



Dây đàn

Phiên 10

Chỉ dành cho Trung tâm Aptech sử dụng



Mục tiêu

- Giải thích các biến và hằng số chuỗi
- Giải thích con trỏ tới chuỗi
- Thực hiện các hoạt động nhập/xuất chuỗi
- Giải thích các hàm chuỗi khác nhau
- Giải thích cách mảng có thể được truyền như là đối số cho các hàm
- Mô tả cách chuỗi có thể được sử dụng làm đối số hàm



Biến chuỗi

- Chuỗi là mảng ký tự kết thúc bằng ký tự NULL ('\0').
- Biến chuỗi có thể được gán hằng chuỗi.
- Hằng chuỗi là một chuỗi ký tự được bao quanh bởi dấu ngoặc kép.
- Ký tự null '\0' được tự động thêm vào biểu diễn bên trong của một chuỗi.
- Khi khai báo một biến chuỗi, hãy chừa thêm một khoảng trống cho ký tự kết thúc null.



Khai báo biến chuỗi

- Một khai báo biến chuỗi điển hình là:

ký tự `str[10];`

- **đường** là một biến mảng ký tự có thể chứa tối đa 10 ký tự bao gồm cả ký tự kết thúc null.



Hoạt động I/O chuỗi-1

- Các hoạt động I/O chuỗi được thực hiện bằng cách sử dụng các hàm từ thư viện I/O chuẩn được gọi là **stdio.h**
- Hàm `gets()` là phương pháp đơn giản nhất để chấp nhận một chuỗi thông qua đầu vào chuẩn
- Các ký tự đầu vào được chấp nhận cho đến khi phím Enter được nhấn
- Hàm `gets()` thay thế ký tự xuống dòng kết thúc '\n' bằng ký tự '\0'
- Cú pháp:

được(chuỗi);



Hoạt động I/O chuỗi-2

- Hàm puts() được sử dụng để hiển thị một chuỗi trên thiết bị đầu ra chuẩn.
- Cú pháp:

đặt(chuỗi);

- Các hàm scanf() và printf() được sử dụng để chấp nhận và hiển thị các kiểu dữ liệu hỗn hợp bằng một câu lệnh duy nhất.
- Cú pháp để chấp nhận một chuỗi như sau:

scanf("%s", chuỗi);

- Cú pháp để hiển thị một chuỗi như sau:

printf("%s", chuỗi);



Các hàm chuỗi

Các hàm xử lý chuỗi được tìm thấy trong tệp tiêu đề chuẩn **chuỗi.h**. Một số thao tác được thực hiện bởi các hàm này là:

- Nối chuỗi
- So sánh các chuỗi
- Xác định vị trí một ký tự trong chuỗi
- Sao chép một chuỗi sang chuỗi khác
- Tính độ dài của một chuỗi



Hàm strcat()

- Nối hai giá trị chuỗi thành một.
- Cú pháp:

chuỗi(chuỗi1, chuỗi2);

- Nối str2 vào cuối str1
- Hàm trả về str1



Hàm strcmp()

- So sánh hai chuỗi và trả về một giá trị số nguyên dựa trên kết quả so sánh.

- Cú pháp:

strcmp(chuỗi1, chuỗi2);

- Hàm trả về giá trị:
 - Nhỏ hơn 0 nếu $str1 < str2$
 - Bằng không nếu $str1$ giống $str2$
 - Lớn hơn 0 nếu $str1 > str2$



Hàm strchr()

- Xác định sự xuất hiện của một ký tự trong một chuỗi.

