Ngôn ngữ lập trình C

B10: Mảng (Arrays)

PHENIKAA UNIVERSITY

Khoa Công nghệ thông tin

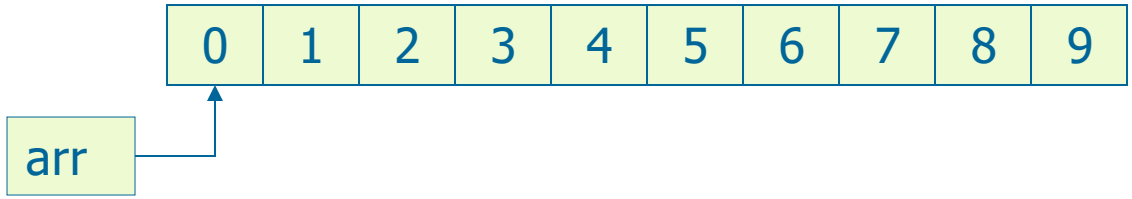
Mảng

- Khái niệm Mảng: là một tập hợp hữu hạn các phần tử có cùng kiểu dữ liệu được lưu trữ liên tiếp nhau trong bộ nhớ.
- Mảng có thể được khai báo cho mọi kiểu dữ liệu của C.
  - Ví dụ: `int Arr[10]` là một mảng có 10 số nguyên kiểu `int`.
- Các ví dụ trong việc sử dụng mảng:
  - Danh sách điểm của sinh viên;
  - Chuỗi các số được nhập bởi người dùng;
  - Véc tơ;
  - Ma trận; …

Mảng trong bộ nhớ

- Các phần tử trong mảng có cùng tên (và cũng là tên mảng) nhưng phân biệt với nhau ở chỉ số cho biết vị trí của chúng trong mảng.
- Ví dụ:

```
int arr[10];
```



- Phần tử thứ n của mảng `arr` có vị trí thứ n-1 (`arr`[n-1]). Vị trí đầu tiên của của mảng bắt đầu từ 0.

Khai báo mảng

- Cú pháp mảng 1 chiều:

```
Kieu_du_lieu ten_mang[kich_thuoc_mang];
```

- Ví dụ:
  - `char c[12];`
    - Phần tử đầu tiên có chỉ số 0
    - n phần tử của mảng có tên c: `c[0], c[1], ..., c[n-1]`
  - `int mang_nguyen[4];`
  - `float mang_thuc[6];`

Khai báo mảng

- Mảng nhiều chiều: mỗi phần tử của mảng cũng là một mảng khác → giống vector trong toán học.
- Ví dụ:
  - Mảng 2 chiều: `int a[6][5];`
  - Mảng 3 chiều: `int b[3][4][5];`

Khởi tạo giá trị mảng

- Mảng có thể được khởi tạo giá trị ngay khi khai báo
  - `int a[4] = {4, 9, 22, 16};`
  - `float b[3] = {40.5, 20.1, 100};`
  - `char c[5] = {'h', 'e', 'l', 'l', 'o'};`

→ Câu lệnh thứ nhất có tác dụng tương đương với 4 lệnh gán:

- `a[0] = 4;`
- `a[1] = 9;`
- `a[2] = 22;`
- `a[3] = 16;`

Khởi tạo giá trị mảng

- Số giá trị khởi tạo không thể lớn hơn số phần tử của mảng nhưng có thể ít hơn.
- Kích thước mảng có thể được suy ra từ số lượng khởi tạo bằng cách để trống dấu ngoặc vuông.
- Ví dụ:
  - `int array1 [8] = {2, 4, 6, 8, 10, 12, 14, 16};`

  tương đương
  - `int array2 []  = {2, 4, 6, 8, 10, 12, 14, 16};`

Sử dụng mảng

- Truy cập vào 1 phần tử của mảng thông qua tên mảng và chỉ số của phần tử đó.
- Cú pháp:

```
ten_mang[chi_so_cua_phan_tu]
```

- Ví dụ 1: `int mang_nguyen[3];`
  - `mang_nguyen[0]`: Phần tử thứ 1.
  - `mang_nguyen[1]`: Phần tử thứ 2.
  - `mang_nguyen[2]`: Phần tử thứ 3

Sử dụng mảng

- Ví dụ 2: `int a[6][5];`
  - a[0] là phần tử đầu tiên của mảng, là 1 mảng 1 chiều;
  - Phần tử đầu tiên của mảng `a[0]` là `a[0][0]`, …
  - …
  - `a[2][3]` sẽ là phần tử thứ 4 của phần tử thứ 3 của a.
  - `a[i][j]` sẽ là phần tử thứ j+1 của `a[i]`, mà phần tử `a[i]` lại là phần tử thứ i+1 của `a`.

