



BÀI THỰC HÀNH SỐ 01

1. MỤC TIÊU

Hiểu và cài đặt được các hàm xử lý trên cấu trúc mảng 2 chiều.

2. LÝ THUYẾT CẦN GHI NHỚ

2.1. Khai báo mảng hai chiều

2.1.1. Cú pháp

2.1.1.1. Khai báo

<kiểu cơ sở> <tên mảng> [<số dòng >] [<số cột >]

- Ý nghĩa

- **Tên mảng**: là tên biến mảng, được đặt đúng theo quy tắc đặt tên trong C/C++.
- **Số dòng**: là một hằng số nguyên, cho biết số lượng dòng tối đa trong mảng.
- **Số cột**: là một hằng số nguyên, cho biết số lượng cột tối đa trong mảng. Như vậy, số phần tử của mảng = **số dòng X số cột**.

- Ví dụ

- Khai báo mảng hai chiều có tên là **Arr** gồm 8 hàng, 14 cột và kiểu cơ sở là **int**. `int Arr [8][14];`
- Khai báo mảng hai chiều có tên là **Mang** gồm 10 hàng, 5 cột và kiểu cơ sở là **float**. `float Mang [10][5];`
- Khai báo mảng hai chiều có tên là **charArray** gồm 12 hàng, 30 cột và kiểu cơ sở là **char**. `char charArray[12][30];`

2.1.1.2. Khai báo đồng thời gán giá trị

- Ví dụ `int A[3][4] = { {2,3,9,4}, {5,6,7,6}, {2,9,4,7} };`

Với khai báo và gán giá trị ở trên, mảng hai chiều sẽ có hình dạng như sau:

	0	1	2	3
0	2	3	9	4
1	5	6	7	6
2	2	9	4	7

2.1.2. Truy cập vào các phần tử của mảng

- Mỗi phần tử của mảng được truy xuất thông qua Tên biến mảng theo sau là chỉ số dòng và chỉ số cột nằm trong cặp dấu ngoặc vuông **[Chỉ số dòng] [Chỉ số cột]**.

Ví dụ với khai báo cho mảng `a [4][6]`. Khi đó, mảng gồm $4 \times 6 = 24$ phần tử. Vị trí các phần tử của mảng theo dòng và cột như hình sau:

	0	1	2	3	4	5
0	[0][0]	[0][1]	[0][2]	[0][3]	[0][4]	[0][5]
1	[1][0]	[1][1]	[1][2]	[1][3]	[1][4]	[1][5]



2	[2][0]	[2][1]	[2][2]	[2][3]	[2][4]	[2][5]
3	[3][0]	[3][1]	[3][2]	[3][3]	[3][4]	[3][5]

Phần tử $a[1][3]$ là phần tử nằm trên dòng 1 cột 3 của mảng hai chiều ở trên.

- Chỉ số của phần tử mảng:
 - Chỉ số của phần tử mảng là một biểu thức mà giá trị là kiểu số nguyên.
 - Chỉ số của mảng có thể là một hằng, một biến hay một biểu thức đại số.
- Với cách truy xuất theo kiểu này, Tên biến mảng[Chỉ số dòng] [Chỉ số cột] có thể coi như là một biến có kiểu dữ liệu là kiểu được chỉ ra trong khai báo biến mảng. Vì vậy, các phép toán (hoặc thao tác) trên kiểu dữ liệu trong C/C++ sẽ được áp dụng cho các phần tử biến có cùng kiểu.
- Ví dụ: với khai báo `int A[3][4] = { {2,3,9,4}, {5,6,7,6}, {2,9,4,7} };`
 Gán giá trị của phần tử $A[1][2]$ cho biến x: `int x=A[1][2]; // => x=7`
 Gán giá trị 100 cho phần tử $A[1][2]$: `A[1][2]=100;`

Trước khi lệnh <code>A[1][2]=100</code> thực hiện					Sau khi lệnh <code>A[1][2]=100</code> thực hiện				
	0	1	2	3		0	1	2	3
0	2	3	9	4	0	2	3	9	4
1	5	6	7	6	1	5	6	100	6
2	2	9	4	7	2	2	9	4	7

