD();
```

```
    ~DSLKD();
```

```
    void Insert(int x);
```

```
    void Display();
```

```
    int Search(int x);
```

```
    void Delete(int x);
```

```
};
```

```
//===chuong trinh chinh===
```

```
int main()
```

```
{
```

```
    DSLKD ds;
```

```
    int n,x,i,tg;
```

```
cout<<"Nhập vào số lượng phần tử của dãy số: "; cin>>n;

cout<<"Nhập vào dãy số:\n";
for(i=0;i<n;i++)
{
    cout<<"Phần tử thứ "<<i+1<<" là: ";
    cin>>tg;
    ds.Insert(tg);
}

cout<<"Danh sách đã nhập là:\n";
ds.Display();

cout<<"\nNhập vào phần tử cần tìm: "; cin>>x;
ds.Search(x);

ds.Delete(x);

cout<<"\n\nDanh sách sau khi xóa phần tử "<<x<<" là:\n";
ds.Display();

cout<<endl;
return 0;
}

//===định nghĩa hàm===

DSLKD::DSLKD():F(NULL)
```



```
{

}

//-----

DSLKD::~~DSLKD()
{
    Node *P;

    while(F)
    {
        P=F;
        F=F->Link;
        delete P;
    }
}

//-----

void DSLKD::Insert(int x)
{
    //buoc 1: tao nut moi
    Node *N=new Node;
    N->Infor=x;
    N->Link=NULL;

    //buoc 2: noi nut moi vao danh sach
    if(F==NULL)
        F=N;
```

```
else
{
    Node *P=F;
    while(P->Link)
        P=P->Link;
    P->Link=N;
}
}
//-----
void DSLKD::Display()
{
    Node *P=F;
    while(P)
    {
        cout<<P->Infor<<" ";
        P=P->Link;
    }
}
//-----
int DSLKD::Search(int x)
{
    Node *P=F;
    if(F==NULL)
    {
        cout<<"Danh sach rong!";
        return 0;
    }
}
```

```
while(P)
{
    if(P->Infor==x)
    {
        cout<<"Tim thay "<<x<<" trong danh sach!\n";
        return 1;
    }
    P=P->Link;
}

return 0;
}
//-----
void DSLKD::Delete(int x)
{
    Node *P=F,*Q;

    //tim nut can xoa
    while(P)
    {
        if(P->Infor==x)
        {
            break;
        }
        Q=P; //giu lai nut P
        P=P->Link;
    }
```

```
//neu tim thay nut can xoa
if(P)
{
    if(P==F)
    {
        F=F->Link;
        delete P;
    }
    else
    {
        Q->Link=P->Link;
        delete P;
    }
}
}
```

//-----

Bài 24(lthdtbai24.cpp): Tính tổng n số phức. Yêu cầu dùng toán tử + để cộng hai số phức, dùng toán tử nhập >> nhập vào số phức, dùng toán tử xuất << để đưa ra số phức ở dạng a + bi, có thể khởi tạo giá trị ban đầu cho phần thực và phần ảo.

```
#include<stdio.h>
```

```
#include<iostream>
```

```
using namespace std;
```

```
//khai bao lop
```

```
class SoPhuc
```

```
{
```

private:

float a,b;

public:

SoPhuc();

friend istream& operator>>(istream &cin, SoPhuc &sp);

friend ostream& operator<<(ostream &cout, SoPhuc &sp);

SoPhuc operator+(const SoPhuc &sp2);

};

//===chuong trinh chinh===

int main()

{

SoPhuc a[10],tong;

int n,i;

cout<<"Nhap vao so luong so phuc: "; cin>>n;

cout<<"Nhap vao so phuc:\n";

for(i=0;i<n;i++)

{

printf("So phuc thu %d: \n",i+1);

cin>>a[i];

}

cout<<"Cac so phuc da nhap la:\n";

for(i=0;i<n;i++)

