# Chuỗi Ký Tự

\* Khái niệm :

+ Chuỗi kí tự là một dãy các ký tự, bao gồm cả chữ cái, chữ số đấu cách và các ký tự đặc biệt. Trong C, chuỗi ký tự được xem như một mảng một chiều kí tự kiểu chả.

+ Ký tự kết thúc : Ký tự NULL (\0) luôn được thêm vào cuối chuỗi để đánh dấu điểm kết thúc.

+ Lưu trữ trong bộ nhớ : Mỗi ký tự trong chuỗi được lưu trữ trong một ô nhớ liên tiếp. Ký tự NULL chiếm thêm một ô nhớ cuối cùng.

+ Ví dụ :

# Mảng kiểu ký tự :

**char** word [ ] = { ‘ H ‘, ‘ e ‘, ‘ l ‘, ‘ l ‘, ‘ o ‘ };

# Chuỗi ký tự :

**char** word [ ] = { ‘ H ‘, ‘ e ‘, ‘ l ‘, ‘ l ‘, ‘ o ‘, ‘ \0’ };

**char** word [ ] = “ Hello “;

=> chuỗi ký tự bao gồm một ký tự đặc biệt nằm cuối chuỗi

+ Báo hiệu kết thúc chuỗi

+ Được hiểu là ký tự NULL, có thể viết là ‘\0’

=> Chuỗi ví dụ “Hello” đã bao hàm ký tự kết thúc tức là char[ 6 ]

=> Kiểu kiệt kê ví dụ { ‘ H ‘, ‘ e ‘, ‘ l ‘, ‘ l ‘, ‘ o ‘, ‘ \0’ }, cũng bao hàm ký tự kết thúc tức là char[ 6 ]

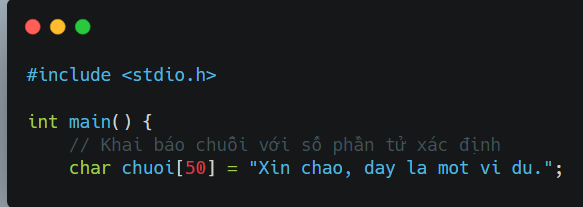
\* Khai Báo :

+ Cú pháp : char **<tên biến> [<Số Ký tự>] = <Chuỗi>**;

+ Cần xác định kích cỡ của chuỗi

# Khai báo <Số ký tự>

# Khởi tạo một giá trị <Chuỗi< cho biến



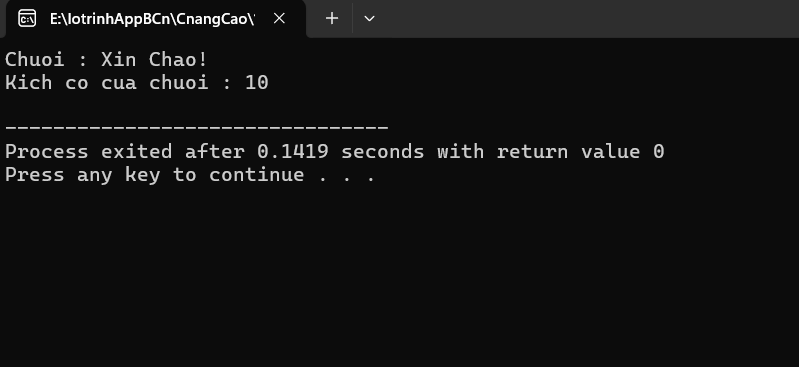
+ <Số ký tự> cần đủ lớn để chứa <Chuỗi>

+ Nếu không khai báo <Số ký tự>

# Cần khởi tạo chuỗi đầu tiên cho biến

# Kích cỡ của biến chính là chiều dài của chuỗi ( gồm ký tự NULL )





+ Chuỗi khởi tạo hợp lệ

# Nằm giữa dấu hai nhấy “”

# Liệt kê { } các ký tự bao gồm ký tự NULL ở cuối cùng

\* Khởi tạo giá trị :

