

Mảng

`huydq@soict.hust.edu.vn`

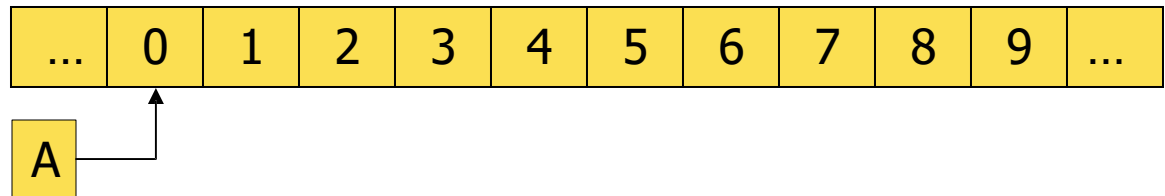
Khái niệm mảng

- Là một khối của các biến có cùng kiểu
- Mảng có thể khai báo cho kiểu dữ liệu bất kì
 - V.d. `int A[10]` tạo mảng 10 số nguyên
- Số phần tử của mảng là một hằng số xác định
- Ví dụ sử dụng mảng để lưu:
 - Danh sách điểm sinh viên
 - Một dãy số nhập bởi người sử dụng
 - Các vector
 - Các ma trận
 - v.v.

Bộ nhớ của mảng

- Mảng là một dãy liên tục của các biến trong bộ nhớ
- Tên mảng là địa chỉ của phần tử đầu tiên trong bộ nhớ
- Ví dụ:

`int A[10];`



- Các phần tử được truy cập theo chỉ số. Phần tử thứ k của mảng A là $A[k-1]$ (chỉ số các phần tử bắt đầu từ 0)

Chương trình dãy số

- Nhập một dãy số nguyên (nhiều nhất 10 phần tử) và in dãy số đó theo thứ tự đảo ngược

```
#include <stdio.h>
int main(void)
{
    int i, n, A[10];
    printf("Nhap so phan tu trong day (n<=10):");
    scanf("%d",&n);

    printf("Nhap cac phan tu trong day:¥n");
    for(i=0; i<n; i++) {
        printf("Phan tu thu %d:", i+1);
        scanf("%d",&A[i]);
    }

    printf("Day so sau khi dao lai:¥n");
    for(i=n-1; i>=0; i--)
        printf("%5d",A[i]);

    return 0;
}
```

Bài tập

- Viết chương trình tính tổng của dãy số theo phương pháp nhập dãy số vào mảng sau đó tính tổng của các số lưu trên mảng.
- Tính min, max của các số trong mảng.

Khởi tạo mảng

- Khi khai báo mảng các phần tử của mảng có thể được khởi tạo giá trị như biến thông thường
- Số phần tử được khởi tạo không được nhiều hơn số phần tử của mảng
 - Nhưng nó có thể ít hơn
 - Khi đó số phần tử còn lại được khởi tạo giá trị 0
- Kích thước của mảng có thể được suy diễn từ số phần tử khởi tạo giá trị

`int A[8] = {2, 4, 6, 8, 10, 12, 14, 16};`

hoặc

`int A[] = {2, 4, 6, 8, 10, 12, 14, 16};`

Ví dụ

```
int    month;
int table[12] = { 30, 40, 45, 95, 130, 220, 210,
                  185, 135, 80, 40, 45 };

printf("Enter a month: ");
scanf("%d", &month);

if (1 <= month && month <= 12)
    printf("Average rainfall for month %d is %d
mm.¥n", month, table[month-1]);
else
    printf("Invalid month¥n");
```

Sắp xếp mảng

- Duyệt các phần tử từ trái sang phải và đổi chỗ nó với phần tử nhỏ hơn nằm bên phải.
 - $A[i]$ là phần tử được duyệt
 - $A[j]$ là phần tử bên phải $A[i]$
 - $A[i]$ được đổi chỗ cho $A[j]$ nếu $A[i] < A[j]$
 - để đổi chỗ giá trị 2 phần tử, ta dùng thêm một bộ nhớ phụ

Chương trình

```
#include <stdio.h>
int main(void)
{
    int A[10], n, i, j, tmp;
    /* nhập dãy số vào mảng */
    for(i=0; i<n-1; i++)
        for(j=i+1; j<n; j++)
            if (a[i]<a[j]) {
                tmp = A[i];
                A[i] = A[j];
                A[j] = tmp;
            }
    printf("Dãy được sắp xếp:\n");
    for(i=0; i<n-1; i++)
        printf("%5d", A[i]);
    return 0;
}
```

