KỸ THUẬT LẬP TRÌNH C

Chương 7: Mảng hai chiều

1. Khái niệm
2. Khai báo mảng hai chiều
3. Khởi tạo mảng hai chiều
4. Nhập xuất mảng hai chiều
5. Dùng mảng hai chiều làm tham số truyền cho hàm
6. Các bài toán cơ bản
7. Mảng vuông

1. Khái niệm

Mảng hai chiều thực chất là mảng một chiều trong đó mỗi phần tử của mảng lại là mảng một chiều, và được truy xuất bởi hai chỉ số dòng và cột.

2. Khai báo mảng hai chiều

Cú pháp:

<Kiểu> <tên mảng>[<Kích thước dòng>][<Kích thước cột>];

Mỗi phần tử của mảng được truy nhập thông qua tên mảng cùng với hai chỉ số: chỉ số dòng (bắt đầu từ 0 đến <Kích thước dòng> – 1) và chỉ số cột (bắt đầu từ 0 đến <Kích thước cột> – 1).

Ví dụ: int a[2][3]

Khai báo mảng hai chiều gồm 6 phần tử kiểu int, bao gồm:

a[0][0]	a[0][1]	a[0][2]
a[1][0]	a[1][1]	a[1][2]

3. Khởi tạo mảng hai chiều

Ví dụ 1:

```
int a[2][3] = {
    {11, 12, 13},
    {21, 22, 23} };
```

Ví dụ 2: Khởi tạo có thể không cần chỉ ra kích thước dòng

```
double b[][3] = {
   {6, 7.8, 8},
   {3, 12.6, 4},
   {6.5, 20, 7} };
```

Ví dụ 3: Sử dụng mảng hai chiều khởi tạo để vẽ chữ H bằng các dấu "*"

```
#define SIZE1 5
#define SIZE2 5
void main() {
    int a[SIZE1][ SIZE2] = {
        {1, 0, 0, 0, 1}, {1, 0, 0, 0, 1}, {1, 1, 1, 1, 1},
        {1, 0, 0, 0, 1}, {1, 0, 0, 0, 1}
    };
    for(i = 0; i < SIZE1; i++) {
        for(j = 0; j < SIZE2; j++) {
            if(a[i][j] == 1) printf("*")
            else printf(" "); //Khoảng trắng
        }
        printf("\n");}}
```

4. Nhập xuất mảng hai chiều

Ví dụ:

```
#define SIZE1 4
#define SIZE2 6
void main() {
    float a[SIZE1][SIZE2]; /*Khai báo mảng hai chiều
          gồm SIZE1*SIZE2 phần tử kiểu thực*/
    int sd, sc; //Số dòng và số cột
    int i, j;
    float tam;
```

```
do {
        printf("Nhap so dong va so cot:");
        scanf("%d%d", &sd, &sc);
} while(sd < 1 || sd > SIZE1 || sc < 1 || sc > SIZE2);
//Nhập dữ liệu mảng từ bàn phím
for(i = 0; i < sd; i++)
        for(j = 0; j < sc; j++)  {
                printf("pt thu [%d][%d]:", i, j);
                scanf("%f", &tam);
                a[i][j] = tam;
        }
```

```
        //Xuất dữ liệu mảng ra màn hình
        for(i = 0; i < sd; i++)  {
                for(j = 0; j < sc; j++)
                        printf("%0.2f\t", a[i][j]);
                printf("\n");
        }
}
```

Lưu ý: Trong C, đối với các phần tử mảng hai chiều không nguyên ta không thể sử dụng toán tử lấy địa chỉ.

5. Dùng mảng hai chiều làm tham số truyền cho hàm

- Tên mảng hai chiều cũng là một **hằng địa chỉ** và nó chính là **địa chỉ phần tử đầu tiên** của mảng.
- Khi dùng tên mảng làm tham số thực truyền cho hàm thì thực chất là địa chỉ phần tử đầu tiên của mảng được truyền cho hàm và như vậy tham số hình thức tương ứng trong định nghĩa hàm phải viết dưới dạng con trỏ.

