

**Tin học cho lớp 11 chuyên Tin**  
**Năm học 2009-2010**  
**Phần II. NHẬP MÔN LẬP TRÌNH THEO NGÔN NGỮ C**

**Bài 4b. Con trỏ và mảng (tiếp theo)**

**1. Mảng hai chiều**

**Cú pháp:** `<kiểu> <tên mảng>[số hàng][số cột];`  
và chỉ số của hàng/cột của mảng sẽ được tính từ 0 trở đi đến số lượng-1.

**Ví dụ:** `float a[3][4]; /*3 hàng: 0,1,2; 4 cột: 0,1,2,3*/`  
Để truy cập đến phần tử hàng *i*, cột *j* ta dùng `a[i][j]`

Tuy nhiên nó cũng có thể khai báo cùng với các giá trị khởi tạo:

```
int a[3][2]={{1,2},{2,3},{3,4}};           bình thường.  
float a[][4]={{1,2,2,3},{3,4,5,6}};       máy hiểu có 2 hàng  
float a[10][4]={{1,2,2,3},{3,4,5,6}};     2 hàng đầu đã có giá trị.
```

**Nhớ là dù sao số cột phải được ấn định ngay!**

**Chú ý:**

**Tên mảng lại chính là một con trỏ giữ địa chỉ của mảng của mảng 1 chiều, và:**

**a** bằng địa chỉ của phần tử đầu tiên của mảng một chiều thứ 0: `a = &a[0][0].`  
**a+1** bằng địa chỉ của phần tử đầu tiên của mảng một chiều thứ 1: `a+1 = &a[1][0].`  
**a+i** bằng địa chỉ của phần tử đầu tiên của mảng một chiều thứ *i*: `a+i = &a[i][0].`  $\forall i < \text{số hàng.}$

Như vậy ta có hệ thức:

$$*(a+i) = a[i][0].$$

Không dùng con trỏ `a+?` để trỏ tới từng phần tử như mảng 1 chiều được, mà dùng một biến con trỏ khác.

**Ví dụ:**

```
int a[3][4], *p; //Số hàng = m = 3, số cột = n = 4.
```

**Tuy a và p đều là kiểu con trỏ, nhưng không thể gán:**

```
p=a;
```

**được, vì p trỏ tới một số nguyên, còn a thì trỏ tới một mảng 1 chiều. Do đó phải ép kiểu:**

```
p=(int *)a;
```

Với cách thức đó, thì sau đó ta có các hệ thức sau: Về phải của phép gán cuối cùng là con trỏ trỏ tới một vùng nhớ 2B thôi, nên gán được cho p. Bước nhảy của p sẽ là 2B (cỡ của số kiểu int).

**p** bằng địa chỉ của phần tử thứ 0 của mảng một chiều thứ 0 = `&a[0][0].`

**p+1** bằng địa chỉ của phần tử thứ 1 của mảng một chiều thứ 0 = `&a[0][1].`

**p+j** bằng địa chỉ của phần tử thứ *j* của mảng một chiều thứ 0 = `&a[0][j].`  $\forall j < \text{số cột.}$

**p+n** bằng địa chỉ của phần tử thứ 0 của mảng một chiều thứ 1 = `&a[1][0].`

**p+n+1** bằng địa chỉ của phần tử thứ 1 của mảng một chiều thứ 1 = `&a[1][1].`

**p+n+j** bằng địa chỉ của phần tử thứ *j* của mảng một chiều thứ 1 = `&a[1][j].`  $\forall j < \text{số cột.}$

**Một cách tổng quát:** Với khai báo

```
<kiểu phần tử> a[m][n], *p;
```

Thì sau phép gán:

```
p=(<kiểu phần tử> *)a;
```

với mọi  $i, j \geq 0$  và  $i < m, j < n$ , ta luôn có:

$$p+i*n+j = \&a[i][j]$$

và

$$a[i][j] = *(p+i*n+j)$$

Sau đây là một ví dụ đơn giản:

Ví dụ 1:

Xét một mảng 3 hàng, 5 cột các số nguyên từ -10 đến 10:

Nhập tự động, xuất thành bảng đẹp mắt.

Tìm max rồi in ra tất cả các phần tử bằng max nếu có cùng các chỉ số hàng, cột của chúng.

```
#include <conio.h>
#include <stdlib.h>
main()
{
int a[3][5],max,dem;
unsigned char i,j;
clrscr();
randomize();
/*Nhap tu dong:*/
for (i=0;i<3;i++)
for (j=0;j<5;j++)
a[i][j]=random(11)-random(11);
/*Xuat thanh bang dep mat:*/
printf("Mang 2 chieu da tu dong nhap la:\n");
for (i=0;i<3;i++)
{
for (j=0;j<5;j++)
printf("%5d",a[i][j]); /*moi so hang chiem 5 vi tri*/
printf("\n");
}
/*Tim max:*/
max=a[0][0];
for (i=0;i<3;i++)
for (j=0;j<5;j++)
if (a[i][j]>max) max=a[i][j];
/*in ra cac phan tu bang max*/
printf("max=%d",max);
printf("Danh sach cac phan tu bang max:")
dem=0;
for (i=0;i<3;i++)
for (j=0;j<5;j++)
if (a[i][j]==max)
{
printf("a[%d,%d], ",i,j);
dem++;
}
printf("Co %d phan tu bang max\n",dem);
getch();
}
```

## 2. Bài tập thực hành:

Nhập bảng gồm 10 hàng, 10 cột các số nguyên, rồi in ra màn hình:

1. Trung bình cộng các phần tử của từng hàng
2. Tổng các số hạng dương của từng cột
3. Số hạng nhỏ nhất và chỉ số của cả mảng.
4. Các số hạng nhỏ nhất và chỉ số của cả mảng.
5. Đảo hàng thành cột và cột thành hàng.
6. Tìm điểm yên ngựa trên mảng