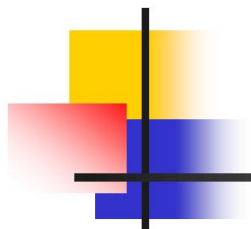




Đầu vào và đầu ra trong 'C'

Chỉ dành cho Trung tâm Aptech sử dụng

Phiên 4



Mục tiêu

Để hiểu các hàm I/O được định dạng - **fomatted**
scanf() và **printf()**

Chỉ dành cho Trung tâm Aptech sử dụng
Để sử dụng các hàm I/O ký tự - **unfomatted**

getchar() và **putchar()**

gets()

puts()



Đầu vào/Đầu ra chuẩn

Trong C, thư viện chuẩn cung cấp các thói quen cho đầu vào và đầu ra

Thư viện chuẩn có các hàm cho I/O xử lý đầu vào,
đầu ra, và thao tác ký tự và chuỗi

Chỉ dành cho Trung tâm Aptech sử dụng

Đầu vào chuẩn thường là bàn phím

Đầu ra chuẩn thường là màn hình (còn gọi là bàn điều khiển)

Đầu vào và đầu ra có thể được định tuyến lại từ hoặc đến các tệp thay vì các thiết bị tiêu chuẩn



Tập Tiêu đề <stdio.h>

```
#include <stdio.h>
```

- Đây là lệnh tiền xử lý

`stdio.h` là một tệp và được gọi là tệp tiêu đề

chứa các macro cho nhiều hàm đầu vào/đầu ra

được sử dụng trong 'C'

Các hàm `printf()`, `scanf()`, `putchar()`, `getchar()` được thiết kế sao cho chúng yêu cầu các macro trong `stdio.h` để thực thi đúng

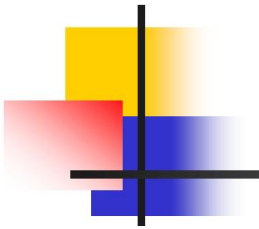


Định dạng đầu vào/đầu ra

`printf()` - cho đầu ra được định dạng

`scanf()` - cho đầu vào được định dạng

Định dạng chỉ định chỉ định định dạng trong đó giá trị của các biến sẽ được nhập vào và in ra



inf () - 1

được sử dụng để hiển thị dữ liệu trên đầu ra chuẩn - console

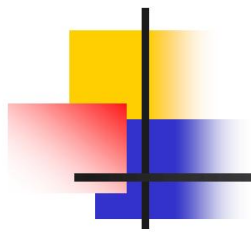
Cú pháp `printf("chuỗi điều khiển", danh sách đối số);`

Chỉ dành cho Trung tâm Aptech sử dụng
Danh sách đối số bao gồm các hằng số, biến, biểu thức hoặc hàm được phân tách bằng dấu phẩy

Phải có một lệnh định dạng trong chuỗi điều khiển cho mỗi đối số trong danh sách

Các lệnh định dạng phải khớp với danh sách đối số trong số, kiểu và thứ tự

Chuỗi điều khiển phải luôn được đặt trong dấu ngoặc kép, là dấu phân cách của nó



lệnh `inf()` -2

Chuỗi điều khiển bao gồm một hoặc nhiều trong ba loại mục sau:

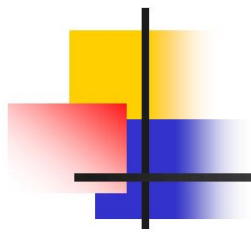
1. Ký tự văn bản : bao

Chỉ dành cho Trung tâm Aptech sử dụng
gồm các ký tự có thể in được

2. Lệnh định dạng: bắt đầu

bằng dấu % và theo sau là mã định dạng - phù hợp với mục dữ liệu

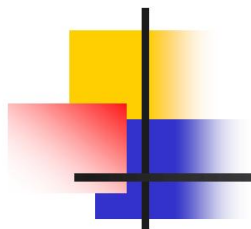
3. Ký tự không in: Bao gồm các tab,
khoảng trống và dòng mới



