

MẢNG MỘT CHIỀU – MẢNG 2 CHIỀU

CÁC BÀI TẬP CƠ BẢN

Bài 1: Tính tổng tất cả các phần tử trong mảng một chiều

```
#include <iostream>
#include <ctime>
using namespace std;

void NhapMang(int[], int&);
void XuatMang(int[], int);
void TinhTongCacPhanTu(int a[], int n);

int main(int argc, char* argv[])
{
    int n;
    int a[100];

    NhapMang(a, n);

    XuatMang(a, n);

    TinhTongCacPhanTu(a, n);

    system("pause");
    return 0;
}

void NhapMang(int a[], int &n){
    do{
        cout << "Nhap so phan tu cua mang: ";
        cin >> n;
    } while (n > 100 || n <= 0);

    srand((unsigned int)time(NULL));
    cout << "\n... Phát sinh tu dong cac phan tu trong mang...\n";
    for (int i = 0; i<n; i++)
    {
        a[i] = rand() % 90 + 10;
    }
}

void XuatMang(int a[], int n){
    cout << "\nCac phan tu hien co trong mang: ";
    for (int i = 0; i<n; i++)
```

```

    {
        cout << a[i] << " ";
    }
    cout << "\n";
}

void TinhTongCacPhanTu (int a[], int n){
    int s = 0;
    for (int i = 0; i<n; i++)
        s = s + a[i];

    cout << "\nTong cac phan tu trong mang = " << s << "\n";
}

```

Bài 2: Đếm số lần xuất hiện một phần tử x bất kỳ.

```

#include <iostream>
#include <ctime>
using namespace std;

void NhapMang(int[], int&);
void XuatMang(int[], int);
void DemSoLanXuatHienMotPhanTu(int[], int, int&);

int main(int argc, char* argv[])
{
    int n, x;
    int a[100];

    NhapMang(a, n);

    XuatMang(a, n);

    DemSoLanXuatHienMotPhanTu(a, n, x);

    system("pause");
    return 0;
}

void NhapMang(int a[], int &n){
    do{
        cout << "Nhap so phan tu cua mang: ";
        cin >> n;
    } while (n > 100 || n <= 0);

    srand((unsigned int)time(NULL));
}

```

```

        cout << "\n... Phát sinh tự động các phần tử trong
mang...\n";
        for (int i = 0; i < n; i++)
        {
            //a[i] = rand() % 90 + 10;
            cin >> a[i];
        }
    }

void XuatMang(int a[], int n){
    cout << "\nCác phần tử hiện có trong mảng: ";
    for (int i = 0; i < n; i++)
    {
        cout << a[i] << " ";
    }
    cout << "\n";
}

void DemSoLanXuatHienMotPhanTu(int a[], int n, int &x){
    int so_lan_xuat_hien = 0;

    cout << "\Nhập phần tử x muốn tìm: ";
    cin >> x;
    for (int i = 0; i < n; i++)
    {
        if (a[i] == x)
            so_lan_xuat_hien++;
    }
    cout << "\nSố lần xuất hiện phần tử " << x << " là " <<
so_lan_xuat_hien << " lần\n";
}

```

Bài 3: Tìm vị trí xuất hiện đầu tiên của một phần tử bất kỳ trong mảng

```
#include <iostream>
```

```
#include <ctime>
```

```
using namespace std;
```

```
void NhapMang(int[], int&);
```

```
void XuatMang(int[], int);
```

```
int TimViTriX(int[], int, int);
```

```

int main(int argc, char* argv[])
{
    int n, x;
    int a[100];

    NhapMang(a, n);

    XuatMang(a, n);

    cout << "Nhap phan tu can tim: ";
    cin >> x;
    int vitri = TimViTriX(a, n, x);
    cout << "Vi tri xuat hien dau tien cua " << x << " trong mang la:" << vitri << endl;

    system("pause");
    return 0;
}

void NhapMang(int a[], int &n){
    do{
        cout << "Nhap so phan tu cua mang: ";
        cin >> n;
    } while (n > 100 || n <= 0);

    srand((unsigned int)time(NULL));
    cout << "\n... Phat sinh tu dong cac phan tu trong mang...\n";
    for (int i = 0; i < n; i++)
    {
        a[i] = rand() % 90 + 10;
    }
}

```

```

    }
}

void XuatMang(int a[], int n){
    cout << "\nCac phan tu hien co trong mang: ";
    for (int i = 0; i<n; i++)
    {
        cout << a[i] << " ";
    }
    cout << "\n";
}

```

```

int TimViTriX(int a[], int n, int x){
    for (int i = 0; i<n; i++)
    {
        if (a[i] == x)
            return i;
    }
    return -1;
}

```

Bài 4: Xóa một phần tử có chỉ số k trong mảng.

```
#include <iostream>
```

```
#include <ctime>
```

```
using namespace std;
```

```
voidNhapMang(int[], int&);
```

```
void XuatMang(int[], int);
```

```
void XoaViTri(int[], int&, int);
```

```
int main(int argc, char* argv[])
```

```

{
    int n, k;