2.2. Nhập dữ liệu cho mảng hai chiều

2.2.1. Khai báo hằng

```
#define m 20 // m là số dòng
#define n 30 // n là số cột
```

2.2.2. Khai báo mảng 2 chiều có kiểu số nguyên

```
int a [m][n];
```

- Tổng quát: nhập dữ liệu cho mảng hai chiều gồm m dòng và n cột

```
void NhapMang (int A[][MAX], int m, int n)
{
    for (int row=0 ; row<m ; row++)
        for (int col=0 ; col<n ; col++)
        {
            printf ("Nhap gia tri cho phan tu [%d][%d]:\n",row,col);
            scanf ("%d" , &a[row][col]) ;
        }
}
```

2.3. Xuất dữ liệu cho mảng hai chiều

- Tổng quát, xuất dữ liệu cho từng phần tử của mảng hai chiều gồm m dòng n cột lưu trữ các số nguyên

```
void XuatMang(int a[][MAX], int m, int n)
{
    for (int i =0 ; i<m ; i++)
```



```

    {
        for (int j=0; j< n ; j++)
            printf ("%4d", a[i][j]);
        printf (" \n");
    }
}

```

3. BÀI TẬP CÓ HƯỚNG DẪN

Bài 1. Thao tác trên mảng 2 chiều

- **Yêu cầu:** Định nghĩa 2 hằng số ROWS=100 và COLS=200. Viết chương trình gồm các hàm chức năng như sau:

- (i)- Viết hàm nhận 2 tham số là chuỗi thông báo và giá trị tối đa (value), hàm cho nhập vào một số nguyên dương X với giá trị X thỏa điều kiện: $0 < X \leq \text{value}$. Nếu nhập sai, chương trình báo lỗi và cho nhập lại cho đến khi đúng. Sau khi nhập hoàn tất, hàm trả về số X vừa nhập. Hàm này sẽ được dùng lần lượt để nhập giá trị cho số dòng và số cột của mảng 2 chiều. Nguyên mẫu hàm có dạng như sau:

```
int NhapSo(char str[], int value)
```

- (ii)- Viết hàm nhập vào mảng hai chiều các số nguyên gồm m dòng, n cột. Với tham số của hàm là tên mảng 2 chiều (A), số dòng (m), số cột (n). Nguyên mẫu hàm có dạng như sau:

```
void NhapMang2Chieu (int a[][COLS], int m, int n)
```

- (iii)- Viết hàm xuất mảng hai chiều các số nguyên vừa nhập ở trên.

```
void XuatMang2Chieu(int a[][SIZE], int m, int n)
```

- (iv)- Viết hàm tính và trả về tổng các phần tử có trong mảng.
- (v)- Viết hàm tính và trả về tổng các phần tử nằm trên đường biên của mảng.
- (vi)- Viết hàm liệt kê các số may mắn có trong mảng.

X	X	X	X	X
X				X
X				X
X	X	X	X	X
Các phần tử trên đường biên				

- Hướng dẫn

Bước 1. Định nghĩa các hằng số sẽ sử dụng trong chương trình.

```
#define ROWS 100
#define COLS 200
```

Bước 2. Viết hàm nhập X ($0 < X \leq \text{value}$).

```

int NhapSo(char str[], int value)
{
    int n;
    do
    {
        printf("Nhap so %s (tu 1-100): ", str);
        scanf("%d", &n);
        if ((n<=0) || (n> value))
            printf("Chi nhan gia tri tu 1 den 100");
    } while ((n<=0) || (n>value));
    return n;
}

```

**Bước 3.** Viết hàm nhập vào mảng gồm m dòng n cột

```
void NhapMang2Chieu (int a[][COLS], int m, int n)
{
    int i,j;
    for (i=0; i<m; i++)
        for (j=0; j<n; j++)
        {
            printf("Nhap gia tri cho phan tu a[%d][%d]: ",i,j);
            scanf("%d",&a[i][j]);
        }
}
```

Bước 4. Viết hàm xuất mảng số nguyên m x n phần tử vừa nhập ở trên.

```
void XuatMang2Chieu(int a[][SIZE], int m, int n)
{
    printf("Ma tran vua tao:\n");
    for (int i=0; i<m; i++)
    {
        for(int j=0; j<n; j++)
            printf("%5d",a[i][j]);
        printf("\n");
    }
}
```