Khởi tạo giá trị char hello[ ] = “Hello!”;

=> Tương đương với

**char** hello[ ] = { ‘ H ‘, ‘ e ‘, ‘ l ‘, ‘ l ‘, ‘ o ‘, ‘ \0’ };

**char** hello[ 7 ] = { ‘ H ‘, ‘ e ‘, ‘ l ‘, ‘ l ‘, ‘ o ‘, ‘ !’, ‘ \0’ };

**char** hello[ 7 ] = “ Hello! “;

Khởi tạo giá trị char empty[ ] = “ “;

=> Tương đương với

char empty[] = " ";

char empty[] = { ' ', '\0' };

empty[2] = " ";

Khởi tạo giá trị :

char hi [ 7 ] = “Hi!”;

char hi [ 7 ] = { ‘H’, ‘i’, ‘!’, ‘\0’ };

{F3CE9235-67ED-4957-B10F-9FD3C8A3560E}

\* Phép gán :

Các phép gán trực tiếp cho chuỗi bị hạn chế => do trình biên dịch xử lý chuỗi như **mảng**

**Khởi tạo giá trị :** char hello[ ] = “Hello!”;

=> Các phép gán mà trình biên dịch báo lỗi

+ hello = “Chao moi nguoi”; //Error

=> do “Chao moi nguoi” là const char[14] không thể gán cho char[7]

\* Làm thế nào để thay đổi giá trị của chuỗi?

> Dùng các hàm xử lí chuỗi như scanf

> Thay đổi từng phần tử

> Các hàm trong thư viện **<string.h>**

> Tự viết hàm xử lí chuỗi kí tự

+ Nhập chuỗi từ bàn phím :

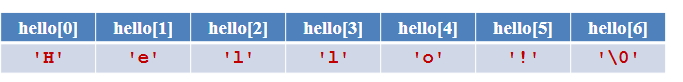
Dùng scanf trong thư viện **<stdio.h>**

**scanf(“%s”, &name);**

**=> scanf tự động thêm ký tự NULL vào sau chuỗi nhận được từ bàn phím**

+ Phần tử của chuỗi :

Khởi tạo : char hello[ ] = “Hello!”;

=> hello[0] = ‘H’... hello[6] = ‘\0 

**+** Từ đó có thể gán chuỗi hello với kí tự bất kì, ví dụ hello[2] = ‘i’

+ Thư viện **<string.h> :**

# strlen

+ Trả về độ dài của chuỗi kí tự

+ Ví dụ :

Strlen ( hello ) = 5.

# char \*strcpy ( char \*destination, const char \*source )

+ Sao chép chuỗi b sang chuỗi a bao gồm kí tự NULL

+ Chuỗi a phải đủ chứa chuỗi b ( bao gồm kí tự NULL )

+ char \*a : Hàm thay đổi giá trị của a

+ char \*b : Hàm không thay đổi giá trị của b

+ Ví dụ : strcpy ( hello, “Hi” ) sao chép “Hi” vào chuỗi hello.

# char \*strcat ( char \*destination, const char \*source)

+ Nối chuỗi b vào cuối chuỗi a

+ Chuỗi a phải đảm bảo đủ chứa chuỗi b ( bao gồm NULL )

+ Ví dụ :

char a[10] = “hello;

Char b[10] = “Hi”

Strcat(a, b) => a = “HelloHi”

# int strcmp và strcoll ( const char \*destination, const char \*source)

+ So sánh chuỗi a và chuỗi b, so sánh từng ký tự từ đầu đến khi phát hiện điểm khác nhau hoặc đến cuối chuỗi ( ‘\0’ )

+ Giá trị trả về > 0 : **a > b**

+ Giá trị trả về = 0 : **a = b**

+ Giá trị trở về < 0 : **a < b**

# strchr ( str, ch )

+ Tìm kiếm ký tự ch đầu tiên trong chuỗi str

+ Trả về con trỏ tới vị trí ký tự nếu tìm thấy, hoặc NULL nếu không tìm thấy