**Mảng trong C**

Mảng là gì, mảng nhiều chiều là gì, cách phân bố trong bộ nhớ của mảng, làm thế nào để khai báo và làm việc với mảng, khi nào sử dụng mảng

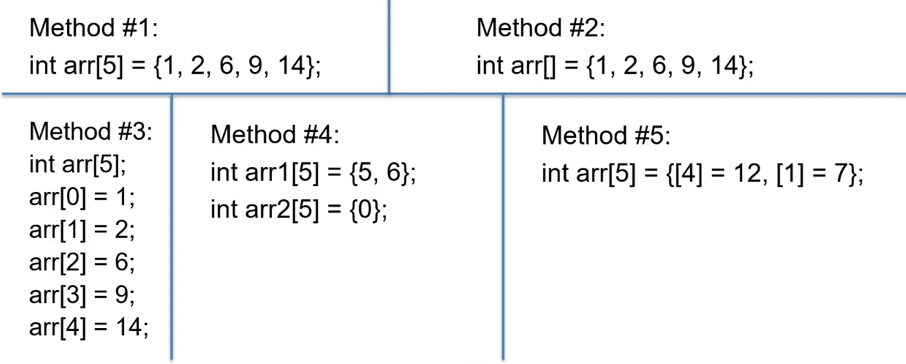
Mảng là 1 cấu trúc dữ liệu lưu trữ 1 tập hợp gồm các phần tử có cùng kiểu có độ dài cố định, có cùng kiểu dữ liệu

Ví dụ: int arr[5];

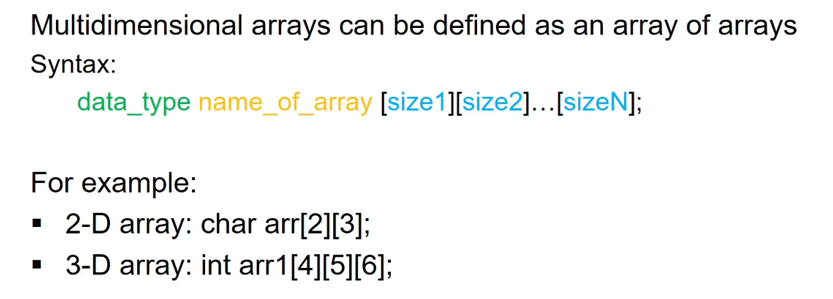
Kích thước của mảng arr là 5\*kích thước interger = 5\*4 bytes = 20 bytes

Kích thước của mảng phải được định nghĩa là 1 số nguyên dương

Cách định nghĩa mảng



Mảng nhiều chiếu được định nghĩa là mảng của các mảng, cú pháp tương tự như mảng 1 chiều



Số lượng chiều của mảng được giới hạn bởi kích thước bộ nhớ và kiểu dữ liệu interger, ví dụ mảng 2 chiều

Char arr[2][3]; đây là mảng gồm 2 hàng và 3 cột

Mảng 3 chiều int arr1[4][5][6]; có thể coi là 4 mảng 2 chiều 5 hàng và 6 cột

Cách khởi tạo tương tự mảng 1 chiều

**Ưu điểm của mảng**

* Có thể dễ dàng truy cập các phần tử trong bộ nhớ của mảng
* Việc tìm kiếm phần tử trong mảng dễ dàng
* Mảng nhiều chiều hỗ trợ các bài toán xử lý bài toán như ma trận
* Việc phân bố bộ nhớ liên tiếp trong mảng giúp chúng ta có thể sử dụng con trỏ để truy cập tất cả các phần tử trong mảng

**Nhược điểm**

* Do kích thước của mảng là cố định cho nên trong quá trình sử dụng ta ko thể tăng hay giảm kích thước của mảng
* Ko thể lưu trữ được các phần tử khác kiểu dữ liệu
* Việc lưu trữ các phần tử khác kiểu dữ liệu có thể cân nhắc sử dụng struct tuỳ vào bài toán
* Việc thêm hoặc xoá phần tử trong mảng không dễ dàng

