



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

C PROGRAMMING INTRODUCTION

TUẦN 3: GIỚI THIỆU ĐẦU RA CHUẨN

Nội dung

- Ôn lại:
 - Hiển thị kết quả
 - **printf**
 - Luồng
 - puts, putchar (in <stdio.h>)
- Thực hành lập trình

Vào/ra trong C

- C không có câu lệnh dựng sẵn cho thao tác vào ra
- `<stdio.h>`: thư viện cung cấp các thao tác vào ra dữ liệu

Luồng

- Luồng:
 - Một chuỗi các kí tự được sắp xếp thành các dòng
 - dòng kết thúc với kí tự xuống dòng
 - ANSI C hỗ trợ dòng có tối đa 254 kí tự
 - Thực hiện tất cả các thao tác vào/ra
 - Thường được điều hướng
 - Đầu vào chuẩn - bàn phím
 - Đầu ra chuẩn - màn hình
 - Lỗi chuẩn - màn hình

Định dạng đầu ra với printf

- **printf**

- định dạng đầu ra chính xác: các cờ, độ rộng các trường, độ chính xác...
- làm tròn, giống cột, căn lề trái/phải, chèn các kí tự đặc biệt, định dạng mũ, định dạng hệ 16, cố định độ rộng và độ chính xác

printf(chuỗi điều khiển định dạng, các tham số khác) ;

- chuỗi điều khiển định dạng: bao gồm danh sách kiểu dữ liệu của các biến cần in ra, kèm theo các nội dung không bắt buộc.
- các tham số khác: các biến tương ứng với các vị trí trong chuỗi điều khiển định dạng
 - mỗi đặc tả kiểu dữ liệu bắt đầu với một dấu %

In số nguyên

- Số nguyên
 - Không chứa phần thập phân: 25, 0, -9
 - Số dương, số âm, hoặc không

Conversion Specifier	Description
d	Display a signed decimal integer.
i	Display a signed decimal integer. (<i>Note:</i> The i and d specifiers are different when used with scanf .)
o	Display an unsigned octal integer.
u	Display an unsigned decimal integer.
x or X	Display an unsigned hexadecimal integer. X causes the digits 0-9 and the letters A-F to be displayed and x causes the digits 0-9 and a-f to be displayed.
h or l (letter l)	Place before any integer conversion specifier to indicate that a short or long integer is displayed respectively. Letters h and l are more precisely called <i>length modifiers</i> .

Ví dụ 1

```
1 #include <stdio.h>
2
3 int main()
4 {
5     printf( "%d\n", 455 );
6     printf( "%i\n", 455 ); /*i same as d*/
7     printf( "%d\n", +455 );
8     printf( "%d\n", -455 );
9     printf( "%hd\n", 32000 );
10    printf( "%ld\n", 20000000000 );
11    printf( "%o\n", 455 );
12    printf( "%u\n", 455 );
13    printf( "%u\n", -455 );
14    printf( "%x\n", 455 );
15    printf( "%X\n", 455 );
16
17    return 0;
18 }
```

```
455
455
455
-455
32000
20000000000
707
455
65081
1c7
1C7
```

In số thực

- Số thực dấu phẩy động
 - Chứa dấu '.' ngăn cách phần nguyên và phần thập phân (33.5)
 - Biểu diễn dạng mũ
 - $150.3 = 1.503 \times 10^2$
 - $150.3 = 1.503\text{E}+02$ (E là viết tắt của exponent)
 - dùng e hoặc E
 - **f** - in số thực dấu phẩy động với ít nhất một chữ số phần thập phân
 - **g** (hoặc **G**) - in theo định dạng **f** hoặc **e(E)** không chứa số 0 ở cuối (1.2300 trở thành 1.23)

Ví dụ 2

```
1 #include <stdio.h>
2
3 int main()
4 {
5     printf( "%e\n", 1234567.89 );
6     printf( "%e\n", +1234567.89 );
7     printf( "%e\n", -1234567.89 );
8     printf( "%E\n", 1234567.89 );
9     printf( "%f\n", 1234567.89 );
10    printf( "%g\n", 1234567.89 );
11    printf( "%G\n", 1234567.89 );
12
13    return 0;
14 }
```

```
1.234568e+006
1.234568e+006
-1.234568e+006
1.234568E+006
1234567.890000
1.23457e+006
1.23457E+006
```

In chuỗi và kí tự

- **c**
 - In kí tự kiểu **char**
 - Không thể dùng để in ra kí tự đầu tiên của chuỗi
- **s**
 - Yêu cầu tham số là con trỏ kiểu **char**
 - In ra các kí tự cho tới khi gặp **NULL** (**'\0'**)
 - Không thể in tham số kiểu **char**
- Ghi nhớ
 - Hằng kí tự (**'z'**)
 - Hằng chuỗi **"z"** (thực chất bao gồm 2 kí tự **'z'** và **'\0'**)

