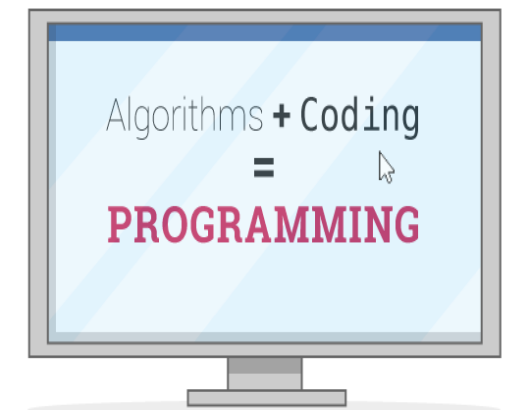
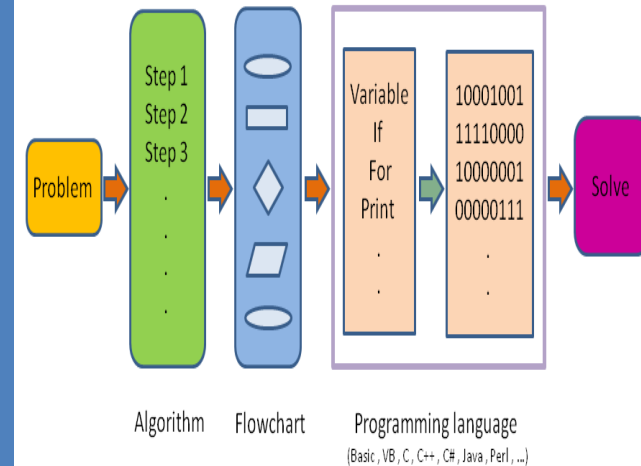


NHẬP MÔN LẬP TRÌNH

Chương 5 MẢNG MỘT CHIỀU

GV: Lê Thị Bích Hằng
Trường Đại học Nha Trang



MẢNG MỘT CHIỀU

Giới thiệu kiểu mảng

Khai báo mảng một chiều

Truy xuất dữ liệu mảng một chiều

Truyền mảng cho hàm

Một số bài toán cơ bản về mảng

Tài liệu tham khảo

■ Đặt vấn đề

- Chương trình cần lưu trữ **3** số nguyên?
=> Khai báo **3** biến **int a1, a2, a3;**
- Chương trình cần lưu trữ **100** số nguyên?
=> Khai báo **100** biến kiểu số nguyên!
- Người dùng muốn nhập **n** số nguyên?
=> Không thực hiện được!

■ Giải pháp

- Kiểu dữ liệu mới cho phép **lưu trữ một dãy** các số nguyên và **dễ dàng truy xuất**.

MẢNG MỘT CHIỀU

Giới thiệu kiểu mảng

Khai báo mảng
một chiều

Truy xuất dữ liệu
mảng một chiều
Truyền mảng cho
hàm

Một số bài toán cơ
bản về mảng
Tài liệu tham khảo

■ Khái niệm kiểu mảng

- Là một **kiểu dữ liệu có cấu trúc** do người lập trình định nghĩa.
- Biểu diễn một **dãy các biến có cùng kiểu**.
 - Ví dụ: dãy các số nguyên, dãy các ký tự...
- Ngôn ngữ C quy định kích thước của mảng là **cố định** và phải **được định nghĩa tại thời điểm khai báo**.
- NNLT C luôn chỉ định **một khối nhớ liên tục** cho một biến kiểu mảng.
- Có hai loại mảng:
 - Mảng một chiều
 - Mảng nhiều chiều

MẢNG MỘT CHIỀU

Giới thiệu kiểu mảng

Khai báo mảng một chiều

Truy xuất dữ liệu mảng một chiều

Truyền mảng cho hàm

Một số bài toán cơ bản về mảng

Tài liệu tham khảo

■ Khai báo mảng với số phần tử xác định

- Cú pháp

Kiểu_dữ_liệu **Tên_biến_mảng**[**Số_phần_tử**];

- **Kiểu_dữ_liệu**: Là kiểu dữ liệu của các phần tử trong mảng (Kiểu cơ sở: char, int, float,...)
- **Tên_biến_mảng**: Là tên của mảng, được đặt đúng theo quy tắc đặt tên định danh.
- **Số_phần_tử**: Là số lượng phần tử tối đa có trong mảng.
- Bộ nhớ sử dụng = **Số_phần_tử** * sizeof (**Kiểu_dữ_liệu**)
- Ví dụ:

○ **int** **a**[10]; /* Khai báo mảng số nguyên a có 10 phần tử */

Vị trí của
phần tử

0 1 2 3 4 5 6 7 8 9



Tên phần
tử

a[0] a[1] a[2] a[3] a[4] a[5] a[6] a[7] a[8] a[9]

