

MỤC LỤC

I. BÀI THỰC HÀNH SỐ 1.....	1
1. Bài 1	1
2. Bài 2	3
II. BÀI THỰC HÀNH SỐ 2.....	7
1. Bài 1	7
2. Bài 2	10
3. Bài 3.1	15
4. Bài 3.2	20
5. Bài 3.3	25
III. BÀI THỰC HÀNH SỐ 3.....	29
6. Bài 1	29
7. Bài 2	34
8. Bài 3.1	37
9. Bài 3.2	41
10. Bài 3.3	47

I. BÀI THỰC HÀNH SỐ 1

1. Bài 1

a. Đề bài: Cài đặt class Complex

complex.h: Viết định nghĩa lớp Complex có hai thành viên dữ liệu private **readPart** và **imaginaryPart** thuộc kiểu **double**, 1 **constructor** và các hàm thành viên public **add** (phép cộng), **sub** (phép trừ), **print**, **nhập** (để nhập dữ liệu), 1 hàm private **setComplexNumber**, các thông tin về kiểu trả về, danh sách tham số có thể xem tại các định nghĩa hàm tại **complex.cpp**.

complex.cpp: hoàn thành các lệnh thuộc phần định nghĩa các hàm add và sub. **add:** viết một lệnh để cộng phần thực của a vào phần thực của lớp; Viết một lệnh để cộng phần ảo của a vào phần ảo của lớp. **sub:** viết một lệnh trừ đi phần thực của s từ phần thực của lớp; viết một lệnh để trừ đi phần ảo của s từ phần ảo của lớp.

Viết chương trình **complex_demo.cpp** để test lại lớp Complex trên. Biên dịch và chạy thử **complex_demo.cpp** - chương trình dùng thử nghiệm class Complex.

b. Thuật toán

- Khai báo lớp Complex gồm 2 thành viên dữ liệu private **readPart** và **imaginaryPart**. Các hàm thành viên public **add**, **sub**, **print**, hàm thành viên private **setComplexNumber**.
- Hàm **setComplexNumber**:
 - Hàm thành viên private có hai tham số là a, b được gán cho phần thực là a và phần ảo là b.
- Hàm **add**:
 - Hàm public có 1 tham số là một Complex x và trả về là một Complex.
 - Phần thực của complex hiện tại cộng với phần thực của complex x, phần ảo của complex hiện tại cộng với phần ảo của complex x.
- Hàm **sub**:
 - Hàm public có 1 tham số là một Complex x và trả về là một Complex.
 - Phần thực của complex hiện tại trừ với phần thực của complex x, phần ảo của complex hiện tại trừ với phần ảo của complex x.
- Hàm **print**:
 - Hàm public không có tham số.
 - In ra 2 giá trị **readPart** và **imaginaryPart** của Complex hiện tại.
- Hàm **nhập**:
 - Hàm public không có tham số.
 - Nhập vào giá trị **readPart** và **imaginaryPart** của Complex hiện tại

c. Mã nguồn

- File **complex.h**:

```
#ifndef COMPLEX_H
#define COMPLEX_H
class complex
{
    private:
        double readPart;
        double imaginaryPart;
        void setComplexNumber(double a,double b);

    public:
        complex(double =0.0,double =0.0);// ham dung
        complex add(complex&);//ham cong hai so phuc
        complex sub(complex&);//ham tru hai so phuc
        void print();//ham in ra so phuc
        void nhap();
};
```

```

#endif
- File complex.cpp:
#include "complex.h"
#include <iostream>
using namespace std;

void complex::setComplexNumber(double a,double b)
{
    this->readPart=a;
    this->imaginaryPart=b;
}
complex::complex(double a,double b)
{
    setComplexNumber(a,b);
}
complex complex::add(complex &a)
{
    this->readPart+=a.readPart;
    this->imaginaryPart+=a.imaginaryPart;
    return *this;
}
complex complex::sub(complex &a)
{
    this->readPart-=a.readPart;
    this->imaginaryPart-=a.imaginaryPart;
    return *this;
}
void complex::nhap()
{
    cout<<"\n nhap vao phan thuc : ";
    cin>>readPart;
    cout<<"\n nhap vao phan ao : ";
    cin>>imaginaryPart;
}
void complex::print()
{
    cout<< readPart;
    if(imaginaryPart>=0) cout<<"+"<<imaginaryPart<<"i"<<endl;
    else cout<<imaginaryPart<<"i"<<endl;
}

- File complex_demo.cpp
#include <iostream>
#include "complex.h"
#include <windows.h>
using namespace std;
/* run this program using the console pauser or add your own getch, system("pause") or input loop */

main() {
    complex a,b;
    a.nhap();
    cout<<"\n so phuc a : ";
    a.print();
    b.nhap();
    cout<<"\n so phuc b : ";
    b.print();
    cout<<"\n so phuc a + b : ";
    a.add(b).print();
    a.sub(b);
    cout<<"\n so phuc a - b : ";
    a.sub(b).print();

    system("pause");
}

```

d. Demo

```
D:\Ky 6\Thuc Hanh Huong Doi Tuong\Thuc Hanh1\Bai C...
nhap vao phan thuc : 5
nhap vao phan ao : 2
so phuc a : 5+2i
nhap vao phan thuc : 3
nhap vao phan ao : -4
so phuc b : 3-4i
so phuc a + b : 8-2i
so phuc a - b : 2+6i
Press any key to continue . . .
```

2. Bài 2

a. Đề bài: Cài đặt class Date

Cài đặt class **Date** gồm hai file **date.h**, **date.cpp** đã hoàn chỉnh gồm các hàm get và các hàm set dùng để truy nhập dữ liệu **private**, hàm **isLeapYear(int)** để kiểm tra xem một năm (là tham số của hàm) có phải là năm nhuận hay không, hàm này trả về **TRUE** nếu là năm nhuận, ngược lại, trả về **FALSE**; hàm **monthDays** trả về số ngày trong tháng hiện tại (thành viên dữ liệu month của lớp). Hàm nhập, xuất dữ liệu ngày tháng năm. Hàm thành viên **nextDay** sửa ngày hiện hành thành ngày hôm sau. Thí dụ, nếu có d là đối tượng Date lưu trữ ngày 27 tháng 3 năm 2003 thì sau lời gọi d.nextDay(), d trở thành ngày 28 tháng 3 năm 2003.

Gợi ý: sử dụng các hàm thành viên có sẵn. Viết file **date_demo.cpp** để in ra 40 ngày liên nhau kể từ ngày 20 tháng 2 năm 2003. Gợi ý: Khai báo biến d là đối tượng kiểu Date khởi tạo tại ngày 20/2/2003; Lặp 40 lần, mỗi lần in d rồi chuyển d thành ngày hôm sau bằng lời gọi đến hàm nextDay.

b. Thuật toán

- Tạo lớp Date gồm 3 thành viên **private** là day, month, year.
- Các hàm getDay() để trả về ngày hiện tại, getMonth() trả về tháng hiện tại, getYear() trả về năm hiện tại, setDay(int) để đặt lại ngày, setMonth(int) để đặt lại tháng, setYear() để đặt lại năm, isLeapYear(int) để kiểm tra năm nhuận, monthDays() để trả về số ngày của tháng hiện tại.
- Các hàm nextDay() để chuyển sang ngày tiếp theo.
- Hàm nhập xuất để nhập, xuất dữ liệu.
- Hàm nextDay():
 - Tăng ngày lên 1.
 - Nếu số ngày lớn hơn số ngày của tháng thì gán lại ngày bằng 1, tăng tháng lên 1.
 - Nếu tháng lớn hơn 12 thì tháng gán lại bằng 1 và tăng năm.

c. Mã nguồn

- File **date.h**

```
#ifndef DATE_H
#define DATE_H

class date
{
```

```

public:
    date(int =0,int =0,int =0);
    void setDate(int d);
    void setMonth(int m);
    void setYear(int y);
    int getDate();
    int getMonth();
    int getYear();
    bool isLeapYear(int y);
    int monthDays(int m, int y);
    date nextDay();
    void nhap();
    void xuat();

private:
    int day,month,year;

};

```

#endif

- File **date.cpp**

```

#include "date.h"
#include <iostream>
using namespace std;

date::date(int d,int m,int y)
{
    this->day=d;
    this->month=m;
    this->year=y;
}

void date::setDate(int d)
{
    this->day=d;
}

void date::setMonth(int m)
{
    this->month=m;
}

void date::setYear(int y)
{
    this->year=y;
}

int date::getDate()
{
    return day;
}

int date::getMonth()
{
    return month;
}

int date::getYear()
{
    return year;
}

bool date::isLeapYear(int y)
{
    if(((y%4==0)&&(y%100!=0))||((y%400==0)))
        return true;
    else return false;
}

int date::monthDays(int m, int y)
{
    int so[13]={0,31,28,31,30,31,30,31,31,30,31,30,31};
    if(isLeapYear(y)==1) so[2]=29;
    int songay=so[m];
    return songay;
}

date date::nextDay()
{
    if(day<monthDays(month,year))
        day+=1;
    else

```

```

        {
            day=1;
            if(month==12)
            {
                month=1;
                year+=1;
            }
            month+=1;
        }
        return *this;
    }
    void date::nhap()
    {
        do
        {
            cout<<"ngay: ";
            cin>>day;
            cout<<"thang: ";
            cin>>month;
            cout<<"nam: ";
            cin>>year;
        } while((day>monthDays(month,year))||((month>12)));
    }
    void date::xuat()
    {
        cout<<"ngay " <<day<<" thang " <<month<<" nam " <<year;
        cout<<"\n";
    }
}

```

- **File *date_demo.cpp***

```

#include <iostream>
#include "date.h"
using namespace std;

```

/* run this program using the console pauser or add your own getch, system("pause") or input loop */

```

main() {
    date a,b;
    a.nhap();
    a.xuat();
    cout<<"40 ngay tiep sau \n";
    for(int lap=1;lap<=40;lap++)
    {
        a.nextDay().xuat();
    }
}

```

d. }Demo

```
D:\Ky 6\Thuc Hanh Huong Doi Tuong\ThucHanh1\date\d...
ngay: 27
thang: 3
nam: 2014
ngay 27 thang 3 nam 2014
40 ngay tiep sau
ngay 28 thang 3 nam 2014
ngay 29 thang 3 nam 2014
ngay 30 thang 3 nam 2014
ngay 31 thang 3 nam 2014
ngay 1 thang 4 nam 2014
ngay 2 thang 4 nam 2014
ngay 3 thang 4 nam 2014
ngay 4 thang 4 nam 2014
ngay 5 thang 4 nam 2014
ngay 6 thang 4 nam 2014
ngay 7 thang 4 nam 2014
ngay 8 thang 4 nam 2014
ngay 9 thang 4 nam 2014
ngay 10 thang 4 nam 2014
ngay 11 thang 4 nam 2014
ngay 12 thang 4 nam 2014
ngay 13 thang 4 nam 2014
ngay 14 thang 4 nam 2014
ngay 15 thang 4 nam 2014
ngay 16 thang 4 nam 2014
ngay 17 thang 4 nam 2014
ngay 18 thang 4 nam 2014
ngay 19 thang 4 nam 2014
ngay 20 thang 4 nam 2014
ngay 21 thang 4 nam 2014
ngay 22 thang 4 nam 2014
ngay 23 thang 4 nam 2014
ngay 24 thang 4 nam 2014
ngay 25 thang 4 nam 2014
ngay 26 thang 4 nam 2014
ngay 27 thang 4 nam 2014
ngay 28 thang 4 nam 2014
ngay 29 thang 4 nam 2014
ngay 30 thang 4 nam 2014
ngay 1 thang 5 nam 2014
ngay 2 thang 5 nam 2014
ngay 3 thang 5 nam 2014
ngay 4 thang 5 nam 2014
ngay 5 thang 5 nam 2014
ngay 6 thang 5 nam 2014

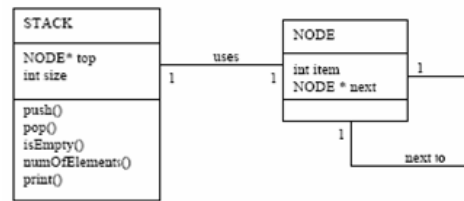
-----
Process exited after 13.23 seconds with return value 0
Press any key to continue . . .
```

II. BÀI THỰC HÀNH SỐ 2

1. Bài 1

e. Đề bài: Thiết kế cấu trúc ngăn xếp

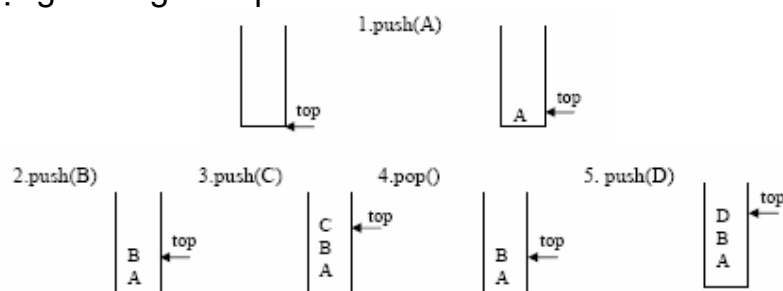
- Thiết kế cấu trúc ngăn xếp như sau:



- **Node:** là một nút trong ngăn xếp chứa 2 thành viên dữ liệu là **item** và **next** để chuyển tới nút tiếp theo.
- **Stack:** là ngăn xếp chứa dữ liệu kiểu **int**
 - Có các thành viên dữ liệu **top** để trỏ đến đỉnh ngăn xếp, **size** để đếm số lượng dữ liệu.
 - Các phương thức: Push() để đẩy một giá trị vào ngăn xếp, Pop() để lấy một giá trị ra khỏi ngăn xếp, numOfElement() để kiểm tra số phần tử trong ngăn xếp, Print() để in các phần tử trong stack, isEmpty() để kiểm tra ngăn xếp có rỗng hay không.