```
{
    cout<<a[i];
    printf("\n");
}

for(i=0;i<n;i++)
    tong=tong+a[i];

cout<<"\nTong cac so phuc la: "; cout<<tong;

cout<<endl;
return 0;
}
//===đinh nghĩa ham===
SoPhuc::SoPhuc():a(0),b(0)
{

}
//-----
istream& operator>>(istream &cin, SoPhuc &sp)
{
    cout<<"\tNhập vào phần thực: "; cin>>sp.a;
    cout<<"\tNhập vào phần ảo: "; cin>>sp.b;

    return cin;
}
//-----
```

```
ostream& operator<<(ostream &cout, SoPhuc &sp)
{
    if(sp.b<0)
        cout<<sp.a<<"-"<<-sp.b<<"i";
    else
        cout<<sp.a<<"+"<<sp.b<<"i";

    return cout;
}
//-----

SoPhuc SoPhuc::operator+(const SoPhuc &sp2)
{
    SoPhuc tg;

    tg.a=a+sp2.a;
    tg.b=b+sp2.b;

    return tg;
}
//-----
```

Bài 25(lthdtbai25.cpp): Tạo lớp đối tượng xâu ký tự có các đặc điểm giống như xâu ký tự trong Pascal: Có thể khởi tạo đối tượng xâu bằng một hằng xâu; có thể gán hằng xâu hoặc đối tượng xâu cho đối tượng xâu; có thể dùng phép toán + để nối xâu; có thể dùng các phép toán so sánh để so sánh xâu; có thể dùng cin và cout để nhập vào và đưa ra đối tượng xâu.

```
#include<iostream>
```

```
#include<stdio.h>
```

```
#include<string.h>
```

```
using namespace std;
```

```
//Khai bao lop xau ky tu
```

```
class XString
```

```
{
```

```
    private:
```

```
        enum {SIZE=256};
```

```
        char str[SIZE]; //Khai bao bien xau ky tu thong thuong
```

```
    public:
```

```
        XString(); //Ham tao khong doi so
```

```
        XString(const char *s); //Ham tao 1 doi so
```

```
        XString operator+(const XString &s2);
```

```
        bool operator>(const XString &s2);
```

```
        bool operator>=(const XString &s2);
```

```
        bool operator<(const XString &s2);
```

```
        bool operator<=(const XString &s2);
```

```
        bool operator==(const XString &s2);
```

```
        bool operator!=(const XString &s2);
```

```
        //Ham ban
```

```
        friend void operator>>(istream &cin,XString &s);
```

```
        friend void operator<<(ostream &cout,XString &s);
```

```
};
```

```
//===chuong trinh chinh===
```



```
int main()
{
    XString s="Ha Noi"; //khởi tạo đối tượng và gán hàng xau

    XString s1,s2;
    s1="Hoang Van Tuan"; //gán hàng xau cho đối tượng
    s2=s1+s2; //nối 2 đối tượng xau

    XString s3;
    s3=s2; //gán 2 đối tượng xau

    XString s4="A",s5="B",s6="B";
    if(s4>s5) //so sánh 2 đối tượng xau
        cout<<"Xau1 > xau2!";
    else
        cout<<"Xau1 < xau2!";

    cout<<s;

    cout<<endl;
    return 0;
}

//===định nghĩa ham===
XString::XString() //Ham tạo không đối số
{
    strcpy(str,""); //Đưa xau rỗng vào biến xau str
}
```

```
//-----  
XString::XString(const char *s) //Ham tao 1 doi so  
{  
    //Neu xau doi so co do dai < SIZE thi copy vao bien xau str, con khong thi cho  
    bien xau str rong  
    if(strlen(s)<SIZE)  
        strcpy(str,s);  
    else strcpy(str,"");  
}  
//-----  
XString XString::operator+(const XString &s2)  
{  
    if(strlen(str)+strlen(s2.str)<strlen(str)-1)  
        strcat(str,s2.str);  
    else  
        cout<<"Khong noi duoc!";  
  
    return *this;  
}  
//-----  
bool XString::operator>(const XString &s2)  
{  
    if(strcmp(str,s2.str)>0)  
        return true;  
    return false;  
}  
//-----
```

```
bool XString::operator>=(const XString &s2)
```

```
{  
    if(strcmp(str,s2.str)>=0)  
        return true;  
    return false;  
}
```

```
//-----
```

```
bool XString::operator<(const XString &s2)
```

```
{  
    if(strcmp(str,s2.str)<0)  
        return true;  
    return false;  
}
```

```
//-----
```

```
bool XString::operator<=(const XString &s2)
```

```
{  
    if(strcmp(str,s2.str)<=0)  
        return true;  
    return false;  
}
```

