

# ARRAY

## Code sample

Một số thao tác cơ bản trên mảng 1 chiều

```
void main()
{
    int n, a[100];
    NhapMang(a,n);
    XuatMang(a,n);
    cout<<"Tong cac phan tu trong mang:" <<SumArray(a,n);
}

void NhapMang(int a[], int &n)
{
    cout<<"Nhap so phan tu cua mang: ";
    cin>>n;

    for (int i = 0; i < n; i++)
    {
        cout<<"a[" <<i <<"]": ";
        cin>>a[i];
    }
}

void XuatMang(int a[], int n)
{
    cout<<"Mang: ";
    for (int i = 0; i < n; i++)
        cout<<a[i] <<" ";
}
```

// tong cac phan tu trong mang

```
long SumArray(int a[], int n)
{
    long s = 0;
    for (int i = 0; i < n; i++)
        s = s + a[i];
    return s;
}
```

// dem so lan xuat hien cua x trong mang

```
int CountX(int a[], int n, int x)
{
    int count = 0;
    for (int i = 0; i < n; i++)
        if (a[i] == x)
            count++;
    return count;
}
```

### Sắp xếp mảng 1 chiều

```
void Sort(int a[], int n, bool bSapTang)
{
    for (int i = 0; i < n - 1; i++)
        for (int j = i + 1; j < n; j++)
        {
            // bSapTang == true -> sắp tăng
            // bSapTang == false -> sắp giảm
            if (bSapTang == true)
            {
                if (a[j] < a[i])
                    Swap(a[i], a[j]);
            }
        }
}
```

```
        else
        {
            if (a[j] > a[i])
                Swap(a[i], a[j]);
        }
    }
}
```

**Trộn 2 mảng một chiều a, b thành một mảng một chiều c, sao cho các phần tử xen kẽ nhau:**

```
void Tron2Mang(int a[], int n, int b[], int m, int c[])
{
    int min = (n > m ? m : n);
    int i = 0, j = 0;
    for (i = 0; i < min; i++, j += 2)
    {
        c[j] = a[i];
        c[j+1] = b[i];
    }
    while (i < n)
        c[j++] = a[i++];
    while (i < m)
        c[j++] = b[i++];
}
```

**Xóa một phần tử mảng:**

```
void Xoa1PhanTu(int a[], int &n, int x)
{
    int b[100]; //mảng trung gian
    for (int i = 0; i < n; i++)
        b[i] = a[i];
    int m = 0;
    for (i = 0; i < n; i++)
        if (b[i] != x)
            a[m++] = b[i];
}
```

## Exercises

Bài 1. VCT nhập một mảng số nguyên có  $n$  phần tử, xây dựng các hàm thực hiện:

Tính tổng tất cả các phần tử không âm.

Tính tổng các phần tử chia hết cho 3.

Tính tổng các phần tử có chữ số đầu tiên là số lẻ.

Bài 2. VCT nhập một mảng số nguyên có  $n$  phần tử, xây dựng các hàm tính tổng các phần tử là:

a) số chính phương

b) số hoàn chỉnh

c) số nguyên tố

d) số dương đối xứng

Bài 3. VCT nhập một mảng có  $n$  phần tử là các số, xây dựng các hàm thực hiện:

a) Kiểm tra mảng có chứa số dương không.

b) Kiểm tra mảng có gồm toàn số dương không.

c) Nếu mảng không toàn dương thì đổi các số dương thành số âm.

Bài 4. VCT nhập một mảng số nguyên có  $n$  phần tử, xây dựng các hàm thực hiện:

a) Kiểm tra mảng có đối xứng không.

b) Kiểm tra mảng có đan xen âm dương không.

c) Kiểm tra mảng có tất cả các cặp phần tử đứng cạnh nhau đều có giá trị khác nhau.

d) Kiểm tra mảng có tất cả bộ 3 phần tử đứng cạnh nhau lập thành cấp số cộng.

Bài 5. VCT nhập một mảng có  $n$  phần tử là các số, xây dựng các hàm thực hiện:

a) Kiểm tra mảng có thứ tự tăng/giảm không.

b) Nếu mảng không có thứ tự thì sắp xếp các phần tử theo thứ tự tăng dần.

Bài 6. VCT nhập một mảng số có  $n$  phần tử và một giá trị  $X$ , xây dựng các hàm thực hiện:

a) Kiểm tra  $X$  có trong mảng không.

b) Nếu  $X$  không có trong mảng thì chèn  $X$  vào mảng tại vị trí sau phần tử đầu tiên.

Bài 7. VCT nhập 2 mảng số  $a$  và  $b$  có  $n$  phần tử, xây dựng các hàm thực hiện:

a) Kiểm tra 2 mảng có thứ tự tăng không, nếu không thì sắp tăng cho 2 mảng.

b) Trộn hai mảng a và b thành một mảng c có thứ tự tăng dần.

Bài 8. VCT nhập vào N số thực từ bàn phím và lưu vào mảng các số thực, cho biết trong mảng có bao nhiêu số âm.

Bài 9. VCT nhập vào 1 số nguyên n. Chuyển đổi n sang hệ nhị phân và xuất ra màn hình. (với giá trị n:  $-255 \leq n \leq 255$ ), sử dụng mảng một chiều.

Bài 10. VCT nhập vào 1 dãy 16 bit nhị phân (có dấu), chuyển đổi dãy bit nhị phân sang số thập phân tương ứng.

Bài 11. VCT nhập vào một mảng gồm n phần tử nguyên ( $1 \leq n \leq 10,000$ ) và một số nguyên X:

- a) Những phần tử nào (in ra cả giá trị và vị trí của phần tử đó) là ước số của X?
- b) Đồn những phần tử là ước số của X về cuối mảng. Sắp xếp các phần tử không phải là ước số của X theo thứ tự tăng dần (giữ nguyên vị trí các phần tử là ước số của X).

Bài 12. VCT nhập vào một mảng các số nguyên dương gồm n phần tử ( $1 \leq n \leq 15$ ):

- a) Đếm số phần tử tận cùng là 6 và chia hết cho 6 trong mảng
- b) Tính trung bình cộng các số nguyên tố hiện có trong mảng
- c) Cho biết trong mảng có bao nhiêu số nguyên tố phân biệt.

Bài 13. Cho mảng A gồm  $n < 1000$  phần tử nguyên  $|A[i]| \leq 10,000$ . Viết hàm thực hiện các công việc sau:

- a) Trích những phần tử trong A không phải số nguyên tố ra mảng B
- b) Sắp giảm các số nguyên trong mảng B
- c) Xóa những số nguyên tố trong mảng A.

Bài 14. VCT nhập 2 dãy số nguyên A, B gồm m, n phần tử ( $1 \leq n, m \leq 25$ ):

- a) Xuất ra những phần tử có trong A mà không có trong B.
- b) Ghép A, B thành C sao cho C không có phần tử trùng nhau.