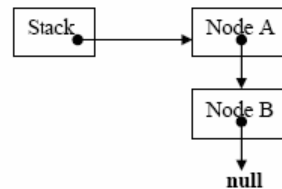
f. Thuật toán

- Khai báo **Node** kiểu **struct**, có 2 thành viên dữ liệu **item** kiểu int và con trỏ **next** kiểu Node.
- Lớp **Stack** có 2 thành viên dữ liệu con trỏ **top** trỏ đến đỉnh ngăn xếp, **size** kiểu int để đếm số lượng phần tử trong stack. Các hàm thành viên Push(int) để đẩy một phần tử vào stack, Pop() để trả về giá trị tại đỉnh stack, numOfElement() trả về số phần tử, các hàm kiểm tra stack rỗng isEmpty().
- Hoạt động của ngăn xếp:

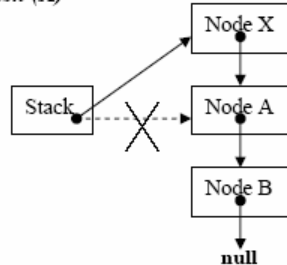


Gợi ý về hoạt động của push và pop

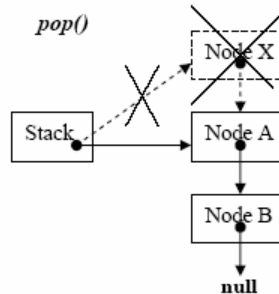
trạng thái hiện tại của ngăn xếp:



push (X)



pop()



- Thuật toán Push(): Đưa **top** trở tới node mới cấp phát vùng nhớ và gán giá trị truyền vào cho node mới này, cho node mới tạo trở tới top cũ, đồng thời tăng biến **size** lên 1.
- Thuật toán Pop(): Nếu Stack không rỗng thì lưu lại giá trị của **top** đang trở tới, đưa **top** trở tới node kế tiếp, trả về giá trị lấy được, giảm **size** đi 1.
- Thuật toán Print(): Lặp trong khi Stack chưa rỗng, Pop() ra lần lượt các node và in giá trị lên màn hình.
- Thuật toán isEmpty(): nếu size=0 trả về **true**, ngược lại trả về **false**.

g. Mã nguồn

- File **stack.h**:

```
#ifndef STACK_H
#define STACK_H
struct node
{
    int item; // item của node
    node *next; // con trỏ node tro den node tiep sau
};
class stack
{
public:
    stack(); // ham dung
    ~stack(); // ham huy
    void push(int x); // ham day node vao stack
    int pop(); // ham lay ra 1 node
    bool isEmpty(); // ham kiem tra ngăn xếp có rỗng hay không
    int numOfElement(); // ham tra về số phần tử trong ngăn xếp
    void print(); // ham in ra các phần tử trong ngăn xếp
private:
    node *top; // con trỏ node tro den dinh của ngăn xếp
    int size; // kích thước ngăn xếp
};

#endif
```

- File **stack.cpp**:

```
#include "stack.h"
#include <iostream>
using namespace std;
stack::stack() // ham dung
{
    this->top = NULL; // cho top = NULL
    size = 0; // kích thước ngăn xếp khởi tạo = 0
}
stack::~~stack() // ham huy
{
    if (this->size) pop(); // nếu size != 0 thì lấy phần tử trong ngăn xếp ra
}
```

```

void stack::push(int x)// lam day node vao ngan xep
{
    size++;//tang size len 1
    node *p = top;// cho con tro p tro den dinh ngan xep
    top=new node;//khoi tao vung nho cho top moi
    top->item=x;//gan x vao item cua node top moi
    top->next=p;//cho top cu bang next cua top moi
}
int stack::pop();//ham lay ra 1 phan tu trong ngan xep
{
    if(isEmpty()) return -1;//neu ngan nho rong thi tra ve -1
    else
    {
        size--;//giam kich co size xuong 1 don vi
        node *p = top;//gan con tro p bang dinh ngan xep
        int item =top->item;//cho item nhan gia tri item cua node top
        top=top->next;//gan top cho node next cua top
        return item;//tra ve gia tri item
    }
}
bool stack::isEmpty();//hamkiem tra ngan xep co rong hay khong
{
    return (this->size==0);//tra ve true neu size=0
}
int stack::numOfElement();//ham tra ve so luong phan tu trong ngan xep
{
    return (this->size);//tra ve size cua ngan xep
}
void stack::print();//ham in cac phan tu trong ngan xep
{
    node *p=top;// gan con tro p tro den dinh cua ngan xep
    while(p!=NULL)//trong khi dinh ngan xep khac rong
    {
        cout<<p->item<<endl;//in ra item cua node
        p=p->next;//gan p bang next cua node p
    }
}

```

- **File *stack_demo.cpp*:**

```

#include <iostream>
#include "stack.h"
using namespace std;

```

/* run this program using the console pauser or add your own getch, system("pause") or input loop */

```

int main() {
    int chon,item;
    stack p;
    do
    {
        cout<<"\n 1.them vao 1 phan tu";
        cout<<"\n 2.lay ra 1 phan tu";
        cout<<"\n 3.in ra danh sach";
        cout<<"\n 4.thoat";
        cout<<"\n moi ban chon : ";
        fflush(stdin); cin>>chon;
        switch(chon)
        {
            case 1:
            {
                cout<<"\n nhap vao item : ";
                cin>>item;
                p.push(item);
                break;
            }
            case 2:
            {
                p.pop(); break;
            }
            case 3:
            {
                p.print();
            }
        }
    }
}

```

```

    }
} while(chon!=4);
return 0;
}

```

h. Demo

```

D:\Ky 6\Thuc Hanh Huong Doi Tuyen...
1.them vao 1 phan tu
2.lay ra 1 phan tu
3.in ra danh sach
4.thoat
moi ban chon : 1

nhap vao item : 3

1.them vao 1 phan tu
2.lay ra 1 phan tu
3.in ra danh sach
4.thoat
moi ban chon : 1

nhap vao item : 5

1.them vao 1 phan tu
2.lay ra 1 phan tu
3.in ra danh sach
4.thoat
moi ban chon : 1

nhap vao item : 4

1.them vao 1 phan tu
2.lay ra 1 phan tu
3.in ra danh sach
4.thoat
moi ban chon : 3
4
5
3

1.them vao 1 phan tu
2.lay ra 1 phan tu
3.in ra danh sach
4.thoat
moi ban chon : 2

1.them vao 1 phan tu
2.lay ra 1 phan tu
3.in ra danh sach
4.thoat
moi ban chon : 3
5
3

1.them vao 1 phan tu
2.lay ra 1 phan tu
3.in ra danh sach
4.thoat
moi ban chon : 4

-----
Process exited after 19.32 seconds with return code 0

```

2. Bài 2

i. Đề bài: Chỉnh sửa lại lớp Stack ở câu 2.1

- Lớp **Node**:

- Bổ sung thành viên dữ liệu id (identification number). Mỗi khi một thẻ hiện mới của Node được tạo, giá trị cho id được sinh tự động theo cơ chế tăng dần (tương tự kiểu auto number của MSAccess) bắt đầu từ 1.

- Bổ sung phương thức print() để in ra nội dung của Node gồm id và item nằm trên 1 dòng, tách nhau bằng 1 dấu tab.
- Bổ sung copy constructor cho Node.
- Lớp **Stack**:
 - Sửa các phương thức bị ảnh hưởng bởi thay đổi của Node.
 - Thêm hằng thành viên (const data member) maxSize để giới hạn kích thước ngăn xếp, giá trị này được người dùng cung cấp khi khai báo một đối tượng ngăn xếp (chú ý constructor)
 - Thêm phương thức isFull() để kiểm tra xem ngăn xếp đã đầy hay chưa. trong các phương thức hiện có, hãy sửa nội dung các phương thức có liên quan đến kích thước tối đa của ngăn xếp, chẳng hạn push().
 - Bổ sung copy constructor cho lớp Stack.

j. Thuật toán

- Sửa lớp **Node**:
 - Sử dụng biến tĩnh **_id** để gán vào **id** của node tương ứng, khi một node được tạo ra, thì **_id** được tăng lên 1 và gán vào **id**, còn **item** được gán bởi biến truyền vào x.
 - Phương thức Print(): in ra **id** và **item** trên cùng một dòng.
 - Copy constructor: tăng **_id** lên 1 và gán cho **id**, **item** được gán bằng **item** của Node truyền vào.
- Sửa lớp **Stack**:
 - Sửa lại phương thức Pop() trả về 1 node và phương thức in các node trong Stack bằng cách Pop() ra từng node và dùng phương thức PrintNode() để in ra.
 - Khai báo thành viên hằng maxSize để giới hạn kích thước của Stack, maxSize được gán bởi giá trị truyền vào từ hàm main thông qua danh sách khởi tạo thành viên, không thể gán trực tiếp trong hàm. Phương thức isFull() được trả về **true** nếu số phần tử hiện tại bằng kích thước của ngăn xếp, ngược lại trả về **false**. Khi Push() một giá trị vào ngăn xếp thì kiểm tra nếu chưa đầy **isFull() == false** thì mới cho phép Push() vào.
 - Copy constructor cho ngăn xếp: Pop() tất cả các phần tử của ngăn xếp ra một mảng a, sau đó Push vào lại từng phần tử cho từng ngăn xếp.

k. Mã nguồn

- File **node.h**

```
#ifndef NODE_H
#define NODE_H

class node
{
public:
    node(const int item=1); // ham dung
    ~node(); // ham huy
    node(const node &x); // ham dung copy
    void printNode(); // ham in ra item va id
    int getItem(); // ham lay gia tri item
    node *getNext(); // ham tra ve con tro this->next
    int getId(); // ham tra ve gia tri id
    void setNext(node *x);
    void dec_ID(); // ham nay se giam thuoc tinh tinh moi khi pop phan tu ra

private:
    int item;
    node *next;
```

```

        int id;
        //thuoc tinh tinh
        static int _id;
    };

#endif
- File node.cpp
#include "node.h"
#include<iostream>
using namespace std;
int node::_id=0;

        node::node(const int item)
        {
            this->item=item;
            this->next=NULL;
            _id++;
            id=_id;
        }
        node::~~node()
        {
            this->next=NULL;
        }
        node::node(const node &x)
        {
            this->item=x.item;
            this->id=x.id;
            this->next=x.next;
        }
        void node::printNode()
        {
            cout<<id<<" "<<item<<endl;
        }
        int node::getItem()
        {
            return item;
        }
        node* node::getNext()
        {
            return next;
        }
        int node::getId()
        {
            return id;
        }
        void node::setNext(node *x)
        {
            this->next=x;
        }
        void node::dec_ID()
        {
            id--;
        }

- File stack.h
#ifndef STACK_H
#define STACK_H
#include "node.h"
class stack
{
    public:
        stack(const int &maxsize = 3);
        ~stack();
        bool isEmpty();
        bool isFull();
        void push(const node &x);
        int pop();
        void print();
        stack(const stack &x);
    private:
        int size;
        node *top;
        const int MAX_SIZE;

```

```

};

#endif
- File stack.cpp
#include "stack.h"
#include "node.h"
#include <iostream>
using namespace std;

stack::stack(const int &maxsize):MAX_SIZE(maxsize)//ham dung
{
    this->top=NULL;// cho dinh top = gia tri NULL
    this->size=0;// size =0
}
stack::~stack()//ham huy
{
    while(size) pop();//neu con size thi lay ra
    system("cls");//xoa man hinh
}
bool stack::isEmpty()//hamkiem tra ngan xep co rong hay khong
{
    return (this->size==0);// tra ve true neu size=0
}
bool stack::isFull()//hamkiem tra ngan xep co day khong
{
    return (this->size>=MAX_SIZE);// tra ve true neu size vuot qua MAX_SIZE
}
void stack::push(const node &x)// ham day node vao ngan xep
{
    node *p=new node(x);// tao con tro node p vung nho sao chep node x
    p->setNext(this->top);//next cua node p la top ngan xep hien tai
    this->top=p;//gan top hien tai bang p
    size++;//tang size len 1
}
int stack::pop()//ham lay ra node
{
    if(isEmpty()) cout<<"ngan xep rong ";
    else{
        node *p=top;//con tro node p tro den top
        int item=top->getItem();//item nhan item cua node top
        top=top->getNext();//gan top bang next cua top
        size--;//giam size di 1
        p->dec_ID();//giam ID xuong 1
        p->setNext(NULL);
        p->~node();//huy node p
        cout<<"da lay ra : "<<item<<endl;
        return item;//tra ve item
    }
}
void stack::print()//ham in ra cac phan tu trong ngan xep
{
    if(isEmpty()) cout<<"ngan xep rong ";
    else
    {
        node *p=top;//con tro node p tro den top
        while(p)//trong khi top != NULL
        {
            p->printNode();//in ra node
            p=p->getNext();//p gan bang next cua p
        }
    }
}
stack::stack(const stack &x):MAX_SIZE(x.MAX_SIZE)
{
    node *p=x.top;//con tro node p bang top cua x
    stack st;//tao stack st rong
    int *item = new int [x.size];//tao mang dong item co kich thuoc bang size cua x
    for(int i=0;i<x.size;i++)
    {
        item[i]=p->getItem();
        p=p->getNext();
    }
}