```
//-----
```

```
bool XString::operator==(const XString &s2)
```

```
{  
    if(strcmp(str,s2.str)==0)  
        return true;  
    return false;  
}
```

```
//-----  
bool XString::operator!=(const XString &s2)  
{  
    if(strcmp(str,s2.str)!=0)  
        return true;  
    return false;  
}  
//-----  
//Định nghĩa hàm thông thường, ngoài đối tượng  
void operator>>(istream &cin,XString &s)  
{  
    //Hủy Enter trong bộ nhớ đệm bàn phím nếu có  
    scanf(" ");  
    //Lấy vào chuỗi ký tự có cả dấu cách  
    cin.get(s.str,sizeof(s.str));  
}  
//-----  
void operator<<(ostream &cout,XString &s)  
{  
    cout<<s.str;  
}  
//-----
```

Bài 26(lthdtbai26.cpp): Một nhân sự luôn có họ tên và ngày sinh.

Sinh viên là một loại nhân sự có thêm mã sinh viên và điểm tbc.

Nhập vào thông tin của một sinh viên. Đưa ra màn hình các thông tin về sinh viên có kèm theo đánh giá, nếu điểm TBC  $\geq 8.0$  thì đánh giá là sinh viên giỏi.

```
#include<iostream>
```

```
#include<stdio.h>
```

```
using namespace std;
```

```
//khai bao lop
```

```
class NhanSu
```

```
{
```

```
    private:
```

```
        char Ten[31];
```

```
        char NS[11];
```

```
    public:
```

```
        void Nhap();
```

```
        void Hien();
```

```
};
```

```
class SinhVien:public NhanSu
```

```
{
```

```
    private:
```

```
        char MaSV[10];
```

```
        float DTB;
```

```
    public:
```

```
        void Nhap();
```

```
        void Hien();
```

```
};
```

```
//===chuong trinh chinh===
```

```
int main()
```

```
{
```

```
    SinhVien SV;
```

```
    cout<<"Nhap thong tin ve sinh vien:\n";
```

```
    SV.Nhap();
```

```
    cout<<"\n\nThong tin sinh vien da nhap:\n";
```

```
    SV.Hien();
```

```
    cout<<endl;
```

```
    return 0;
```

```
}
```

```
//===dinh nghia ham===
```

```
//ham trong lop nhan su
```

```
void NhanSu::Nhap()
```

```
{
```

```
    cout<<"Nhap vao ho ten: ";
```

```
    scanf(" ");
```

```
    cin.get(Ten, sizeof(Ten));
```

```
    cout<<"Nhap vao ngay sinh: ";
```

```
    cin>>NS;
```

```
}
```

```
//-----
```

```
void NhanSu::Hien()
{
    cout<<"Ho ten: "<<Ten;
    cout<<"\nNgày sinh: "<<NS;
}
//-----
//ham trong lop sinh vien
void SinhVien::Nhap()
{
    NhanSu::Nhap();

    cout<<"Nhap ma sinh vien: ";
    cin>>MaSV;
    cout<<"Nhap diem trung binh: ";
    cin>>DTB;
}
//-----
void SinhVien::Hien()
{
    NhanSu::Hien();

    cout<<"\nMa sinh vien: "<<MaSV;
    cout<<"\nDiem trung binh chung: "<<DTB;

    if(DTB>=8)
        cout<<"\nDanh gia: Gioi!";
}
```

Bài 27(lthdtbai27.cpp): Một nhân sự nói chung có họ tên và ngày sinh. Nhân viên trong một công ty là một loại nhân sự nhưng có thêm mã nhân viên và hệ số lương. Lãnh đạo trong công ty là một loại nhân viên có thêm chức vụ. Nhập vào thông tin của một số nhân viên và lãnh đạo trong công ty. Đưa ra các thông tin đã nhập.

