# MÃ NGUỒN MẪU – MẢNG MỘT CHIỀU

## 1. File header

```
// array.h
#define MAX_SIZE 100

// Nhập xuất mảng một chiều ---------------------------------
// Nhập mảng
void inputArray(int a[], int &n);
// Xuất mảng
void outputArray(int a[], int n);

// Một số thao tác cơ bản -----------------------------------
// Tính tổng các phần tử trong mảng
int calcSum(int a[], int n);
// Tính tổng các phần tử là số chẵn trong mảng
int calcSumOfEven(int a[], int n);
// Tìm vị trí của phần tử có giá trị val, return -1 nếu không tìm thấy
int findIndexOfVal(int a[], int n, int val, int start=0);

// Chèn, xóa phần tử trong mảng ---------------------------
// Chèn giá trị val vào mảng tại vị trí pos
bool insertElement(int a[], int &n, int val, int pos);
// Xóa một phần tử tại ví trí pos ra khỏi mảng
bool deleteElement(int a[], int &n, int pos);

// Làm việc trên 2 mảng (ứng dụng cho phân số) có cùng số phần tử
// Nhập mảng phân số (num: numerator(tử số), den: denominator (mẫu số), n: số lượng phân số)
void inputArrayOfFractions(int num[], int den[], int &n);
// Xuất mảng phân số (num: numerator, den: denominator)
void outputArrayOfFractions(int num[], int den[], int &n);
// Xuất một phân số
void outputFraction(int num, int den);
// Tìm vị trí và giá trị phân số nhỏ nhất mảng (num: numerator, den: denominator)
int findIndexMinFraction(int num[], int den[], int n);

// Làm việc trên 2 mảng khác số lượng phần tử
// Nối 2 mảng thành 1 mảng
bool concatTwoArrays(int a[], int &na, int b[], int nb);
bool concatTwoArrays(int a[], int na, int b[], int nb, int res[], int &nres);

// Nâng cao
// Thêm một phần tử vào mảng đang tăng
void insertIntoAscendingArray(int a[], int &n, int val);
void inputAndCreateAscendingArray(int a[], int &n);
```

## 2. File source

```
// array.cpp

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

// Nhập xuất mảng một chiều ---------------------------------
// Nhập mảng
void inputArray(int a[], int &n) {
    cout << "Nhap vao so luong phan tu: ";
    cin >> n;

    // Chỉ số mảng bắt đầu từ 0 và kết thúc tại n - 1
    for (int i = 0; i < n; i++) {
```

```
        cout << "a[" << i << "] = ";
        cin >> a[i];
    }
}

// Xuất mảng
void outputArray(int a[], int n) {
    for (int i = 0; i < n; i++)
        cout << a[i] << " ";

    cout << endl;
}

// Một số thao tác cơ bản ---------------------------------
// Tính tổng các phần tử trong mảng
int calcSum(int a[], int n) {
    int sum = 0;
    for (int i = 0; i < n; i++)
        sum += a[i];

    return sum;
}

// Tính tổng các phần tử là số chẵn trong mảng
int calcSumOfEven(int a[], int n) {
    int sumEven = 0;
    for (int i = 0; i < n; i++)
        if (a[i] % 2 == 0)
            sumEven += a[i];

    return sumEven;
}

// Tìm vị trí của phần tử có giá trị val
int findIndexOfVal(int a[], int n, int val, int start) {
    for (int i = start; i < n; i++)
        if (a[i] == val)
            return i;

    return -1; // Không tìm thấy val trong mảng
}

// Chèn, xóa phần tử trong mảng ---------------------------
// Chèn giá trị val vào mảng tại vị trí pos
bool insertElement(int a[], int &n, int val, int pos) {
    // Kiểm tra hợp lệ
    if (pos < 0 || pos >= MAX_SIZE)
        return false;

    // Tăng số lượng phần tử
    n = n + 1;

    // Dời các phần tử qua phải
    for (int i = n - 1; i > pos; i--)
        a[i] = a[i - 1];

    // Chèn val vào vị trí pos
    a[pos] = val;

    return true;
}

// Xóa một phần tử tại ví trí pos ra khỏi mảng
bool deleteElement(int a[], int &n, int pos) {
    // Kiểm tra hợp lệ
    if (pos < 0 || pos >= n)
        return false;


    for (int i = pos; i < n - 1; i++)
        a[i] = a[i + 1];
```

```
    // Giảm số lượng phần tử
    n = n - 1;

    return true;
}

// Làm việc trên 2 mảng (ứng dụng cho phân số) ---------------
// Nhập mảng phân số (num: numerator(tử số), den: denominator (mẫu số), n: số lượng phân số)
void inputArrayOfFractions(int num[], int den[], int &n) {
    cout << "Nhap so luong phan so: ";
    cin >> n;

    for (int i = 0; i < n; i++) {
        cout << "Nhap tu so thu " << i << ": ";
        cin >> num[i];
        cout << "Nhap mau so thu " << i << ": ";
        cin >> den[i];
        cout << "------------------------------" << endl;
    }
}


void outputFraction(int num, int den) {
    cout << num << "/" << den << "  ";
}

// Xuất mảng phân số (num: numerator, den: denominator)
void outputArrayOfFractions(int num[], int den[], int &n) {
    cout << "Noi dung mang phan so la: ";
    for (int i = 0; i < n; i++)
    {
        outputFraction(num[i], den[i]);
    }

    cout << endl;
}

// Tìm vị trí và giá trị phân số nhỏ nhất mảng (num: numerator, den: denominator)
int findIndexMinFraction(int num[], int den[], int n) {
    // Giả sử phần tử đầu tiên là min
    float valTemp, valMin = float(num[0])/den[0];
    int idxMin = 0;
    int numMin = num[0];
    int denMin = den[0];

    for (int i = 1; i < n; i++) {
        valTemp = float(num[i])/den[i];

        if (valTemp < valMin) {
            valMin = valTemp;
            idxMin = i;
            numMin = num[i];
            denMin = den[i];
        }
    }

    return idxMin;
}



// Làm việc trên 2 mảng khác số lượng phần tử
// Nối 2 mảng. Mảng đầu tiên sẽ nhận kết quả trả về
    // Cách này không cần tạo thêm mảng mới
    // Phù hợp cho trường hợp số lượng phần tử lớn
bool concatTwoArrays(int a[], int &na, int b[], int nb) {
    if (na + nb > MAX_SIZE)
        return false;
```

```
    // Gộp mảng b vào cuối mảng a
    for (int i = 0; i < nb; i++)
        a[na + i] = b[i];

    na += nb; // Tăng số lượng phần tử
    return true;
}

// Nối 2 mảng. Kết quả trả về qua mảng res
bool concatTwoArrays(int a[], int na, int b[], int nb, int res[], int &nres) {
    if (na + nb > MAX_SIZE)
        return false;

    for (int i = 0; i < na; i++)
        res[i] = a[i];
    for (int i = 0; i < nb; i++)
        res[i + na] = b[i];

    nres = na + nb;
    return true;
}


// Nâng cao
// Thêm một phần tử vào mảng đang tăng
void insertIntoAscendingArray(int a[], int &n, int val) {
    int i = 0; // Nếu mảng chưa có phần tử nào, mặc định sẽ thêm vào vị trí 0

    // Tìm vị trí thích hợp
    if (n > 0) { // Mảng đã có phần tử
        while (i < n) {
            if (val < a[i])
                break;

            i++;
        }
    }

    insertElement(a, n, val, i);
}

void inputAndCreateAscendingArray(int a[], int &n) {
    int temp;
    n = 0;
    do {
        cout << "Nhap vao mot phan tu (nhap -1 de dung): ";
        cin >> temp;

        if (temp == -1)
            break;

        insertIntoAscendingArray(a, n, temp);
    } while (temp != -1);
}
```