Ví dụ 10.1

- Tạo một mảng các số chẵn từ 2 đến 20. Rồi in mảng đó ra màn hình.

```
#include <stdio.h>
#define arraySize 10

void main()
{
        int s[ arraySize ];  // array S has 10 elements
        int i;
        for ( i = 0; i < arraySize; i++ )
            s[ i ] = 2 + 2 * i;
        printf("Element \t Value\n");
        for ( i = 0; i < arraySize; i++ )
                    printf("%d\t%d\n", i, s[i]);
}
```

Các thao tác cơ bản làm việc trên mảng

- Nhập dữ liệu cho mảng;
- Tìm kiếm trên mảng;
- Sắp xếp mảng;

Các thao tác cơ bản làm việc trên mảng

- Nhập dữ liệu cho mảng:
  - Nhập dữ liệu cho từng phần tử của mảng.
  - Ví dụ 1:

```
float a[10]; int i;
scanf("%f",&a[1]);
a[2] = a[1] + 5;
```

  - Ví dụ 2:

```
int b[10], i;
//Nhap gia tri tu ban phim cho tat ca cac phan tu mang b
for(i = 0; i < 10; i++)
{
 printf("\n Nhap gia tri cho b[%d]", i);
 scanf("%d",&b[i]);
}
```

Các thao tác cơ bản làm việc trên mảng

- Nhập dữ liệu cho mảng:
  - Trường hợp không biết mảng sẽ có bao nhiêu phần tử mà chỉ biết số phần tử tối đa có thể có của mảng.
  - Ví dụ:

```
int a[100];//Khai bao mang, so phan tu toi da la 100
int n; // Bien luu giu so phan tu thuc su cua mang
int i;
printf("\n Cho biet so phan tu cua mang: ");
scanf("%d",&n);
for(i = 0; i < n; i++)
 {
  printf("\n a[%d] = ", i);
  scanf("%d",&a[i]);
 }
```

Bài tập 10.1

- Tạo mảng quản lý lượng mưa trong năm:
  - Viết hàm Nhập số liệu từ bàn phím vào mảng;
  - Viết hàm In số liệu ra màn hình;
  - Gợi ý sử dụng mảng khai báo biến toàn cục.

Tháng	Lượng mưa (in mm)
1	40
2	45
3	95
4	130
5	220
6	210
7	185
8	135
9	80
10	40
11	45
12	30

Bảng lượng mưa trong năm

Bài tập 10.2

- Lập chương trình thực hiện các công việc sau:
  - Nhập một dãy n số bất kỳ từ bàn phím, với n < 10 nhập từ bàn phím.
  - Đưa dãy số đã nhập ra màn hình.
  - Tính: trung bình cộng các số âm, tổng các số dương, đếm số các số 0 trong dãy.
  - Nhập một số x bất kỳ từ bàn phím. In ra vị trí các số trong dãy có giá trị bằng x.

Bài tập 10.3

- Lập chương trình thực hiện các công việc sau:
  - Nhập một dãy n số bất kỳ vào 1 mảng, với n <= 10 nhập từ bàn phím.
  - Sắp xếp dãy số đã nhập theo thứ tự tăng dần, đưa kết quả ra màn hình.
  - Sắp xếp dãy số đã nhập theo thứ tự giảm dần, đưa kết quả ra màn hình.
  - Sắp xếp dãy số giảm dần theo giá trị tuyệt đối, đưa kết quả ra màn hình.

Bài tập 10.4

- Lập chương trình thực hiện các công việc sau:
  - Nhập một dãy n số bất kỳ từ bàn phím vào mảng a với n <= 10 nhập từ bàn phím.
  - Sao chép tất cả các số dương trong mảng a sang mảng b, tất cả các số âm sang mảng c.
  - Đưa các mảng b, c ra màn hình.