```
#include<iostream>
```

```
#include<stdio.h>
```

```
using namespace std;
```

```
//khai bao lop
```

```
class NhanSu
```

```
{
```

```
private:
```

```
    char Ten[31];
```

```
    char NS[11];
```

```
public:
```

```
    void Nhap();
```

```
    void Hien();
```

```
};
```

```
class NhanVien:public NhanSu
```

```
{
```

```
private:
```

```
    char Ma[7];
```

```
    float HeSo;
```



```
public:
```

```
    void Nhap();
```

```
    void Hien();
```

```
};
```

```
class LanhDao:public NhanVien
```

```
{
```

```
private:
```

```
    char ChucVu[61];
```

```
public:
```

```
    void Nhap();
```

```
    void Hien();
```

```
};
```

```
//===chuong trinh chinh===
```

```
int main()
```

```
{
```

```
    NhanVien nv1,nv2;
```

```
    LanhDao ld;
```

```
    //nhap nhan vien
```

```
    cout<<"Nhap vao thong tin cua nhan vien 1:\n";
```

```
    nv1.Nhap();
```

```
    cout<<"Nhap vao thong tin cua nhan vien 2:\n";
```

```
    nv2.Nhap();
```

```
//nhap lanh dao
cout<<"Nhap vao thong tin cua lanh dao:\n";
ld.Nhap();

//hien thi danh sach nhan vien va lanh dao
cout<<"\n\nThong tin nhan vien la:\n";
nv1.Hien();
cout<<endl;
nv2.Hien();

cout<<"\nThong tin lanh dao:\n";
ld.Hien();

cout<<endl;
return 0;
}

//===định nghĩa hàm===
//ham thanh vien trong lop nhan su
void NhanSu::Nhap()
{
    cout<<"\tNhap vao ho ten: "; scanf(" "); cin.get(Ten,sizeof(Ten));
    cout<<"\tNhap vao ngay sinh: "; cin>>NS;
}

//-----
void NhanSu::Hien()
{
    cout<<"\tHo ten: "<<Ten;
```

```
        cout<<"\n\tNgày sinh: "<<NS;
    }
    //-----

    //ham thanh vien trong lop nhan vien
    void NhanVien::Nhap()
    {
        NhanSu::Nhap();

        cout<<"\tNhap vao ma nhan vien: "; cin>>Ma;
        cout<<"\tNhap vao he so luong: "; cin>>HeSo;
    }
    //-----

    void NhanVien::Hien()
    {
        NhanSu::Hien();

        cout<<"\n\tMa nhan vien: "<<Ma;
        cout<<"\n\tHe so luong: "<<HeSo;
    }
    //-----

    //ham thanh vien trong lop lanh dao
    void LanhDao::Nhap()
    {
        NhanVien::Nhap();
```

```
    cout<<"\tNhap vao chuc vu: "; cin>>ChucVu;
}
//-----

void LanhDao::Hien()
{
    NhanVien::Hien();

    cout<<"\n\tChuc vu: "<<ChucVu;
}
//-----
```

Bài 28(lthdtbai28.cpp): Thời điểm là một loại ngày tháng và cũng là một loại thời gian.

Ngày tháng có ngày, tháng, năm. Thời gian có giờ và phút. Nhập vào một thời điểm,

đưa ra ngày và thời gian của thời điểm đó ở dạng dd/mm/yy - h:mm.

```
#include<iostream>
```

```
#include<stdio.h>
```

```
using namespace std;
```

```
//khai bao lop
```

```
class NgayThang
```

```
{
```

```
private:
```

```
    int Ngay,Thang,Nam;
```

```
public:
```

```
    void Nhap();
```

```
void Hien();  
};  
class ThoiGian  
{  
private:  
    int Gio,Phut;  
  
public:  
    void Nhap();  
    void Hien();  
};  
class ThoiDiem:public NgayThang,public ThoiGian  
{  
private:  
  
public:  
    void Nhap();  
    void Hien();  
};  
//===chương trình chính===  
int main()  
{  
    ThoiDiem td;  
  
    cout<<"Nhập vào 1 thời điểm:\n";
```

```
td.Nhap();

cout<<"Thoi diem da nhap la:\n";
td.Hien();

cout<<endl;
return 0;
}

//===định nghĩa ham===
//ham thanh vien lop Ngay thang
void NgayThang::Nhap()
{
    cout<<"Nhap vao ngay (dd): "; cin>>Ngay;
    cout<<"Nhap vao thang (mm): "; cin>>Thang;
    cout<<"Nhap vao nam (yy): "; cin>>Nam;
}