Ví dụ 3

```
1 #include <stdio.h>
2
3 int main()
4 {
5     char character = 'A';
6     char string[] = "This is a string";
7     const char *stringPtr = "This is also a string";
8
9     printf( "%c\n", character );
10    printf( "%s\n", "This is a string" );
11    printf( "%s\n", string );
12    printf( "%s\n", stringPtr );
13
14    return 0;
15 }
```

```
A
This is a string
This is a string
This is also a string
```

Các đặc tả khác

- **p**
 - Hiển thị giá trị con trỏ (địa chỉ)
- **n**
 - Lưu số kí tự được in ra bởi lệnh `printf` hiện thời
 - Lấy tham số là con trỏ kiểu số nguyên
 - `%n` không in ra nội dung gì
 - Lệnh `printf` trả về giá trị
 - Số kí tự in ra
 - Giá trị âm nếu có lỗi
- **%**

Ví dụ 4

```
1 #include <stdio.h>
2
3 int main()
4 {
5     int *ptr;
6     int x = 12345, y;
7
8     ptr = &x;
9     printf( "The value of ptr is %p\n", ptr );
10    printf( "The address of x is %p\n\n", &x );
11
12    printf("Total characters printed on this line is:%n",&y );
13    printf( " %d\n\n", y );
14
15    y = printf( "This line has 28 characters\n" );
16    printf( "%d characters were printed\n\n", y );
17
18    printf( "Printing a %% in a format control string\n" );
19
20    return 0;
21 }
```

The value of ptr is 0065FDF0
The address of x is 0065FDF0

Total characters printed on this line is: 41

This line has 28 characters
28 characters were printed

Printing a % in a format control string

Tùy chỉnh độ rộng

- Độ rộng
 - Độ rộng của trường dữ liệu
 - Nếu độ rộng lớn hơn dữ liệu, mặc định căn lề trái
 - Dấu - chiếm một vị trí trong độ rộng
 - Độ rộng của số nguyên được đặt giữa % và kí tự đặc tả
 - %4d - độ rộng 4

Tùy chỉnh độ chính xác

- Độ chính xác
 - Ý nghĩa phụ thuộc vào kiểu dữ liệu
 - Số nguyên (mặc định 1) - số chữ số tối thiểu
 - Nếu giá trị quá nhỏ, thêm tiền tố 0
 - Số thực dấu phẩy động - số chữ số phần thập phân (e and f)
 - Với g - số lượng tối đa chữ số có ý nghĩa
 - Chuỗi - số lượng tối đa kí tự được in ra của chuỗi

Tùy chỉnh độ chính xác (tiếp)

- Định dạng
 - Độ chính xác: dùng dấu chấm (.) theo sau là độ chính xác
 $\% .3f$
 - Có thể kết hợp với độ rộng
 $\%5.3f$
 - Có thể sử dụng biểu thức số nguyên cho độ chính xác và độ rộng
 - Sử dụng *
 - Độ rộng âm - căn lề trái
 - Độ rộng dương - căn lề phải
 - Độ chính xác phải là số dương

Example 5

```
1 #include <stdio.h>
2
3 int main()
4 {
5     int i = 873;
6     double f = 123.94536;
7     char s[] = "Happy Birthday";
8
9     printf( "Using precision for integers\n" );
10    printf( "\t%.4d\n\t%.9d\n\n", i, i );
11    printf( "Using precision for floating-point numbers\n" );
12    printf( "\t%.3f\n\t%.3e\n\t%.3g\n\n", f, f, f );
13    printf( "Using precision for strings\n" );
14    printf( "\t%.11s\n", s );
15
16    return 0;
17 }
```

Using precision for integers

0873

000000873

Using precision for floating-point numbers

123.945

1.239e+02

124

Using precision for strings

Happy Birth

Sử dụng cờ trong chuỗi điều khiển định dạng của hàm `printf`

- Cờ
 - Bổ sung định dạng
 - Đặt cờ ngay sau kí tự %
 - Có thể kết hợp đồng thời nhiều cờ

Flag	Description
- (minus sign)	Left-justify the output within the specified field.
+ (plus sign)	Display a plus sign preceding positive values and a minus sign preceding negative values.
<i>space</i>	Print a space before a positive value not printed with the + flag.
#	Prefix 0 to the output value when used with the octal conversion specifier o.
	Prefix 0x or 0X to the output value when used with the hexadecimal conversion specifiers x or X.
	Force a decimal point for a floating-point number printed with e, E, f, g or G that does not contain a fractional part. (Normally the decimal point is only printed if a digit follows it.) For g and G specifiers, trailing zeros are not eliminated.
0 (zero)	Pad a field with leading zeros.