Mã định dạng-1

Định dạng	inf()	quét()
Ký tự đơn	%c	%c
Sợi dây	%S	%S
Số nguyên thập phân có dấu	%d	%d
Dấu chấm động (ký hiệu thập phân)	%f	%f hoặc %e
Dấu chấm động (ký hiệu thập phân)	%lf	%lf
Dấu chấm động (ký hiệu mũ)	%e	%f hoặc %e
Dấu chấm động (viết tắt là , bất cứ cái nào là %f hoặc %e)	%g	
Số nguyên thập phân không dấu	%u	%u
Số nguyên thập lục phân không dấu (sử dụng "ABCDEF")	%X	%X
Số nguyên bát phân không dấu	%O	%O

Trong bảng trên c, d, f, lf, e, g, u, s, o và x là các chỉ định loại



Mã định dạng-2

Mã định dạng	Quy ước in ấn
%d	Số chữ số trong số nguyên
%f	Phần nguyên của số sẽ được in ra như vậy. Phần thập phân sẽ bao gồm 6 chữ số. Nếu phần thập phân của số nhỏ hơn 6, nó sẽ được đệm bằng số 0 ở bên phải, nếu không nó sẽ được làm tròn ở bên phải.
%e	Một chữ số bên trái dấu thập phân và 6 chữ số bên phải, như trong %f ở trên



Chuỗi điều khiển ký tự đặc biệt

\\	để in ký tự \
\"	để in tính cách
%%	để in ký tự %

chuỗi điều khiển & mã định dạng

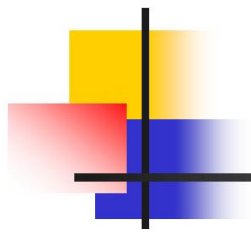
KHÔNG	Các tuyên bố	Điều khiển Sợi dây	Chuỗi điều khiển chứa những gì	Lý lẽ <small>Danh sách</small>	Giải thích về danh sách đối số	Màn hình Trưng bày
1.	<code>printf("%d",300);</code>	%d	Chỉ bao gồm lệnh định dạng	300	Không thay đổi	300
2.	<code>printf("%d",10+5);</code>	%d	Chỉ bao gồm lệnh định dạng	10 + 5	Sự biểu lộ	15
3.	<code>printf("Chào buổi sáng ông. Lý.");</code>	Chào buổi sáng ông Lee.	Chỉ bao gồm các ký tự văn bản	Không	Không	Chào buổi sáng ông. Lý.
4.	<code>int đếm = 100; printf("%d",đếm);</code>	%d	Chỉ bao gồm lệnh định dạng	đếm	biến đổi	100
5.	<code>printf("\nhello");</code>	\nhào	Bao gồm các ký tự không in và ký tự văn bản	Không	Không	xin chào trên một dòng mới
6.	<code>#define str "Táo ngon" printf("%s",chuỗi);</code>	%S	Chỉ bao gồm lệnh định dạng	Đường	Hàng số biểu tượng	Quả táo tốt
7.	<code>.... int đếm,số_con; đếm=0; stud_nim=100; printf("%d %d\n",đếm, số lượng học sinh);</code>	%d %d Bao gồm	lệnh định dạng và chuỗi thoát	đếm, stud_num	hai biến	0 , 100



Ví dụ cho printf()

Chương trình hiển thị số nguyên, số thực , ký tự và chuỗi

```
#include <stdio.h> void  
main() {  
    Chỉ dành cho Trung tâm Aptech sử dụng  
    int a = 10;  
    float b = 24.67892345; char ch  
    = 'A';  
  
    printf("Dữ liệu số nguyên = %d", a);  
    printf("Dữ liệu số thực = %f", b);  
    printf("Ký tự = %c", ch); printf("Lệnh  
    này in ra chuỗi"); printf("%s", "Lệnh này cũng  
    in ra một chuỗi");  
}
```



Các trình sửa đổi trong printf()-1

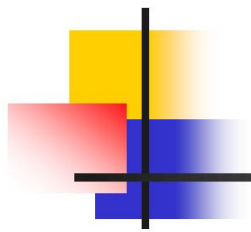
1. '-' Bỏ ngữ

Mục dữ liệu sẽ nằm trong trường của nó, mục sẽ là được in bắt đầu từ vị trí ngoài cùng bên trái của trường.

2. Bộ điều chỉnh độ rộng trường

Có thể sử dụng với kiểu float, double hoặc mảng char (chuỗi).

Bộ điều chỉnh độ rộng trường, là một số nguyên, xác định độ rộng trường cho mỗi dữ liệu.



