Ngôn ngữ lập trình C

B3: Các hàm vào-ra chuẩn

Khoa Công nghệ thông tin

Chủ đề chính

- Vào-ra trong C
  - Trình bày kết quả
    - *printf*
    - *puts*
  - Nhập số liệu
    - *scanf*
    - *gets, getch*
  - Bài tập thực hành

Vào-Ra trong C

- C không có câu lệnh dựng sẵn cho các thao tác vào-ra.
- Một thư viện các hàm được cung cấp để thực hiện các hoạt động này. Để sử dụng Thư viện các hàm Vào-ra ta cần khai báo tệp tiêu đề <stdio.h>.
- Các bạn không cần phải ghi nhớ chúng, chỉ cần làm quen với chúng.

Hàm `printf`

- **Mục đích:**
  - Hiển thị ra màn hình các loại dữ liệu cơ bản như: Số, kí tự và xâu kí tự;
  - Một số hiệu ứng hiển thị đặc biệt như xuống dòng, sang trang, tab, …
- **Cú pháp:**
  - `printf(xâu_định_dạng,[danh_sách_tham_số]);`
  - `xâu_định_dạng`: là xâu dùng để qui định cách thức hiển thị dữ liệu ra màn hình máy tính.
  - `danh_sách_tham_số`: danh sách các biến sẽ được hiển thị giá trị lên màn hình theo cách thức được qui định trong xâu_định_dạng.

Hàm `printf`

- **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'), tab ('\t'), …

Hàm `printf`

- Chương trình ví dụ 3.1:

```
#include <stdio.h>
void main()
{
  int a = 5;
  float x = 1.234;
  printf("Hien thi mot so nguyen %d và mot so thuc %f",a,x);
}
```

- Kết quả:

```
Hien thi mot so nguyen 5 và mot so thuc 1.234000
```

Hàm `printf`

- **Trong ví dụ trên:**
  - "Hien thi mot so nguyen %d và mot so thuc %f" là `xâu_định_dạng`
  - `a,x` là `danh_sách_tham_số`
  - `%d` dùng để báo cho máy biết rằng cần phải hiển thị tham số kiểu nguyên (biến a)
  - `%f` dùng để báo cho máy cần hiển thị tham số tương ứng (biến x) theo định dạng số thực.

Hàm `printf`

- Nhóm kí tự định dạng thứ k trong `xâu_định_dạng` dùng để xác định quy cách hiển thị tham số thứ k trong `danh_sách_tham_số`.
- Số lượng tham số trong `Danh_sách_tham_số` bằng số lượng nhóm các kí tự định dạng trong `xâu_định_dạng`. Trong ví dụ trên là 2.
- Mỗi nhóm kí tự định dạng chỉ dùng cho một kiểu dữ liệu. Ví dụ:
  - `%d` dùng cho kiểu số nguyên;
  - `%f` dùng cho kiểu số 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.

Bài tập 3.1

- Sử dụng lại ví dụ ở trên nhưng thay đổi vị trí của %d và %f. Biên dịch và chạy chương trình thu kết quả và so sánh với kết quả trước khi thay đổi.

Hàm `printf`

- Một số nhóm kí tự định dạng phổ biến:

Nhóm kí tự định dạng	Áp dụng cho kiểu dữ liệu	Ghi chú
`%d`	**int, long, char**	Hiển thị tham số tương ứng dưới dạng số nguyên có dấu hệ đếm thập phân
`%i`	**int, long, char**	Hiển thị tham số tương ứng dưới dạng số nguyên có dấu hệ đếm thập phân
`%o`	**int, long, char**	Hiển thị tham số tương ứng dưới dạng số nguyên không dấu trong hệ đếm cơ số 8.

