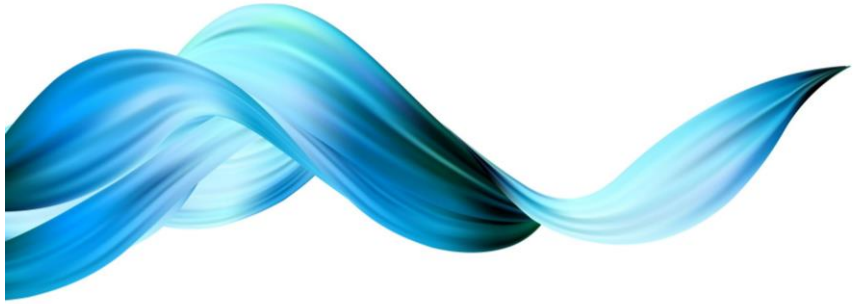

- Xác định độ rộng trường và căn lề
- Làm tròn số
- Các cờ
- Các kí tự điều khiển
- Đọc định dạng đầu vào với scanf
- Tập các kí tự cần đọc hoặc bỏ qua

Chuẩn định dạng vào ra trong C

Xác định độ rộng trường, căn lề

- Hàm printf, scanf, fprintf, fscanf dùng các định dạng chuyển đổi để nhập xuất dữ liệu các kiểu tương ứng khi thao tác với các chuẩn vào ra khác nhau: màn hình-bàn phím-file
- Để in ra giá trị x trên một vùng với độ rộng tương đương n kí tự ta chèn n vào sau %, trước định dạng của kiểu tương ứng
- Ví dụ:
  - %10d sẽ dành 10 vị trí in ra số nguyên
  - %20s sẽ dành 20 vị trí in ra chuỗi kí tự
  - %8f sẽ dành 8 vị trí in ra số thực kiểu float
- Để căn lề nội dung sang trái vùng in ra, ta thêm dấu – trước n: %-10d, %-20s, …

Ví dụ áp dụng

- Ví dụ in ra số nguyên:

```
int a = 12345;
// in ra binh thuong:
printf("a = %d\n", a);
// in ra gia tri a voi do rong 10:
printf("a = %10d\n", a);
printf("a = %-10d\n", a);
```

```
a = 12345
a =      12345
a = 12345
```

Làm tròn số

- Với số thực, nếu muốn làm tròn số đến k chữ số sau phần nguyên, ta dùng %.kf với số kiểu float và %.klf với số kiểu double
- Ví dụ: %.2f làm tròn 2 chữ số sau dấu phẩy
- Kết hợp với độ rộng trường: %n.kf hoặc %n.klf có ý nghĩa là dành n vị trí in ra số thực x, giá trị in ra được làm tròn k chữ số sau dấu phẩy

Ví dụ áp dụng

- Ví dụ sau áp dụng cả độ rộng trường, căn lề, làm tròn số:

```
float PI = 3.141592f;
// in ra binh thuong:
printf("PI = %f\n", PI);
// in ra gia tri a voi do rong 10
// do chinh xac 2 chu so dang sau dau ,
printf("PI = %10.2f\n", PI);
printf("PI = %-10.2f\n", PI);
```

```
PI = 3.141592
PI =       3.14
PI = 3.14
```

Mở rộng

- Có thể xác định độ rộng trường và độ chính xác qua đối số.
- Ví dụ:

```
printf("PI = %*.*f\n", 10, 2, PI);
printf("PI = %*.*f\n", -10, 2, PI);
```

Các cờ

- Các cờ và ý nghĩa sử dụng

Cờ	Mô tả
-	Căn lề trái với độ rộng trường cho trước
+	Hiển thị giá trị số với dấu + hoặc trừ đằng trước
Dấu cách	In một giá trị dấu cách trước giá trị dương mà không in dấu +
0	Điền 0 vào trước giá trị gốc nếu độ rộng trường thừa chỗ in ra giá trị gốc

```
int a = 120;
int b = -123;
// dien 0 vao truoc gia tri goc
printf("%08d\n", a);
printf("%08d\n", b);
// hien thi dau cua gia tri
printf("%+8d\n", a);
printf("%+8d\n", b);
```

```
00000120
-0000123
    +120
    -123
```