Bài tập 10.5

- Lập chương trình thực hiện các công việc sau:
  - Nhập một dãy n số bất kỳ từ bàn phím vào mảng a với n <= 10 nhập từ bàn phím.
  - Tính tổng của các cực đại địa phương của dãy (cực đại địa phương của dãy số là số lớn hơn hai số lân cận nó, số phía trước và số phía sau).

Mảng là đối số của hàm

- Hàm chấp nhận mảng làm đối số của hàm
- Khi mảng làm đối số cho một hàm thì kích thước của mảng cần được xác định.

Mảng là đối số của hàm

- Hàm có đối là mảng một chiều: Hai cách khai báo hàm:
  - `Kiểu_hàm Tên_hàm(Kiểu_dữ_liệu *pa,...)`: pa là một con trỏ.
  - `Kiểu_hàm Tên_hàm(Kiểu_dữ_liệu pa[],...)`: pa coi như một mảng hình thức.
- Nếu tham số là tên mảng a một chiều có kiểu dữ liệu phù hợp thì ta gọi:

```
Tên_hàm(a,...).
```

Mảng là đối số của hàm

- **Hàm có đối là mảng 2 chiều**: Giả sử a là mảng 2 chiều float a[20][30]. Ta có thể khai báo theo hai cách sau:
- Cách 1:
  - `void Tên_hàm(float (*pa)[50], int m, int n)`
  - `void Tên_hàm(float pa[][50], int m, int n)`
  - `pa` là con trỏ kiểu `float[50]` (chứa địa chỉ đầu của vùng nhớ dành cho 50 số thực, tức là vùng nhớ 200 byte), m là số hàng và n là số cột
  - Trong thân hàm, để truy nhập tới phần tử hàng i cột j, ta dùng kí hiệu `pa[i][j]`.
  - Theo cách này, ta chỉ làm việc với mảng 2 chiều có tối đa 50 cột.
  - Lời gọi hàm:

```
Tên_hàm(a, m, n)
```

Mảng là đối số của hàm

- **Hàm có đối là mảng 2 chiều**:
  - Cách 2:
    - `float *pa;` biểu thị địa chỉ đầu của mảng a
    - `int N;` biểu thị số cột của mảng a
    - Khai báo: `Tên_hàm(float *pa, int N,...)`
    - Lời gọi: `Tên_hàm(a, 30,...)`
    - Trong thân hàm, để truy nhập tới phần tử `a[i][j]` ta dùng công thức `*(pa + i*N + j)`
    - Theo cách này, mảng 2 chiều quy về mảng 1 chiều, hàm có thể dùng cho bất kỳ mảng 2 chiều nào.

Mảng là đối số của hàm

- Khi hàm bắt đầu làm việc thì giá trị của a (địa chỉ phần tử a[0]) được truyền cho pa.
- Bên trong hàm, các mảng được sử dụng và truy cập như bình thường với bên ngoài hàm.
- Các thay đổi mảng bên trong hàm ảnh hưởng đến mảng gốc. (*Sinh viên tự giải thích???*)

Ví dụ 10.2

```
int calc_sum(int arr[], int size)
{
  int i = 0;
  int sum = 0;

  for (i = 0; i < size; ++i)
      sum += arr[i];

  return sum;
}
```

Ví dụ 10.3

- Viết và chạy chương trình để kiểm nghiệm việc sử dụng mảng làm đối số của hàm:

```c
#include<stdio.h>
#define N 8
//void maximum(int a[], int N){
void test(int a[][N], int m, int n){
 int i,j;
 printf("\nMang a duoc truyen vao ham: ");
 for(i=0; i<m; i++){
    printf("\n");
    for(j=0; j<n; j++) printf("%5d",a[i][j]);
 }
 for(i=0; i<m; i++)
    for(j=0; j<n; j++) a[i][j] +=2;
 printf("\nMang a ham sau khi bi thay doi trong ham: ");
 for(i=0; i<m; i++){
    printf("\n");
    for(j=0; j<n; j++) printf("%5d",a[i][j]);
 }
}
```

```c
void main(){
 int i,j,m,n,a[N][N];
 m = 5;
 n = 1;
 for(i=0; i<m; i++)
    for(j=0; j<n; j++) a[i][j] = i;
 printf("\nMang a duoc tao ra trong ham main(): ");
 for(i=0; i<m; i++){
    printf("\n");
    for(j=0; j<n; j++) printf("%5d",a[i][j]);
 }
// maximum(a,N);
 test(a,m,n);
// maximum(a);
 printf("\nMang a sau khi goi ham: ");
 for(i=0; i<m; i++){
    printf("\n");
    for(j=0; j<n; j++) printf("%5d",a[i][j]);
 }
 printf("\n");
}
```

