



25 YEARS ANNIVERSARY
SOICT

HA NOI UNIVERSITY OF SCIENCE AND TECHNOLOGY
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY



HA NOI UNIVERSITY OF SCIENCE AND TECHNOLOGY
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

Tuần 3. Lệnh xuất dữ liệu `printf`

Nội dung

- Xuất dữ liệu
 - ²Hiển thị dữ liệu ra màn hình
 - **printf**
 - Streams
 - puts, putchar (in <stdio.h>)
 - Bài tập

Nhập/Xuất trong C

- C không có sẵn các câu lệnh để nhập/xuất thông tin.
- Thư viện chứa các hàm để thực hiện các thao tác.
- Các hàm của thư viện I/O được liệt kê trong file “header” `<stdio.h>`.
- Không cần phải nhớ các hàm này, chỉ cần làm quen với chúng.

Các luồng vào ra (Streams)

Một luồng đọc dữ liệu



Một luồng ghi dữ liệu



Các luồng vào ra (Streams)

- Streams

- Chuỗi các ký tự được tổ chức thành các dòng (line)
 - Kết thúc bằng ký tự xuống dòng mới
 - Trong chuẩn ANSI C, hệ thống phải hỗ trợ các dòng có độ dài ít nhất 254 ký tự
- Thực hiện mọi việc vào ra chính
- Có thể được định hướng lại
 - Luồng nhập tiêu chuẩn - keyboard
 - Luồng ra tiêu chuẩn - screen
 - Luồng báo lỗi tiêu chuẩn - screen

Xác lập khuôn dạng xuất thông tin với `printf`

- `printf`

- Thiết lập xuất thông tin chuẩn xác

- Các đặc tả: cờ, độ rộng các trường, độ chính xác.
 - Có thể thực hiện các phép làm tròn, xếp cột, căn trái/phải, thêm các ký tự, dạng mũ, dạng hệ 16 và độ rộng, độ chính xác cố định

- Khuôn dạng câu lệnh

`printf`(xâu-điều-khiển-khuôn-dạng, các-tham-số-khác) ;

- xâu-điều-khiển-khuôn-dạng : bao gồm danh sách các kiểu dữ liệu của các biến sẽ được xuất và có thể là văn bản và các ký tự điều khiển.
 - Các tham số khác: tương ứng với từng đặc tả chuyển đổi trong xâu-điều-khiển-khuôn-dạng
 - Mỗi đặc tả bắt đầu với một dấu phân tách, kết thúc với đặc tả cụ thể

In các số nguyên (Integer)

- Integer

- Tất cả các số (không gồm dấu thập phân): **25, 0, -9**
- Số dương, số âm hoặc số 0
- Chỉ có dấu “-” (của số âm) được mặc định in ra

Chỉ định	
d	Hiển thị số nguyên thập phân có dấu
i	Hiển thị số nguyên thập phân có dấu (chỉ định i và d khác nhau khi dùng với hàm scanf)
o	Hiển thị số nguyên hệ 8 không dấu
u	Hiển thị số nguyên hệ 10 không dấu
X hoặc x	Hiển thị số nguyên hệ 16 không dấu. X hiển thị các chữ cái A-F và x hiển thị chữ cái a-f
h hoặc l	Đứng trước các chỉ định để chỉ ra số nguyên short hay long sẽ được hiển thị.