//-----
void NgayThang::Hien()
{
    printf("%d/%d/%d",Ngay,Thang,Nam);
}

//-----

//ham thanh vien lop Thoi gian
void ThoiGian::Nhap()
{
    do
```

```
{
    cout<<"Nhap vao gio (h): "; cin>>Gio;

    if(Gio<0 || Gio>=24)
        cout<<"Gio khong thuoc 0-23. Hay nhap lai gio!\n";
}
while(Gio<0 || Gio>=24);

do
{
    cout<<"Nhap vao phut (mm): "; cin>>Phut;

    if(Phut<0 || Phut>=60)
        cout<<"Phut khong thuoc 0-60. Hay nhap lai phut!\n";
}
while(Phut<0 || Phut>=60);
}
//-----
void ThoiGian::Hien()
{
    printf("%d:%d",Gio,Phut);
}
//-----

//ham thanh vien lop Thoi diem
void ThoiDiem::Nhap()
{
```

```
    ThoiGian::Nhap();  
    NgayThang::Nhap();  
}
```

```
//-----
```

```
void ThoiDiem::Hien()
```

```
{  
    NgayThang::Hien();  
    cout<<"-";  
    ThoiGian::Hien();  
}
```

```
//-----
```

Bài 29(lthdtbai29.cpp): Tính diện tích và chu vi các hình: Hình chữ nhật biết 2 cạnh, hình tam giác biết 3 cạnh, hình tròn biết bán kính. Nhập vào một số hình trong đó có cả hình chữ nhật, hình tam giác và hình tròn. Đưa ra diện tích và chu vi các hình đã nhập. Y/c cài đặt đa hình động và lớp trừu tượng; viết 1 hàm lựa chọn hình, nhập kích thước cho hình và trả về hình đã nhập; viết 1 hàm đưa ra diện tích và chu vi một hình truyền vào qua đối số.

```
#include<stdio.h>
```

```
#include<iostream>
```

```
#include<math.h>
```

```
using namespace std;
```

```
//khai báo lớp
```

```
class Hình
```

```
{
```

```
private:
```

```
public:
```



```
virtual void Nhap()=0;
virtual float TinhDT()=0;
virtual float TinhCV()=0;
};
class HinhCN:public Hinh
{
private:
    float a,b;

public:
    void Nhap();
    float TinhDT();
    float TinhCV();
};
class HinhTG:public Hinh
{
private:
    float a,b,c;

public:
    void Nhap();
    float TinhDT();
    float TinhCV();
};
class HinhTR:public Hinh
{
private:
```

```
float r;  
  
public:  
    void Nhap();  
    float TinhDT();  
    float TinhCV();  
};  
  
//khai bao ham thong thuong  
Hinh* LuaChonHinh();  
void Hien(Hinh *h);  
  
//===chuong trinh chinh===  
int main()  
{  
    Hinh *DSHinh[5];  
    int n,i;  
  
    cout<<"Nhap so luong hinh can tinh: "; cin>>n;  
  
    cout<<"Nhap thong tin cac hinh:\n";  
    for(i=0;i<n;i++)  
    {  
        cout<<"Hinh thu "<<i+1<<":\n";  
        DSHinh[i]=LuaChonHinh();  
    }  
}
```

```
cout<<"Dien tich va chu vi cac hinh da nhap la:\n";
for(i=0;i<n;i++)
{
    cout<<"Hinh thu "<<i+1<<":\n";
    Hien(DSHinh[i]);
}

cout<<endl;
return 0;
}

//===định nghĩa ham===
//ham thanh vien lop hinh chu nhac
void HinhCN::Nhap()
{
    cout<<"Nhap vao 2 canh: "; cin>>a>>b;
}

//-----
float HinhCN::TinhDT()
{
    return a*b;
}

//-----
float HinhCN::TinhCV()
{
    return (a+b)*2;
}

//-----
```

```
//ham thanh vien lop hinh tam giac
```

```
void HinhTG::Nhap()
```

```
{
```

```
do
```

```
{
```

```
cout<<"Nhap vao 3 canh: "; cin>>a>>b>>c;
```

```
if(a+b<=c || a+c<=b || b+c<=a)
```

```
cout<<"Tam giac khong thao man. Hay nhap lai!\n";
```

```
}
```

```
while(a+b<=c || a+c<=b || b+c<=a);
```

```
}
```

```
//-----
```

```
float HinhTG::TinhDT()
```

```
{
```

```
float p=(a+b+c)/2;
```

```
return sqrt(p*(p-a)*(p-b)*(p-c));
```

```
}
```

```
//-----
```

```
float HinhTG::TinhCV()
```

```
{
```

```
return a+b+c;
```

```
}
```

```
//-----
```

```
//ham thanh vien lop hinh tron
```

```
void HìnhTR::Nhap()
{
    cout<<"Nhap vao ban kinh: "; cin>>r;
}
//-----

float HìnhTR::TinhDT()
{
    return 3.14*r*r;
}
//-----

float HìnhTR::TinhCV()
{
    return 2*3.14*r;
}
//-----

//định nghĩa hàm thông thường
Hình* LuaChonHình()
{
    Hình *h;
    int chon;