Ví dụ:

```
#define SIZE1 4
#define SIZE2 6
// khai báo các nguyên mẫu hàm
void Nhap(float (*a)[SIZE2], int *sd, int *sc);
void Xuat(float (*a)[SIZE2], int sd, int sc);
// Định nghĩa các hàm
```

```
void Nhap(float (*a)[SIZE2], int *sd, int *sc) {
        int i, j; float tam;
        do {
                printf("Nhap so dong va so cot:");
                scanf("%d%d", &(*sd), &(*sc));
        } while(*sd < 1 || *sd > SIZE1 || *sc < 1 || *sc >
                SIZE2);
        for(i = 0; i < *sd; i++)
                for(j = 0; j < *sc; j++)  {
                        printf("pt thu [%d][%d]:", i, j);
                        scanf("%f", &tam);
                        a[i][j] = tam;
                }
}
```

```
void Xuat(float (*a)[SIZE2], int sd, int sc) {
        int i, j;
      for(i = 0; i < sd; i++) {
              for(j = 0; j < sc; j++)
                      printf("%0.2f\t", a[i][j]);
              printf("\n");
      }
}
void main() {
      float a[SIZE1][SIZE2];
      int sd, sc;
      Nhap(a, &sd, &sc); Xuat(a, sd, sc);
}
```

Lưu ý: Tham số hình thức tương ứng với tham số thực là tên mảng hai chiều cũng có thể viết như sau:

```
void Nhap(float a[][SIZE2], int *sd, int *sc);
void Xuat(float a[][SIZE2], int sd, int sc);
```

6. Các bài toán cơ bản trên mảng hai chiều

- Duyệt hết các phần tử của mảng
  - Duyệt theo từng dòng từ trên xuống dưới, với mỗi dòng duyệt từ trái sang phải

```
for(i = 0; i < sd; i++)
        for(j = 0; j < sc; j++) {
                //Xử lý a[i][j]
                ...
        }
```

- Duyệt theo từng cột từ trái sang phải, với mỗi cột duyệt từ trên xuống dưới

```
for(j = 0; j < sc; j++)
        for(i = 0; i < sd; i++) {
                //Xử lý a[i][j]
                ...
        }
```

- Duyệt các phần tử nằm trên cùng một dòng hay trên cùng một cột
  - Duyệt các phần tử trên dòng có chỉ số k (0 ≤ k < sd)

```
for(i = 0; i < sc; i++) {
        <Xử lý a[k][i]> }
```

  - Duyệt các phần tử trên cột có chỉ số k (0 ≤ k < sc)

```
for(i = 0; i < sd; i++) {
        <Xử lý a[i][k]>}
```

7. Mảng vuông (ma trận vuông)

a. Tính chất

Mảng vuông là mảng hai chiều có số dòng = số cột = n, n gọi là cấp ma trận. Sau đây là một số đặc tính của ma trận vuông:

- Các phần tử nằm trên đường chéo chính là các phần tử a[i][i] (0 ≤ i < n)
- Các phần tử nằm trên đường chéo phụ là các phần tử a[i][n – 1- i] (0 ≤ i < n)

- Các phần tử nằm trong nửa mảng vuông phía trên đường chéo chính là các phần tử a[i][j] với i ≤ j
- Các phần tử nằm trong nửa mảng vuông phía dưới đường chéo chính là các phần tử a[i][j] với i ≥ j
- Các phần tử nằm trong nửa mảng vuông phía trên đường chéo phụ là các phần tử a[i][j] với i + j ≤ n – 1
- Các phần tử nằm trong nửa mảng vuông phía dưới đường chéo phụ là các phần tử a[i][j] với i + j ≥ n – 1

b. Duyệt mảng vuông

- Duyệt các phần tử nằm trên ĐCC

```
for(i = 0; i < n; i++)
        <Xử lý a[i][i]>
```

- Duyệt các phần tử nằm trên ĐCP

```
for(i = 0; i < n; i++)
        <Xử lý a[i][n – 1 – i]>
```

- Duyệt các phần tử nằm trong nửa mảng vuông.
  - Phía trên ĐCC

```
for(i = 0; i < n; i++)
        for(j = 0; j < n; j++)
                if(i <= j) {
                        <Xử lý a[i][j]> }
```

  - Phía dưới ĐCC

```
for(i = 0; i < n; i++)
        for(j = 0; j < n; j++)
        if(i >= j) {
                <Xử lý a[i][j]> }
```

- Phía trên ĐCP

```
for(i = 0; i < n; i++)
        for(j = 0; j < n; j++)
                if(i + j <= n – 1)
                        <Xử lý a[i][j]>
```

- Phía dưới ĐCP

```
for(i = 0; i < n; i++)
        for(j = 0; j < n; j++)
                if(i  + j >= n – 1)
                        <Xử lý a[i][j]>
```

Hết