```

```

        top=NULL;
        size=0;
        for(int i=size-1;i>0;i++)
        {
            this->push(item[i]);
        }
        delete item;
    }
}

```

- File ***stack_demo.cpp***

```

#include <iostream>
#include "node.h"
#include "stack.h"
using namespace std;
/* run this program using the console pauser or add your own getch, system("pause") or input loop */

```

```

int main() {
    int chon,item,maxSize;

    cout<<" nhap so phan tu toi da cho stack:";
    cin>>maxSize;
    stack p(maxSize);
    do
    {
        cout<<"\n 1.them 1 phan tu vao stack";
        cout<<"\n 2.lay ra 1 phan tu trong stack";
        cout<<"\n 3.copy stack nay voi stack khac";
        cout<<"\n 4.in ra cac phan tu trong stack";
        cout<<"\n 5.thoat";
        cout<<"\n moi ban chon : ";
        cin>>chon;
        switch(chon)
        {
            case 1:
            {
                cout<<"\n nhap vao item : ";
                cin>>item;
                node *_node = new node(item);
                p.push(*_node);
                break;
            }
            case 2:
            {
                p.pop();
                cout<<"\n Done! "<<endl;
                p.print();
                break;
            }
            case 3:
            {
                cout<<"\n copy thong tin qua stack ";
                stack *k;
                k = new stack(p);
                k->print();
                break;
            }
            case 4:
            {
                cout<<"\n cac phan tu trong stack "<<endl;
                p.print();
                break;
            }
        }
    } while(chon!=5);
    return 0;
}

```

Demo

```
2.lay ra 1 phan tu trong stack
3.copy stack nay voi stack khac
4.in ra cac phan tu trong stack
5.thoat
moi ban chon : 1

nhap vao item : 6

1.them 1 phan tu vao stack
2.lay ra 1 phan tu trong stack
3.copy stack nay voi stack khac
4.in ra cac phan tu trong stack
5.thoat
moi ban chon : 1

nhap vao item : 4

1.them 1 phan tu vao stack
2.lay ra 1 phan tu trong stack
3.copy stack nay voi stack khac
4.in ra cac phan tu trong stack
5.thoat
moi ban chon : 1

nhap vao item : 8

1.them 1 phan tu vao stack
2.lay ra 1 phan tu trong stack
3.copy stack nay voi stack khac
4.in ra cac phan tu trong stack
5.thoat
moi ban chon : 4

cac phan tu trong stack
4 8
3 4
2 6
1 4

1.them 1 phan tu vao stack
2.lay ra 1 phan tu trong stack
3.copy stack nay voi stack khac
4.in ra cac phan tu trong stack
5.thoat
moi ban chon : 2
da lay ra : 8

Done!
3 4
2 6
1 4

1.them 1 phan tu vao stack
2.lay ra 1 phan tu trong stack
3.copy stack nay voi stack khac
4.in ra cac phan tu trong stack
5.thoat
moi ban chon :
```

3. Bài 3.1

I. Đề bài: Cài đặt lớp Vecto

Cài đặt lớp Vecto với các phép toán: +, -, *, =, <<, >>, [] để truy xuất tọa độ vecto, ~ để tính độ dài vecto.

m.Thuật toán

- Khai báo số chiều của vecto **n** kiểu int, mảng **data** được cấp phát động để lưu tọa độ của vecto.
- Toán tử +(tính tổng 2 vecto trả về là một vecto): dựng một vecto t có số chiều bằng số chiều của vecto đang tham chiếu. Nếu số chiều của vecto đang tham chiếu bằng số chiều của vecto truyền vào x thì cộng lần lượt từng tọa độ của vecto hiện tại với vecto x lưu vào vecto t. Sau đó trả về vecto t. Nếu không cùng số chiều thì in ra dòng không cùng số chiều

- Toán tử - (tính hiệu 2 vecto trả về là một vecto): giống như toán tử cộng, nhưng thay bằng phép trừ.
- Toán tử * (tính tích vô hướng 2 vecto trả về là một giá trị): khai báo biến **k** kiểu dữ liệu float, nếu vecto đang tham chiếu và vecto x cùng số chiều thì nhân lần lượt tọa độ vecto đang tham chiếu với tọa độ của vecto x rồi cộng vào k, trả về biến k. Nếu không cùng số chiều thì trả về giá trị 0
- Toán tử = (gán vecto x cho vecto đang tham chiếu trả về vecto đang tham chiếu): nếu x gán cho x thì không làm gì cả, ngược lại thu hồi vùng nhớ cũ của vecto đang tham chiếu rồi cấp phát vùng nhớ mới, copy từng giá trị của vecto x vào vecto đang tham chiếu.
- Toán tử << (in ra các tọa độ của vecto x trả về biến tham chiếu kiểu ostream): dùng vòng for in ra tất cả các giá trị của mảng data.
- Toán tử >> (nhập các tọa độ của vecto x trả về biến tham chiếu kiểu istream): dùng vòng for nhập vào từng tọa độ của vecto x.
- Toán tử [] (truy xuất tọa độ của vecto): trả về giá trị tọa độ của vecto x
- Toán tử ~ (tính độ dài của vecto): dùng vòng lặp for và tính độ dài của

vecto theo công thức $\sqrt{\sum_{i=0}^{n-1} x_i^2}$

n. Mã nguồn

- File *vecto.h*

```
#ifndef VECTO_H
#define VECTO_H
#include <iostream>
using namespace std;
class vecto
{
    public:
        vecto(int n=3);
        ~vecto();
        friend istream& operator >> (istream &in, vecto &x);
        friend ostream& operator << (ostream &out, vecto &x);
        vecto operator +(const vecto &x);
        vecto operator -(const vecto &x);
        float operator *(const vecto &x);
        vecto(const vecto &x);
        vecto& operator = (const vecto &x);
        int operator [] (int i);
        float operator ~ ();

    private:
        int n;
        int *data;
};

#endif
```

- File *vecto.cpp*

```
#include "vecto.h"
#include <iostream>
#include <math.h>
using namespace std;

vecto::vecto(int x):n(x)
{
    data=new int [n];
    for (int i=0;i<n;i++)
    {
        data[i]=0;
    }
}

vecto::~~vecto()
{
    delete [] data;
}

istream& operator >> (istream &in, vecto &x)
```

```

{
    for(int i=0;i<x.n;i++)
    {
        in>>x.data[i];
    }
    return in;
}
ostream& operator << (ostream &out, vecto &x)
{
    for(int i=0;i<x.n;i++)
    {
        out<<x.data[i]<<" ";
    }
    out<<endl;
    return out;
}
vecto vecto::operator +(const vecto &x)
{
    vecto t(n);
    if(n==x.n)
    {
        for(int i=0;i<n;i++)
        {
            t.data[i]=data[i]+x.data[i];
        }
    }
    else
    {
        cout<<" khong cung so chieu ";
    }
    return t;
}
vecto vecto::operator - (const vecto &x)
{
    vecto t(n);
    if(n==x.n)
    {
        for(int i=0;i<n;i++)
        {
            t.data[i]=data[i]-x.data[i];
        }
    }
    else
    {
        cout<<" khong cung so chieu ";
    }
    return t;
}
float vecto::operator * (const vecto &x)
{
    float k=0;
    if(n==x.n)
    {
        for(int i=0;i<n;i++)
        {
            k+=data[i]*x.data[i];
        }
    }
    else cout<<"khong cung so chieu";
    return k;
}
vecto::vecto(const vecto &x):n(x.n)
{
    n=x.n;
    for (int i=0;i<n;i++)
    {
        data[i]=x.data[i];
    }
}
vecto& vecto::operator = (const vecto &x)

```

```

{
    if(this!=&x)
    {
        int n=x.n;
        delete [] data;
        data = new int [n];
        for(int i=0;i<n;i++)
        {
            data[i]=x.data[i];
        }
        return *this;
    }
}
int vecto::operator [] (int i)
{
    return data[i];
}
float vecto::operator ~ ()
{
    float k=0;
    for(int i=0;i<n;i++)
    {
        k+=data[i]*data[i];
    }
    return sqrt(k);
}

```

- File ***demo_vecto.cpp***

/* run this program using the console pauser or add your own getch, system("pause") or input loop */

```

#include <iostream>

#include "vecto.h"

using namespace std;

main() {
    int n;

    cout<<"\n nhap vao so chieu cua vecto thu 1 : ";
    cin>>n;

    vecto a(n);

    cout<<"\n nhap vao cac phan tu trong vecto thu 1 \n";
    cin>>a;

    cout<<"vecto thu 1 so chieu : "<<n<<endl;
    cout<<"cac phan tu cua vecto : "<<a;

    int m;

    cout<<"\n nhap vao so chieu cua vecto thu 2 : ";
    cin>>m;

    vecto b(m);

    cout<<"\n nhap vao cac phan tu trong vecto thu 1 \n";
    cin>>b;

    cout<<"vecto thu 2 co so chieu : "<<m<<endl;
    cout<<"cac phan tu cua vecto "<<b;

    vecto c(n);

    c=a+b;

    cout<<"\n tong cua hai vecto a va b : "<<endl;
}

```

```

cout<<"cac phan tu cua vecto tong : "<<c;

vecto d(n);

d =a-b;

cout<<"\n hieu cua hai vecto a va b : "<<endl;

cout<<"cac phan tu cua vecto hieu : "<<d;

float k;

k=a*b;

cout<<"\n tich cua hai vecto a va b la : "<<k;

cout<<endl;

vecto h(n);

h=a;

cout<<"vecto h gan bang vecto a co so chieu la : "<<n<<endl;

cout<<h;

float dai= ~a;

cout<<"do dai cua vecto a : "<<dai;

cout<<endl;

int tc=a[2];

cout<<"toa do vecto o i = 2 la : "<<tc;

}

```

d. Demo

```

D:\Ky 6\Thuc Hanh Huong Doi Tuong\ThucHanh1\vect...
nhap vao so chieu cua vecto thu 1 : 3
nhap vao cac phan tu trong vecto thu 1
1
2
3
vecto thu 1 so chieu : 3
cac phan tu cua vecto : 1 2 3

nhap vao so chieu cua vecto thu 2 : 3
nhap vao cac phan tu trong vecto thu 1
4
2
5
vecto thu 2 co so chieu : 3
cac phan tu cua vecto 4 2 5

tong cua hai vecto a va b :
cac phan tu cua vecto tong : 5 4 8

hieu cua hai vecto a va b :
cac phan tu cua vecto hieu : -3 0 -2

tich cua hai vecto a va b la : 23
vecto h gan bang vecto a co so chieu la : 3
1 2 3
do dai cua vecto a : 3.74166
toa do vecto o i = 2 la : 3

Process exited after 9.65 seconds with return value 0
Press any key to continue . . .

```

4. Bài 3.2

O. Đề bài:Cài đặt lớp Matran

Cài đặt lớp Matran với các phép toán: +, -, *, =, <<, >>, () để truy xuất phần tử của ma trận, ~ để tính định thức của ma trận vuông.

