



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

# TIN HỌC ĐẠI CƯƠNG

Phần II: LẬP TRÌNH C

# Nội dung chính

- Chương 1: Tổng quan về ngôn ngữ C
- Chương 2: Kiểu dữ liệu và biểu thức trong C
- Chương 3: Vào ra dữ liệu
- Chương 4: Cấu trúc điều khiển
- Chương 5: Mảng, con trỏ và chuỗi ký tự
- Chương 6: Cấu trúc
- Chương 7: Hàm
- Chương 8: Tập dữ liệu

# Chương 3: Vào ra dữ liệu

3.1. Xuất dữ liệu với printf()

3.2. Nhập dữ liệu với scanf()

# Chương 3: Vào ra dữ liệu

3.1. Xuất dữ liệu với printf()

3.2. Nhập dữ liệu với scanf()

# Các hàm vào ra cơ bản

- In dữ liệu:
  - `printf()`
- Nhập dữ liệu
  - `scanf()`
- Cần khai báo tệp tiêu đề:  
`#include <stdio.h>`

# Hàm in dữ liệu (ra màn hình)

- printf()

# Mục đích

- Hiển thị ra màn hình các loại dữ liệu cơ bản  
Số nguyên, số thực, kí tự, chuỗi kí tự
- Tạo một số hiệu ứng hiển thị đặc biệt  
Xuống dòng, sang trang,...
- Cú pháp  
`printf(xau_dinh_dang [, DS_tham_so]);`

# Cú pháp

```
printf(xau_dinh_dang [, DS_tham_so]);
```

- Xau\_dinh\_dang: Là một xâu qui định cách thức hiển thị dữ liệu ra màn hình máy tính.
  - Bao gồm các nhóm kí tự định dạng
  - Nhóm kí tự định dạng thứ k xác định quy cách hiển thị tham số thứ k trong DS\_tham\_số
    - Số lượng tham số trong DS\_tham\_số bằng số lượng nhóm các kí tự định dạng trong xau\_dinh\_dang.
- DS\_tham\_so: Danh sách các biến/biểu thức sẽ được hiển thị giá trị lên màn hình theo cách thức được qui định trong xau\_dinh\_dang.



# Ví dụ

```
#include <stdio.h>
void main() {
    int a = 5;
    float x = 1.234;
    printf("Hien thi mot bieu thuc
nguyen %d và mot so thuc %f", 2 * a,
x);
}
```

**Kết quả:** Hien thi mot bieu thuc nguyen 10 va mot so thuc 1.234000

# Xâu định dạng

- Các kí tự thông thường:
  - Được hiển thị ra màn hình.
- Các kí tự điều khiển:
  - Dùng để tạo các hiệu ứng hiển thị đặc biệt như xuống dòng ('\n').
- Các nhóm kí tự định dạng:
  - Xác định quy cách hiển thị các tham số trong phần danh\_sach\_tham\_so.

# Nhóm ký tự định dạng

- Mỗi nhóm ký tự định dạng chỉ dùng cho một kiểu dữ liệu”

Ví dụ: %d dùng cho kiểu nguyên, %f dùng cho kiểu thực

- DS\_tham\_so phải phù hợp với các nhóm ký tự định dạng trong xau\_dinh\_dang về:
  - Số lượng;
  - Thứ tự;
  - Kiểu dữ liệu;

Nếu không phù hợp sẽ hiển thị ra kết quả không như ý

`printf(" %d ", 3.14);` → -31457

# Các ký tự định dạng

| Ký tự  | Kiểu dữ liệu      | Kết quả                            |
|--------|-------------------|------------------------------------|
| %i, %d | int, char         | Số thập phân                       |
| %o     | int, char         | Số hệ 8<br>(không có 0 đằng trước) |
| %x %X  | int, char         | Số hệ hexa<br>(chữ thường/chữ hoa) |
| %u     | unsigned int/char | Số thập phân                       |

# Các ký tự định dạng

| Ký tự    | Kiểu dữ liệu  | Kết quả                            |
|----------|---------------|------------------------------------|
| %ld, %li | long          | Số thập phân                       |
| %lo      | long          | Số hệ 8<br>(không có 0 đằng trước) |
| %lx, %LX | long          | Số hệ hexa<br>(chữ thường/chữ hoa) |
| %lu      | unsigned long | Số thập phân                       |