Chỉ số mảng

- Một mảng luôn có kích thước cố định
- Không có một phương thức cài đặt sẵn nào để kiểm tra chỉ số trong phạm vi của mảng
- Người lập trình phải tự kiểm soát chỉ số trong phạm vi của mảng

Ví dụ:

```
int i, n=5;  
int A[n];  
int B[5];  
for (i=1; i<=n; i++)  
    B[i] = 0;
```

Kích thước mảng
phải là hằng số

Lỗi chỉ số
vượt phạm vi
của mảng

Truyền mảng cho hàm

- Một mảng có thể được truyền cho hàm và giá trị của nó có thể bị thay đổi
- để truyền một mảng cho hàm ta dùng các tham số như sau:
 - `f (int array[], int n)`

Chương trình dãy số (hàm)

```
#include <stdio.h>

void nhapMang(int array[], int num)
{
    int i;
    for(i=0; i<num; i++) {
        printf("Phan tu thu %d:", i+1);
        scanf("%d", &array[i]);
    }
}

void inNguoc(int array[], int num)
{
    int i;
    for(i=num-1; i>=0; i--)
        printf("%5d", array[i]);
}
```

Số phần tử chính xác của mảng phải được truyền qua một tham số khác

Chương trình dãy số (tiếp)

```
int main(void)
{
    int n, A[10];
    printf("Nhap so phan tu trong day (n<=10):");
    scanf("%d",&n);

    printf("Nhap cac phan tu trong day:¥n");
    nhapMang(A, n);

    printf("Day so sau khi dao lai:¥n");
    inNguoc(A, n);

    return 0;
}
```

Mảng hai chiều

- Một phần tử của mảng có kiểu dữ liệu xác định
- Nhưng một mảng có thể coi là kiểu cho một phần tử của mảng.
 - Vì thế một phần tử của mảng có thể là một mảng khác
- Một mảng hai chiều là một mảng của mảng

Ví dụ:

```
int A[5][3];  
for (i=0; i<=5; i++)  
    for (j=0; j<=3; j++)  
        A[i][j] = 10;
```

Ví dụ bảng cửu chương

```
#include <stdio.h>

void main()
{
    /* bảng cửu chương cho cả số 0 */
    int cuuchuong[10][10];
    int i, j;

    /* tạo giá trị cho bảng cửu chương */
    for (i=0; i<=9; i++)
        for (j=0; j<=9; j++)
            cuuchuong[i][j] = i*j;

    printf("Nhap hai so cua bang cuu chuong\n");
    printf("So 1: "); scanf("%d", &i);
    printf("So 2: "); scanf("%d", &j);

    printf("Giá trị trong bang la %d", cuuchuong[i][j]);
}
```

Ví dụ (tiếp)

```
#define NYEARS 5
#define NMONTHS 12
int table[NYEARS][NMONTHS] = {
    {30,40,75,95,130,220,210,185,135,80,40,45},
    {25,25,80,75,115,270,200,165, 85, 5,10, 0},
    {35,45,90,80,100,205,135,140,170,75,60,95},
    {30,40,70,70, 90,180,180,210,145,35,85,80},
    {30,35,30,90,150,230,305,295, 60,95,80,30}
};
```

table[i][j] cho phép truy cập vào giá trị lượng mưa tháng thứ (j+1) và năm (2000+i)

Ví dụ (tiếp)

```
printf("Enter a year: ");
sscanf("%d", &year);
if (month < 2000 && month > 2004) {
    printf("Year between 2000 and 2004\n");
    return 1;
}
printf("Enter a month: ");
scanf("%d", &month);
if (month < 1 && month > 12) {
    printf("Invalid month\n");
    return 1;
}
printf("Average rainfall is mm.\n", table[year-
2000][month-1]);
```

Bài tập

- Cho khai báo nguyên mẫu các hàm như sau:
`int nhapdayso(int a[]);`
`int timmax(int a[], int n);`
- Hàm `nhapdayso()` cho phép nhập số phần tử và các con số trong dãy số để lưu vào mảng `a`. Hàm trả về số phần tử của dãy số đã nhập. Hàm `timmax()` tìm giá trị lớn nhất trong một dãy số truyền vào.
- Viết định nghĩa các hàm và chương trình chính cho phép nhập vào hai dãy số khác nhau và tìm giá trị lớn nhất của các số trong cả hai dãy số đó.