- i và d là tương tự nhau và chỉ khác nhau khi sử dụng hàm `scanf`

Hàm `printf`

- Một số nhóm kí tự định dạng phổ biến:

`%u`	**int, long, char**	Hiển thị tham số tương ứng dưới dạng số nguyên không dấu.
`%x`	**int, long, char**	Hiển thị tham số tương ứng dưới dạng số nguyên hệ đếm 16 (không có 0x đứng trước), sử dụng các chữ cái a b c d e f
`%X`	**int, long, char**	Hiển thị tham số tương ứng dưới dạng số nguyên hệ đếm 16 (không có 0x đứng trước), sử dụng các chữ cái A B C D E F
`%e`	**float, double**	Hiển thị tham số tương ứng dưới dạng số thực dấu phẩy động
`%f`	**float, double**	Hiển thị tham số tương ứng dưới dạng số thực dấu phẩy tĩnh

Hàm `printf`

- Một số nhóm kí tự định dạng phổ biến:

`%g`	**float, double**	Hiển thị tham số tương ứng số thực dưới dạng ngắn gọn hơn trong 2 dạng dấu phẩy tĩnh và dấu phẩy động
`%c`	**int, long, char**	Hiển thị tham số tương ứng dưới dạng kí tự
`%s`	**char *** (xâu kí tự)	Hiển thị tham số tương ứng dưới dạng xâu kí tự

- Thêm ký tự 'h' hoặc 'l': ngụ ý là các biến số ở dạng `short` hoặc `long` (giải thích kỹ hơn ở buổi sau).

Ví dụ 3.2

```
1 #include <stdio.h>
2
3 void main()
4 {
5         printf( "%d\n", 455 );
6         printf( "%i\n", 455 );/*i giong nhu 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    }
```

```
455
455
455
-455
32000
2000000000
707
455
65081
1c7
1C7
```

Ví dụ 3.3

```
1 #include <stdio.h>
2
3 void 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     }
```

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

Ví dụ 3.4

```
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
```

Ví dụ 3.5

```
1  #include <stdio.h>
2
3  void 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 }
```

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

Một số định dạng đặc biệt

- **`p`**
  - Hiển thị giá trị con trỏ (địa chỉ)
- **`n`**
  - Lưu số ký tự đã xuất ra bởi lệnh **`printf`** sử dụng
  - Cần một con trỏ số nguyên tương ứng làm tham số
  - Không có gì được in ra bởi %n
  - Với mỗi câu lệnh **`printf`** được gọi có %n sẽ trả về:
    - Số ký tự đã được hàm `printf` in ra;
    - Số âm nếu có lỗi.
- **`%`**
  - In ra ký tự '`%`' bằng `%%`

Ví dụ 3.6

```
1#include <stdio.h>
2
3void 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}
```

```
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
```

Bổ sung về định dạng

- C cho phép đưa thêm một số thuộc tính định dạng dữ liệu khác vào trong xâu định dạng như:
  - Độ rộng để hiển thị (độ rộng tối thiểu);
  - Căn lề trái;
  - Căn lề phải.

Bổ sung về định dạng

- Độ rộng để hiển thị:
  - Đối với số nguyên hoặc ký tự hoặc xâu ký tự: có dạng %md, với m là số nguyên không âm;
  - Đối với số thực: m, n là 2 số nguyên không âm: %m.nf ngụ ý rằng cần dành:
    - m vị trí để hiển thị số thực;
    - n vị trí trong m vị trí đó để hiển thị phần thập phân.
  - Khi số chỗ cần để hiển thị nội dung dữ liệu lớn hơn trong định dạng: tự động cung cấp thêm chỗ mới để hiển thị chứ không cắt bớt nội dung của dữ liệu.

Ví dụ 3.7

```
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
```

Bổ sung về định dạng

- Căn lề phải, căn 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 %.

Ví dụ 3.8

```
1 #include <stdio.h>
2
3 void 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 }
```

```
hello         7         a  1.230000

hello     7         a         1.230000
```

Bài tập 3.2

- Viết chương trình hiển thị được các con số như sau:
  - 1.034
  - 1.03400
  - 1.034

In ra một số ký tự đặc biệt

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 3.3

- Viết chương trình in ra màn hình các ký tự:
  - /
  - "
  - `

Bài tập 3.4

- Viết chương trình in ra kích thước của các dạng dữ liệu cơ bản như: int, long short, double, char, …
- Sử dụng hàm **sizeof** để lấy kích thước của dữ liệu, ví dụ:

```
sizeof(int);
```