```

```
int a[100];
```

```
NhapMang(a, n);
```

```
XuatMang(a, n);
```

```
cout << "Nhap chi so k:";
```

```
cin >> k;
```

```
XoaViTri(a, n, k);
```

```
XuatMang(a, n);
```

```
system("pause");
```

```
return 0;
```

```
}
```

```
void NhapMang(int a[], int &n){
```

```
do{
```

```
    cout << "Nhap so phan tu cua mang: ";
```

```
    cin >> n;
```

```
} while (n > 100 || n <= 0);
```

```
srand((unsigned int)time(NULL));
```

```
cout << "\n... Phat sinh tu dong cac phan tu trong mang...\n";
```

```
for (int i = 0; i<n; i++)
```

```
{
```

```
    a[i] = rand() % 90 + 10;
```

```
}
```

```
}
```

```

void XuatMang(int a[], int n){
    cout << "\nCac phan tu hien co trong mang: ";
    for (int i = 0; i<n; i++)
    {
        cout << a[i] << " ";
    }
    cout << "\n";
}

```

```

void XoaViTri(int a[], int& n, int k)
{
    for (int i = k; i < n-1; i++)
    {
        a[i] = a[i + 1];
    }
    a[n-1] = NULL;
    n--;
}

```

Bài 5: Viết chương trình nhập, xuất một mảng số nguyên hai chiều có m dòng và n cột. Xác định phần tử lớn nhất và nhỏ nhất trong mảng

```

#include <iostream>

#define max_dong 100
#define max_cot 100
using namespace std;

void NhapMang(int[][max_cot], int&, int&);
void XuatMang(int[][max_cot], int, int);
void TimMinMax(int[][max_cot], int, int);

int main(int argc, int argv[])
{
    int m, n;
    int a[max_dong][max_cot];
    //Nhap mang hai chieu
    NhapMang(a, m, n);
    //Xuat mang
    XuatMang(a, m, n);
}

```

```

// tim phan tu lon nhat va nho nhat trong mang
TimMinMax(a, m, n);

system("pause");
return 0;
}

void NhapMang(int a[][max_cot], int &m, int &n){
    int i, j;

    // nhap cac phan tu cho mang 2 chieu co m dong va n cot
    cout << " Nhap so dong : m = ";
    cin >> m;

    cout << "Nhap so cot : n = ";
    cin >> n;
    for (i = 0; i<m; i++)
        for (j = 0; j<n; j++)
        {
            cout << "\n a[" << i << "," << j << "] = ";
            cin >> a[i][j];
        }
}

void XuatMang(int a[][max_cot], int m, int n){
    int i, j;
    //xuat cac phan tu cua mang 2 chieu
    cout << "\n Cac phan tu cua mang A:\n";
    for (i = 0; i<m; i++)
    {
        for (j = 0; j<n; j++)
            cout << a[i][j] << " ";
        cout << endl;
    }
}

void TimMinMax(int a[][max_cot], int m, int n){
    int i, j;
    int max = a[0][0], min = a[0][0], x_max = 0, y_max = 0,
    x_min = 0, y_min = 0;
    for (i = 0; i<m; i++)
        for (j = 0; j<n; j++)
        {
            if (max <= a[i][j])
            {
                max = a[i][j];
                x_max = i;
            }
        }
}

```



```

        y_max = j;
    }

    if (min >= a[i][j])
    {
        min = a[i][j];
        x_min = i;
        y_min = j;
    }
}

cout << "\n Phan tu lon nhat cua mang la " << max << " tai
vi tri " << x_max << "," << y_max << endl;
cout << "\n Phan tu nho nhat cua mang la " << min << " tai
vi tri " << x_min << "," << y_min << endl;
}

```

CÁC BÀI TẬP THÊM CÓ ĐỘ KHÓ TRUNG BÌNH

1. Tìm giá trị lớn nhất, nhỏ nhất trong mảng 1 chiều
2. Tìm giá trị lớn thứ k trong mảng 1 chiều
3. Chèn 1 phần tử vào trong mảng 1 chiều không sử dụng mảng phụ
4. Nối 2 mảng một chiều thành một mảng 1 chiều.
5. Tính tổng các phần tử trên đường chéo chính trong mảng 2 chiều
6. Tính tổng các phần tử trên đường chéo phụ trong mảng 2 chiều
7. Sắp xếp các phần tử trên mảng 2 chiều tăng dần trên từng dòng.
8. Tính tổng và tích 2 ma trận.
9. Đếm số phần tử là số nguyên tố và tính tổng các phần tử này trong 1 mảng 2 chiều
10. Đếm số phần tử là số chính phương và tính tổng các phần tử này trong 1 mảng 2 chiều

CÁC BÀI TẬP THÊM CÓ ĐỘ KHÓ CAO

1. Kiểm tra xem mảng 1 chiều có tăng dần hay giảm dần không.
2. Đếm số mảng con tăng dần hoặc giảm dần trong mảng 1 chiều.
3. Cho mảng 1 chiều có n phần tử và $k < n$. In ra tổng lớn nhất của k phần tử liên tiếp xuất hiện trên mảng.
4. Đếm số lượng các phần tử khác nhau xuất hiện trong mảng 1 chiều.
5. Mảng x và y chứa hoành độ và tung độ của các điểm trên mặt phẳng hai chiều. In ra khoảng cách xa nhất giữa 2 điểm.

ĐỀ QUY

Xây dựng công thức truy hồi và viết hàm dùng phương pháp đệ quy cho các bài toán sau:

1. Kiểm tra n có phải là số nguyên tố hay không bằng đệ quy
2. Tìm số dư của phép chia x cho y (dùng đệ quy, không dùng toán tử $\%$ cho bài này)
3. Tìm ước số chung lớn nhất của hai số
4. Tính tổng n số tự nhiên đầu tiên
5. Tính tổng của n số lẻ đầu tiên
6. Tính tổng của n số nguyên tố đầu tiên (với hàm kiểm tra số nguyên tố đã được viết trước đó)