Các trình sửa đổi trong printf()-2

3. Bộ điều chỉnh độ chính

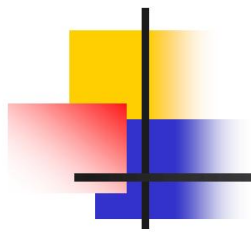
xác Bộ điều chỉnh này có thể được sử dụng với kiểu float, double hoặc mảng char (chuỗi). Nếu được sử dụng với kiểu dữ liệu float hoặc double, chuỗi chữ số sẽ chỉ ra số tối đa chữ số được in ở bên phải dấu thập phân.

4. Bộ điều chỉnh

'0' Đệm mặc định trong một trường được thực hiện bằng khoảng trắng. Nếu người dùng muốn đệm một trường bằng số không thì phải sử dụng bộ điều chỉnh này

5. 'l' Modifier

Modifier này có thể được sử dụng để hiển thị số nguyên dưới dạng long int hoặc đối số có độ chính xác kép. Mã định dạng tương ứng là %ld



Các trình sửa đổi trong printf()-3

6. Bộ ngữ 'h'

Bộ điều chỉnh này được sử dụng để hiển thị số nguyên ngắn.
Mã định dạng tương ứng là %hd

7. Bộ sửa đổi '*'

Nếu người dùng không muốn chỉ định trước độ rộng trường nhưng muốn chương trình chỉ định thì sử dụng trình sửa đổi này

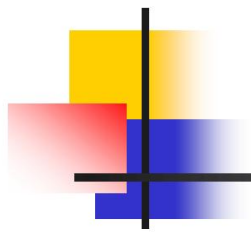


Ví dụ cho các trình sửa đổi

```
/* Chương trình này trình bày cách sử dụng Modifiers trong printf() */ #include <stdio.h>
void main() {
```

```
    printf("Số 555 ở nhiều dạng khác nhau:\n"); printf("Không có bất  
    ký ký tự sửa đổi nào:\n"); printf("[%d]\n",555);  
    printf("Có ký tự sửa đổi - :  
    \n"); printf("[%d]\n",555); printf("Với  
    chuỗi chữ số 10 làm ký tự sửa  
    đổi :\n"); printf("[%10d]\n",555); printf("Với 0 làm ký tự sửa đổi:  
    \n"); printf("[%0d]\n",555);  
    printf("Với 0 và chuỗi chữ số 10 làm ký tự sửa  
    đổi :\n"); printf("[%010d]  
    \n",555); printf("Với -, 0 và chuỗi chữ số 10 làm ký tự sửa đổi: \n");  
    printf("[%010d]\n",555);
```

```
}
```

quét ()

Được sử dụng để chấp nhận dữ liệu

Định dạng chung của hàm scanf()

`scanf("chuỗi điều khiển", danh sách đối số);`

Định dạng được sử dụng trong câu lệnh printf() được sử dụng với cú pháp tương tự trong các câu lệnh scanf()

Sự khác biệt trong danh sách đối số giữa printf() và scanf()

printf() sử dụng tên biến, hằng số, hằng số tượng trưng và biểu thức

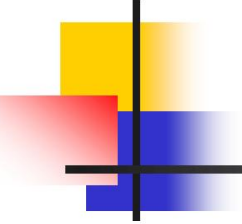
scanf() sử dụng con trỏ tới các biến

Khi sử dụng scanf() hãy tuân theo các quy tắc sau đối với danh sách đối số:

Nếu bạn muốn đọc giá trị của một biến có kiểu dữ liệu cơ bản, đặt trước tên biến bằng ký hiệu &

Khi đọc giá trị của một biến có kiểu dữ liệu phái sinh, không sử dụng & trước tên biến

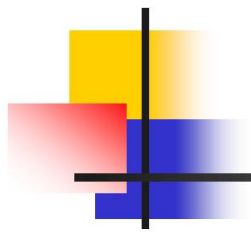
Sự khác biệt trong các lệnh định dạng của `printf()` và `scanf()`



Không có tùy chọn %g

Chỉ dành cho Trung tâm Aptech sử dụng

Mã định dạng %f và %e có hiệu lực giống nhau



Ví dụ cho scanf()