Nhận xét: Với kiểu long, thêm ký tự l ngay sau dấu %

# Các ký tự định dạng

| <b>Ký tự</b> | <b>Kiểu dữ liệu</b> | <b>Kết quả</b>                         |
|--------------|---------------------|--|
| %f           | float/double        | Số thực dấu phẩy tĩnh                  |
| %e, %E       | float/double        | Số thực dấu phẩy động                  |
| %c           | int, char           | Kí tự đơn lẻ                           |
| %s           | char []             | Hiển thị chuỗi kí tự kết thúc bởi '\0' |
| %%           |                     | Hiển thị kí tự %                       |

# Độ rộng hiển thị

- Có dạng “%**m**”,
  - m là một giá trị nguyên, không âm.
  - m cho biết số chỗ trống dành cho hiển thị biểu thức tương ứng

- Ví dụ:

int a = 1234;

printf(“%5d”,a) → □ 1234

printf(“%5d”,34)→ □ □ □ 34

□ ký hiệu cho dấu trắng (space)

# Ví dụ

- `printf("\n%3d %15s %3c", 1, "nguyen van a", 'g');`
- `printf("\n%3d %15s %3c", 2, "tran van b", 'k');`

□ □ 1 □ □ □ □ nguyen □ van □ a □ □ □ g

□ □ 2 □ □ □ □ □ tran □ van □ b □ □ □ k



# Độ rộng hiển thị với số thực

- Có dạng “%m.n”,
  - m, n là 2 giá trị nguyên, không âm.
  - m cho biết **kích thước để hiển thị số thực**
  - n cho biết kích thước dành cho phần thập phân, nếu không đủ C sẽ làm tròn khi hiển thị
- Ví dụ:
  - `printf("\n%f", 17.346);` → 17.346000
  - `printf("\n%.2f", 17.346);` → 17.35
  - `printf("\n%.2f", 17.345);` → 17.34 !?
  - `printf("\n%8.2f", 17.346);` → □□□17.35
  - `printf("\n%8.2f", 17.344);` → ???

# Chú ý

- Nếu số chỗ cần để hiển thị dữ liệu lớn hơn được cung cấp trong định dạng  $\Rightarrow$  Tự động cung cấp thêm chỗ mới để hiển thị đầy đủ, không cắt bớt nội dung của dữ liệu.
- Ví dụ:

`printf("%2d", 1234);`  $\rightarrow$  1234

`printf("%6.3f", 123.456);`  $\rightarrow$  123.456

`printf("%12.6e", 123.456);`  $\rightarrow$  1.234560e+02

`printf("%12.3e", 123.456);`  $\rightarrow$  1.235e+02

# Căn lề trái - căn lề phải

%-

- Khi hiển thị dữ liệu có sử dụng tham số độ rộng, để căn lề trái cần thêm dấu trừ - vào ngay sau dấu %:
  - Ngầm định, căn lề phải
- Ví dụ:

```
printf("%-3d%-10s%-5.2f%-3c",5,"Hello",7.5,'g')
```

→5□□Hello□□□□□7.50□g□□

# Chương 3: Vào ra dữ liệu

3.1. Xuất dữ liệu với printf()

3.2. Nhập dữ liệu với scanf()

# Mục đích

- Dùng để nhập dữ liệu từ bàn phím
- Ký tự đơn lẻ
- Chuỗi ký tự
- Số nguyên: hệ 10, 8, 16
- Số thực

Dấu phẩy tĩnh; Dấu phẩy động

- Cú pháp

```
scanf(xau_dinh_dang[,DS_dia_chi]);
```

# Cú pháp

```
scanf(xau_dinh_dang [, DS_dia_chi]);
```

- Xau\_dinh\_dang: Gồm các ký tự được qui định cho từng loại dữ liệu được nhập vào.
  - Ví dụ: dữ liệu định nhập kiểu nguyên thì xâu định dạng là : %d
- DS\_dia\_chi: bao gồm địa chỉ của các biến (toán tử &), phân tách nhau bởi dấu phẩy (,)
- Phải phù hợp với các kí tự định dạng trong xau\_dinh\_dang về số lượng, kiểu, thứ tự