Ví dụ 6

```
1#include <stdio.h>
2
3 int main()
4 {
5     printf( "%10s%10d%10c%10f\n\n", "hello", 7, 'a', 1.23 );
6     printf( "%-10s%-10d%-10c%-10f\n", "hello", 7, 'a', 1.23 );
7     return 0;
8}
```

```
hello          7          a  1.230000
hello         7          a      1.230000
```

Ví dụ 7

```
1 #include <stdio.h>
2
3 int main()
4 {
5     int c = 1427;
6     double p = 1427.0;
7
8     printf( "%#o\n", c );
9     printf( "%#x\n", c );
10    printf( "%#X\n", c );
11    printf( "\n%g\n", p );
12    printf( "%#g\n", p );
13
14    return 0;
15 }
```

02623

0x593

0X593

1427

1427.00

In ký tự

- In kí tự
 - Có thể in phần lớn kí tự
 - Các kí tự đặc biệt phải kèm theo dấu \ phía trước

In ký tự (tiếp)

Escape sequence	Description
\'	Output the single quote (') character.
\"	Output the double quote (") character.
\?	Output the question mark (?) character.
\\	Output the backslash (\) character.
\a	Cause an audible (bell) or visual alert.
\b	Move the cursor back one position on the current line.
\f	Move the cursor to the start of the next logical page.
\n	Move the cursor to the beginning of the next line.
\r	Move the cursor to the beginning of the current line.
\t	Move the cursor to the next horizontal tab position.
\v	Move the cursor to the next vertical tab position.

Bài tập 3.1

- Viết chương trình in ra kích thước của các kiểu dữ liệu cơ sở: int, long short, double, char...
- Sử dụng hàm **sizeof**
- Vd: sizeof(int);

Lời giải

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

```
main()
```

```
{
```

```
    printf("    THE SIZE OF BASIC DATA TYPES\n\n");
```

```
    printf("int  %d\n",sizeof(int));
```

```
    printf("short int  %d\n",sizeof(short int));
```

```
    printf("long int  %d\n",sizeof(long int));
```

```
    printf("unsigned int  %d\n",sizeof(unsigned int));
```

```
    printf("unsigned short  %d\n",sizeof(unsigned short));
```

```
    printf("unsigned long  %d\n",sizeof(unsigned long));
```

```
}
```


Bài tập 3.2

- Viết chương trình sau. Biên dịch, liên kết và chạy.

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

```
void main ()
```

```
{
```

```
    int year;
```

```
    float height;
```

```
    year = 21;
```

```
    height = 1.77;
```

```
    printf("Ali is %d years old and %f meter height\n", year, height);
```

Bài tập 3.3

- Viết chương trình hỏi tên người dùng và chào hỏi.
- Sử dụng hàm scanf() để đọc dữ liệu có kiểu cho trước
- Vd:
char word[20];
scanf("%19s", word);

Lời giải

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

```
int main(void) {
```

```
    char name[16]; /* string to hold name */
```

```
    printf("What's your name? ");
```

```
    scanf("%15s", name);
```

```
    printf("Hi there, %s!\n", name);
```

```
    return 0;
```

```
}
```

Bài tập 3.4

- Viết chương trình nhận hai số nguyên m và n và in ra tổng của hai số
- VD:
Enter first number: 3
Enter second number: 5
Sum $3+5 = 8$

Lời giải

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

```
int main(void) {  
    int n, m; /* lower and upper bounds */  
    int sum; /* sum */  
  
    /*  
     * Get the numbers  
     */  
    printf("Enter first number: ");  
    scanf("%d", &n);  
    printf("Enter second number: ");  
    scanf("%d", &m);
```

Lời giải

```
/*  
 * Compute sum of n and m  
 * (also, display inputs for user to check)  
 */  
sum = n+m;  
  
/*  
 * Print results  
 */  
printf("Sum of %d and %d = %d\n", n, m, sum);  
  
return 0;
```

Bài tập 3.5

- Viết chương trình nhận hai số nguyên m và n ($m < n$) và in ra tổng các số trong $[m, n]$
- VD:
Enter first number: 3
Enter second number: 5
Accumulated sum = 12



25 YEARS ANNIVERSARY
SOICT

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

**Thank you
for your
attentions!**



soict.hust.edu.vn/



fb.com/groups/soict