P. Thuật toán

- Khai báo **n** là số hàng, **m** là số cột của ma trận, **data** là mảng 2 chiều kiểu **int** cấp phát động.
- Toán tử + (Tính tổng hai ma trận trả về một ma trận): nếu số hàng và số cột của ma trận đang tham chiếu và ma trận x bằng nhau thì dựng ma trận t có số hàng và số cột bằng ma trận đang tham chiếu, dùng 2 vòng for lồng nhau để cộng các phần tử của từng hàng từng cột tương ứng cho nhau lưu vào ma trận t. Cuối cùng trả về ma trận t.
- Toán tử - (Tính hiệu hai ma trận trả về là một ma trận): thuật toán tương tự toán tử +, chỉ thay phép cộng bởi phép trừ.
- Toán tử * (nhân 2 ma trận, trả về một ma trận): nếu số hàng của ma trận đang tham chiếu bằng số cột của ma trận x thì dựng ma trận t có số hàng bằng số hàng ma trận đang tham chiếu, số cột bằng số cột của x. Tính tích hai ma trận theo công thức sau:
 - Lặp i = 0 ... số hàng-1
Lặp j = 0 ... số cột của x-1
$$t(i,j) = 0$$
$$\text{Lặp } k = 0 \dots \text{số cột}-1$$
$$t(i,j) += \text{data}[i][k] * x(k,j)$$
- Toán tử = (gán ma trận x vào ma trận đang tham chiếu): nếu ma trận đang tham chiếu khác x thì thu hồi vùng nhớ cũ của ma trận đang tham chiếu, gán lại số hàng, số cột của ma trận đang tham chiếu bằng số hàng, số cột của x, cấp phát mảng data với số hàng số cột mới, chép dữ liệu từ ma trận x sang ma trận đang tham chiếu và trả về ma trận đang tham chiếu.
- Toán tử << (in ra ma trận x trả về biến ostream):
 - Lặp i = 0 ... số hàng của x -1
Lặp j = 0 ... số cột của x -1
In ra x(i,j)
Xuống dòng
- Toán tử >> (nhập các hệ số của ma trận x trả về biến istream):
 - Lặp i = 0 ... số hàng của x -1
Lặp j = 0 ... số cột của x -1
Nhập vào x(i,j)
- Toán tử () (truy xuất hệ số của ma trận): trả về giá trị hệ số của ma trận
- Toán tử ~ (tính định thức ma trận vuông, trả về một giá trị kiểu float): nếu số hàng bằng số cột thì:
 - Xét n=1 thì trả về data[0][0]
 - Xét n=2 thì trả về data[0][0]*data[1][1]-data[0][1]*data[1][0];
 - Cho 1 giá trị tem=0;
 - Cho l chạy từ 0->n
Tem +=data[0][l]*BU(0,i)
 - Trả về tem
 - Tính phần Bù đại số qua hàm BU(l,j)

q. Mã nguồn

- File **matran.h**

```

#ifndef MATRAN_H
#define MATRAN_H
#include<iostream>
using namespace std;
class matran
{
    public:
        matran(int n=2,int m=2);//ham dung
        ~matran();//ham huy
        friend istream& operator >>(istream &in,const matran &x);//ham nhap
        friend ostream& operator <<(ostream &out,const matran &x);//ham xuat
        matran operator +(const matran &x);//ham cong hai ma tran
        matran operator -(const matran &x);//ham tru hai ma tran
        matran operator *(const matran &x);//ham nhan hai ma tran
        matran& operator =(const matran &x);//ham gan ma tran
        int operator()(int i,int j);//ham truy xuat gia tri he so cua ma tran
        matran(const matran &x);//ham dung sao chep
        float operator ~();//ham tinh dinh thuc cua ma tran vuong
        float BU(int i,int j)const;//tinh phan bu dai so
        matran mini_matran(int indexrow,int indexcolumn)const;//tim ma tran mini de tinh phan
        tu dai so
    private:
        int m,n;
        int **data;
};

#endif
- File matran.cpp
#include "matran.h"
#include<iostream>
#include <istream>
#include<cmath>
using namespace std;
matran::matran(int n,int m)
{
    this->n=n;
    this->m=m;
    data = new int* [n];
    if(!data){
        cout<<"Error!";
        exit(1);
    }
    for(int i=0;i<n;i++){
        data[i] =new int [m];
        if(!data[i]){
            cout<<"Error!";
            exit(1);
        }
        for(int j=0;j<m;j++){
            data[i][j] = 0;
        }
    }
}
matran::~matran()
{
    for(int i=0;i<n;i++) delete [] data[i];
    delete data;
}
int matran::operator()(int i,int j)
{
    return data[i][j];
}
istream& operator >> (istream &in,const matran &x)
{
    for(int i=0;i<x.n;i++)
        for(int j=0;j<x.m;j++)
        {
            cout<<"a["<<i<<"]["<<j<<"]=" ";
            fflush(stdin);
            cin>>x.data[i][j];
        }
}

```

```

        return in;
    }
    ostream& operator <<(ostream &out,const matran &x)
    {
        for(int i=0;i<x.n;i++)
        {
            cout<<"\n";
            for(int j=0;j<x.m;j++)
            {
                cout<<" "<<x.data[i][j];
            }
        }
        return out;
    }
    matran matran::operator +(const matran &x)
    {
        matran k(n,m);
        if((this->n==x.n)&&(this->m==x.m))
        {
            for(int i=0;i<n;i++)
            {
                for(int j=0;j<m;j++)
                {
                    k.data[i][j]=(*this).data[i][j]+x.data[i][j];
                }
            }
        }
        return k;
    }
    matran matran::operator -( matran &x)
    {
        matran k(n,m);
        if((this->n==x.n)&&(this->m==x.m))
        {
            for(int i=0;i<n;i++)
            {
                for(int j=0;j<m;j++)
                {
                    k.data[i][j]=(*this).data[i][j]-x.data[i][j];
                }
            }
        }
        else cout<<"\n ko the tru dc"<<endl;
        return k;
    }
    matran matran::operator *(const matran &x)
    {
        if(m==x.n)
        {
            matran tem(n,x.m);
            for(int i=0;i<n;i++)
            {
                for(int j=0;j<m;j++)
                {
                    tem.data[i][j]=0;
                }
                for(int k=0;k<n;k++)
                {
                    tem.data[i][j]+=(*this).data[i][k]*x.data[k][j];
                }
            }
            return tem;
        }
    }
    matran& matran::operator =(const matran &x)
    {
        for(int i=0;i<m;i++) delete data[i];
        delete data;
        m=x.m;
        n=x.n;
        data=new int*[n];
    }

```

```

        for(int i=0;i<n;i++) data[i]=new int [m];
        for(int i=0;i<n;i++)
        {
            for(int j=0;j<n;j++)
            {
                data[i][j]=x.data[i][j];
            }
        }

        return *this;
    }

    matran::matran(const matran &x)
    {
        n=x.n;
        m=x.m;
        data=new int *[n];
        for(int i=0;i<n;i++) data[i]=new int[m];
        for(int i=0;i<n;i++)
        {
            for(int j=0;j<m;j++)
            {
                data[i][j]=x.data[i][j];
            }
        }
    }

    float matran::operator ~ ()
    {
        if(m!=n) return 0;
        if(n==1) return data[0][0];
        if(n==2) return data[0][0]*data[1][1]-data[0][1]*data[1][0];
        float tem=0;
        for(int i=0;i<n;i++) tem+=data[0][i]*BU(0,i);    //tinh de quy theo hang 0
        return tem;
    }

    float matran::BU(int i,int j)const
    {
        return pow(-1,(i+j))*(~mini_matran(i,j));
    }

    matran matran::mini_matran(int indexrow,int indexcolumn)const
    {
        matran t(n-1,m-1);
        for(int i=0;i<n;i++){
            if(i!=indexrow)
            {
                for(int j=0;j<n;j++)
                {
                    if(j!=indexcolumn)
                    {
                        t.data[i-(i>indexrow)][j-
(j>indexcolumn)]=data[i][j];
                    }
                }
            }
        }

        return t;
    }
}

```

- **File *demo_matran.cpp***

```

#include <iostream>
#include "matran.h"
using namespace std;
/* run this program using the console pauser or add your own getch, system("pause") or input loop */

main() {
    int n1,m1;
    cout<<"nhap vao so hang cua ma tran 1: ";
    cin>>n1;
    cout<<"\n nhap vao so cot cua ma tran 1: ";
    cin>>m1;
    matran a(n1,m1);
}

```



```

cin>>a;
cout<<a;
int n2,m2;
cout<<"\n nhap vao hang cua ma tran 2: ";
cin>>n2;
cout<<"\n nhap vao so cot cua ma tran 2: ";
cin>>m2;
matran b(n2,m2);
cin>>b;
cout<<b;
if((n1!=n2)||m1!=m2)
{
    cout<<"\n khong the cong hai ma tran";
}
else
{
    cout<<"\n tong cua hai ma tran a va b "<<a+b;
}
if((n1!=n2)||m1!=m2)
{
    cout<<"\n khong the cong hai ma tran";
}
else
{
    cout<<"\n tong cua hai ma tran a va b "<<a-b;
}
matran k;
if(m1!=n2)
{
    cout<<"\n khong the nhan hai vecto nay";
}
else
{
    cout<<"\n tinh cua hai ma tran a va b "<<a*b;
}
b=a;
cout<<"\n ma tran b duoc gan bang ma tran a ";
cout<<b;
if(a.n==a.m){
cout<<"\n dinh thuc cua ma tran a : "<<~a;}

```

Demo

```
D:\Ky 6\Thuc Hanh Huong Doi Tuong\Thuc Hanh...
nhap vao so hang cua ma tran 1: 2
nhap vao so cot cua ma tran 1: 2
a[0][0]= 1
a[0][1]= 2
a[1][0]= 3
a[1][1]= 4

1 2
3 4
nhap vao hang cua ma tran 2: 3
nhap vao so cot cua ma tran 2: 3
a[0][0]= 1
a[0][1]= 2
a[0][2]= 3
a[1][0]= 4
a[1][1]= 5
a[1][2]= 6
a[2][0]= 4
a[2][1]= 3
a[2][2]= 2

1 2 3
4 5 6
4 3 2
khong the cong hai ma tran
khong the cong hai ma tran
khong the nhan hai vecto nay
ma tran b duoc gan bang ma tran a
1 2
3 4
dinh thuc cua ma tran a : -2
-----
Process exited after 12.2 seconds with return value 0
Press any key to continue . . .
```

5. Bài 3.3

r. Đề bài: Cài đặt lớp đa thức Polynomial

Cài đặt lớp đa thức Polynomial với các phép toán: +, -, *, =, <<, >>, [] để truy xuất hệ số của đa thức, () để tính giá trị của đa thức.

s. Thuật toán

- Khai báo **n** kiểu int là bậc của đa thức, mảng một chiều **data** cấp phát động để lưu hệ số của đa thức.
- Toán tử + (Tính tổng 2 đa thức trả về một đa thức): dựng đa thức k có bậc bằng n của đa thức hiện tại
So sánh nếu $n=n$ của đa thức x thì cho i chạy từ n đến 0, cộng các hệ số data tương ứng của this và x vào k.
Nếu $n>n$ của x thì cho i chạy từ n đến 0 và cộng các hệ số tương ứng của this và x vào k. Nếu ngược lại thì cho i chạy từ x.n đến 0 và cộng tương tự. Kết quả trả về k
- Toán tử - (Tính hiệu 2 đa thức trả về một đa thức): thuật toán giống toán tử +, thay phép cộng bằng phép trừ.
- Toán tử * (Nhân đa thức đang tham chiếu với một số): Dựng đa thức k có bậc bằng $n+x.n$, lặp từ $i=n \rightarrow 0$, và j chạy từ $x.n \rightarrow 0$ nhân hệ số của đa thức đang tham chiếu với giá trị x cộng và lưu vào đa thức k có bậc $i+j$, trả về đa thức k.
- Toán tử = (Gán đa thức x cho đa thức đang tham chiếu trả về một tham chiếu): Nếu đa thức cần gán khác đa thức x thì: thu hồi vùng nhớ cũ, gán

lại bậc đa thức bằng bậc của x, cấp phát vùng nhớ mới, chép dữ liệu từ đa thức x sang đa thức đang tham chiếu.

- Toán tử << (Toán tử trích): Lặp $i = n \rightarrow 0$, in ra các hệ số của đa thức trả về biến kiểu ostream.
- Toán tử >> (Toán tử chèn): Lặp $i = n \rightarrow 0$, nhập các hệ số của đa thức của đa thức x, trả về biến kiểu istream.
- Toán tử [] (truy xuất hệ số của đa thức): cần viết 2 phiên bản trả về một giá trị khi tham chiếu đến dữ liệu hằng, trả về địa chỉ nếu tham chiếu đến dữ liệu cần sửa đổi.
- Toán tử () (Tính giá trị của đa thức)
Khai báo 1 biến float gtri ;
Cho i chạy từ $n \rightarrow 0$: $gtri += data[i] * pow(x, i)$
Kết quả trả về gtri

- Mã nguồn

- File **dathuc.h**

```
#ifndef DATHUC_H
#define DATHUC_H
#include<iostream>
#include<cmath>
#include<istream>
using namespace std;
class dathuc
{
public:
    dathuc(int n=3);//ham khai dung
    ~dathuc();//ham huy
    dathuc operator + (const dathuc &x);// ham toan tu cong hai da thuc
    dathuc operator - (const dathuc &x);// toan tu tru hai da thuc
    dathuc operator * (const dathuc &x);// toan tu nhan hai ma thuc
    friend ostream &operator << (ostream &out,const dathuc &x);// ham nhap cac phan tu
    cua da thuc
    friend istream &operator >> (istream &in,dathuc &x);// ham xuat cac phan tu cua da thuc
    float operator [] (int i);//ham lay ra he so cua da thuc
    float operator () (int x);// ham tinh gia tri cua da thuc voi gia tri x
    dathuc(const dathuc &x);// ham dung sao chep
    dathuc& operator = (const dathuc &x);//ham toan tu gan

private:
    int n;// bac cua da thuc
    float *data;// mang dong luu gia tri cua
};
```

