

Các khái niệm cơ bản

1. Tập ký tự
2. Tên
3. Tên chuẩn
4. Từ khóa
5. Từ vựng quy ước
6. Kiểu dữ liệu
7. Hằng
8. Biến
9. Lỗi cú pháp
10. Lỗi ý nghĩa

Tập ký tự

- Mọi ngôn ngữ lập trình đều được xây dựng từ một bộ ký tự nào đó.
- Bộ ký tự của C:
 - 26 chữ cái hoa (tiếng Anh): A, B, C,..., Z
 - 26 chữ cái thường (tiếng Anh): a, b, c,...,z
 - 10 chữ số: 0,1,2,...,9
 - Các ký hiệu toán học: +, -, *, /,()
 - Các ký hiệu đặc biệt khác: . , ; : { } [] ? ! | % \$ #...
 - Dấu cách
- *Chú ý: Khi viết chương trình, không sử dụng các ký hiệu ngoài các ký tự nêu trên, ví dụ: ϵ , ξ ...*

Tên

- Tên là một dãy các ký tự: chữ, số và dấu gạch nối (ký tự đầu tiên của tên là chữ hoặc dấu gạch nối)
- Các loại thường gặp: tên hằng, tên biến, tên hàm, tên mảng ...
- *Chú ý:*
 - Độ dài của tên không được quá 32 ký tự
 - Tên không được trùng với từ khóa
 - Chữ hoa và chữ thường khác nhau
- Ví dụ đúng: a_1 a1 delta1 ALPHA _x
- Ví dụ sai: 3xy
 int
 be ta
 X-1

Tên chuẩn

- là các tên đã được định nghĩa hay được gán cho các ý nghĩa chuyên biệt.
- Các tên này có thể được dùng lại cho một ý nghĩa khác tùy theo người viết mã.
- Thường các tên được phép định nghĩa lại thuộc một trong hai loại chính là:
 - Các hàm hay thủ tục chuẩn.
 - Các biến toàn cục (*global*)
- Ví dụ
 - Trong C thì `sin` là tên của một hàm tính giá trị `sin` (trong thư viện *math.h*) nhưng người lập trình hoàn toàn có thể định nghĩa lại hàm này để cho nó có chức năng khác.

Từ khóa

- Là những từ có ý nghĩa hoàn toàn xác định
- Thường được dùng làm:
 - tên kiểu dữ liệu: int, float, void...
 - Câu lệnh: if, do, while, for...
- Chú ý:
 - Không dùng từ khóa để đặt tên biến, hằng....
 - Từ khóa phải được viết bằng chữ thường.

Từ vựng quy ước

- Từ vựng qui ước là những dãy các kí tự nối nhau và được ngôn ngữ cho sử dụng như là tên, giá trị hay một luật nào đó.
- Người lập trình nên tránh sử dụng các từ qui ước này vào việc đặt tên.
- Thí dụ trong C thì việc viết:
 - `#define MYVALUE 10`; thì dãy kí tự `"#define"` là một từ vựng qui ước.
 - `#include`

Kiểu dữ liệu

- Kiểu ký tự (char):** biểu diễn một số nguyên hoặc một ký tự (biểu diễn được một ký tự trong bảng mã ASCII)

Kiểu	Phạm vi biểu diễn	Số ký tự	Kích thước
char (signed char)	-128 -> 127	256	1 byte
unsigned char	0 -> 255	256	1 byte

- Kiểu nguyên:** Biểu diễn một số nguyên.

Kiểu	Phạm vi biểu diễn	Kích thước
int	-32768 -> 32767	2 byte
unsigned int	0 -> 65535	2 byte
Long	-2.147.483.648 -> 2.147.483.647	4 byte
unsigned long	0 -> 4.294.967.295	4 byte

Kiểu dữ liệu (tiếp)

3. Kiểu thực: biểu diễn một số thực

Kiểu	Phạm vi	Kích thước
float	$3.4E-38 \rightarrow 3.4E+38$	4 byte
double	$1.7E-308 \rightarrow 1.7E+308$	8 byte
long double	$3.4E-4932 \rightarrow 1.1E+4932$	10 byte

Hằng

- Hằng là đại lượng mà giá trị của nó không thay đổi trong quá trình tính toán.
- Các loại hằng trong C **Thử in ra màn hình kí tự A, và xâu "Ha Noi"**
 - Hằng thực, hằng nguyên: `1.5`, `100`, `1`, `1000000`
 - Hằng ký tự là một ký tự riêng biệt được viết trong hai dấu nháy đơn: `'a'` ; `'9'`
 - Một vài hằng ký tự phải viết: `'\'` ; `'\b'` ; `'\0'` (null – kết thúc xâu) khác `'0'`; `'A'` = `'\65'`
 - Hằng xâu ký tự: `"Ha noi"` ; `"a"` khác `'a'`
- Tên hằng: `#define MAX 1000`

Biến

- Là đại lượng có thể thay đổi giá trị nhiều lần trong một chương trình
- Khai báo tên biến:

Kiểu dữ liệu tên biến;

- Ví dụ: `int a;`
`float x,y;`
`char beta;`
- Vị trí khai báo biến: ngay sau dấu { đầu tiên của mỗi hàm
- Khởi đầu giá trị cho biến ngay khi khai báo:
`int a=23;`
- Lấy địa chỉ của biến: & tên biến

Lỗi cú pháp

- Vi phạm khi đặt hay gọi tên biến và hàm
- Lỗi chính tả: người viết mã có thể viết hay gọi sai tên hàm, tên biến.
- Vượt quá khả năng tính toán:
 - Lỗi thời gian (*timing error*)
 - Lỗi chia cho 0: *division by 0*.

Lỗi cú pháp (tiếp)

- Gán sai dữ liệu: dùng một dữ liệu có kiểu khác với kiểu của biến để gán cho biến đó một cách không chủ ý.
 - Các lỗi biên giá trị vượt quá phạm vi biểu diễn
 - Ví dụ: Gán một giá trị lớn hơn 255 cho một biến có kiểu là short trong ngôn ngữ C
 - Thí dụ: đoạn mã C/C++ sau đây sẽ gây ra lỗi biên -- Chia cho 0
- ```
for (m=10; m >= 0, m--) { x= 8+ 2/m; }
```

# Lỗi ý nghĩa

- Lỗi về quản lý bộ nhớ: số lượng bộ nhớ xin đăng ký quá nhiều gây ra treo máy - lỗi rò rỉ bộ nhớ (*memory leaking*).
- Sai sót trong thuật toán: một thuật toán không chặt chẽ, xử lý không hết mọi trường hợp có thể xảy ra, không dự đoán được sự thay đổi trong lúc thực thi.