

Trường ĐH CNTP TP.HCM Khoa: CNTT Bộ môn: CNPM Môn: TH NMLT	BÀI 7. MẢNG 2 CHIỀU	
---	----------------------------	--

A. MỤC TIÊU:

- Khai báo được biến kiểu mảng 2 chiều (ma trận).
- Phân tích được tình huống sử dụng ma trận để lưu trữ dữ liệu.
- Cài đặt được hàm nhập, xuất, xử lý dữ liệu bằng cách sử dụng ma trận.
- Phân tích được tình huống sử dụng ma trận vuông để lưu trữ dữ liệu.
- Cài đặt được hàm nhập, xuất, xử lý dữ liệu bằng cách sử dụng ma trận vuông.
- Vận dụng được ma trận, ma trận vuông trong từng bài toán cụ thể.

B. DỤNG CỤ - THIẾT BỊ THÍ NGHIỆM CHO MỘT SV:

STT	Chủng loại – Quy cách vật tư	Số lượng	Đơn vị	Ghi chú
1	– Máy tính (có thể là máy bàn hoặc laptop).	1	cái	

C. VẬT LIỆU

- Visual C++.Net.
- Phần mềm hỗ trợ dạy học: NetOp School hoặc NetSupport School.

D. NỘI DUNG THỰC HÀNH

I. Tóm tắt lý thuyết

1. Định nghĩa:

Mảng hai chiều (còn gọi là ma trận) thực chất là mảng một chiều trong đó mỗi phần tử của mảng là một mảng một chiều, và được truy xuất bởi chỉ số dòng và chỉ số cột.

2. Cú pháp:

<tên kiểu dữ liệu> <tên mảng> [<số dòng>] [<số cột>];

Lưu ý: Chỉ số cột và chỉ số dòng luôn luôn bắt đầu từ 0.

Ví dụ:

- ☐ Khai báo mảng 2 chiều a có 3 dòng 2 cột chứa số nguyên:

int a[3][2];

- ☐ Khai báo mảng 2 chiều ch có 50 dòng 100 cột chứa ký tự:

char ch[50][100];

3. Truy xuất phần tử của mảng:

Tên_mảng [chỉ_số_dòng][chỉ số cột]

Ví dụ: Cho ma trận A như sau:

		Chỉ số cột			
		0	1	2	3
Chỉ số dòng	0	2	3	9	4
	1	5	6	7	6
	2	2	9	4	7

$A[0][0]=2; A[0][1]=3;$
 $A[1][1]=6; A[1][3]=6;$

4. Tham số mảng của một hàm

<Kiểu dữ liệu trả về> <Tên hàm> (<kiểu dữ liệu ma trận> <Tên ma trận> [<Số dòng tối đa>] [<Số cột tối đa>], **int m**, **int n**)

```
{  
    <Khởi tạo các biến>  
    <Duyệt mảng để tính toán>  
    <Trả về kết quả>  
}
```

Trong đó, m là số dòng thực sự, n là số cột thực sự sử dụng lưu trữ trong ma trận.

Qui ước: Trong phần thực hành chúng ta thống nhất dùng **Ma trận** thay cho **Mảng 2 chiều**.

II. Bài tập mẫu hướng dẫn thực hành.

1. Viết chương trình nhập các giá trị cho ma trận chứa các số nguyên, sau đó xuất mảng vừa nhập (Lưu ý: Viết chương trình dạng hàm).

Yêu cầu:

- Viết 2 hàm con Nhập và Xuất ma trận chứa các số nguyên.
- Gọi thực hiện 2 hàm con trong hàm main.

Hướng dẫn:

- Khai báo thêm các thư viện cơ bản cho chương trình.

```
#include <conio.h>
```

```
#include <stdio.h>
```

- Khai báo hằng số cho chương trình.

```
#define MAXROW 100
```

```
#define MAXCOL 100
```

- Viết các định nghĩa hàm cần thiết cho chương trình như sau:

```
//=====  
voidNhapMaTran_SoNguyen(int a[][MAXCOL], int &m, int &n)  
{  
    do  
    {  
        printf("Cho biet so dong cua mang: ");  
        scanf_s("%d", &m);
```

```

    } while(m <= 0);
do
{
    printf("Cho biet so cot cua mang: ");
    scanf_s("%d", &n);
}while(n <= 0);
for(int i=0; i< m; i++)
{
    for(int j=0; j < n; j++)
    {
        printf("Gia tri phan tu a[%d][%d]=", i, j);
        scanf_s("%d", &a[i][j]);
    }
}
}

//=====
void XuatMaTran_SoNguyen(int a[][MAXCOL], int m, int n)
{
    for(int i=0; i<m; i++)
    {
        for(int j=0; j<n; j++)
            printf("%5d", a[i][j]);
        printf("\n");
    }
}

//=====
void main()
{
    int A[MAXROW][MAXCOL];
    int M, N;
   NhapMaTran_SoNguyen(A, M, N);
    printf("\nNoi dung cua mang la: \n");
    XuatMaTran_SoNguyen(A, M, N);
    _getch();
}

```

2. Cho ma trận chứa các số nguyên. Viết chương trình theo kiểu menu cho phép người dùng chọn thực hiện một trong các thao tác sau (Lưu ý: Viết chương trình dạng hàm):

- Nhập ma trận số nguyên bằng cách tạo giá trị ngẫu nhiên cho các phần tử của ma trận
- Xuất ma trận số nguyên
- Xuất các giá trị của ma trận ra màn hình.
- Tính và xuất tổng các phần tử ma trận.
- Tìm max/min của các phần tử ma trận.