#endif

- File **dathuc.cpp**

```
#include<iostream>
#include"dathuc.h"
using namespace std;
dathuc::dathuc(int n)//ham khai dung
{
    this->n=n;
    data=new float [n+1];
    for(int i=n;i>=0;i--)
    {
        data[i]=0;
    }
}
dathuc::~dathuc();//ham huy
{
    delete [] data;
}
dathuc dathuc::operator + (const dathuc &x)// ham toan tu cong hai da thuc
{
    dathuc k(n);
    if(n==x.n)
    {
        for(int i=n;i>=0;i--)
```

```

        {
            k.data[i]=data[i]+x.data[i];
        }
    }
    if(n<x.n)
    {
        k.n=x.n;
        for(int i=x.n;i>=0;i--)
        {
            k.data[i]=x.data[i];
        }
        for(int i=n;i>=0;i--)
        {
            k.data[i]+=data[i];
        }
    }
    if(n>x.n)
    {
        for(int i=n;i>=0;i--)
        {
            k.data[i]=data[i];
        }
        for(int i=x.n;i>=0;i--)
        {
            k.data[i]+=x.data[i];
        }
    }
    return k;
}
dathuc dathuc::operator - (const dathuc &x)// toan tu tru hai da thuc
{
    dathuc k(x);
    for(int i=x.n;i>=0;i--)
    {
        k.data[i]=-k.data[i];
    }
    return k+(*this);
}
dathuc dathuc::operator * (const dathuc &x)// toan tu nhan hai ma thuc
{
    dathuc k(n+x.n);
    for(int i=n;i>=0;i--)
        for(int j=x.n;j>=0;j--)
        {
            k.data[i+j]+=data[i]*x.data[j];
        }
    return k;
}
ostream &operator << (ostream &out,const dathuc &x)// ham nhap cac phan tu cua da
thuc
{
    cout<<"\n cac he so cua da thuc la ( ";
    for(int i=x.n;i>0;i--)
    {
        out<<x.data[i]<<" , ";
    }
    cout<<x.data[0]<<" )";
    return out;
}
istream &operator >> (istream &in,dathuc &x)// ham xuat cac phan tu cua da thuc
{
    for(int i=x.n;i>=0;i--)
    {
        cout<<"\n he so tuong ung voi x*"<<i<<" la : ";
        in>>x.data[i];
    }
    return in;
}
float dathuc::operator [] (int i)//ham lay ra he so cua da thuc
{
    return data[i];
}

```

```

float dathuc::operator () (int x)// ham tinh gia tri cua da thuc voi gia tri x
{
    float gtri;
    for(int i=n;i>=0;i--)
    {
        gtri+=data[i]*pow(x,i);
    }
    return gtri;
}
dathuc::dathuc(const dathuc &x)// ham dung sao chep
{
    n=x.n;
    data=new float[n+1];
    for(int i=0;i<=n;i++){
        data[i]=x.data[i];
    }
}
dathuc& dathuc::operator = (const dathuc &x)//ham toan tu gan
{
    delete data;
    n=x.n;
    data=new float[n+1];
    for(int i=0;i<=n;i++){
        data[i]=x.data[i];
    }
}

```

- **File *dathuc_demo.cpp***

```

#include <iostream>
#include "dathuc.h"
using namespace std;
/* run this program using the console pauser or add your own getch, system("pause") or input loop */

main() {
    int n;

    cout<<"nhap vao bac cua da thuc 1 : ";
    cin>>n;
    dathuc a(n);
    cout<<"\n nhap vao cac phan tu cua da thuc 1 ";
    cin>>a;
    cout<<a;
    int m;
    cout<<"\n nhap vao bac cua da thuc 2 : ";
    cin>>m;
    dathuc b(m);
    cout<<"\n nhap vao cac phan tu cua da thuc 2 ";
    cin>>b;
    cout<<b;
    cout<<"\n cong hai da thuc \n";
    cout<<a+b;
    cout<<"\n tru hai da thuc \n";
    cout<<a-b;
    cout<<"\n nhan hai da thuc \n";
    cout<<a*b;
    cout<<"\n gan da thuc thu 2 bang da thuc thu 1 \n";
    b=a;
    cout<<b;
    cout<<"\n he so thu 2 cua da thuc 1 : "<<a[2];
    cout<<"\n gia tri cua da thu 1 tai gia tri x ";
    int x;
    cin>>x;
    cout<<" la : "<<a(x);
}

```

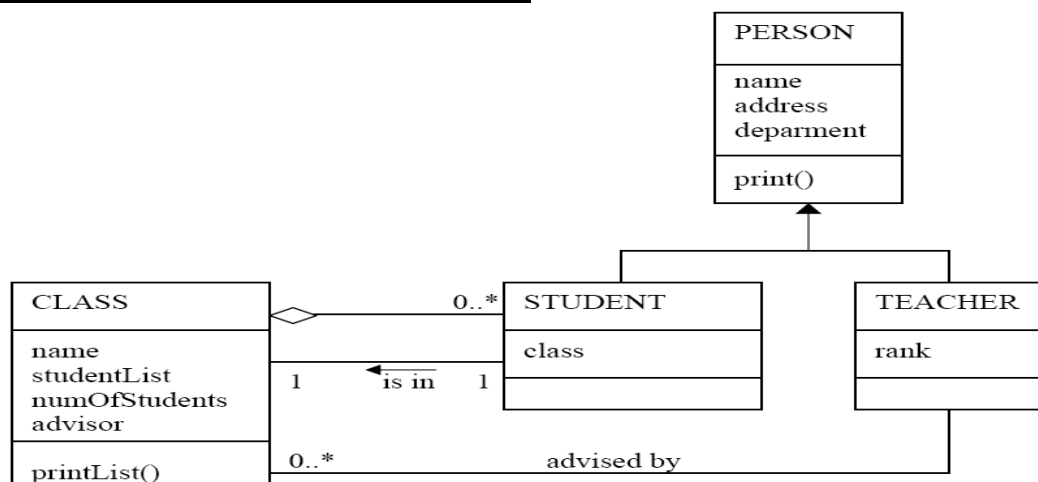
t. Demo

```
D:\Ky 6\Thuc Hanh Huong Doi Tuong\ThucHanh1\dathuc\dathu... - [X]
nhap vao bac cua da thuc 1 : 3
  nhap vao cac phan tu cua da thuc 1
  he so tuong ung voi x*3 la : 2
  he so tuong ung voi x*2 la : 3
  he so tuong ung voi x*1 la : 1
  he so tuong ung voi x*0 la : 4
  cac he so cua da thuc la < 2 , 3 , 1 , 4 >
  nhap vao bac cua da thuc 2 : 2
  nhap vao cac phan tu cua da thuc 2
  he so tuong ung voi x*2 la : 1
  he so tuong ung voi x*1 la : 2
  he so tuong ung voi x*0 la : 3
  cac he so cua da thuc la < 1 , 2 , 3 >
  cong hai da thuc
  cac he so cua da thuc la < 2 , 4 , 3 , 7 >
  tru hai da thuc
  cac he so cua da thuc la < 2 , 2 , -1 , 1 >
  nhan hai da thuc
  cac he so cua da thuc la < 2 , 7 , 13 , 15 , 11 , 12 >
  gan da thuc thu 2 bang da thuc thu 1
  cac he so cua da thuc la < 2 , 3 , 1 , 4 >
  he so thu 2 cua da thuc 1 : 3
  gia tri cua da thu 1 tai gia tri x 2
  la : 34
-----
Process exited after 12.62 seconds with return value 0
Press any key to continue . . .
```

III. BÀI THỰC HÀNH SỐ 3

6. Bài 1

u. Đề bài: Cài đặt chương trình quản lý lớp học



V. Thuật toán

- Lớp **person** khai báo 3 thành viên protected là **name**(tên), **address** (địa chỉ), **deparment** kiểu **char**, hàm dựng truyền vào 3 tham số để gán vào 3 biến trên. Hàm thuần ảo nhập và print.
- Lớp **teacher** : kế thừa public từ lớp **person**, khai báo thêm biến **rank** là cấp bậc của giáo viên, override lại 2 phương thức nhập và xuất thông tin của **teacher**.
- Lớp **student** : kế thừa public từ lớp **person**, khai báo thêm biến **class** để lưu lớp đang học, override lại 2 phương thức nhập và xuất thông tin của **student**.
- Lớp **class** : gồm có 4 thành viên dữ liệu **name** (tên lớp), **studentList** (là mảng có 50 con trỏ trỏ tới từng student được cấp phát động, **numOfStudents** (số lượng sinh viên), **advisor** (tên giáo viên chủ nhiệm). Có 2 phương thức là **nhappt** để nhập tên lớp, thông tin của giáo viên chủ nhiệm (cho advisor trỏ tới **teacher**), nhập thông tin sinh viên trong lớp bằng cách cho các con trỏ của **studentList** trỏ tới từng thành phần **student**, phương thức **printList** in ra thông tin của lớp bao gồm tên giáo viên, số lượng và danh sách sinh viên.

W. Mã nguồn

- File **person.h**

```
#ifndef PERSON_H
#define PERSON_H
#include<string.h>
#include<iostream>
class person
{
    public:
        person(char s1[]="",char s2[]="",char s3[]="");
        ~person();
        virtual void print()=0;
        virtual void nhap()=0;

    protected:
        char name[30];
        char address[30];
        char deparment[30];
};

#endif
```
- File **person.cpp**

```
#include "person.h"
person::person(char *s1,char *s2,char *s3)
{
    strcpy(name,s1);
    strcpy(address,s2);
    strcpy(deparment,s3);
}
person::~~person(){
}
```
- File **teacher.h**

```
#ifndef TEACHER_H
#define TEACHER_H
#include "person.h"
class teacher: public person
{
    public:
        teacher(char s1[]="",char s2[]="",char s3[]="",char s4[]="");
        ~teacher();
        virtual void nhap();
        virtual void print();

    private:
        char rank[20];
};
```

```

    #endif
-   File teacher.cpp
#include "teacher.h"
#include <iostream>
using namespace std;

teacher::teacher(char *s1,char *s2,char *s3,char *s4)
{
    strcpy(name,s1);
    strcpy(address,s2);
    strcpy(deparment,s3);
    strcpy(rank,s4);
}

teacher::~teacher()
{

}

void teacher::nhap()
{
    cout<<"thong tin cua giang vien "<<endl;
    cout<<"name : "; fflush(stdin); cin.getline(name,30);
    cout<<"address : "; fflush(stdin); cin.getline(address,30);
    cout<<"deparment : "; fflush(stdin); cin.getline(deparment,30);
    cout<<"rank : "; fflush(stdin); cin.getline(rank,20);
}

void teacher::print()
{
    cout<<"Name : "<<name<<endl;
    cout<<"Address : "<<address<<endl;
    cout<<"Deparment : "<<deparment<<endl;
    cout<<"Rank : "<<rank<<endl;
}
}

```

- File **student.h**

```

#ifndef STUDENT_H
#define STUDENT_H

#include <iostream>
#include <string.h>
#include "person.h"

class student:public person
{
public:
    student(char s1[]=" ",char s2[]=" ",char s3[]=" ",char s4[]=" ");
    ~student();
    virtual void nhap();
    virtual void print();

private:
    char Class[30];
};

#endif

#endif // STUDENT_H

```

- File **student.cpp**

```

#include "student.h"
#include <iostream>
using namespace std;

student::student(char *s1,char *s2,char *s3,char *s4)
{

```



```

        strcpy(name,s1);
        strcpy(address,s2);
        strcpy(deparment,s3);
        strcpy(Class,s4);
    }
    student::~student(){
    }
    void student::nhap()
    {
        cout<<"thong tin cua hoc sinh "<<endl;
        cout<<"Name : "; fflush(stdin); cin.getline(name,30);
        cout<<"Address : "; fflush(stdin); cin.getline(address,30);
        cout<<"Deparment : "; fflush(stdin); cin.getline(deparment,30);
        cout<<"Class: "; fflush(stdin); cin.getline(Class,30);
    }
    void student::print()
    {
        cout<<"Name : "<<name<<endl;
        cout<<"\t\tAddress : "<<address<<endl;
        cout<<"\t\tDeparment : "<<deparment<<endl;
        cout<<"\t\tClass : "<<Class<<endl;
    }
}

```

- File **cclass.h**

```

#ifndef CCLASS_H
#define CCLASS_H
#include "person.h"
#include "student.h"
#include "teacher.h"
class cclass
{
    public:
        cclass();
        ~cclass();
        void nhaptt();
        void printList();
    private:
        char name[30];
        person *studentList[50];
        int numofStudents;
        person *advisor;
};

#endif

```

- File **cclass.cpp**

```

#include "cclass.h"
#include <iostream>
using namespace std;
cclass::cclass()
{
    for(int i=0;i<50;i++)
    {
        studentList[i]=NULL;
    }
}
cclass::~cclass(){
}
void cclass::nhaptt()
{
    cout<<"Advisor. "<<endl;
    advisor = new teacher();
    advisor->nhap();
    cout<<"thong tin cua lop "<<endl;
    cout<<"Name Class: "; fflush(stdin); cin.getline(name,30);
    cout<<"Number of Student : "; cin>>numofStudents;
    for(int i=0;i<numofStudents;i++)
    {
        studentList[i]=new student(name);
        studentList[i]->nhap();
    }
}

```

```

void cclass::printList()
{
    cout<<endl;
    cout<<endl;

    cout<<"Advisor ";
    advisor->print();
    for(int i=0;i<numOfStudents;i++)
    {
        cout<<i+1<<"\t";
        studentList[i]->print();
    }
}

```

- File ***Class_Demo.cpp***

```

#include <iostream>
#include<windows.h>
#include"cclass.h"
using namespace std;

```

/* run this program using the console pauser or add your own getch, system("pause") or input loop */

```

main() {
    cclass a;
    a.nhaptt();
    cout<<"=====THONG TIN LOP HOC=====";
    a.printList();
    system("pause");
}

```

X. Demo

```

D:\Ky 6\Thuc Hanh Huong Doi Tuong\ThucHanh1\dathuc\dathu...
nhap vao hac cua da thuc 1 : 3
nhap vao cac phan tu cua da thuc 1
he so tuong ung voi x*3 la : 2
he so tuong ung voi x*2 la : 3
he so tuong ung voi x*1 la : 1
he so tuong ung voi x*0 la : 4
cac he so cua da thuc la < 2 , 3 , 1 , 4 >
nhap vao hac cua da thuc 2 : 2
nhap vao cac phan tu cua da thuc 2
he so tuong ung voi x*2 la : 1
he so tuong ung voi x*1 la : 2
he so tuong ung voi x*0 la : 3
cac he so cua da thuc la < 1 , 2 , 3 >
cong hai da thuc
cac he so cua da thuc la < 2 , 4 , 3 , 7 >
tru hai da thuc
cac he so cua da thuc la < 2 , 2 , -1 , 1 >
nhan hai da thuc
cac he so cua da thuc la < 2 , 7 , 13 , 15 , 11 , 12 >
gan da thuc thu 2 bang da thuc thu 1
cac he so cua da thuc la < 2 , 3 , 1 , 4 >
he so thu 2 cua da thuc 1 : 3
gia tri cua da thu 1 tai gia tri x 2
la : 34
-----
Process exited after 12.62 seconds with return value 0
Press any key to continue . . .