MẢNG MỘT CHIỀU

Giới thiệu kiểu
mảng

Khai báo mảng
một chiều

Truy xuất dữ liệu
mảng một chiều

Truyền mảng cho
hàm

Một số bài toán cơ
bản về mảng

Tài liệu tham khảo



■ Khởi tạo giá trị của mảng trong lúc khai báo

- Khởi tạo giá trị cho mọi phần tử của mảng

```
int a[4] = {10, 69, 39, -15};
```

0 1 2 3

a 10 69 39 -15

- Khởi tạo giá trị cho một số phần tử đầu mảng

```
int a[4] = {10, 69};
```

0 1 2 3

a 10 69 0 0

MẢNG MỘT CHIỀU

Giới thiệu kiểu
mảng

Khai báo mảng
một chiều

Truy xuất dữ liệu
mảng một chiều
Truyền mảng cho
hàm

Một số bài toán cơ
bản về mảng

Tài liệu tham khảo

■ Khởi tạo giá trị của mảng trong lúc khai báo

- Khởi tạo giá trị **0** cho mọi phần tử của mảng

```
int a[4] = {0};
```

	0	1	2	3
a	0	0	0	0

MẢNG MỘT CHIỀU

Giới thiệu kiểu
mảng

Khai báo mảng
một chiều

Truy xuất dữ liệu
mảng một chiều

Truyền mảng cho
hàm

Một số bài toán cơ
bản về mảng

Tài liệu tham khảo

■ Khai báo mảng với số phần tử không xác định

- Cú pháp

Kiểu_dữ_liệu **Tên_biến_mảng**[]

- Kiểu khai báo này thường được áp dụng trong các trường hợp:
 - Vừa khai báo vừa gán giá trị
 - Khai báo mảng là tham số hình thức của hàm.

- Vừa khai báo vừa gán giá trị:

Kiểu_dữ_liệu **Tên_biến_mảng**[] = { Giá_trị_1, Giá_trị_2,..., Giá_trị_n };

- Số phần tử của mảng là số giá trị trong cặp dấu { }.
- Số phần tử = $\text{sizeof}(\text{Tên_biến_mảng}) / \text{sizeof}(\text{Kiểu_dữ_liệu})$

```
int a[] = {10, 69, 39, -15};
```

0

1

2

3

a

10

69

39

-15

MẢNG MỘT CHIỀU

Giới thiệu kiểu mảng

Khai báo mảng một chiều

Truy xuất dữ liệu mảng một chiều

Truyền mảng cho hàm

Một số bài toán cơ bản về mảng

Tài liệu tham khảo

■ Khai báo mảng với số phần tử không xác định

- Cú pháp

Kiểu_dữ_liệu **Tên_biến_mảng**[]

- Kiểu khai báo này thường được áp dụng trong các trường hợp:
 - Vừa khai báo vừa gán giá trị
 - Khai báo mảng là tham số hình thức của hàm.

- Vừa khai báo vừa gán giá trị:

Kiểu_dữ_liệu **Tên_biến_mảng**[] = { **Giá_trị_1**, **Giá_trị_2**, ..., **Giá_trị_n** };

- Số phần tử của mảng là số giá trị trong cặp dấu {}.

```
int a[] = {10, 69, 39, -15};
```

	0	1	2	3
a	10	69	39	-15

MẢNG MỘT CHIỀU

Giới thiệu kiểu
mảng

Khai báo mảng
một chiều

Truy xuất dữ liệu
mảng một chiều
Truyền mảng cho
hàm

Một số bài toán cơ
bản về mảng

Tài liệu tham khảo



■ Khai báo mảng với số phần tử không xác định

- Khai báo mảng là tham số hình thức của hàm:
 - Vì tham số kiểu mảng truyền cho hàm chính là **địa chỉ của phần tử đầu tiên của mảng**, nên chúng ta có thể bỏ đi số lượng tối đa của phần tử trong khai báo hàm (Trong trường hợp này ta không cần chỉ định số phần tử của mảng là bao nhiêu), và có thể đưa số lượng phần tử thực sự qua một biến khác.
 - Ví dụ:

```
void NhapMang(int a[], int &n);  
void SapXepTang(int a[], int n);
```

MẢNG MỘT CHIỀU

Giới thiệu kiểu
mảng

Khai báo mảng
một chiều

Truy xuất dữ liệu
mảng một chiều

Truyền mảng cho
hàm

Một số bài toán cơ
bản về mảng

Tài liệu tham khảo



■ Lưu ý khi khai báo mảng

- Phải xác định cụ thể số phần tử ngay lúc khai báo
 - `int a[];`
`=> int a[100];`
- Không khởi tạo cách biệt với khai báo
 - `int a[4];`
`a = {30, 15, 8, -3};`
`=> int a[4] = {30, 15, 8, -3};`
- Nên sử dụng chỉ thị tiền xử lý `#define` để định nghĩa số phần tử mảng
 - `#define MAX 10`
`int a[MAX];` `// int a[10]`

MẢNG MỘT CHIỀU

Giới thiệu kiểu
mảng

Khai báo mảng
một chiều

Truy xuất dữ liệu
mảng một chiều

Truyền mảng cho
hàm

Một số bài toán cơ
bản về mảng

Tài liệu tham khảo



Truy xuất từng phần tử của mảng

- Cú pháp:

Tên_biến_mảng[Chỉ_số]

- Mỗi phần tử của mảng được truy xuất thông qua *Tên_biến_mảng* theo sau là *Chỉ_số* nằm trong cặp dấu ngoặc vuông **[]**.