Bước 5. Viết hàm main () kết 3 hàm trên sao cho chạy ổn định rồi mới viết tiếp hàm khác.

```
int main()
{
    int m,n, a[ROWS][COLS];
    m=NhapSo("dong", ROWS);
    n=NhapSo("cot", COLS);
    NhapMang2Chieu (a, m, n);
    XuatMang2Chieu(a, m, n);
    return 0;
}
```

Bước 6. Viết hàm tính tổng tất cả các phần tử của mảng. Gọi và nhận kết quả trả về từ hàm này trong ham main để xuất kết quả.

```
int TongMang(int a[][SIZE], int m, int n)
{
    int t=0;
    for (int i=0; i<m; i++)
        for(int j=0; j<n; j++)
            t+=a[i][j];
    return t;
}
```

Bước 7. Viết hàm Tính tổng các phần tử nằm trên đường biên của mảng. Gọi và nhận kết quả trả về từ hàm này trong ham main để xuất kết quả.

```
int TongBien(int a[][SIZE], int m, int n)
{
    int t=0;
    for (int i=0; i<m; i++)
```



```
{
    for(int j=0;j<n;j++)
        if ( (i==0) || (i==m-1) || (j==0) || (j==n-1) )
            t+=a[i][j];
}
return t;
}
```

Bước 8. Viết hàm liệt kê các số may mắn có trong mảng. Gọi hàm này trong hàm main để xuất kết quả.

```
bool LaSoMayMan(int k)
{
    while (k>0)
    {
        int so=k%10;
        if ( (so!=6) && (so!=8))
            return false;
        k=k/10;
    }
    return true;
}

void LietKeSoMayMan(int a[][SIZE], int m, int n)
{
    printf("\nCac so may man co trong ma tran: ");
    for (int i=0; i<m; i++)
    {
        for(int j=0;j<n;j++)
            if (LaSoMayMan(a[i][j]))
                printf("%5d",a[i][j]);
    }
}
```

Sau khi hoàn tất, hàm main sẽ có dạng như sau:

```
#define SIZE 100
int NhapSo(char *str);
void NhapMang2Chieu (int a[][SIZE], int m, int n);
void XuatMang2Chieu(int a[][SIZE], int m, int n);
int TongMang(int a[][SIZE], int m, int n);
int TongBien(int a[][SIZE], int m, int n);
bool LaSoMayMan(int k);
void LietKeSoMayMan(int a[][SIZE], int m, int n);

int main()
{
    int m,n, a[SIZE][SIZE];
    m=NhapSo("dong");
    n=NhapSo("cot");
    NhapMang2Chieu (a, m, n);
    XuatMang2Chieu(a, m, n);
    printf("\nTong cac so co trong mang la: %d",TongMang(a,m,n));
    printf("\nTong cac so tren duong bien cua mang:
                                                %d",TongBien(a,m,n));

    LietKeSoMayMan(a, m, n);
    return 0;
}
```



4. BÀI TẬP DO SINH VIÊN TỰ THỰC HIỆN

4.1. Nhập (hoặc tạo) giá trị cho các phần tử trong mảng hai chiều

Bài 1. Viết hàm nhập mảng hai chiều các số nguyên dương gồm n dòng và m cột ($5 < n, m < 200$). Giá trị nhập Khi nhập sai, chương trình báo lỗi và sẽ yêu cầu nhập lại).

Bài 2. Viết hàm khởi tạo giá trị các phần tử là ngẫu nhiên cho mảng hai chiều các số nguyên kích thước n dòng và m cột ($2 \leq n, m < 50$) cho các trường hợp sau:

- Giá trị của các phần tử được phát sinh ngẫu nhiên và nằm trong khoảng từ -100 đến 100.
- Giá trị của các phần tử là $a[i][j] = i * j$.
- Giá trị của các phần tử được phát sinh ngẫu nhiên và nằm trong khoảng từ 0 đến 99.
- Giá trị của các phần tử được phát sinh ngẫu nhiên và phải toàn là số lẻ.
- Giá trị của các phần tử được phát sinh ngẫu nhiên và phải toàn là số nguyên tố.
- Giá trị của các phần tử được phát sinh ngẫu nhiên sao cho phần tử $a[i][j]$ chỉ nhận số chẵn khi $i+j$ là số lẻ và chỉ nhận số lẻ khi $i+j$ là số chẵn.