```

7. Bài 2

y. Đề bài: Thiết kế và cài đặt lớp MyString

- Hãy thiết kế và cài đặt lớp MyString (có hoạt động tương tự lớp string trong thư viện chuẩn C++).
- Lớp MyString sử dụng bộ nhớ động để lưu trữ chuỗi ký tự (chú ý việc cấp phát và thu hồi bộ nhớ tại các phương thức).
- Lớp MyString cần có:
 - constructor mặc định tạo chuỗi rỗng ("")
 - constructor chuyển đổi từ char* hoặc char[] sang MyString
 - copy constructor
- Cài các phép toán sau (overloaded operator) trên MyString:
 - +, & nối chuỗi
 - = gán chuỗi
 - ==, != so sánh
 - >= <=, so sánh chuỗi theo thứ tự từ điển
 - [] lấy ký tự nằm trong chuỗi, ví dụ s[2] cho ta ký tự số 2 trong chuỗi (đánh số các ký tự bắt đầu từ 0)
 - >>, << dùng để ghi/đọc chuỗi

z. Thuật toán

- Tạo lớp MyString gồm 2 thành viên dữ liệu: **n** dùng để đếm chiều dài chuỗi, mảng **data** kiểu char để lưu chuỗi vào được cấp phát động, dùng hàm hủy để thu hồi vùng nhớ đã cấp phát.
- Hàm dựng tạo mặc định chuỗi rỗng bằng cách gán **n** = 0, data rỗng. Hàm dựng chuyển từ chuỗi char*s sang MyString: cho data là mảng có độ dài bằng maxSize, nội dung mảng data được chép từ chuỗi s.
- Hàm dựng sao chép: gán **n** bằng **n** của x, cấp phát vùng nhớ cho data và sao chép nội dung data của x sang.
- Định nghĩa các phép toán:
 - Toán tử +, & dùng để nối chuỗi đang thao tác với chuỗi x: dựng một MyString t sao chép chuỗi hiện tại và có chiều dài bằng tổng của chuỗi đang thao tác với chuỗi x, cho biến i chạy từ 0->x.n rồi cộng từ phần tử vào chuỗi t. Kết quả trả về chuỗi t.
 - Toán tử gán = : Nếu chuỗi cần gán khác chuỗi x thì thu hồi vùng nhớ cũ, gán n bằng n của x, cấp phát vùng nhớ mới, sao chép dữ liệu từ x sang chuỗi đang thao tác.
 - Toán tử so sánh == : nếu chiều dài 2 chuỗi khác nhau thì trả về false, ngược lại kiểm tra từng vị trí tương ứng nếu nội dung khác nhau thì trả về false, nếu 2 chuỗi có chiều dài bằng nhau và các ký tự tương ứng giống nhau thì trả về true. Toán tử so sánh != : tương tự toán tử ==, chỉ đổi ngược true, false.
 - Toán tử so sánh theo từ điển <, <=, >, >= : dùng biến i để đếm, so sánh từ đầu đến vị trí cuối của chuỗi ngắn hơn, nếu có 1 cặp ký tự khác nhau thì trả về kết quả so sánh thứ tự của cặp ký tự trong từ điển, nếu giống nhau hoàn toàn thì trả về kết quả so sánh của độ dài chuỗi.
 - Toán tử lấy giá trị [] : trả về ký tự tại vị trí thứ i

- Toán tử >> dùng để đọc xâu vào dùng hàm getline để lấy một xâu sau đó đưa vào trong mảng data. Toán tử << dùng để in xâu data ra màn hình.

aa. Mã nguồn

- File *mystring.h*

```
#ifndef MYSTRING_H
#define MYSTRING_H
#include<iostream>
using namespace std;
class mystring
{
    public:
        mystring();//ham dung tao xau rong
        ~mystring();//ham huy
        mystring operator + (const mystring &x);//ham cong hai xau
        mystring operator & (const mystring &x);//ham cong hai xau
        mystring& operator = (const mystring &x);//ham gan
        bool operator == (const mystring &x);//ham so sanh ==
        bool operator != (const mystring &x);//ham so sanh !=
        bool operator > (const mystring &x);//ham so sanh >
        bool operator >= (const mystring &x);//ham so sanh >=
        bool operator <= (const mystring &x);//ham so sanh <=
        bool operator < (const mystring &x);//ham so sanh <
        char operator [] (int i);//ham lay ki tu cua xau tai vi tri i
        friend ostream& operator << (ostream &out, const mystring &x);//ham nhap
        friend istream& operator >> (istream &in, mystring &x);//ham xuat
        mystring(const mystring &x);//ham dung sao chep
        mystring(const char *str="");

    private:
        int n;
        char *data;
};

#endif
```

- File *mystring.cpp*

```
#include "mystring.h"
#include<iostream>
#include<stdio.h>
#define maxSize 32567
using namespace std;

mystring::mystring()
{
    this->n=maxSize;
    data = new char [n];
    data[0]='\0';
}

mystring::~mystring()
{
    delete [] data;
}

mystring mystring::operator + (const mystring &x)
{
    mystring tem(*this);
    tem.n+=x.n;
    for(int i=0;i<x.n;i++){
        tem.data[n+i]=x.data[i];
    }
    tem.data[tem.n]='\0';
    return tem;
}

mystring mystring::operator & (const mystring &x)
{
    mystring t(*this);
    t.n+=x.n;
    for(int i=0;i<x.n;i++){
        t.data[i]+=x.data[i];
    }
    return t;
}

mystring& mystring::operator = (const mystring &x)
{

```

```

        delete [] data;
        this->n=x.n;
        data=new char [n];
        for(int i=0;i<n;i++)
        {
            data[i]=x.data[i];
        }
        return *this;
    }
    bool mystring::operator == (const mystring &x)
    {
        if(n!=x.n) return 0;
        for(int i=0;i<n;i++)
            if(data[i]!=x.data[i]) return 0;
        return 1;
    }
    bool mystring::operator != (const mystring &x)
    {
        return !(*this==x);
    }
    bool mystring::operator > (const mystring &x)
    {
        return !(*this<=x);
    }
    bool mystring::operator >= (const mystring &x)
    {
        return !(*this<x);
    }
    bool mystring::operator <= (const mystring &x)
    {
        return (*this<x)||(*this==x);
    }
    bool mystring::operator < (const mystring &obj)
    {
        if(n<obj.n){
            for(int i=0;i<n;i++) if(data[i]>obj.data[i]) return 0;
            return 1;
        }
        else{
            for(int i=0;i<obj.n;i++) if(data[i]<obj.data[i]) return 1;
            return 0;
        }
    }
    char mystring::operator [] (int i)
    {
        return (*this).data[i];
    }
    ostream& operator << (ostream &out, const mystring &x)
    {
        out<<x.data;
        return out;
    }
    istream& operator >> (istream &in, mystring &x)
    {
        gets(x.data);
        for(x.n=0;x.n<maxSize;x.n++) if(x[x.n]=='\0') break;
        return in;
    }
    mystring::mystring(const mystring &x){
        data=new char[maxSize];
        n=x.n;
        for(int i=0;i<n;i++) data[i]=x.data[i];
        data[n]='\0';
    }

    mystring::mystring(const char *s)
    {
        data=new char[maxSize];
        for(int n=0;n<maxSize;n++)
        {
            data[n]=s[n];
            if(s[n]=='\0') break;

```

```

    }
}

- File mystring_demo.cpp
#include <iostream>
#include "mystring.h"
/* run this program using the console pauser or add your own getch, system("pause") or input loop */
using namespace std;
main() {
    mystring a,b;
    cout<<"\n nhap vao chuoi thu 1 : ";
    cin>>a;
    cout<<"\n nhap vao chuoi thu 2: ";
    cin>>b;
    cout<<"cac chuoi vua nhap la : \n";
    cout<<a<<endl;
    cout<<b;
    cout<<"\n ta co : "<<endl;
    if(a==b) cout<<"chuoi thu 1 == chuoi thu 2 \n";
    if(a>=b) cout<<"chuoi thu 1 >= chuoi thu 2 \n";
    if(a!=b) cout<<"chuoi thu 1 != chuoi thu 2 \n";
    if(a>b) cout<<"chuoi thu 1 > chuoi thu 2 \n";
    if(a<=b) cout<<"chuoi thu 1 <= chuoi thu 2 \n";
    if(a<b) cout<<"chuoi thu 1 < chuoi thu 2 \n";
    cout<<"\n cong chuoi thu 1 voi chuoi thu 2 : "<<a+b;
    cout<<"\n cong chuoi thu 2 voi chuoi thu 1 : "<<b+a;

}

```

bb.Demo

```

D:\Ky 6\Thuc Hanh Huong Doi Tuong\ThucHanh1\mystring\mys...
nhap vao chuoi thu 1 : bach
nhap vao chuoi thu 2: khoa
cac chuoi vua nhap la :
bach
khoa
ta co :
chuoi thu 1 != chuoi thu 2
chuoi thu 1 <= chuoi thu 2
chuoi thu 1 < chuoi thu 2

cong chuoi thu 1 voi chuoi thu 2 : bachkhoa
cong chuoi thu 2 voi chuoi thu 1 : khoabach
-----
Process exited after 2.888 seconds with return value 0
Press any key to continue . . .

```

8. Bài 3.1

cc.Đề bài

Cài đặt lớp Vecto là lớp template.

dd.Thuật toán

- Số chiều của vecto được truyền vào khi khởi tạo lớp template, khai báo mảng data kiểu T được cấp phát động để lưu tọa độ của vecto. Phương thức getN() để lấy số chiều vecto.
- Toán tử + để tính tổng hai vecto: dựng một vecto t có số chiều bằng số chiều của vecto đang tham chiếu. Nếu số chiều của vecto đang tham chiếu bằng số chiều của vecto truyền vào x thì cộng lần lượt từng tọa độ của vecto hiện tại với vecto x lưu vào vecto t. Sau đó trả về vecto t.
- Toán tử - để tính hiệu 2 vecto trả về là một vecto: giống như toán tử cộng, nhưng thay bằng phép trừ.

- Toán tử * để tính tích vô hướng 2 vecto trả về là một giá trị kiểu T: khai báo biến k kiểu float, nếu vecto đang tham chiếu và vecto x cùng số chiều thì nhân lần lượt tọa độ vecto đang tham chiếu với tọa độ của vecto x rồi cộng vào k, trả về biến k.
- Toán tử = (gán vecto x cho vecto đang tham chiếu trả về vecto đang tham chiếu): nếu x gán cho x thì không làm gì cả, ngược lại thu hồi vùng nhớ cũ của vecto đang tham chiếu rồi cấp phát vùng nhớ mới, copy từng giá trị của vecto x vào vecto đang tham chiếu.
- Toán tử << : dùng vòng for in ra tất cả các giá trị của mảng data.
- Toán tử >> (nhập các tọa độ của vecto x trả về biến tham chiếu kiểu istream): dùng vòng for nhập vào từng tọa độ của vecto x.
- Toán tử [] (truy xuất tọa độ của vecto): cần viết 2 phiên bản trả về một giá trị khi tham chiếu đến dữ liệu hằng, trả về địa chỉ nếu tham chiếu đến dữ liệu cần sửa đổi.
- Toán tử ~ (tính độ dài của vecto): dùng vòng lặp for và tính độ dài của

vecto theo công thức
$$\sqrt{\sum_{i=0}^{n-1} x_i^2}$$

ee. Mã nguồn

- File **vecto.h**

```
#ifndef VECTO_H
#define VECTO_H
#include <iostream>
using namespace std;
template <typename T>
class vecto
{
public:
    vecto(int n=3);
    ~vecto();
    vecto operator +(const vecto &x);
    vecto operator -(const vecto &x);
    float operator * (const vecto &x);
    vecto(const vecto &x);
    vecto& operator = (const vecto &x);
    T& operator [] (int i);
    void setN(int i);
    int getN() const;
    T operator [] (const int i)const;
    float operator ~ ();

private:
    int n;
    T *data;
};
```