**Làm thế nào để làm việc với mảng**

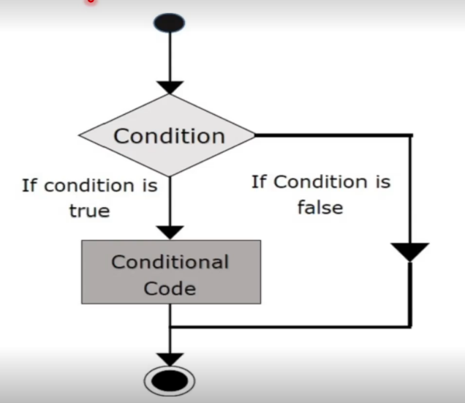
Các bài toán làm việc với mảng là truy cập các phần tử của mảng, tìm kiếm, thêm bớt phần tử trong mảng, sắp xếp phần tử trong mảng, sử dụng mảng character và hay là chuỗi ký tự.

Khi nào sử dụng mảng ứng với các bài toán yêu cầu như lưu trữ 1 số liệu nhẩt định các dữ liệu có cùng kiểu và ko yêu cầu phải thêm bớt dữ liệu trong quá trình sử dụng.

Tuỳ thuộc vào bài toán có thể lựa chọn sử dụng các cấu trúc dữ liệu khác như liên kết chuỗi, stack, heap, Q, tree, (đây là các cấu trúc dữ liệu)

**Cấu trúc điều kiện rẽ nhánh trong C**

* Giới thiệu về cấu trúc điều kiện rẽ nhánh
* Làm thế nào để xây dựng biểu thức điều kiện
* Các câu lệnh if, else, các câu lệnh switch case



Các bước thực hiện bài toán: chấp nhận số truyền vào từ người dùng, chia số đó với 2 và lấy phần dư nếu phần dư bằng 0 thì số đó là số chẵn, khác 0 là số lẻ

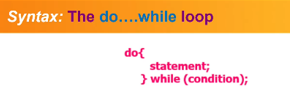
**Vòng lặp trong C**

* Cách vào và thoát vòng lặp
* Làm cách nào sử dụng vòng lặp

Vòng lặp trong C là 1 phần mã code chương trình được thực hiện nhiều lần cho đến khi 1 điều kiện cụ thể được thoả mãn

Các loại vòng lặp trong C là for, while, do … while

Vòng lặp while được thực hiện đến khi điều kiện đó vẫn còn dc thoả mãn.



Khác vòng lặp while ở chỗ vòng lặp do while sẽ thực hiện khối lệnh ít nhất 1 lần

**Từ khoá trong vòng lặp**

* **Goto**
* **Continue**
* **Exit()**
* **Break**
* **Return**

**Goto label**

Dc sử dụng để chuyển đến 1 lệnh bất kỳ có nhãn, ko nên sử dụng goto trong chương trình bởi vì goto phá vỡ chương trình có cấu trúc và cũng giảm khả năng maintain chương trình

**Continue**

Giúp chúng ta bỏ qua các câu lệnh phía sau và thực hiện vòng lặp mới

Tuỳ thuộc vào chương trình chúng ta có thể sử dụng continue khi mà 1 một điều kiện xảy ra mà chúng ta không muốn thực hiện những cái khối lệnh còn lại trong vòng lặp

Lệnh **exit()** giúp chúng ta thoát khỏi toàn bộ chương trình

Lệnh **break** giúp chúng ta thoát khỏi vòng lặp chương trình và tiếp tục

Lệnh **returrn** giúp chúng ta thoát khỏi chương trình con và quay trở lại chương trình đã gọi chương trình con đó ví dụ hàm main gọi đến 1 hàm con/ function, lệnh returrn sẽ giúp chúng ta thoát khỏi function và trở lại hàm main

**Câu hỏi:** có 1 con trỏ p trỏ vào phần tử đầu của mảng a thì vấn đề con trỏ p ấy và ký tự a đấy có thể thao tác 1 vài thứ khá là giống nhau nhưng mà có gì p làm dc mà a ko làm dc và điểm khác nhau giữa 2 thg đó? Chịu, hỏi chung chung quá