# Hoạt động

- Đọc các ký tự được gõ vào từ bàn phím
- Căn cứ vào xâu định dạng, chuyển thông tin đã nhập sang kiểu dữ liệu phù hợp
- Gán những giá trị vừa nhập vào các biến tương ứng trong DS\_dia\_chi
- Ví dụ:

```
int a;
```

```
scanf("%d",&a); → 1234_ → a = 1234
```

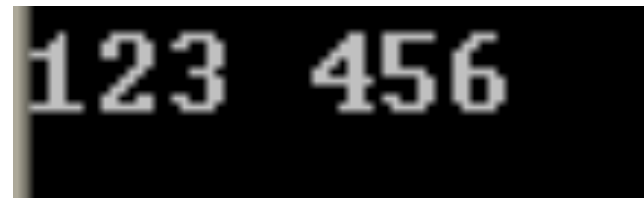
# Ghi chú

- Thông tin được gõ vào từ bàn phím, được lưu ở vùng đệm trước khi được xử lý bởi hàm scanf() → Hàm scanf() đọc từ vùng đệm

```
#include <stdio.h>
int main(){
    int a, b;
    scanf("%d",&a);
    scanf("%d",&b);
    printf ("%d %d", a, b);
    return 0;
}
```



123 456\_



123 456



# Các ký tự định dạng

| <b>Kí tự</b> | <b>Khuôn dạng dữ liệu nhập</b> |
|--------------|--------------------------------|
| <b>%c</b>    | Đọc kí tự đơn lẻ               |
| <b>%d</b>    | Đọc số thập phân               |
| <b>%o</b>    | Đọc số hệ 8                    |
| <b>%x</b>    | Đọc số hệ hexa                 |
| <b>%u</b>    | Đọc số thập phân không dấu     |

# Các ký tự định dạng

| Kí tự | Chú thích                               |
|-------|---|
| %s    | Đọc xâu kí tự tới khi gặp dấu phân cách |
| %f    | Đọc số thực dấu phẩy tĩnh (float)       |
| %ld   | Đọc số nguyên kiểu long                 |
| %lf   | Đọc số thực dấu phẩy tĩnh (double)      |
| %e    | Đọc số thực dấu phẩy động               |
| %%    | Đọc ký tự %                             |