#endif

- File **vecto.cpp**

```
#include "vecto.h"
#include <iostream>
#include <math.h>
using namespace std;
template <typename T>
vecto<T>::vecto(int x):n(x)
{
    data=new T [n];
    for (int i=0;i<n;i++)
    {
        data[i]=0;
    }
}
template <typename T>
vecto<T>::~~vecto()
{
    delete [] data;
```

```

}
template <typename T>
istream& operator >> (istream &in, vecto<T> &x)
{
    for(int i=0;i<x.getN();i++)
    {
        in>>x[i];
    }
    return in;
}
template <typename T>
ostream& operator << (ostream &out, vecto<T> &x)
{
    for(int i=0;i<x.getN();i++)
    {
        out<<x[i]<<" ";
    }
    out<<endl;
    return out;
}
template <typename T>
vecto<T> vecto<T>::operator +(const vecto<T> &x)
{
    vecto<T> t(n);
    if(n==x.n)
    {
        for(int i=0;i<n;i++)
        {
            t.data[i]=data[i]+x.data[i];
        }
    }
    else
    {
        cout<<" khong cung so chieu ";
    }
    return t;
}
template <typename T>
vecto<T> vecto<T>::operator - (const vecto<T> &x)
{
    vecto<T> t(n);
    if(n==x.n)
    {
        for(int i=0;i<n;i++)
        {
            t.data[i]=data[i]-x.data[i];
        }
    }
    else
    {
        cout<<" khong cung so chieu ";
    }
    return t;
}
template <typename T>
float vecto<T>::operator * (const vecto &x)
{
    float k=0;
    if(n==x.n)
    {
        for(int i=0;i<n;i++)
        {
            k+=data[i]*x.data[i];
        }
    }
    else cout<<"khong cung so chieu";
    return k;
}
template <typename T>
vecto<T>::vecto(const vecto<T> &x):n(x.n)
{

```



```

        n=x.n;
        for (int i=0;i<n;i++)
        {
            data[i]=x.data[i];
        }
    }
    template <typename T>
    vecto<T>& vecto<T>::operator = (const vecto<T> &x)
    {
        if(this!=&x)
        {
            int n=x.n;
            delete [] data;
            data = new int [n];
            for(int i=0;i<n;i++)
            {
                data[i]=x.data[i];
            }
            return *this;
        }
    }
    template <typename T>
    T& vecto<T>::operator [] (int i)
    {
        return data[i];
    }
    template <typename T>
    float vecto<T>::operator ~ ()
    {
        float k=0;
        for(int i=0;i<n;i++)
        {
            k+=data[i]*data[i];
        }
        return sqrt(k);
    }
    template <typename T>
    void vecto<T>::setN(int i)
    {
        this->n=i;
    }
    template <typename T>
    int vecto<T>::getN()const
    {
        return n;
    }
    template <typename T>
    T vecto<T>::operator [] (const int i)const
    {
        static T tmp = 0;
        return (i>=0 && i<n)? data[i]:tmp;
    }
}

```

- **File template *Vecto_demo.cpp***

/* run this program using the console pauser or add your own getch, system("pause") or input loop */

```
#include <iostream>
```

```
#include "vecto.cpp"
```

```
using namespace std;
```

```
main() {
```

```
    int n;
```

```
    cout<<"\n nhap vao so chieu cua vecto thu 1 : ";
```

```
    cin>>n;
```

```
    vecto<int> a(n);
```

```
    cout<<"\n nhap vao cac phan tu trong vecto thu 1 \n";
```

```
    cin>>a;
```

```
    cout<<"vecto thu 1 so chieu : "<<n<<endl;
```

```
    cout<<"cac phan tu cua vecto : "<<a;
```

```
    int m;
```

```
    cout<<"\n nhap vao so chieu cua vecto thu 2 : ";
```

```
    cin>>m;
```

```
    vecto<int> b(m);
```

```
    cout<<"\n nhap vao cac phan tu trong vecto thu 1 \n";
```

```

cin>>b;
cout<<"vecto thu 2 co so chieu : "<<m<<endl;
cout<<"cac phan tu cua vecto "<<b;
vecto<int> c(n);
c=a+b;
cout<<"\n tong cua hai vecto a va b : "<<endl;
cout<<"cac phan tu cua vecto tong : "<<c;
vecto<int> d(n);
d=a-b;
cout<<"\n hieu cua hai vecto a va b : "<<endl;
cout<<"cac phan tu cua vecto hieu : "<<d;
float k;
k=a*b;
cout<<"\n tich cua hai vecto a va b la : "<<k;
cout<<endl;
vecto<int> h(n);
h=a;
cout<<"vecto h gan bang vecto a co so chieu la : "<<n<<endl;
cout<<h;
float dai= ~a;
cout<<"do dai cua vecto a : "<<dai;
cout<<endl;
int tc=a[2];
cout<<"toa do vecto o i = 2 la : "<<tc;
}

```

ff. Demo

```

D:\Ky 6\Thuc Hanh Huong Doi Tuong\ThucHa...
nhap vao so chieu cua vecto thu 1 : 3
nhap vao cac phan tu trong vecto thu 1
2
1
2
vecto thu 1 so chieu : 3
cac phan tu cua vecto : 2 1 2

nhap vao so chieu cua vecto thu 2 : 3
nhap vao cac phan tu trong vecto thu 1
2
1
4
vecto thu 2 co so chieu : 3
cac phan tu cua vecto 2 1 4

tong cua hai vecto a va b :
cac phan tu cua vecto tong : 4 2 6

hieu cua hai vecto a va b :
cac phan tu cua vecto hieu : 0 0 -2

tich cua hai vecto a va b la : 13
vecto h gan bang vecto a co so chieu la : 3
2 1 2
do dai cua vecto a : 3
toa do vecto o i = 2 la : 2

-----
Process exited after 7.508 seconds with return value
Press any key to continue . . .

```

9. Bài 3.2

gg.Đề bài

Cài đặt lại lớp Matrix là lớp template.

hh.Thuật toán

- Khai báo data là mảng 2 chiều kiểu T cấp phát động. Phương thức getN() dùng để lấy cấp của ma trận

- Toán tử + (Tính tổng hai ma trận trả về một ma trận): Dựng ma trận template t, Nếu số hàng của this bằng với số cột của x, đồng thời số hàng của this bằng số cột của this thì dùng 2 vòng for lồng nhau để cộng các phần tử của từng hàng từng cột tương ứng cho nhau lưu vào ma trận t. Cuối cùng trả về ma trận t.
- Toán tử - (Tính hiệu hai ma trận trả về là một ma trận): thuật toán tương tự toán tử +, chỉ thay phép cộng bởi phép trừ.
- Toán tử * (nhân 2 ma trận, trả về một ma trận): So sánh nếu số cột của this bằng số hàng của ma trận x thì lập ma trận template t có số hàng là n của this, cột bằng m của x. Sau đó cho ba vòng for $i=0 \rightarrow n$, $j=0 \rightarrow m$, $k=0 \rightarrow n$ để data của tem nhận giá trị tổng giá trị data của this với giá trị data của x
- Toán tử = (gán ma trận x vào ma trận đang tham chiếu): nếu ma trận đang tham chiếu khác x thì thu hồi vùng nhớ cũ của ma trận đang tham chiếu, cấp phát lại mảng data, chép dữ liệu từ ma trận x sang ma trận đang tham chiếu và trả về ma trận đang tham chiếu.
- Toán tử << (in ra ma trận x trả về biến ostream):
 Lặp $i = 0 \dots N-1$
 Lặp $j = 0 \dots N-1$
 In ra $x(i,j)$
 Xuống dòng
- Toán tử >> (nhập các hệ số của ma trận x trả về biến istream):
 Lặp $i = 0 \dots N-1$
 Lặp $j = 0 \dots N-1$
 Nhập vào $x(i,j)$
- Toán tử () (truy xuất hệ số của ma trận): cần viết 2 phiên bản trả về một giá trị khi tham chiếu đến dữ liệu hằng, trả về địa chỉ nếu tham chiếu đến dữ liệu cần sửa đổi.

ii. Mã nguồn

- File **matran.h**

```
#ifndef MATRAN_H
#define MATRAN_H
#include<iostream>
using namespace std;
template <typename T>
class matran
{
public:
    matran(int n=2,int m=2);//ham dung
    ~matran();//ham huy
    matran operator +(const matran &x);//ham cong hai ma tran
    matran operator -(const matran &x);//ham tru hai ma tran
    matran operator *(const matran &x);//ham nhan hai ma tran
    matran& operator =(const matran &x);//ham gan ma tran
    T& operator()(int i,int j);//ham truy xuất gia trị he so cua ma tran
    matran(const matran &x);//ham dung sao chép
    float operator ~();//ham tính dinh thuc cua ma tran vuong
    float BU(int i,int j)const;//tính phân bố đại số
    matran mini_matran(int indexrow,int indexcolumn)const;//tim ma tran mini de tính phân
    tu đại số

    void setN(int i);
    int getN()const;
    T operator()(const int i,const int j)const;
    void setM(int j);
    int getM()const;

private:
    int m,n;
    T **data;
};
```

```

#endif
- File matran.cpp
#include "matran.h"
#include <iostream>
#include <istream>
#include <cmath>
using namespace std;

template <typename T>
matran<T>::matran(int n,int m)
{
    this->n=n;
    this->m=m;
    data = new T* [n];
    if(!data){
        cout<<"Error!";
        exit(1);
    }
    for(int i=0;i<n;i++){
        data[i]=new T [m];
        if(!data[i]){
            cout<<"Error!";
            exit(1);
        }
        for(int j=0;j<m;j++){
            data[i][j] = 0;
        }
    }
}

template <typename T>
matran<T>::~~matran()
{
    for(int i=0;i<n;i++) delete [] data[i];
    delete data;
}

template <typename T>
T& matran<T>::operator ()(int i,int j)
{
    return data[i][j];
}

template <typename T>
istream& operator >> (istream &in,matran<T> &x)
{
    for(int i=0;i<x.getN();i++)
        for(int j=0;j<x.getM();j++)
        {
            cout<<"a["<<i<<"]["<<j<<"]= ";
            fflush(stdin);
            in>>x(i,j);
        }

    return in;
}

template <typename T>
ostream& operator <<(ostream &out,const matran<T> &x)
{
    for(int i=0;i<x.getN();i++)
    {
        cout<<"\n";
        for(int j=0;j<x.getM();j++)
        {
            cout<<" "<<x(i,j);
        }
    }
    return out;
}

template <typename T>
matran<T> matran<T>::operator +(const matran<T> &x)
{
    matran<T> k(n,m);
    if((this->n==x.n)&&(this->m==x.m))
    {

```

```

        for(int i=0;i<n;i++)
        {
            for(int j=0;j<m;j++)
            {
                k.data[i][j]=(*this).data[i][j]+x.data[i][j];
            }
        }
        return k;
    }
}
template <typename T>
matran<T> matran<T>::operator -( matran<T> &x)
{
    matran<T> k(n,m);
    if((this->n==x.n)&&(this->m==x.m))
    {
        for(int i=0;i<n;i++)
        {
            for(int j=0;j<m;j++)
            {
                k.data[i][j]=(*this).data[i][j]-x.data[i][j];
            }
        }
    }
    else cout<<"\n ko the tru dc"<<endl;
    return k;
}
template <typename T>
matran<T> matran<T>::operator *(const matran<T> &x)
{
    if(m==x.n)
    {
        matran<T> tem(n,x.m);
        for(int i=0;i<n;i++)
        {
            for(int j=0;j<m;j++)
            {
                tem.data[i][j]=0;
            }
            for(int k=0;k<n;k++)
            {
                tem.data[i][j]+=(*this).data[i][k]*x.data[k][j];
            }
        }
        return tem;
    }
}
template <typename T>
matran<T>& matran<T>::operator =(const matran<T> &x)
{
    for(int i=0;i<m;i++) delete data[i];
    delete data;
    m=x.m;
    n=x.n;
    data=new T*[n];
    for(int i=0;i<n;i++) data[i]=new T [m];
    for(int i=0;i<n;i++)
    {
        for(int j=0;j<n;j++)
        {
            data[i][j]=x.data[i][j];
        }
    }

    return *this;
}
template <typename T>
matran<T>::matran(const matran<T> &x)
{
    n=x.n;
    m=x.m;
    data=new T *[n];

```

```

        for(int i=0;i<n;i++) data[i]=new T[m];
        for(int i=0;i<n;i++)
        {
            for(int j=0;j<m;j++)
            {
                data[i][j]=x.data[i][j];
            }
        }
    }
    template <typename T>
    float matran<T>::operator ~ ()
    {
        if(m!=n) return 0;
        if(n==1) return data[0][0];
        if(n==2) return data[0][0]*data[1][1]-data[0][1]*data[1][0];
        float tem=0;
        for(int i=0;i<n;i++) tem+=data[0][i]*BU(0,i);    //tinh de quy theo hang 0
        return tem;
    }
    template <typename T>
    float matran<T>::BU(int i,int j)const
    {
        return pow(-1,(i+j))*(-mini_matran(i,j));
    }
    template <typename T>
    matran<T> matran<T>::mini_matran(int indexrow,int indexcolumn)const
    {
        matran<T> t(n-1,m-1);
        for(int i=0;i<n;i++){
            if(i!=indexrow)
            {
                for(int j=0;j<n;j++)
                {
                    if(j!=indexcolumn)
                    {
                        t.data[i-(i>indexrow)][j-
(j>indexcolumn)]=data[i][j];
                    }
                }
            }
        }
        return t;
    }
    template <typename T>
    void matran<T>::setN(int i)
    {
        this->n=i;
    }
    template <typename T>
    int matran<T>::getN()const
    {
        return n;
    }
    template <typename T>
    T matran<T>::operator()(const int i,const int j)const
    {
        int tmp=0;
        return(i>=0 && i<n && j>=0 && j<m) ?data[i][j]:tmp;
    }
    template <typename T>
    void matran<T>::setM(int j)
    {
        this->m=j;
    }
    template <typename T>
    int matran<T>::getM()const
    {
        return m;
    }

