

Bài 1: TRINOMIAL

Xây dựng lớp Tam thức bậc hai với các thuộc tính là các hệ số a , b , c kiểu thực và các phương thức:

Phương thức khởi tạo: khởi gán các giá trị của các hệ số a , b , c .

Hàm toán tử xuất: in tam thức lên màn hình (có dạng $ax^2 + bx + c$).

Phương thức toán tử: “Đổi dấu tam thức” (-): đổi dấu các hệ số a , b , c .

Phương thức toán tử: cộng, trừ hai tam thức (cộng và trừ các hệ số tương ứng).

Xây dựng chương trình chính khai báo hai tam thức. Khởi gán giá trị cho các hệ số và đảo dấu của hai tam thức. In các tam thức đã đảo dấu ra màn hình. Tính và in ra màn hình các tam thức là tổng và hiệu của hai tam thức đã đảo dấu ở trên.

Bài 2: MATRIX

Xây dựng lớp ma trận gồm các thuộc tính: `double a[][]` là một mảng hai chiều chứa các phần tử của ma trận; m , n là các thuộc tính kích thước thực tế của ma trận và các phương thức:

Hàm toán tử nhập: nhập các giá trị m , n ; cấp phát bộ nhớ và nhập ma trận a .

Hàm toán tử xuất: xuất các giá trị của ma trận a lên màn hình.

Phương thức toán tử “Đổi dấu ma trận” (-): đổi dấu tất cả các phần tử của ma trận; **cộng, trừ** hai ma trận (cộng trừ các phần tử tương ứng của ma trận).

Xây dựng chương trình chính trong đó khai báo và nhập các giá trị cho hai ma trận P và Q . Đổi dấu các ma trận và in các ma trận đã đổi dấu ra màn hình. Tính và in ra màn hình các ma trận là tổng, hiệu của các ma trận đã đổi dấu ở trên.

Gợi ý cấp phát động mảng 2 chiều: sử dụng con trỏ cấp 2

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main(){
6      int dong = 3; //số dòng ma trận
7      int cot = 4;  //số cột ma trận
8      int **p;      //khai báo con trỏ cấp 2
9      // cấp phát
10     p = new int*[dong]; //cấp phát cho mảng <dong> dòng
11     //cấp phát cho mỗi dòng <cot> cột
12     for(int i = 0; i < dong; i++){
13         p[i] = new int[cot]; //p[i] chính là con trỏ cấp 1
14     }
```