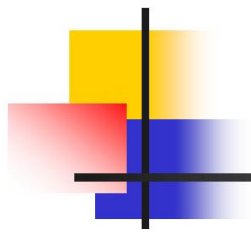
```
#include <stdio.h> void  
main() {
```

```
    int a;  
    float d;  
    char ch, tên[40];
```

```
    printf("Vui lòng nhập dữ liệu\n"); scanf("%d %f %c  
    %s", &a, &d, &ch, name);
```

```
    printf("\n Các giá trị được chấp nhận là: %d, %f, %c,  
    %s", a, d, ch, name);
```

```
}
```



Bộ đệm I/O

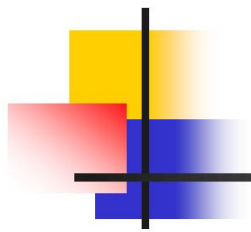
được sử dụng để đọc và ghi các ký tự ASCII

Bộ **đệm** là vùng lưu trữ tạm thời, trong bộ nhớ hoặc trên thẻ điều khiển của thiết bị

I/O đệm có thể được chia nhỏ thành:

I/O của bảng điều khiển

Tập đệm I/O



Giao diện điều khiển I/O

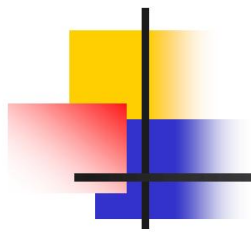
Các chức năng I/O của Console hướng các hoạt động của chúng đến chuẩn đầu vào và đầu ra của hệ thống

Chỉ dành cho Trung tâm Aptech sử dụng

Trong 'C' các hàm I/O giao diện điều khiển đơn giản nhất là:

`getchar()` - đọc một (và chỉ một) ký tự từ bàn phím

`putchar()` - xuất ra một ký tự duy nhất
trên màn hình



lấy char()

Được sử dụng để đọc dữ liệu đầu vào, từng ký tự một từ bàn phím

Chỉ dành cho Trung tâm Aptech sử dụng

Đệm các ký tự cho đến khi người dùng nhập một ký tự đầu dòng trở lại

Hàm `getchar()` không có đối số, nhưng dấu ngoặc đơn - vẫn phải có

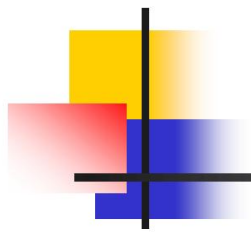


Ví dụ cho getchar()

```
/* Chương trình để chứng minh cách sử dụng getchar() */ #include <stdio.h> void  
main() {
```

Chỉ dành cho Trung tâm Aptech sử dụng

```
    char letter;  
    printf("\nVui lòng nhập bất kỳ ký tự nào : "); letter = getchar();  
    printf("\nKý tự bạn nhập là  
    %c", letter);  
}
```

putchar()

Hàm xuất ký tự trong 'C'

yêu cầu một lập luận

Chỉ dành cho Trung tâm Aptech sử dụng

Đối số của hàm putchar() có thể là:

Hằng số ký tự đơn

Một chuỗi thoát

Một biến ký tự



tùy chọn và hiệu ứng putchar()

Lý lẽ	Chức năng	Tác dụng
biến ký tự	<code>putchar(c)</code>	Hiển thị nội dung của biến ký tự c
hằng số ký tự	<code>putchar('A')</code>	Hiển thị chữ A
hằng số số	<code>putchar('5')</code>	Hiển thị chữ số 5
trình tự thoát	<code>putchar('\t')</code>	Chèn một ký tự khoảng cách tab vào vị trí con trỏ
trình tự thoát	<code>putchar('\n')</code>	Chèn một ký tự trả về tại vị trí con trỏ



putchar()

```
/* Chương trình này trình bày cách sử dụng hằng số và chuỗi thoát trong putchar()
*/
```

```
#include <stdio.h> void
```

```
main() {
```

```
    putchar('H'); putchar('\n'); putchar  
    ('\t');
```

```
    putchar('E'); putchar('\n');
```

```
    putchar('\t'); putchar('\t');
```

```
    putchar('L'); putchar('\n');
```

```
    putchar('\t'); putchar('\t'); putchar('\t'); putchar('L');
```

```
    putchar('\n'); putchar('\t');
```

```
    putchar('\t'); putchar('\t'); putchar('\t'); putchar('\t');
```

```
    putchar('0');
```

```
}
```

Ví dụ

Chỉ dành cho Trung tâm Aptech sử dụng