```

- File **templateMatran_demo.cpp**

```

#include <iostream>
#include "matran.cpp"

```

```

using namespace std;
/* run this program using the console pauser or add your own getch, system("pause") or input loop */

main() {
    int n1,m1;
    cout<<"nhap vao so hang cua ma tran 1: ";
    cin>>n1;
    cout<<"\n nhap vao so cot cua ma tran 1: ";
    cin>>m1;
    matran<int> a(n1,m1);
    cin>>a;
    cout<<a;
    int n2,m2;
    cout<<"\n nhap vao hang cua ma tran 2: ";
    cin>>n2;
    cout<<"\n nhap vao so cot cua ma tran 2: ";
    cin>>m2;
    matran<int> b(n2,m2);
    cin>>b;
    cout<<b;
    if((n1!=n2)||m1!=m2)
    {
        cout<<"\n khong the cong hai ma tran";
    }
    else
    {
        cout<<"\n tong cua hai ma tran a va b "<<a+b;
    }
    if((n1!=n2)||m1!=m2)
    {
        cout<<"\n khong the cong hai ma tran";
    }
    else
    {
        cout<<"\n tong cua hai ma tran a va b "<<a-b;
    }
    matran<int> k;
    if(m1!=n2)
    {
        cout<<"\n khong the nhan hai vecto nay";
    }
    else
    {
        cout<<"\n tich cua hai ma tran a va b "<<a*b;
    }
    b=a;
    cout<<"\n ma tran b duoc gan bang ma tran a ";
    cout<<b;
    if(n1==m1){
        cout<<"\n dinh thuc cua ma tran a : "<<~a;}
}

```

jj. Demo

```
D:\Ky 6\Thuc Hanh Huong Doi Tuong\ThucHanh1\Te...
nhap vao so hang cua ma tran 1: 2
nhap vao so cot cua ma tran 1: 2
a[0][0]= 1
a[0][1]= 2
a[1][0]= 3
a[1][1]= 4

1 2
3 4
nhap vao hang cua ma tran 2: 3
nhap vao so cot cua ma tran 2: 3
a[0][0]= 1
a[0][1]= 2
a[0][2]= 3
a[1][0]= 4
a[1][1]= 5
a[1][2]= 6
a[2][0]= 7
a[2][1]= 8
a[2][2]= 9

1 2 3
4 5 6
7 8 9
khong the cong hai ma tran
khong the cong hai ma tran
khong the nhan hai vecto nay
ma tran b duoc gan bang ma tran a
1 2
3 4
dinh thuc cua ma tran a : -2
-----
Process exited after 9.824 seconds with return value 0
Press any key to continue . . .
```

10. Bài 3.3

kk. Đề bài

Cài đặt lại lớp đa thức Polynomial là lớp template.

II. Thuật toán

- Khai báo mảng một chiều data kiểu T cấp phát động để lưu hệ số của đa thức. Phương thức getN() dùng để lấy bậc của đa thức
- Toán tử + (Tính tổng 2 đa thức trả về một đa thức): dựng đa thức template k có bậc là n
So sánh nếu $n = x.n$ thì cho l chạy từ $n \rightarrow 0$ cộng từ phần tử data của this với x vào k
So sánh nếu $n < x.n$ thì cho l chạy từ $x.n \rightarrow 0$ cho data[i] của k nhận giá trị của data[i] của x. Sau đó cho l chạy từ $n \rightarrow 0$ cộng các phần tử của data[i] của this vào k
- Toán tử - (Tính hiệu 2 đa thức trả về một đa thức): thuật toán giống toán tử +, thay phép cộng bằng phép trừ.
- Toán tử * (Nhân đa thức đang tham chiếu với một số): Dựng đa thức k có bậc bằng tổng n với x.n, lặp từ $i = n \rightarrow 0$ và $j = x.n \rightarrow 0$, nhân hệ số thứ i của this với hệ số thứ j của x rồi cộng lại vào hệ số data[i+j] của k. Kết quả trả về đa thức t.

- Toán tử = (Gán đa thức x cho đa thức đang tham chiếu trả về một tham chiếu): Nếu đa thức cần gán khác đa thức x thì: thu hồi vùng nhớ cũ, gán lại bậc đa thức bằng bậc của x, cấp phát vùng nhớ mới, chép dữ liệu từ đa thức x sang đa thức đang tham chiếu.
- Toán tử << (Toán tử trích): Lặp $i = 0 \dots N$ của x, in ra các hệ số của đa thức trả về biến kiểu ostream.
- Toán tử >> (Toán tử chèn): Lặp $i = 0 \dots N$ của x, nhập các hệ số của đa thức của đa thức x, trả về biến kiểu istream.
- Toán tử [] (truy xuất hệ số của đa thức): cần viết 2 phiên bản trả về một giá trị khi tham chiếu đến dữ liệu hằng, trả về địa chỉ nếu tham chiếu đến dữ liệu cần sửa đổi.
- Toán tử () (Tính giá trị của đa thức):
Khai báo 1 biến float gtri ;
Cho i chạy từ $n \rightarrow 0$: $gtri += data[i] * pow(x, i)$
Kết quả trả về gtri

- Mã nguồn

- File **dathuc.h**

```
#ifndef DATHUC_H
#define DATHUC_H
#include<iostream>
#include<cmath>
#include<istream>
using namespace std;
template <typename T>
class dathuc
{
public:
    dathuc(int n=3);//ham khai dung
    ~dathuc();//ham huy
    dathuc operator + (const dathuc &x);// ham toan tu cong hai da thuc
    dathuc operator - (const dathuc &x);// toan tu tru hai da thuc
    dathuc operator * (const dathuc &x);// toan tu nhan hai ma thuc
    T& operator [] (int i);//ham lay ra he so cua da thuc
    T operator [] (int i)const;//ham set gia tri cho data
    void setN(int i);//set gia tri cho n
    int getN()const;//ham lay gia tri cua n
    float operator () (float x);// ham tinh gia tri cua da thuc voi gia tri x
    dathuc(const dathuc &x);// ham dung sao chep
    dathuc& operator = (const dathuc &x);//ham toan tu gan

private:
    int n;// bac cua da thuc
    T *data;// mang dong luu gia tri cua
};

#endif
```

- File **dathuc.cpp**

```
#include<iostream>
#include"dathuc.h"
using namespace std;
template <typename T>
dathuc<T>::dathuc(int n)//ham khai dung
{
    this->n=n;
    data=new T [n+1];
    for(int i=n;i>=0;i--)
    {
        data[i]=0;
    }
}
template <typename T>
dathuc<T>::~~dathuc();//ham huy
{
    delete [] data;
}
```

```

template <typename T>
dathuc<T> dathuc<T>::operator + (const dathuc<T> &x)// ham toan tu cong hai da thuc
{
    dathuc<T> k(n);
    if(n==x.n)
    {
        for(int i=n;i>=0;i--)
        {
            k.data[i]=data[i]+x.data[i];
        }
    }
    if(n<x.n)
    {
        k.n=x.n;
        for(int i=x.n;i>=0;i--)
        {
            k.data[i]=x.data[i];
        }
        for(int i=n;i>=0;i--)
        {
            k.data[i]+=data[i];
        }
    }
    if(n>x.n)
    {
        for(int i=n;i>=0;i--)
        {
            k.data[i]=data[i];
        }
        for(int i=x.n;i>=0;i--)
        {
            k.data[i]+=x.data[i];
        }
    }
    return k;
}

template <typename T>
dathuc<T> dathuc<T>::operator - (const dathuc<T> &x)// toan tu tru hai da thuc
{
    dathuc<T> k(x);
    for(int i=x.n;i>=0;i--)
    {
        k.data[i]=-k.data[i];
    }
    return k+(*this);
}

template <typename T>
dathuc<T> dathuc<T>::operator * (const dathuc<T> &x)// toan tu nhan hai ma thuc
{
    dathuc<T> k(n+x.n);
    for(int i=n;i>=0;i--)
        for(int j=x.n;j>=0;j--)
        {
            k.data[i+j]+=data[i]*x.data[j];
        }
    return k;
}

template <typename T>
ostream& operator <<(ostream &out, const dathuc<T> &x)
{
    cout<<"\n cac phan tu cua da thuc ( ";
    for(int i=x.getN();i>0;i--)
    {
        out<<x[i]<<" , ";
    }
    out<<x[0]<<" )";
    return out;
}

template <typename T>
istream &operator >> (istream &in,dathuc<T> &x)// ham xuat cac phan tu cua da thuc
{
    for(int i=x.getN();i>=0;i--)

```

```

{
    cout<<"\n he so tuong ung voi x*"<i<<" la : ";
    in>>x[i];
}
return in;
}
template <typename T>
T& dathuc<T>::operator [] (int i)//ham lay ra he so cua da thuc
{
    return data[i];
}
template <typename T>
float dathuc<T>::operator () (float x)// ham tinh gia tri cua da thuc voi gia tri x
{
    float gtri;
    for(int i=n;i>=0;i--)
    {
        gtri+=data[i]*pow(x,i);
    }
    return gtri;
}
template <typename T>
dathuc<T>::dathuc(const dathuc<T> &x)// ham dung sao chep
{
    n=x.n;
    data=new T[n+1];
    for(int i=0;i<=n;i++){
        data[i]=x.data[i];
    }
}
template <typename T>
dathuc<T>& dathuc<T>::operator = (const dathuc<T> &x)//ham toan tu gan
{
    delete data;
    n=x.n;
    data=new T[n+1];
    for(int i=0;i<=n;i++){
        data[i]=x.data[i];
    }
}
template <typename T>
T dathuc<T>::operator [] (int i)const//ham set gia tri cho data
{
    int tmp=0;
    return ((i>=0)&&(i<=n))?data[i]:tmp;
}
template <typename T>
void dathuc<T>::setN(int i)//set gia tri cho n
{
    this->n=i;
}
template <typename T>
int dathuc<T>::getN()const//ham lay gia tri cua n
{
    return n;
}
}

```

- **File `templatedathuc_demo.cpp`**

```

#include <iostream>
#include "dathuc.cpp"
using namespace std;
/* run this program using the console pauser or add your own getch, system("pause") or input loop */

main() {
    int n;

    cout<<"nhap vao bac cua da thuc 1 : ";
    cin>>n;
    dathuc<int> a(n);
    cout<<"\n nhap vao cac phan tu cua da thuc 1 ";
    cin>>a;
    cout<<a;
}

```

```

int m;
cout<<"\n nhap vao bac cua da thuc 2 : ";
cin>>m;
dathuc<int> b(m);
cout<<"\n nhap vao cac phan tu cua da thuc 2 ";
cin>>b;
cout<<b;
cout<<"\n cong hai da thuc \n";
cout<<a+b;
cout<<"\n tru hai da thuc \n";
cout<<a-b;
cout<<"\n nhan hai da thuc \n";
cout<<a*b;
cout<<"\n gan da thuc thu 2 bang da thuc thu 1 \n";
b=a;
cout<<b;
cout<<"\n he so thu 2 cua da thuc 1 : "<<a[2];
cout<<"\n gia tri cua da thu 1 tai gia tri x ";
int x;
cin>>x;
cout<<" la : "<<a(x);

```

DEMO

```

D:\Ky 6\Thuc Hanh Huong Doi Tuong\ThucHanh1\Te...
nhap vao bac cua da thuc 1 : 3

nhap vao cac phan tu cua da thuc 1
he so tuong ung voi x*3 la : 2
he so tuong ung voi x*2 la : 3
he so tuong ung voi x*1 la : 1
he so tuong ung voi x*0 la : 2
cac phan tu cua da thuc < 2 , 3 , 1 , 2 >
nhap vao bac cua da thuc 2 : 2

nhap vao cac phan tu cua da thuc 2
he so tuong ung voi x*2 la : 1
he so tuong ung voi x*1 la : 3
he so tuong ung voi x*0 la : 2
cac phan tu cua da thuc < 1 , 3 , 2 >
cong hai da thuc

cac phan tu cua da thuc < 2 , 4 , 4 , 4 >
tru hai da thuc

cac phan tu cua da thuc < 2 , 2 , -2 , 0 >
nhan hai da thuc

cac phan tu cua da thuc < 2 , 9 , 14 , 11 , 8 , 4 >
gan da thuc thu 2 bang da thuc thu 1

cac phan tu cua da thuc < 2 , 3 , 1 , 2 >
he so thu 2 cua da thuc 1 : 3
gia tri cua da thu 1 tai gia tri x 2
la : 32

-----
Process exited after 23.76 seconds with return value 0
Press any key to continue . . .

```