- Sắp xếp các giá trị của ma trận theo thứ tự tăng dần.

Hướng dẫn:

- Khai báo thêm các thư viện cơ bản cho chương trình.

```
#include<conio.h>
#include<stdio.h>
#include<stdlib.h>
#include<time.h>
#include<math.h>
```

- Khai báo hằng số cho chương trình.

```
#define MAXROW 100 //số dòng tối đa
#define MAXCOL 100 //số cột tối đa
```

- Viết các định nghĩa hàm cần thiết cho chương trình như sau:

```
//=====
void TaoMaTran_SoNguyen(int a[][MAXCOL], int &m, int &n)
{
    //SV tự làm
}
//=====
void XuatMaTran_SoNguyen(int a[][MAXCOL], int m, int n)
{
    //SV tự làm
}
//=====
int TongMaTran(int a[][MAXCOL], int m, int n)
{
    //SV tự làm
}
//=====
int MaxMaTran(int a[][MAXCOL], int m, int n)
{
    //SV tự làm
}
//=====
int MinMaTran(int a[][MAXCOL], int m, int n)
{
    //SV tự làm
}
//=====
void HoanVi(int &x, int &y)
{
    int temp=x;
    x=y;
```

```

        y=temp;
    }
    //=====
void SapXepMaTranTang(int a[][MAXCOL], int m, int n)
{
    int sopt=m*n;
    for(int i=0; i<sopt-1; i++)
        for(int j=i+1; j<sopt; j++)
            if(a[i/n][i%n]>a[j/n][j%n])
                HoanVi(a[i/n][i%n], a[j/n][j%n]);
}
-   Viết hàm main để thực thi chương trình.
//=====
void Menu()
{
    printf("\n*****");
    printf("\n*      MENU      *");
    printf("\n*-----*");
    printf("\n* 1. Khoi tao ma tran      *");
    printf("\n* 2. Xuat noi dung cua ma tran *");
    printf("\n* 3. Tong gia tri cac phan tu cua ma tran *");
    printf("\n* 4. Gia tri phan tu lon nhat, nho nhat *");
    printf("\n* 5. Sap xep ma tran tang dan *");
    printf("\n* 0. Thoat chuong trinh *");
    printf("\n*****");
}
//=====
void main()
{
    // SV tự tổ chức các lệnh cho chương trình thực thi theo yêu cầu:
    cho người dùng lựa chọn câu/bài muốn thực thi
}

```

III. Bài tập ở lớp

Bài 1. Cho một ma trận chứa các số nguyên. Viết một chương trình (*dạng menu*) thực hiện các yêu cầu sau:

1. Hoàn thành các bài tập ở phần (II)
2. Tạo ngẫu nhiên giá trị cho các phần tử của ma trận.
3. Xuất giá trị các phần tử của ma trận ra màn hình.
4. Tìm vị trí của giá trị x trong ma trận, x nhập từ phím.
5. Tìm phần tử max của dòng k trong ma trận.

6. Đếm phần tử dương trong ma trận.
7. Đếm số lần phần tử x xuất hiện trong ma trận.
8. Kiểm tra ma trận có vuông không?
9. Kiểm tra ma trận có toàn là số chẵn không?
10. Tìm dòng có tổng lớn nhất.
11. Tìm số chẵn lớn nhất trong ma trận.

Bài 2. Cho một ma trận vuông chứa các số nguyên. Viết một chương trình (dạng menu) thực hiện các yêu cầu sau:

12. Nhập ma trận vuông, các giá trị người dùng nhập từ phím
13. Nhập ma trận vuông, các giá trị tạo ngẫu nhiên
14. Xuất ma trận vuông
15. Tính tổng đường chéo chính
16. Đếm số phần tử chính phương trên đường chéo phụ
17. Kiểm tra ma trận dương không?
18. Kiểm tra ma trận có đường chéo chính toàn là số 1 không?
19. Kiểm tra ma trận đối xứng không?
20. Tìm vị trí phần tử có số lần xuất hiện nhiều nhất trong ma trận không?
21. Tìm số nguyên tố lớn nhất trong ma trận

IV. Bài tập về nhà

Cho một ma trận **vuông** chứa các số nguyên. Viết một chương trình (*dạng menu*) thực hiện các yêu cầu sau:

1. Xuất các phần tử trên đường chéo phụ
2. Xuất các phần tử thuộc tam giác trên.
3. Tìm cột có tổng lớn nhất trong ma trận.
4. Tìm dòng có tổng chẵn nhỏ nhất trong ma trận.
5. Tính tổng các phần tử trên đường chéo chính.
6. Tính tổng các phần tử chẵn thuộc tam giác trên.
7. Tính tổng các phần tử lẻ thuộc tam giác dưới.
8. Xuất các phần tử thuộc các đường biên của ma trận
9. Xuất các dòng tăng dần
10. Xuất các cột giảm dần
11. Chuyển đổi vị trí 2 dòng k và l trong ma trận.
12. Chuyển đổi vị trí 2 cột k và l trong ma trận.
13. Sắp xếp ma trận theo quy tắc các phần tử trên dòng được sắp tăng dần
14. Sắp xếp ma trận theo quy tắc tổng các dòng tăng dần
15. Đếm số phần tử cực đại trong ma trận. biết phần tử cực đại là phần tử lớn hơn các phần tử xung quanh nó.

--Hết--