



TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

TIN HỌC ĐẠI CƯƠNG

Bài 6. Vào ra dữ liệu trong C

Các lệnh vào ra dữ liệu

- C cung cấp 2 hàm vào ra cơ bản:
 - printf()
 - scanf()
- Muốn sử dụng 2 hàm `printf()` và `scanf()` ta cần khai báo tệp tiêu đề `stdio.h`:
`#include <stdio.h>`
Hoặc
`#include "stdio.h"`

1. Hàm printf()

- Mục đích:
 - Hiển thị ra màn hình các loại dữ liệu cơ bản như: Số, kí tự và chuỗi kí tự
 - Một số hiệu ứng hiển thị đặc biệt như xuống dòng, sang trang,...

3

Mục đích và cú pháp

- Cú pháp:
printf(xâu_định_dạng , danh_sách_tham_số);
 - ***xâu_định_dạng***: Qui định cách thức hiển thị dữ liệu ra màn hình máy tính.
 - ***danh_sách_tham_số***: Danh sách các giá trị được hiển lên màn hình theo cách thức được qui định trong *xâu_định_dạng*. Có thể sử dụng:
 - Biến số
 - Hằng số
 - Biểu thức
 - Hàm
 - Giá trị cụ thể

4

Mục đích và cú pháp

- Ví dụ: Chương trình sau

```
#include <conio.h> //Khai bao tep tieu de
#include <stdio.h> //Khai bao tep tieu de
int main(){ //Viet chuong trinh chinh
    int a = 5;
    float x = 1.234;
    printf("Hien thi mot so nguyen a = %d và
           mot so thuc      x = %f",a,x);
    getch();
    return 0
}
```

- Sẽ cho ra kết quả:

Hien thi mot so nguyen a = 5 va mot so thuc x =1.234000

Mục đích và cú pháp

- Trong **xâu_định_dạng** chứa:
 - Các kí tự thông thường: Được hiển thị ra màn hình.
 - Các nhóm kí tự định dạng: Xác định quy cách hiển thị các tham số trong phần `danh_sách_tham_số`.
 - Các kí tự điều khiển: Dùng để tạo các hiệu ứng hiển thị đặc biệt như xuống dòng (`'\n'`) hay sang trang (`'\f'`)...

Mục đích và cú pháp

- Mỗi nhóm kí tự định dạng chỉ dùng cho một kiểu dữ liệu
Ví dụ: %d dùng cho kiểu nguyên
%f dùng cho kiểu thực
- Nếu giữa nhóm kí tự định dạng và tham số tương ứng không phù hợp với nhau thì sẽ hiển thị ra kết quả không như ý.

7

Mục đích và cú pháp

- *danh_sách_tham_số* phải phù hợp với các nhóm kí tự định dạng trong *xâu_định_dạng* về:
 - Số lượng
 - Kiểu dữ liệu
 - Thứ tự

8

Một số nhóm định dạng phổ biến

Nhóm kí tự định dạng	Kiểu dữ liệu	Kết quả
%c	char	Kí tự đơn lẻ
%i, %d	int, char	Số thập phân
%o	int, char	Số bát phân (không có 0 đằng trước)
%x, %X	int, char	Số hexa (chữ thường/chữ hoa)
%u	unsigned int/char	Số thập phân

9

Một số nhóm định dạng phổ biến

Nhóm kí tự định dạng	Kiểu dữ liệu	Kết quả
%ld, %li	long	Số thập phân
%lo	long	Số bát phân (không có 0 đằng trước)
%lx, %LX	long	Số hexa (chữ thường/chữ hoa)
%lu	unsigned long	Số thập phân

10

Một số nhóm định dạng phổ biến

Nhóm kí tự định dạng	Kiểu dữ liệu	Kết quả
%s	char []	Hiển thị xâu kí tự kết thúc bởi '\0'
%f	float/double	Số thực dấu phẩy tĩnh
%e, %E	float/double	Số thực dấu phẩy động
%		Hiển thị kí tự %

11

Độ rộng hiển thị - số nguyên

- Đối với **số nguyên**

- Có dạng %md, với m là số nguyên không âm
- Ví dụ: Có số a = 1234

Lệnh:

```
printf("%5d",a); //danh 5 cho de hien thi a  
printf("\n%5d",34);
```

Cho ra kết quả: □1234

□□□34

(□ kí hiệu cho dấu cách đơn (space))

12

Độ rộng hiển thị - số nguyên

- Ví dụ:

```
printf("\n%3d%15s%3c", 1, "nguyen van a", 'g');  
printf("\n%3d%15s%3c", 2, "tran van b", 'k');
```

- Kết quả:

```
  1          nguyen van a  g  
  2          tran van b  k
```

13

Độ rộng hiển thị - số thực

- m, n là 2 số nguyên không âm

`%m.nf`

Trong đó:

- m vị trí để hiển thị số thực (dấu '.' được tính 1 vị trí)
- n vị trí trong m vị trí đó để hiển thị phần thập phân.

14

Độ rộng hiển thị - số thực

- Ví dụ:

```
printf("\n%f", 17.345);  
printf("\n%.2f", 17.345);  
printf("\n%7.2f", 17.345);
```
- Kết quả:

```
17.345000  
17.35  
□□17.35
```

15

Độ rộng hiển thị - Chú ý

- Khi số chỗ cần thiết để hiển thị nội dung dữ liệu lớn hơn trong định dạng → Tự động cung cấp thêm chỗ mới để hiển thị trong các trường hợp sau:
 - Số nguyên
 - Phần nguyên của số thực
 - Xâu ký tự
- Ví dụ: `a = 1000`

```
printf("Số a là: %1d", a);
```

 - Kết quả:

```
Số a là: 1000
```

16

Căn lề phải, lề trái

- **Căn lề phải:**
 - Khi hiển thị dữ liệu, mặc định C căn lề phải
- **Căn lề trái:**
 - Nếu muốn căn lề trái khi hiển thị dữ liệu ta chỉ cần thêm dấu trừ - vào ngay sau dấu %.