Đáp án

```
#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 3.5

- Viết chương trình in ra 4 ký tự ‘B’, ‘I’, ‘D’, ‘V’ dưới dạng ký tự và dạng số nguyên.

- Nhớ những giá trị số nguyên của các ký tự BIDV vừa in ra và viết một chương trình dùng các số nguyên đó nhưng sử dụng định dạng %c để in được chữ BIDV

Đáp án

```
#include <stdio.h>
void main(){
    printf("%c %c %c %c", 'B', 'I', 'D',
    'V');
    printf("%d %d %d %d", 'B', 'I', 'D',
    'V');
}
```

Đáp án

Giả sử số nguyên nhận được của:

- B là xb; I là xi; D là xd, V là xv.

```
#include <stdio.h>
void main(){
    printf("%c %c %c %c", xb, xi, xd, xv);
}
```

Hàm `scanf`

- **Mục đích:**
  - dùng để nhập dữ liệu từ bàn phím.
- **Cú pháp:**
  - `scanf(xâu_định_dạng,[danh_sách_địa_chỉ]);`
  - Địa chỉ của một biến được viết bằng cách đặt dấu & trước tên biến. Ví dụ:
    - các biến có tên là `a, x, ten_bien` thì địa chỉ của chúng lần lượt sẽ là: `&a, &x, &ten_bien;`
    - `scanf("%d%f", &a, &b);`

Hàm `scanf`

- **Xâu_định_dạng:**
  - Tương tự như quy định đã trình bày với hàm `printf`.
- **Danh_sách_địa_chỉ:**
  - Bao gồm các địa chỉ của các biến, các địa chỉ này được phân tách nhau bởi dấu phẩy (,)
  - 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

Ví dụ 3.9

```
#include <conio.h>
#include <stdio.h>
void main()
{
 // khai bao bien
 int a;
 float x;
 char ch;
 char* str;
 // Nhap du lieu
 printf(“Nhap vao mot so nguyen”);
 scanf(“%d”,&a);
 printf(“\n Nhap vao mot so thuc”);
 scanf(“%f”,&x);
 printf(“\n Nhap vao mot ki tu”);
 fflush(stdin); scanf(“%c”,&ch);
 printf(“\n Nhap vao mot xau ki tu”);
 fflush(stdin); scanf(“%s”,str);
 // 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);
 printf(“\n Xau ki tu: %s”,str);
}
```

Kết quả

```
Nhap vao mot so nguyen: 2007
Nhap vao mot so thuc: 18.1625
Nhap vao mot ki tu: b
Nhap vao mot xau ki tu: ngon ngu lap trinh C
Nhung du lieu vua nhap vao
So nguyen: 2007
So thuc: 18.16
Ki tu: b
Xau ki tu: ngon
```

Hàm `scanf`
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 (spacebar) thì `scanf()` sẽ hiểu là kết thúc nhập dữ liệu cho một số.

Hàm `scanf`
Một số quy tắc cần lưu ý

- **Quy tắc 2:** Khi đọc ký 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.

Hàm `scanf`
Một số quy tắc cần lưu ý

- **Quy tắc 3:** Khi đọc xâu kí tự
  - Hàm scanf() nếu gặp các kí tự dấu trắng, dấu tab hay dấu xuống dòng thì nó sẽ hiểu là kết thúc nhập dữ liệu cho một xâu kí tự.
  - Trước khi nhập dữ liệu kí tự hay xâu kí tự ta nên dùng lệnh `fflush(stdin)` để xóa bộ đệm.

Bài tập 3.6

- Viết chương trình hỏi tên tên của bạn và sau đó in ra lời chào đến bạn.
  - `Nhap vao ten cua minh: Ng_Van_A`
  - `Nhap vao lop cua ban: CNTT_VJ_K13`
  - `Chao ban Ng_Van_A lop CNTT_VJ_K13`

Đáp án

```
#include <stdio.h>

void main() {
   char name[16];
   printf(“Ten ban la gi? ");
   scanf("%15s", name);
   printf(“Chao ban, %s!\n", name);
}
```

Bài tập 3.7

- Viết chương trình nhập vào hai số từ bàn phím, sau đó tính tổng và in kết quả ra màn hình:

```
Nhap vao so thu nhat: 3
Nhap vao so thu hai: 5
Tong hai so 3+5 = 8
```

Đáp án

```
#include <stdio.h>

void main(void) {
    int n, m;
    int sum;

    printf(“Nhap vao so thu nhat: ");
    scanf("%d", &n);
    printf(“Nhap vao so thu hai: ");
    scanf("%d", &m);
    sum = n+m;
    printf(“Tong hai so  %d + %d = %d\n", n, m, sum);

}
```

Các lệnh vào ra khác
gets()

- Dùng để nhập vào từ bàn phím một xâu kí tự bao gồm cả dấu cách, điều mà hàm **scanf()** không làm được.
- Cú pháp :

```
gets (xâu_kí_tự);
```

- Ví dụ:

```
char* str;
puts("Nhap vao mot xau ki tu:");
fflush(stdin); gets(str);
```

Các lệnh vào ra khác
puts()

- Hiển thị ra màn hình nội dung xâu_kí_tự và sau đó đưa con trỏ xuống dòng mới.
- Cú pháp :

puts(xâu_kí_tự);

- Ví dụ: `puts("Nhap vao xau ki tu:");`
- Tương đương với lệnh:

```
printf("%s\n","Nhap vao xau ki tu:");
```

Các lệnh vào ra khác
getch()

- Nhập một ký tự từ bàn phím. Thường dùng để chờ người sử dụng ấn một phím bất kì rồi sẽ kết thúc chương trình.
- Cú pháp :

```
getch();
```

- Ví dụ: `ch = getch();`
- Tương đương với lệnh: `scanf("%c",&ch);`
- Để sử dụng các hàm `gets()`, `puts()`, `getch()` ta cần khai báo tệp tiêu đề `conio.h`.

Ví dụ 3.10

```
#include <conio.h>
#include <stdio.h>
void main()
{
 // khai bao bien
 char* str;
 // Nhap du lieu
 puts("Nhap vao mot xau ki tu:");
 fflush(stdin); gets(str);
 // Hien thi du lieu vua nhap vao
 puts("Xau vua nhap vao: ");
 puts(str);
 puts("An phim bat ki de ket thuc...");
 getch();
}
```

Kết quả

```
Nhap vao mot xau ki tu:
ngon ngu lap trinh C
Xau vua nhap vao:
ngon ngu lap trinh C
An phim bat ki de ket thuc ...
```

Bài tập về nhà

1. Viết chương trình nhập vào một số a bất kỳ và in ra giá trị a^2, a^3, a^4
2. Viết chương trình đọc từ bàn phím 3 số nguyên biểu diễn ngày, tháng, năm và xuất ra màn hình dưới dạng "dd/mm/yyyy".
3. Viết chương trình đọc và 2 số nguyên và in ra kết quả của phép (+), phép trừ (-), phép nhân (*), phép chia (/). Nhận xét kết quả chia 2 số nguyên.
4. Viết chương trình nhập vào bán kính hình cầu, tính và in ra diện tích, thể tích của hình cầu đó. Hướng dẫn: S = 4πR^2 và V = (4/3)πR^3.
5. Nhập vào một số là số giây, đổi số giây này ra giờ phút giây và xuất theo dạng gio:phut:giay, mỗi thành phần có 2 chữ số. Ví dụ 3661 = 01:01:01.

Bài tập về nhà

6. Viết chương trình nhập vào thông tin của 2 sinh viên (tên, quê quán, năm sinh), sau đó in tất cả thông tin của 5 sinh viên này ra màn hình.

```
Nhap thong tin sinh vien 1:
-  Ten: Nguyen Manh Hai
-  Que quan: Hai Duong
-  Tuoi: 18
Nhap thong tin sinh vien 2:
-  Ten: Tran Duc Anh
-  Que quan: Ha Noi
-  Tuoi: 20
Cac sinh vien da nhap vao:
*-----------------------------------------*
|       Ten          |  Que quan  |  Tuoi  |
*-----------------------------------------*
| Nguyen Manh Hai | Hai Duong    | 18      |
  ---------------------------------------
| Tran Duc Anh      | Ha Noi       | 20      |
*-----------------------------------------*
```