- Ví dụ:

	0	1	2	3
<code>int a[4];</code>	<div></div>	<div></div>	<div></div>	<div></div>

- Các truy xuất hợp lệ: `a[0]`, `a[1]`, `a[2]`, `a[3]`
 - Các truy xuất không hợp lệ: `a[-1]`, `a[4]`, `a[5]`, ...
- => Cho kết quả không như mong muốn

MẢNG MỘT CHIỀU

Giới thiệu kiểu
mảng

Khai báo mảng
một chiều

Truy xuất dữ liệu
mảng một chiều

Truyền mảng cho
hàm

Một số bài toán cơ
bản về mảng

Tài liệu tham khảo

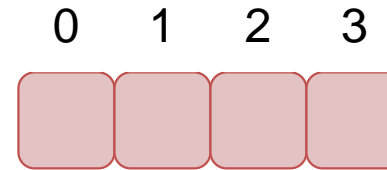
■ Gán dữ liệu kiểu mảng

- Cú pháp:

`Tên_biến_mảng[Chỉ_số] = Giá_trị;`

- Ví dụ.

`int a[4], b[4];`



- Các phép gán hợp lệ:

- `a[0] = 7;`

- `b[2] = a[1];`

- Phép gán không hợp lệ:

- `b = a;`

⇒ Sử dụng vòng lặp lần lượt gán từng phần tử trong mảng *a* vào mảng *b*:

```
for(int i = 0; i < 4; i++)
```

```
    b[i] = a[i];
```

MẢNG MỘT CHIỀU

Giới thiệu kiểu
mảng

Khai báo mảng
một chiều

Truy xuất dữ liệu
mảng một chiều

Truyền mảng cho
hàm

Một số bài toán cơ
bản về mảng

Tài liệu tham khảo



■ Truyền mảng cho hàm

- Tham số kiểu mảng trong khai báo hàm **giống như khai báo biến** mảng

```
void SapXepTang(int a[100]);
```

- Tham số kiểu mảng truyền cho hàm chính là **địa chỉ của phần tử đầu tiên của mảng**

- Có thể **bỏ số lượng phần tử** hoặc **sử dụng con trỏ**.
- Mảng **có thể thay đổi nội dung** sau khi thực hiện hàm.

```
void SapXepTang(int a[]);
```

```
void SapXepTang(int *a);
```

- Số lượng phần tử thực sự truyền qua biến khác

```
void SapXepTang(int a[100], int n);
```

```
void SapXepTang(int a[], int n);
```

```
void SapXepTang(int *a, int n);
```

MẢNG MỘT CHIỀU

Giới thiệu kiểu
mảng

Khai báo mảng
một chiều

Truy xuất dữ liệu
mảng một chiều

Truyền mảng cho
hàm

Một số bài toán cơ
bản về mảng

Tài liệu tham khảo



■ Viết hàm thực hiện từng yêu cầu sau:

- Nhập mảng
- Xuất mảng
- Tìm kiếm một phần tử trong mảng
- Kiểm tra tính chất của mảng
- Tách mảng
- Gộp mảng
- Tìm giá trị nhỏ nhất/lớn nhất của mảng
- Sắp xếp mảng giảm dần/tăng dần
- Thêm một phần tử vào mảng
- Xóa một phần tử ra khỏi mảng
- Sửa một phần tử vào mảng

MẢNG MỘT CHIỀU

Giới thiệu kiểu mảng

Khai báo mảng một chiều

Truy xuất dữ liệu mảng một chiều

Truyền mảng cho hàm

Một số bài toán cơ bản về mảng

Tài liệu tham khảo



- [1] Quách Tuấn Ngọc, *Ngôn ngữ lập trình C*, NXB Giáo dục, 2003
- [2] Trần Đan Thư, *Giáo trình lập trình C tập 1&2*, NXB ĐHQG TP. Hồ Chí Minh
- [3] Trần Đan Thư, *Nhập môn lập trình*, NXB KH&KT, 2019
- [4] Phạm Văn Ất, *Kỹ thuật lập trình C căn bản và nâng cao*, NXB Hồng Đức 2009
- [5] B. W. Kernighan and D. M. Ritchie, *C Programming Language, 2nd edition*, Prentice Hall, 1998
- [6] Lê Thị Bích Hằng, *Tài liệu tham khảo Nhập môn lập trình*, Lưu hành nội bộ
- [7] Đặng Bình Phương, *Bài giảng Nhập môn lập trình*