- Cú pháp:

strchr(chuỗi, chuỗi);

- Hàm trả về giá trị:

- Con trỏ đến lần xuất hiện đầu tiên của ký tự (được trỏ bởi **ký tự**) trong chuỗi, **đường**
- NULL nếu nó không có mặt



Hàm strcpy()

- Sao chép giá trị trong một chuỗi vào chuỗi khác

- Cú pháp:

strcpy(chuỗi1, chuỗi2);

- Giá trị của str2 được sao chép vào str1

- Hàm trả về **chuỗi 1**



Hàm strlen()

- Xác định độ dài của một chuỗi

- Cú pháp:

strlen(chuỗi);

- Hàm trả về một giá trị số nguyên cho độ dài của **đường**



Truyền Mảng tới Chức năng-1

- Khi một mảng được truyền như một đối số cho một hàm, chỉ có địa chỉ của mảng được truyền
- Tên mảng không có chỉ số dưới tham chiếu đến địa chỉ của mảng

hàm main() không có giá trị

```
{  
  int ary[10];  
  .  
  .  
  fn_ary(ary);  
  .  
  .  
}
```



Truyền Mảng tới Chức năng-2

bao gồm <stdio.h>

hàm main() không có giá trị

```
{  
int số[5], ctr, tổng=0;  
int sum_arr(int num_arr[]); /* Khai báo hàm */  
  
clrscr();  
  
for(ctr=0;ctr<5;ctr++) /* Chấp nhận các số vào mảng */ {  
  
    printf("\nNhập số %d: ", ctr+1);  
    scanf("%d", &num[ctr]);  
}
```



Truyền Mảng tới Chức năng-3

```
sum=sum_arr(num); /* Gọi hàm */

printf("\nTổng của mảng là %d", sum);

lấy();
}

int sum_arr(int num_arr[]) /* Định nghĩa hàm */ {

    int i, tổng;

    for(i=0,total=0;i<5;i++) /* Tính tổng */
        tổng cộng+=số_mảng[i];

    trả về tổng số; /* Trả về tổng cho hàm main() */
}
```



Truyền Mảng tới Chức năng-4

Mẫu đầu ra của chương trình

Nhập số 1: 5

Nhập số 2: 10

Nhập số 3: 13

Nhập số 4: 26

Nhập số 5: 21

Tổng của mảng là 75

Ví dụ về việc truyền chuỗi đến Chức năng

```
# bao gồm <stdio.h>
# bao gồm <string.h>
```

hàm main() không có giá trị

```
{
dòng char[5][20];
int ctr, longctr=0;
int longest(char lines_arr[][20]);
* Khai báo hàm */

clrscr();
đối với (ctr = 0; ctr < 5; ctr++)
/* Chấp nhận các giá trị chuỗi vào mảng */ {

    printf("\nNhập chuỗi %d: ", ctr+1);
    scanf("%s", dòng[ctr]);

}
```

Ví dụ về việc truyền chuỗi đến Chức năng 2

```
longctr=dài nhất(các dòng);  
/* Truyền mảng vào hàm */  
  
printf("\nChuỗi dài nhất là %s", lines[longctr]);  
  
lấy();  
}  
  
int longest(char lines_arr[][20]) /* Định nghĩa hàm */ {  
  
    int i=0, l_ctr=0, độ dài_trước, độ dài_mới;  
  
    prev_len=strlen(dòng_mảng[i]);  
    /* Xác định độ dài của phần tử đầu tiên */
```

Ví dụ về việc truyền chuỗi đến Chức năng

```
đối với (i++;i<5;i++)
{
    new_len=strlen(dòng_mảng[i]);
    /* Xác định độ dài của phần tử tiếp theo */

    nếu(new_len>prev_len)
        l_ctr=i;
    /* Lưu trữ chỉ số dưới của chuỗi dài hơn */

    prev_len=new_len;
}

trả về l_ctr;
/* Trả về chỉ số dưới của chuỗi dài nhất */
}
```



Ví dụ về việc truyền chuỗi đến Chức năng-4

Mẫu đầu ra của chương trình

Nhập chuỗi 1:

Nhập chuỗi 2: Sigma

Nhập chuỗi 3: Giao thức

Nhập chuỗi 4: Robert

Nhập chuỗi 5: Ludlum

Chuỗi dài nhất là Protocol