

## Bài thực hành 8

### MẢNG MỘT CHIỀU

Sinh viên tạo project mang tên **BTH8** tại ổ đĩa D. Sau đó thực hiện các bài tập dưới đây, mỗi bài tập lưu lại với định dạng tên *BTH8\_BT01.cpp*, *BTH8\_BT02.cpp*, ... *BTH8\_BT13.cpp*.

#### A. Bài tập cơ bản

1. Viết chương trình thực hiện các công việc sau đây:
  - a) Khai báo một mảng alpha lưu 50 phần tử kiểu **double**.
  - b) Gán giá trị của 25 phần tử đầu tiên là bình phương của chỉ số phần tử, 25 phần tử còn lại có giá trị là lập phương của chỉ số.
  - c) In các giá trị trong mảng alpha ra màn hình, mỗi dòng gồm 10 phần tử.
2. Viết chương trình thực hiện các công việc sau đây:
  - a) Định nghĩa hằng MAXSIZE là 100.
  - b) Khai báo mảng values có thể lưu số phần tử tối đa là MAXSIZE, kiểu của mỗi phần tử là **double**.
  - c) Khai báo currentSize lưu số phần tử hiện có trong mảng.
  - d) Nhập vào giá trị của currentSize.
  - e) Nhập vào các giá trị trong mảng values, số phần tử đọc vào là currentSize.
  - f) In các phần tử trong mảng values trên một dòng, cách nhau bởi dấu phẩy.
3. Viết chương trình nhập vào một mảng số nguyên (tối đa 100 phần tử), tìm và in ra phần tử lớn nhất của mảng.
4. Viết chương trình nhập vào một mảng số nguyên (tối đa 100 phần tử), tính và in ra tổng và giá trị trung bình của các phần tử trong mảng.
5. Viết chương trình nhập vào một mảng số nguyên (tối đa 100 phần tử) và vị trí của phần tử cần xóa, xóa phần tử tại vị trí đã nhập và in ra mảng sau khi xóa.
6. Viết chương trình gồm các hàm sau đây:
  - a) Nhập một dãy số nguyên vào mảng arr, số phần tử nhập vào là arrSize:  
**void nhap(int arr[], int arrSize);**

- b) In dãy số nguyên trong arr, cách nhau bởi dấu phẩy, arrSize là số phần tử của dãy:

```
void xuat(const int arr[], int arrSize);
```

- c) Tìm một giá trị x trong mảng arr, trả về true tìm thấy, ngược lại trả về false, arrSize là số phần tử của mảng:

```
bool tim(const int arr[], int arrSize, int x);
```

- d) Đảo ngược các phần tử trong mảng arr, arrSize là số phần tử của mảng:

```
void daonguoc(int arr[], int arrSize)
```

- e) Xóa phần tử tại một vị trí p trong mảng arr, arrSize là số phần tử của mảng:

```
void xoa(int arr[], int& arrSize, int p)
```

7. Viết hàm trả về vị trí xuất hiện đầu tiên của phần tử nhỏ nhất trong mảng. Hàm nhận vào mảng số nguyên và số phần tử của mảng. Viết chương trình kiểm tra hàm.
8. Viết hàm đếm số lần xuất hiện của một phần tử trong một mảng số nguyên. Hàm nhận vào ba tham số: mảng a, số phần tử n và phần tử x. Hàm trả về số lần xuất hiện của x trong mảng a. Viết chương trình kiểm tra hàm.

## B. Bài làm thêm

9. Các hệ số của đa thức  $a_n x^n + a_{n-1} x^{n-1} + \dots + a_0$  được lưu trong mảng a. Viết hàm nhận vào mảng a, bậc của đa thức n, giá trị x và trả về giá trị của đa thức.
10. Viết chương trình in các số nguyên tố từ 2 đến n bằng cách dùng sàng Eratosthenes (*sieve of Eratosthenes*). Thuật giải như sau:

Bước 1. Tạo mảng chứa các số nguyên từ 2..n.

Bước 2. Bắt đầu từ số nguyên tố đầu tiên  $p = 2$ .

Bước 3. Đánh dấu tất cả bội số của p không phải là số nguyên tố ( $2p, 3p, 4p, \dots$ ).

Bước 4. Tìm số kế tiếp lớn hơn p chưa bị đánh dấu. Nếu không có thì thuật giải kết thúc. Ngược lại đặt p là số này và quay lại bước 3.

Ví dụ dùng thuật giải để tìm các số nguyên tố đến  $n = 25$ :

2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
2	3		5		7		9		11		13		15		17		19		21		23		25
2	3		5		7				11		13				17		19				23		25
2	3		5		7				11		13				17		19				23		

11. Viết hàm sắp xếp các phần tử trong mảng tăng dần.
12. Viết hàm xóa các phần tử trùng nhau trong mảng.
13. Cho mảng gồm các số nguyên có giá trị trong khoảng từ 1 đến  $k$  ( $k < 100$ ). Viết hàm in ra số lần xuất hiện của mỗi phần tử trong mảng.