# 3. File main.cpp

```
// main.cpp
#include <iostream>
#include "array.h"
using namespace std;

int main() {
    int a[MAX_SIZE];
    int n;
```

```
    // Gọi hàm nhập mảng
    inputArray(a, n);

    // Gọi hàm xuất nội dung mảng
    cout << "Noi dung mang la: ";
    outputArray(a, n);

    // Tính tổng các phần tử trong mảng
    cout << "Tong phan tu trong mang: " << calcSum(a, n) << endl;

    // Tìm vị trí của phần tử có giá trị 5 xuất hiện đầu tiên và tiếp theo trong mảng
    int firstIdx = findIndexOfVal(a, n, 5);
    int secondIdx = findIndexOfVal(a, n, 5, firstIdx + 1);
    cout << "Vi tri xuat hien dau tien cua 5 la: " << firstIdx << endl;
    cout << "Vi tri xuat hien tiep theo cua 5 la: " << secondIdx << endl;

    // Xuất mảng từ vị trí xuất hiện đầu tiên của 5 đến vị trí xuất hiện tiếp theo
    cout << "Mang con giua 2 vi tri xuat hien cua 5: ";
    outputArray(a + (firstIdx + 1), secondIdx - (firstIdx + 1));

    // Chèn giá trị 10 vào vị trị thứ 1 trong mảng
            // giả sử giá trị là hợp lệ
    insertElement(a, n, 10, 1);
    cout << "Sau khi chen 10 vao vi tri 1: ";
    outputArray(a, n);

    // Xóa phần tử thứ 0 khỏi mảng
    deleteElement(a, n, 0);
    cout << "Sau khi xoa phan tu tai vi tri 0: ";
    outputArray(a, n);

    return 0;
}
```

Nội dung khi chạy chương trình main với các nội dung nhập vào tương ứng là:

```
Nhap vao so luong phan tu: 5
a[0] = 5
a[1] = 1
a[2] = 2
a[3] = 5
a[4] = 3
Noi dung mang la: 5 1 2 5 3
Tong phan tu trong mang: 16
Vi tri xuat hien dau tien cua 5 la: 0
Vi tri xuat hien tiep theo cua 5 la: 3
Mang con giua 2 vi tri xuat hien cua 5: 1 2
Sau khi chen 10 vao vi tri 1: 5 10 1 2 5 3
Sau khi xoa phan tai vi tri 0: 10 1 2 5 3
```

HẾT