Các kí tự điều khiển

- Các kí tự điều khiển

Kí tự	Mô tả
\’	Xuất ra kí tự nháy đơn ‘
\”	Xuất ra kí tự nháy kép “
\?	Xuất ra kí tự dấu hỏi ?
\\	Xuấ ra kí tự \
\a	Bật chuông cảnh báo
\b	Di chuyển con trỏ chuột về trước 1 vị trí trên dòng hiện tại
\n	Đưa con trỏ chuột xuống đầu dòng kế tiếp
\f	Đưa con trỏ chuột đến vị trí bắt đầu của trang logic kế tiếp
\r	Đưa con trỏ chuột đến vị trí bắt đầu của dòng hiện tại
\t	Đưa con trỏ chuột đến vị trí cách vị trí hiện tại 1 tab ngang
\v	Đưa con trỏ chuột đến vị trí cách vị trí hiện tại 1 tab dọc

Đọc định dạng đầu vào với hàm scanf

- Khi đọc dữ liệu nên đưa ra lời nhắc để người dùng nhập đúng kiểu dữ liệu tương ứng
- Tránh yêu cầu người dùng nhập nhiều kiểu dữ liệu khác nhau trong 1 lần nhập
- Mỗi lần chỉ nên đọc vào một số ít giá trị nhất định
- Luôn có kế hoạch dự phòng trong trường hợp dữ liệu nhận được không như mong muốn: đưa ra quy ước, lời nhắc, ngoại lệ…
- Ta có thể giới hạn lượng giá trị nhập vào với một số định dạng kiểu nguyên và chuỗi kí tự

Ví dụ áp dụng

- Các ví dụ sau sẽ dùng để nhập vào một số lượng giới hạn các giá trị:

```
int x;
printf("Enter an integer number: ");
scanf("%5d", &x);
printf("Value of x = %d\n", x);
```

```
Enter an integer number: 123456768576586758
Value of x = 12345
```

```
char name[21];
printf("%s", "What your name? ");
scanf("%20s", name);
```

Tập các giá trị cần đọc

- Để xác định tập các kí tự cần đọc ta để chúng ở trong cặp dấu [], kết hợp với độ rộng trường, bắt đầu bởi %. Ví dụ:

```
char s[10];
printf("%s", "Enter a string: ");
scanf("%9[aeiou]", s);
printf("s = \"%s\"", s);
```

```
Enter a string: eoeiehhhh
s = "eoeie"
```

Tập các giá trị cần bỏ qua

- Để xác định tập các kí tự cần bỏ qua ta làm y hệt việc xác định các kí tự cần đọc, bổ sung thêm dấu ^ ở sau [
- Việc đọc sẽ dừng khi gặp một trong các kí tự trong tập đã chỉ ra

```
char s[10];
printf("%s", "Enter a string: ");
scanf("%9[^aeiou]", s);
printf("s = \"%s\"", s);
```

```
Enter a string: This is a string
s = "Th"
```

Bỏ qua việc đọc 1 kí tự

- Nếu muốn bỏ qua việc đọc một kí tự và chương trình không kết thúc khi gặp kí tự đó, ta sẽ dùng %*c
- Ví dụ đọc ngày, tháng, năm sinh nhưng bỏ qua việc đọc kí tự phân tách / hoặc –

```
int day, month, year;
printf("%s", "Enter your birth day in format: dd/mm/yyyy\n");
scanf("%d%*c%d%*c%d", &day, &month, &year);
printf("day = %d\nmonth = %d\nyear = %d\n", day, month, year);
```

```
Enter your birth day in format: dd/mm/yyyy
27-04-2001
day = 27
month = 4
year = 2001

Enter your birth day in format: dd/mm/yyyy
16*09*2004
day = 16
month = 9
year = 2004
```

Tiếp theo

Kiểu dữ liệu struct