Bài tập 10.6

- Viết một hàm chấp nhận 2 mảng số nguyên, trả về 1 nếu 2 mảng đó bằng nhau, trả về 0 nếu hai mảng đó có 1 phần tử khác nhau (không bằng nhau).
- Viết chương trình nhập vào 2 mảng số nguyên và cho biết 2 mảng đó có bằng nhau hay không?

Bài tập 10.7

- Viết hai hàm:
  - Sắp xếp một mảng được truyền vào theo thứ tự tăng dần.
  - Sắp xếp một mảng được truyền vào theo thứ tự giảm dần.
- Viết một chương trình cho phép người dùng nhập vào một mảng có N phần tử nhập từ bàn phím. Xây dựng menu để người dùng lựa chọn sắp xếp mảng đó theo thứ tự tăng dần hoặc giảm dần.

Bài tập 10.8

Phòng Lab 502 chương trình Việt Nhật chứa 50 máy tính để bàn. Hàng tuần các máy sẽ được sử dụng để thực tập. Biết một lần được sử dụng – điện năng tiêu thụ là 400 Watt. Viết chương trình quản lý việc sử dụng máy với các menu sau:

- Dùng máy: Khi một người vào phòng máy – chỉ định số hiệu máy muốn sử dụng. Nếu máy còn trống – được cấp máy
- Rời máy: máy về trạng thái rỗi
- In ra trạng thái các máy (đang sử dụng – tắt)
- In ra điện năng tiêu thụ tích lũy trên các máy cho đến thời điểm hiện tại
- In ra tổng điện năng tiêu thụ - và tiền điện thanh toán (750VNĐ/KW)
- In ra những máy được dùng nhiều nhất và ít nhất

Bài tập 10.9

- Viết hàm cho phép nhập vào môt ma trận kích thước nxn.
- Viết hàm cho phép in ma trận ra màn hình từ ma trận được truyền vào.
- Viết 3 hàm cho phép truyền vào 2 ma trận và thực hiện:
  - Cộng 2 ma trận;
  - Trừ 2 ma trận;
  - Nhân 2 ma trận.
- Viết chương trình cho phép nhập vào 2 ma trận kích thước nxn, với n nhập từ bàn phím, sau đó viết menu cho phép người dùng thực hiện việc chọn các phép toán: cộng, trừ, nhân 2 ma trận đó. In ma trận kết quả ra màn hình.

Bài tập 10.10 – Công tắc đèn

- Tưởng tượng dàn đèn trong phòng học thực hành C như một ma trận 5*3.
- Đèn sáng: Giá trị phần tử ma trận là 1 và ngược lại là 0
- Công suất tiêu thụ của các đèn ở vị trí hàng lẻ cột lẻ là 10 W, hàng chẵn cột chẵn là 20 W còn lại là 15 W.

- Viết chương trình menu:
  - 1. Cho người dùng bật tắt đèn theo hàng
  - 2. Cho người dùng bật tắt đèn theo cột
  - 3. Cho người dùng bật tắt đèn theo vị trí
  - 4. Xem công suất tiêu thụ của dàn đèn theo trạng thái hiện thời

Bài tập 10.11: Lăng súc sắc

- Viết chương trình mô phỏng việc lăng một cặp quân súc sắc, tổng số điểm thu được sẽ nằm trong miền giá trị từ 2 đến 12.
- Với mỗi giá trị điểm, chương trình thống kê số lần lăng súc sắc đạt được số điểm đó trong 100 lần tung.
- ***Gợi ý***: *sử dụng hàm rand() để sinh số nguyên ngẫu nhiên của C. Dùng biểu thức rand() % 6 sinh ra các số ngẫu nhiên từ 0 đến 5, và rand() % 6 + 1 sinh ra các số nguyên ngẫu nhiên từ 1 đến 6.*