4.2. Xuất

Bài 3. Viết hàm xuất mảng hai chiều các số thực A(m x n) trong các trường hợp sau:

- Thay giá trị số nguyên tố bằng các khoảng trắng, các số khác giữ nguyên giá trị.
- Thay giá trị của phần tử không nằm trên đường biên bằng các khoảng trắng, các giá trị trên đường biên được giữ nguyên giá trị.

5	18	4	7
2	6	11	8
19	15	20	1
16	3	17	14
10	13	9	12
Mảng ban đầu			

	18	4	
	6		8
	15	20	1
16			14
10		9	12
Kết quả thực hiện câu a			

5	18	4	7
2			8
19			1
16			14
10	13	9	12
Kết quả thực hiện câu b			

	18		7
2		11	
	15		1
16		17	
	13		12
Kết quả thực hiện câu c			

- Thay giá trị của phần tử có chỉ số dòng + chỉ số cột là số chẵn bằng các khoảng trắng, các số khác giữ nguyên giá trị.

4.3. Tính tổng/trung bình

4.3.1. Thao tác trên toàn bộ các phần tử có trong mảng hai chiều

Viết hàm thực hiện những yêu cầu sau (mỗi yêu cầu được viết thành 1 hàm riêng):

Bài 4. Tính tổng tất cả các số

Bài 5. Tính tổng các số chẵn.

Bài 6. Tính giá trị trung bình cộng

Bài 7. Tính giá trị trung bình nhân

Bài 8. Tính tổng các số nguyên tố.



Bài 9. Tính tổng các số hoàn thiện. Biết rằng số hoàn thiện là số (n) có tổng các ước số không kể n bằng chính n. Ví dụ **6 là số hoàn thiện** vì $1+2+3 = 6$

4.3.2. Thao tác trên dòng có trong mảng hai chiều

Viết hàm thực hiện những yêu cầu sau (mỗi yêu cầu được viết thành 1 hàm riêng):

Bài 10. Tính tổng tất cả các số trên mỗi dòng. Sau đó in ra giá trị tổng dòng lớn nhất.

Bài 11. Tính tổng các số chẵn trên mỗi dòng. Sau đó in ra giá trị tổng các số chẵn lớn nhất.

Bài 12. Tính tổng các số âm (<0) trên mỗi dòng. Sau đó in ra giá trị tổng âm lớn nhất.

Bài 13. Tính giá trị trung bình cộng trên mỗi dòng. Sau đó in ra giá trị tổng trung bình lớn nhất.

Bài 14. Tính tổng các số nguyên tố trên mỗi dòng. Nếu dòng không có số nguyên tố, xem như tổng của dòng đó là 0.

Từ bài 10 đến 14 thực hiện in ra màn hình theo dạng:

Mảng hai chiều ban đầu

4	2	9	7
2	2	9	4
7	8	3	4
4	3	5	6

Kết quả in ra màn hình \Rightarrow

4	2	9	7	Tổng dòng =	22
2	2	9	4	Tổng dòng =	17
7	8	3	4	Tổng dòng =	22
4	3	5	6	Tổng dòng =	18

Tổng dòng lớn nhất là 22, dòng lớn nhất là dòng 0 và 2

Minh họa kết quả in ra màn hình của bài 103 (ký hiệu \cup đại diện cho khoảng trắng)

Mảng hai chiều ban đầu

4	2	9	7
2	2	9	4
7	8	3	4
4	3	5	6

Kết quả in ra màn hình \Rightarrow

\cup	\cup	\cup	7
2	\cup	\cup	\cup
7	\cup	\cup	\cup
\cup	\cup	5	\cup

4.3.3. Thao tác trên các cột có trong mảng hai chiều

Thực hiện tương tự như các yêu cầu có trong bài tập 4.3.2, nhưng thay dòng thành cột.

4.3.4. Thao tác trên các phần tử nằm trên đường biên trong mảng hai chiều

Bài 15. Tính tổng:

- Tất cả các số trên đường biên.
- Tất cả các số KHÔNG thuộc đường biên.