    cout<<"Ban chon hinh nao (1-CN, 2-TG, 3-TR): "; cin>>chon;

    switch(chon)
    {
        case 1:
```

```
        h=new HìnhCN;
        break;
    case 2:
        h=new HìnhTG;
        break;
    case 3:
        h=new HìnhTR;
        break;
    default:
        h=new HìnhCN;
    }

    h->Nhap(); //da hình xong

    return h;
}

//-----
void Hien(Hình *h)
{
    printf("\tDiện tích là: %0.2f",h->TinhDT());
    printf("\n\tChu vi là: %0.2f",h->TinhCV());
    cout<<endl;
}

//-----
```

Bài 30(lthdtbai30.cpp): Viết chương trình quản lý nhân sự của một trường học; nhân sự gồm có giảng viên và sinh viên; giảng viên có mã gv, họ tên và số bài báo;

sinh viên có mã sv, họ tên và điểm tbc. Nhập vào một số nhân sự trong đó có cả giảng viên và sinh viên. Đưa ra thông tin của các nhân sự đã nhập kèm theo đánh giá: giảng viên đánh giá là giỏi nếu số bài báo  $\geq 5$ , sinh viên đánh giá là giỏi nếu điểm TBC  $\geq 8.0$ . Y/c cài đặt đa hình động.

```
#include<stdio.h>
```

```
#include<iostream>
```

```
using namespace std;
```

```
//khai bao lop
```

```
class NhanSu
```

```
{
```

```
private:
```

```
    char Ten[31];
```

```
public:
```

```
    virtual void Nhap();
```

```
    virtual void Hien();
```

```
};
```

```
class GiangVien:public NhanSu
```

```
{
```

```
private:
```

```
    char MaGV[7];
```

```
    int SoBB;
```

```
public:
```

```
    void Nhap();
```

```
    void Hien();
```

```
};  
class SinhVien:public NhanSu  
{  
private:  
    char MaSV[7];  
    float DiemTBC;  
  
public:  
    void Nhap();  
    void Hien();  
};  
  
//khai bao ham thong thuong  
NhanSu *NhapNS();  
void HienNS(NhanSu *ns);  
  
//===chuong trinh chinh===  
int main()  
{  
    NhanSu *DSNhanSu[5];  
    int n,i;  
  
    cout<<"Nhap vao so luong nhan su: "; cin>>n;  
  
    cout<<"Nhap vao thong tin nhan su:\n";  
    for(i=0;i<n;i++)  
    {
```



```
        cout<<"\tNhan su thu "<<i+1<<":\n";
        DSNhanSu[i]=NhapNS();
    }

    cout<<"\n\nThong tin nhan su la:\n";
    for(i=0;i<n;i++)
    {
        cout<<"\tNhan su thu "<<i+1<<" la:\n";
        HienNS(DSNhanSu[i]);
    }

    cout<<endl;
    return 0;
}

//===định nghĩa ham===
//ham thanh vien lop nhan su
void NhanSu::Nhap()
{
    cout<<"\tNhap vao ten: "; scanf(" "); cin.get(Ten,sizeof(Ten));
}
//-----

void NhanSu::Hien()
{
    cout<<"\tHo ten: "<<Ten;
}
//-----
```

```
//ham thanh vien lop giang vien
void GiangVien::Nhap()
{
    NhanSu::Nhap();

    cout<<"\tNhap vao ma giang vien: "; cin>>MaGV;
    cout<<"\tNhap vao so bai bao: "; cin>>SoBB;
}
//-----
void GiangVien::Hien()
{
    NhanSu::Hien();

    cout<<"\n\tMa giang vien: "<<MaGV;
    cout<<"\n\tSo bai bao: "<<SoBB;

