

BÀI 3: VÀO RA

MỤC TIÊU

Người học sau khi học xong bài 3 sẽ có các khái niệm cơ bản về các vấn đề sau:

- Hàm in chuẩn printf và fprintf;
- Đưa dữ liệu ra máy in;
- Dòng stdin và các hàm scanf, gets;
- Hàm xuất ký tự puts và putchar.

KIẾN THỨC CẦN CÓ

- Học xong bài 2.
- Hiểu biết cơ bản về thao tác bàn phím liên quan đến các lệnh in ấn & làm việc được với máy in.
- Khuyến nghị học môn Tin Học Cơ bản.

NỘI DUNG

1. Hàm printf()
2. Hàm scanf()
3. Đưa ra máy in
4. Ví dụ minh họa các quy tắc vào ra
5. Dòng vào STDIN và các hàm scanf(), gets(), getchar()
6. Các hàm xuất ký tự puts() và putchar()
7. Các hàm vào ra trên màn hình, bàn phím

1. HÀM PRINTF()

- Hàm printf() dùng để chuyển dạng, tạo khuôn và đưa giá trị các đối số ra màn hình.
- Cú pháp tổng quát:

```
int printf(const char *dk, [, danh sách các đối số]);
```

- dk: chuỗi điều khiển, có nhiều lựa chọn cho chuỗi điều khiển như sau:
 - Các ký tự điều khiển: Cho phép hiển thị các ký tự đặc biệt, thông qua 1 ký tự \ đứng trước.

| Viết: | Hiển thị: |
|-------|-----------|
| \' | ' |
| \" | " |
| \\ | \ |

1. HÀM PRINTF (tiếp theo)

➤ Các ký tự tạo khuôn dạng:

`%[-][fw][.pp] ký_tự_chuyển_dạng`

- Dấu % và ký_tự_chuyển_dạng là 2 thành phần bắt buộc phải có
- Dấu - Dùng để lấp đầy khoảng trống của kết quả hiển thị khi trường thông tin dành cho hiển thị chưa được lấp đầy
- fw: Xác định độ rộng tối thiểu của dãy ra
- pp: Chỉ sử dụng khi đối tượng có kiểu dữ liệu là float hoặc double

1. HÀM PRINTF (tiếp theo)

- Ký tự chuyển dạng: Dùng để chuyển dạng hiển thị tín hiệu ra, có 1 số định dạng tiêu biểu như sau:

| Ký tự chuyển dạng | Kiểu của đối | Cách chuyển dạng |
|-------------------|--------------|---------------------|
| c | char | ký tự |
| di | int | số nguyên có dấu |
| ld,li | long | số nguyên có dấu |
| u | int | số nguyên không dấu |
| o | int | hệ cơ số 8 |
| lo | long | hệ cơ số 8 (dài) |
| x | int | hệ 16 |

2. HÀM SCANF()

- Hàm scanf có nhiều chức năng tương tự như hàm printf nhưng được tiến hành theo thứ tự ngược lại.
- Cú pháp hàm scanf:

```
int scanf(const char *dk [, danh sách các đối]);
```

- scanf: tên hàm
- dk: chuỗi điều khiển
- danh sách các đối: chứa địa chỉ vùng nhớ để lưu trữ giá trị đọc vào từ bàn phím, các đối được viết cách nhau bởi 1 dấu phẩy

2. HÀM SCANF() (tiếp theo)

Thông tin đặc tả của hàm scanf:

`% [*][d....d]ký_tự_chuyển_dạng`

- [*] Trường vào được đọc bình thường, nhưng giá trị bị bỏ qua và không lưu vào bộ nhớ
- [d...d] Dãy số xác định chiều dài cực đại của trường vào
 - Nếu không có [d...d] thì dãy số vào sẽ được đọc hoàn toàn
 - Nếu có [d...d] thì bộ nhớ chỉ lưu thông tin với kích thước cực đại đã khai báo. Phần còn lại tùy thuộc các đặc tả phía sau

Ví dụ: dòng vào là:

`13.2 abc\t 25\n`

- Đầu vào có 3 trường 13.2, abc và 25
- Độ dài lần lượt là 4, 3 và 2

2. HÀM SCANF() (tiếp theo)

Ký tự chuyển dạng: Dùng để chuyển dạng tín hiệu vào, nó tương đương với hàm printf với 1 số định dạng tiêu biểu như sau:

| Ký tự chuyển dạng | Kiểu của đối | Cách chuyển dạng |
|-------------------|--------------|---------------------|
| c | char | ký tự |
| di | int | số nguyên có dấu |
| ld,li | long | số nguyên có dấu |
| u | int | số nguyên không dấu |
| o | int | hệ cơ số 8 |
| lo | long | hệ cơ số 8 (dài) |
| x | int | hệ 16 |