Example 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", 2000000000 );
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
2000000000
707
455
65081
1c7
1C7
```

In các số thực dấu phẩy động

- **số thực dấu phẩy động**
- Có dấu thập phân (Ví dụ: **33.5**)
 - Định nghĩa của mũ (phiên bản máy tính của định nghĩa khoa học)
 - **150.3** là **1.503 $\times 10^2$** trong khoa học
 - **150.3** là **1.503E+02** trong biểu diễn với số mũ
 - dùng **e** hoặc **E**
 - **f** – in số thực với ít nhất 1 chữ số bên trái hoặc phải của dấu thập phân
 - **g** (hoặc **G**) - in **f** hoặc **e(E)** mà không có các số 0 đằng sau (**1.2300** trở thành **1.23**)
 - Sử dụng biểu diễn mũ nếu số mũ nhỏ hơn **-4**, hoặc lớn hơn hoặc bằng độ chính xác (mặc định là 6 chữ số)

Example 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 Xâu và các Ký tự

- **c**

- In tham số kiểu ký tự **char**
- Không thể sử dụng để in ra ký tự đầu tiên của một xâu

- **s**

- Cần khai báo như một tham số kiểu con trỏ (**pointer**) trỏ tới kiểu **char** làm tham số
- In ra các ký tự cho tới khi gặp **NULL** (' \0 ')
- Không thể in tham số kiểu **char**

- Lưu ý

- Dấu nháy đơn dùng cho các hằng ký tự (' z ')
- Nháy kép cho các xâu "z" (về bản chất sẽ bao gồm 2 ký tự: ' z ' và ' \0 ')

Example 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
8 string";
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 chỉ thị biểu diễn khác

- **p**
 - Hiển thị giá trị của con trỏ (địa chỉ)
- **n**
 - Chứa số các ký tự đã xuất ra bởi câu lệnh **printf** hiện tại
 - Cần dùng một con trỏ trỏ tới số nguyên như là tham số để chứa giá trị này
 - **%n** Không in ra giá trị gì
 - Tất cả lời gọi **printf** đều trả về một giá trị
 - Số ký tự đã xuất ra
 - Giá trị âm nếu có lỗi xảy ra
- **%**
 - In ra dấu %
 - **%%**

Example 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     return 0;
20
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

In ra với độ rộng và độ chính xác

- Độ rộng trường
 - Kích thước của trường mà dữ liệu được in ra
 - Nếu độ rộng lớn hơn dữ liệu, mặc định căn theo bên phải
 - Nếu độ rộng trường quá nhỏ, tăng để khớp với dữ liệu
 - Dấu “-” chiếm một vị trí ký tự trong trường
 - Độ rộng số nguyên được chèn vào giữa dấu % và chỉ thị in
 - **%4d** – in số nguyên với độ rộng trường là 4

In ra với độ rộng và độ chính xác(II)

- Độ chính xác
 - Ý nghĩa thay đổi tùy theo kiểu dữ liệu
 - Số nguyên (mặc định độ c/x là 1) – số chữ số nhỏ nhất được in ra
 - Nếu dữ liệu quá nhỏ, thêm vào các chữ số 0
 - Số thực – số chữ số xuất hiện sau dấu thập phân (**e** và **f**)
 - Với **g** – số tối đa các chữ số có ý nghĩa
 - Xâu – số tối đa các ký tự được viết ra từ xâu

In ra với độ rộng và độ chính xác(III)

- Khuôn dạng
 - Độ chính xác: dùng một dấu chấm (.) sau đó là độ chính xác sau dấu %
Ví dụ: %.3f
 - Có thể kết hợp với trường độ rộng
%5.3f
 - Có thể sử dụng biểu thức số nguyên để xác định độ rộng và độ chính xác
 - Dùng dấu *
 - Độ rộng trường là số âm: căn theo bên trái
 - Độ rộng trường là số dương: căn theo bên phải
 - Độ chính xác phải là số dương

```
printf( "%*.*f", 7, 2, 98.736 );
```

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 các cờ trong chuỗi-điều khiển-khuôn-dạng `printf`

- Các cờ (Flags)
 - Hỗ trợ khả năng định dạng
 - Đặt các cờ ngay bên phải của dấu %
 - Có thể kết hợp 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 <code>o</code> .
	Prefix 0x or 0X to the output value when used with the hexadecimal conversion specifiers <code>x</code> or <code>X</code> .
	Force a decimal point for a floating-point number printed with <code>e</code> , <code>E</code> , <code>f</code> , <code>g</code> or <code>G</code> that does not contain a fractional part. (Normally the decimal point is only printed if a digit follows it.) For <code>g</code> and <code>G</code> specifiers, trailing zeros are not eliminated.
0 (zero)	Pad a field with leading zeros.

Example 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
```

Example 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 các chữ cái và ký tự thoát xâu

- In các chữ cái
 - Hầu hết các ký tự đều có thể được in ra
 - Một số ký tự "có vấn đề" như là dấu nháy kép "
 - Phải được biểu diễn bởi ký tự thoát xâu
 - Biểu diễn bởi dấu \ và một ký tự thoát xâu

Printing Literals and Escape Sequences (II)

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.

Exercises 3.1

- Viết một chương trình hiển thị kích thước của các kiểu dữ liệu cơ bản như : int, long short, double, char...
- Có thể dùng hàm **sizeof**.
- Ví dụ: sizeof(int);

Solution

```
#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));
```

```
}
```

Exercises 3.2

- Gõ lại chương trình sau, biên dịch, 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);
```

```
}
```

Exercises 3.3

- Viết một chương trình hỏi tên của bạn và sau đó chào bạn
- Có thể dùng hàm scanf() để đọc dữ liệu theo kiểu chỉ định từ bàn phím
- E.g:
char word[20];
scanf("%19s", word);

Solution

```
#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;  
}
```

Exercises 3.4

- Viết chương trình đọc vào 2 số nguyên n và m và in ra tổng giá trị của n và m.
- The program should look like this when it's working:

Enter first number: 3

Enter second number: 5

Sum $3+5 = 8$

Solution

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

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

Solution

```
/*
```

```
 * 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;
```

```
}
```


Exercise 3.5

- The BK library™ DVD shop has three rental rates

Type of rent	Rent per disk
--------------	---------------

Overnight	\$7.00
-----------	--------

Three-day	\$5.00
-----------	--------

Weekly	\$3.00
--------	--------

- Write a simple C program to input the day of the week, and the number of overnight, three-day and weekly DVDs the customer is renting. Compile this program, and print out the input values to ensure that they are read correctly.
- Update your program to compute the total cost of renting the DVDs

Hint

- Note: since the day of the week is indicated by a single character, you will need to define a set of characters, e.g., ‘m’ for Monday, ‘t’ for Tuesday, and ‘h’ for Thursday.
- When reading a single character, use scanf(“%c”,&day) to skip leading blanks.



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





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