    if(SoBB>=5)
        cout<<"\n\t==> Giang vien gioi!";
}
//-----
//ham thanh vien lop sinh vien
void SinhVien::Nhap()
{
    NhanSu::Nhap();

    cout<<"\n\tNhap vao ma sinh vien: "; cin>>MaSV;
```

```
    cout<<"\n\tNhap vao diem tbc: "; cin>>DiemTBC;
}
//-----

void SinhVien::Hien()
{
    NhanSu::Hien();

    cout<<"\n\tMa sinh vien: "<<MaSV;
    cout<<"\n\tDiem tbc: "<<DiemTBC;

    if(DiemTBC>=8)
        cout<<"\n\t==> Sinh vien gioi!";
}
//-----

//kđinh nghĩa hàm thông thường
NhanSu *NhapNS()
{
    NhanSu *ns;
    int chon;

    cout<<"\tNhap loai nhan su (1-GV, 2-SV): "; cin>>chon;

    switch(chon)
    {
    case 1:
        ns=new GiangVien;
```

```
        break;
    case 2:
        ns=new SinhVien;
        break;
    default:
        ns=new GiangVien;
    }

    ns->Nhap(); //da hinh dong

    return ns;
}
//-----
void HienNS(NhanSu *ns)
{
    ns->Hien();
}
//-----
```

Bài 31(lthdtbai31.cpp): Đọc vào một dãy số nguyên từ tệp văn bản, trong đó có cả số dương và số âm. Sử dụng hàng đợi lưu trữ kế tiếp theo kiểu quay vòng để tách, đưa ra màn hình đồng thời ghi ra tệp văn bản dãy các số dương và dãy các số âm.

```
#include<iostream>
```

```
#include<stdio.h>
```

```
#include<fstream>
```

```
using namespace std;
```

```
//khai bao lop
class Queue
{
private:
    enum{SIZE = 50};
    int a[SIZE];
    int F,R;

public:
    Queue(); //ham khoi tao

    void CQInsert(int x);
    int CQDelete();
    int IsCQEmpty();
};

//===chuong trinh chinh===
int main()
{
    Queue Q;
    int n,i,x[10];

    ifstream fin("dayso.txt");
    ofstream fout("ketqua.txt");

    fin>>n;
```

```
for(i=0;i<n;i++)
    fin>>x[i];

cout<<"Day so doc vao tu tep la:\n";
for(i=0;i<n;i++)
    cout<<x[i]<<" ";

fout<<"Day so duong la: ";
for(i=0;i<n;i++)
    if(x[i]>=0)
    {
        Q.CQInsert(x[i]);
        fout<<x[i]<<" ";
    }

fout<<"\nDay so am la: ";
for(i=0;i<n;i++)
    if(x[i]<0)
    {
        Q.CQInsert(x[i]);
        fout<<x[i]<<" ";
    }

cout<<"\nHang doi la:\n";
while(!Q.IsCQEmpty())
    cout<<Q.CQDelete()<<" ";
```

```
    cout<<endl;
    return 0;
}
//===định nghĩa ham===
Queue::Queue():F(-1),R(-1) //khi hàng đợi rỗng thì F=R=0
{

}
//-----
void Queue::CQInsert(int x)
{
    //buoc 1: kiểm tra hàng đợi đầy
    if(F==0 && R==SIZE-1 || R+1==F)
    {
        cout<<"hàng đợi đầy!";
        return;
    }

    //buoc 2: thay đổi chỉ số R
    if(R==SIZE-1)
        F=R=0;
    else if(R==SIZE-1)
        R=0;
    else
        R++;
}
```

```
//buoc 3: bo sung x vao hang doi
a[R]=x;
}
//-----
int Queue::CQDelete()
{
    //buoc 1: kiem tra hang doi rong
    if(F==-1)
    {
        cout<<"Hang doi rong!";
        return -1;
    }

    //buoc 2: luu lai phan tu loai bo
    int y=a[F];

    //buoc 3: thay doi chi so F
    if(F==R)
        F=R=-1;
    else if(F==SIZE-1)
        F=0;
    else
        F++;

    //buoc 4: tra ve phan tu loai bo
    return y;
}
```



```
//-----  
int Queue::IsCQEmpty()  
{  
    if(F==-1)  
        return 1;  
    return 0;  
}  
//-----
```

Hoang Van Tuan