3. ĐƯA RA MÁY IN

Việc xuất tín hiệu ra máy in thông qua hàm chuẩn sau:

```
fprintf(stdprn, const char *dk [, danh sách các đối]);
```

- fprintf: tên hàm
- stdprn: xác định thiết bị đưa ra là máy in
- *dk: biến con trỏ kiểu char, dùng để chứa địa chỉ các chuỗi điều khiển
- danh sách các đối: tương tự như đối với hàm printf

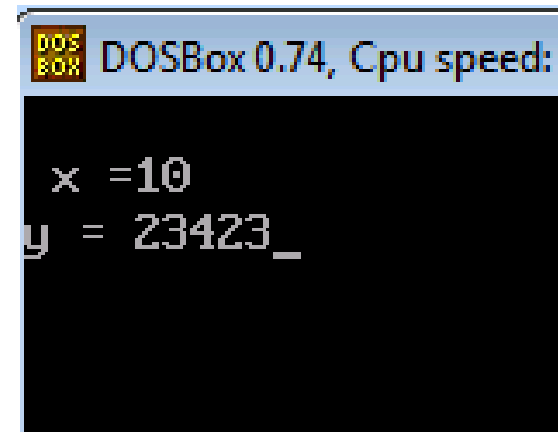
Ví dụ: chương trình in ra các phần tử của mảng kích thước 8x6

```
float a[8][6];  
int j,i;  
for(i = 0, i < 8, i++)  
{  
    for (j = 0, j < 6, j++)  
    {  
        fprintf(stdprn, "%10.2f", a[i][j]);  
    }  
}
```

4. VÍ DỤ MINH HỌA CÁC QUY TẮC VÀO RA

Ví dụ 1:

```
#include <stdio.h>
#include <conio.h>
int main(void)
{
    int x = 10;
    long y = 23423
    printf("\n x =%d \n y = %ld", x, y);
    getch();
}
```

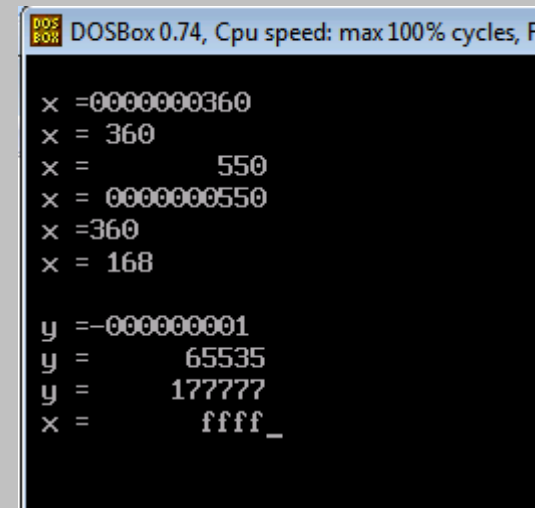


4. VÍ DỤ MINH HỌA CÁC QUY TẮC VÀO RA (tiếp theo)

Ví dụ 2:

```
#include <stdio.h>
#include <conio.h>
void main(void)
{
    int x = 360, y = -1;
    long yl = -11;
    char *vb = "Programmer";
    double dx = 3.22312;
```

```
    printf("\n x =%010d \n x = %-010d \n x = %10o \n x = %010o",x,x,x,x);
    printf("\n x =%-10d \n x = %-010x ",x,x);
    printf("\n \n y =%010d \n y = %10u \n y = %10o \n x = %10x",y,y,y,y);
    getch();
}
```



```
DOSBox 0.74, Cpu speed: max 100% cycles, F
x =0000000360
x = 360
x = 550
x = 0000000550
x =360
x = 168
y =-000000001
y = 65535
y = 177777
x = ffff_
```

5. DÒNG VÀO STDIN VÀ CÁC HÀM SCANF() , GETS(), GETCHAR ()

- Hàm gets() nhập 1 chuỗi ký tự từ stdin.
- Cú pháp:

```
char *gets(char *s);
```

s: con trỏ trỏ tới vùng nhớ chứa ký tự nhận được

- Công dụng:
 - Nhận dãy ký tự từ stdin, nếu gặp ký tự \n thì không lưu vào bộ nhớ
 - Chuỗi được bổ xung thêm ký tự kết thúc \0 và đặt vào cùng vùng nhớ
 - Hàm trả về địa chỉ chuỗi nhận được

5. DÒNG VÀO STDIN VÀ CÁC HÀM SCANF() , GETS(), GETCHAR () (tiếp theo)

- Hàm getchar(): Nhận 1 ký tự từ stdin
- Cú pháp:

```
int getchar(void);
```

Công dụng:

- Nhận 1 ký tự từ stdin
- Trả về ký tự nhận được
- Có thể xóa sạch thông tin có trong stdin bằng lệnh:

```
fflush(stdin);
```

- Lệnh xóa này sẽ cho phép tránh được cách ảnh hưởng do quá trình nhập liệu bởi hàm gets() và scanf()

6. CÁC HÀM XUẤT KÝ TỰ PUTS() VÀ PUTCHAR()

- Hàm puts(): Đưa 1 chuỗi ký tự ra stdout
- Cú pháp:

```
int puts(const char *s);
```

s: kiểu con trỏ trỏ tới vị trí vùng nhớ cần xuất ra

- Công dụng:
 - Đưa chuỗi ký tự và \n nếu có ra stdout
 - Nếu thành công, trả về ký tự \n
 - Nếu không thành công, trả về lỗi EOF

Ví dụ:

```
puts("\n Hanoi")
```

- Đưa chữ Hanoi lên 1 dòng mới
- Chuyển con trỏ xuống dòng dưới

6. CÁC HÀM XUẤT KÝ TỰ PUTS() VÀ PUTCHAR() (tiếp theo)

- Hàm putchar(): Đưa 1 ký tự ra stdout
- Cú pháp:

```
int putchar(int ch);
```

ch: chứa mã ký tự cần xuất ra

- Công dụng:
 - Đưa ký tự và \n nếu có ra stdout
 - Nếu thành công, trả về ký tự \n
 - Nếu không thành công, trả về lỗi EOF

Ví dụ:

```
putchar("\n A")
```

- Đưa chữ A lên 1 dòng mới
- Chuyển con trỏ xuống dòng dưới

7. CÁC HÀM VÀO RA TRÊN MÀN HÌNH, BÀN PHÍM

- Hàm getch(): nhận 1 ký tự từ bộ đệm bàn phím, không cho hiện lên màn hình.
- Cú pháp:

```
int getch(void);
```

- Công dụng:
 - Nhận 1 ký tự trong bộ đệm bàn phím
 - Dừng màn hình cho tới khi có 1 ký tự mới được nhập vào
 - Không hiện ký tự lên màn hình
- Trả về: ký tự nhận được

7. CÁC HÀM VÀO RA TRÊN MÀN HÌNH, BÀN PHÍM (tiếp theo)

- Hàm `getche()`: nhận 1 ký tự từ bộ đệm bàn phím, cho hiện lên màn hình.
- Cú pháp:

```
int getche(void);
```

- Công dụng:
 - Nhận 1 ký tự trong bộ đệm bàn phím
 - Dừng màn hình cho tới khi có 1 ký tự mới được nhập vào
 - Hiện ký tự lên màn hình
- Trả về: ký tự nhận được

7. CÁC HÀM VÀO RA TRÊN MÀN HÌNH, BÀN PHÍM (tiếp theo)

- Hàm `putch()`: Đưa 1 ký tự ra cửa sổ văn bản màn hình.
- Cú pháp:

```
int putch(inh ch);
```

- Công dụng:
 - Đưa ký tự cửa sổ văn bản màn hình;
 - Ký tự có thể có màu khác nhau, thông qua hàm `textcolor()`;
 - Hàm `putch()` luôn hiển thị ký tự màu trắng.
- Trả về: ký tự được hiển thị.

7. CÁC HÀM VÀO RA TRÊN MÀN HÌNH, BÀN PHÍM (tiếp theo)

- Hàm kbhit(): Kiểm tra bộ đệm bàn phím:
- Cú pháp:

```
int kbhit(void);
```

Công dụng:

- Hàm có giá trị = 0 nếu bộ đệm rỗng;
 - Hàm có giá trị != 0 nếu bộ đệm có thông tin;
 - Ký tự có thể có màu khác nhau, thông qua hàm textcolor();
 - Hàm putch() luôn hiển thị ký tự màu trắng.
- Trả về: giá trị 0 hoặc số khác 0.
 - Hàm xóa màn hình: Dùng để xóa các thông tin hiển thị trên màn hình.
 - Cú pháp:

```
Clrscr();
```

TÓM LƯỢC CUỐI BÀI

- Thông qua các kiến thức học được trong bài 3, người học có thể biết được các thao tác cơ bản liên quan đến các thủ tục vào/ra mà chương trình C có thể thực hiện được.
- Lệnh in ra màn hình hoặc in ra máy in tương đối giống nhau.
- Để có thể nhận thông tin từ bàn phím thì có thể sử dụng 2 lệnh gets hoặc scanf.
- Khi cần xuất ký tự thì dùng lệnh puts hoặc putchar.