17

Căn lề phải, lề trái

- Ví dụ:

```
printf("\n%-3d %-15s %5.2f %-3c",
      9, "nguyen van a", 7.5, 'g');
printf("\n%-3d %-15s %5.2f %-3c",
      10, "nguyen ha", 6.75, 'k');
```
- Kết quả:

```
9   nguyen van a   7.50g
10  nguyen ha     6.75k
```

18

2. Hàm scanf()

- Mục đích:
 - Hàm scanf() dùng để nhập dữ liệu từ bàn phím
- Cú pháp:
`scanf(xâu_định_dạng,danh_sách_địa_chỉ);`
- Ví dụ:
`scanf ("%d %f" , &a , &b) ;`

19

Mục đích và cú pháp

- `xâu_định_dạng`:
 - Gồm các ký tự được qui định cho từng loại dữ liệu được nhập vào.
 - Ví dụ: Với dữ liệu định nhập vào là kiểu nguyên thì `xâu_định_dạng` là : `%d`
- `danh_sách_địa_chỉ`:
 - Bao gồm các địa chỉ của các biến (toán tử `&`), phân tách nhau bởi dấu phẩy (,)

20

Mục đích và cú pháp

- **danh_sách_địa_chỉ** phải phù hợp với các nhóm kí tự định dạng trong **xâu_định_dạng** về:
 - Số lượng
 - Kiểu dữ liệu
 - Thứ tự

21

Một số nhóm định dạng phổ biến

Nhóm kí tự định dạng	Kiểu dữ liệu	Chú thích
%c	char	Kí tự đơn lẻ
%d	int	Số thập phân
%o	int	Số bát phân
%x	int	Số hexa
%u	unsigned int	Số thập phân

22

Một số nhóm định dạng phổ biến

Nhóm kí tự định dạng	Kiểu dữ liệu	Chú thích
%s	char[]	Hiển thị xâu kí tự kết thúc bởi '\0'
%f	float	Số thực dấu phẩy tĩnh
%ld	long	Số nguyên
%lf	double	Số thực dấu phẩy tĩnh
%		Đọc ký tự %

23

Ví dụ

```
#include <conio.h>
#include <stdio.h>
int main()
{
    // khai bao bien
    int a;
    float x;
    char ch;
    // Nhap du lieu
    printf("Nhap vao mot so nguyen");
    scanf("%d", &a);
    printf("\n Nhap vao mot so thuc");
    scanf("%f", &x);
```

24

Ví dụ

```
printf("\n Nhap vao mot ki tu");  
fflush(stdin); scanf("%c",&ch);  
  
// Hien thi du lieu vua nhap vao  
printf("\n Nhung du lieu vua nhap vao");  
printf("\n So nguyen: %d",a);  
printf("\n So thuc : %.2f",x);  
printf("\n Ki tu: %c: ",ch);  
getch();  
return 0;  
}
```

25

Ví dụ

- Kết quả:
Nhập vào một số nguyên: 2007
Nhập vào một số thực: 17.1625
Nhập vào một ký tự: b
Nhưng dữ liệu vừa nhập vào
Số nguyên: 2007
Số thực: 17.16
Ký tự: b

26

Một số quy tắc cần lưu ý

- Quy tắc 1: Khi đọc số
 - Hàm `scanf()` quan niệm rằng mọi kí tự số, dấu chấm ('.') đều là kí tự hợp lệ.
 - Khi gặp các dấu phân cách như tab, xuống dòng hay dấu cách (space bar) thì `scanf()` sẽ hiểu là kết thúc nhập dữ liệu cho một số

27

Một số quy tắc cần lưu ý

- Quy tắc 2: Khi đọc kí tự:
 - Hàm **`scanf()`** cho rằng mọi kí tự có trong bộ đệm của thiết bị vào chuẩn đều là hợp lệ, kể cả các kí tự tab, xuống dòng hay dấu cách.
 - Trước khi nhập dữ liệu kí tự hay chuỗi kí tự ta nên dùng lệnh **`fflush(stdin)`** để xóa bộ đệm.

28

3. Các lệnh vào ra khác

- Hàm `getch()`: thường dùng để chờ người sử dụng ấn một phím bất kì rồi sẽ kết thúc chương trình.
- Cú pháp

`getch () ;`

- Để sử dụng các hàm `getch ()` ta cần khai báo tệp tiêu đề `conio.h`.

29

Ví dụ

```
#include <conio.h>
#include <stdio.h>
int main()
{
    int a, b;
    printf("Nhap a: "); scanf("%d", &a);
    printf("Nhap b: "); scanf("%d", &b);
    printf("Tong hai so la: %d", a+b);
    printf("Nhan mot phim bat ky de ket
        thuc!");
    getch();
    return 0;
}
```

30

Ví dụ

- Kết quả:

Nhap a: 1

Nhap b: 9

Tong hai so la: 10

Nhan mot phim bat ky de ket thuc!

31

Bài tập

- Viết chương trình tính tổng, hiệu, tích của 2 số thực a và b
- Viết chương trình tính giá trị hàm số

$$f(x) = 3\sin(x) + 4\cos(x)$$

$$g(x, y) = 3|x| + y^2$$

32