

**6.1.** Xây dựng lớp VECTO có các thông tin về hai tọa độ trong mặt phẳng hai chiều: x, y và các phương thức:

- **Hàm toán tử nhập** để nhập tọa độ x, y.
- **Hàm toán tử xuất** để hiển thị tọa độ của véc tơ ra màn hình.
- Các phép toán cộng, trừ hai véc tơ (Cho hai véc tơ A(x1, y1) và B(x2, y2) thì A + B là véc tơ có tọa độ (x1+x2, y1+y2), A – B là véc tơ có tọa độ (x1-x2, y1-y2).

Viết chương trình chính thực hiện nhập vào hai véc tơ A và B. Tính tổng, hiệu của chúng và in kết quả ra màn hình.

## **6.2. FRACTION**

Phép nhân, chia, cộng, trừ hai phân thức được định nghĩa như sau:

$$\frac{a}{b} \times \frac{c}{d} = \frac{ac}{bd} \qquad \frac{a}{b} : \frac{c}{d} = \frac{ad}{bc} \qquad \frac{a}{b} + \frac{c}{d} = \frac{ad+bc}{bd} \qquad \frac{a}{b} - \frac{c}{d} = \frac{ad-bc}{bd}$$

Hãy xây dựng một lớp Phân số với các thuộc tính Tử số, Mẫu số và các phương thức:

- **Hàm toán tử nhập:** nhập các giá trị của tử số và mẫu số.
- **Hàm toán tử xuất:** đưa phân số ra màn hình (dưới dạng Tử\_Số/ Mẫu\_số).
- **Các phương thức toán tử:** nhân, chia, cộng, trừ hai phân số.
- **Tính giá trị:** trả về giá trị kiểu thực là Tử\_Số/ Mẫu\_Số.

Viết chương trình chính nhập hai phân số, đưa ra màn hình các phân số là tích, thương, tổng, hiệu của hai phân số vừa nhập kèm theo giá trị của phân số kết quả.

## **6.3. COMPLEX NUMBER**

Cho hai số phức dạng: SP1 = a1+ i\*b1; SP2 = a2+ i\*b2 với a1, a2 là các phần thực và b1, b2 là các phần ảo; Phép cộng, trừ hai số phức được định nghĩa như sau:

$$SP3 = SP1 + SP2 = (a1+a2) + i*(b1+b2);$$

$$SP4 = SP1 - SP2 = (a1-a2) + i*(b1-b2);$$

Hãy xây dựng lớp số phức với các thuộc tính Thực, Ảo và các phương thức:

- **Phương thức khởi tạo:** khởi gán phần thực và phần ảo của số phức.
- **Hàm toán tử xuất:** in số phức lên màn hình theo định dạng <Thực> + i\* <Ảo>.
- **Phương thức toán tử:** + và - hai số phức.

Viết hàm **main** nhập vào hai số phức SP1 và SP2. Tính và in ra số phức SP3 và SP4 là tổng và hiệu của hai số phức SP1, SP2.

#### 6.4. 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.

- 6.5.** Xây dựng lớp ma trận gồm các thuộc tính: `double a[][]` là mảng 2 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 của 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.

**6.6.** Ta định nghĩa phương thức toán tử sắp xếp mảng 1 chiều như sau:

**Phương thức ++** sắp xếp mảng theo chiều tăng dần.

**Phương thức --** sắp xếp mảng theo chiều giảm dần.

Hãy định nghĩa một lớp Mảng gồm: thuộc tính a[] kiểu float, biến kích thước mảng n kiểu nguyên và các phương thức:

**Hàm toán tử nhập:** nhập kích thước mảng n, cấp phát bộ nhớ và nhập các giá trị cho mảng.

**Hàm toán tử xuất:** xuất các giá trị của mảng ra màn hình.

**Phương thức toán tử ++ và --** như trên để sắp xếp mảng tăng dần và giảm dần.

Viết chương trình chính sử dụng lớp trên để nhập vào một mảng n phần tử thực, sau đó sắp xếp mảng theo chiều tăng dần (giảm dần) và in mảng đã sắp xếp lên màn hình.