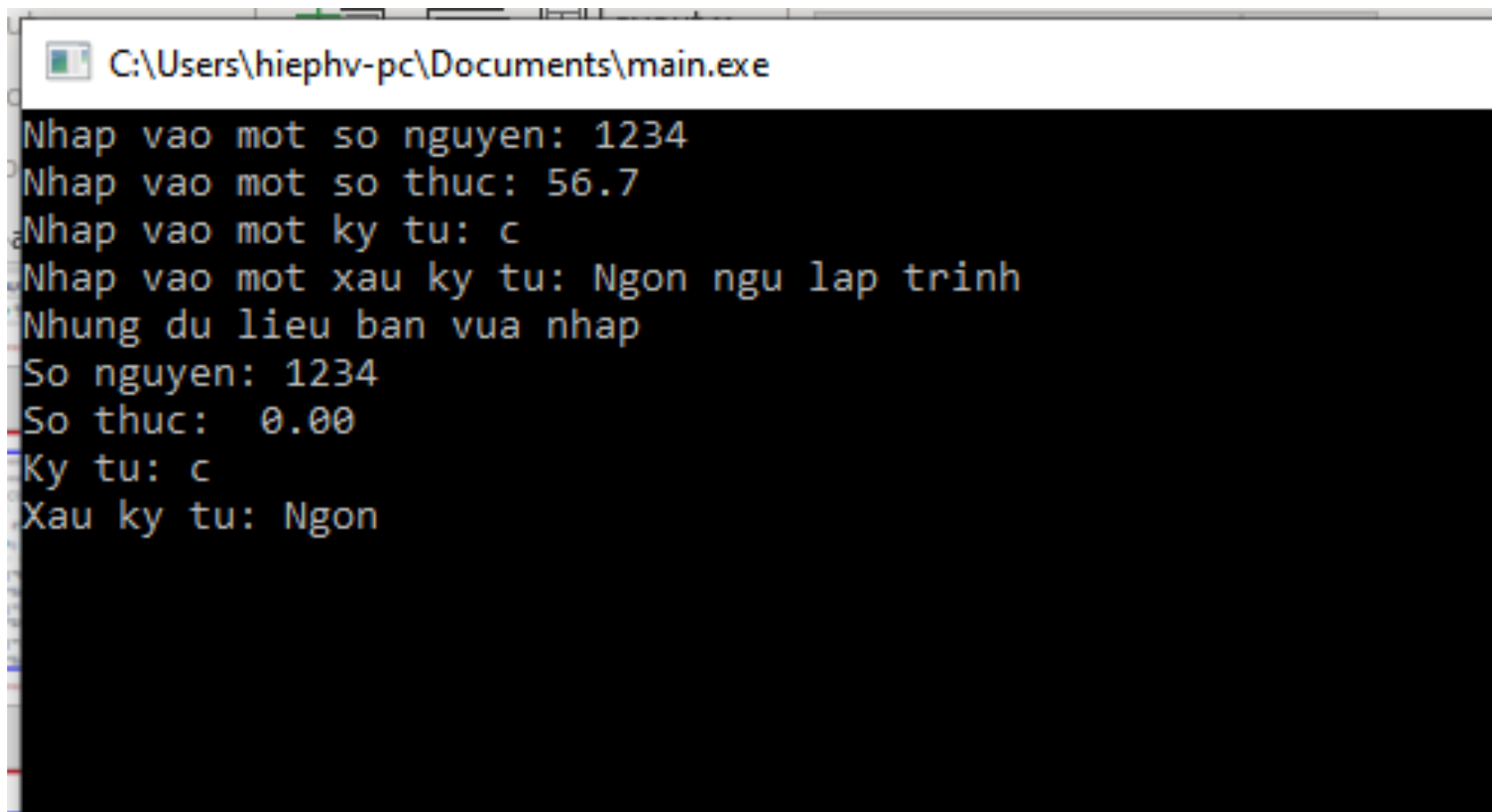
# Ví dụ

```
#include <conio.h>
#include <stdio.h>
void main(){
    // khai bao bien
    int a; float x;
    char ch; char str[30];
    // Nhap du lieu
    printf("Nhap vao mot so nguyen:");
    scanf("%d",&a);
    printf("\nNhap vao mot so thuc:");
    scanf("%f",&x);
    printf("\n Nhap vao mot ki tu:");
    scanf("%c",&ch);
```

# Ví dụ

```
printf("\nNhap vao mot xau ki tu:");  
scanf("%s",str);  
  
// Hien thi du lieu vua nhap vao  
printf("\nNhưng du lieu vua nhap vao");  
printf("\nSố nguyên : %d",a);  
printf("\nSố thực : %5.2f",x);  
printf("\nKý tự : %c",ch);  
printf("\nXâu ký tự : %s",str);  
}
```

# Ví dụ → Kết quả thực hiện



```
C:\Users\hiephv-pc\Documents\main.exe
Nhap vao mot so nguyen: 1234
Nhap vao mot so thuc: 56.7
Nhap vao mot ky tu: c
Nhap vao mot xau ky tu: Ngon ngu lap trinh
Nhưng dữ liệu bạn vừa nhập
Số nguyên: 1234
Số thực: 0.00
Ký tự: c
Xâu ký tự: Ngon
```

# Các quy tắc cần lưu ý

- Khi đọc số
- Hàm scanf() quan niệm rằng mọi kí tự số, dấu chấm (‘.’) đều là kí tự hợp lệ.  
Số thực dấu phẩy động, chấp nhận ký tự e/E
- Khi gặp các dấu phân cách như tab, xuống dòng hay dấu cách (space bar), scanf() sẽ hiểu là kết thúc nhập dữ liệu cho một số

# Các quy tắc cần lưu ý

- Khi đọc kí 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

- 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.

# Ví dụ: Đọc 2 số nguyên, đưa ra tổng, hiệu, tích...

```
#include <stdio.h>

int main(){
    int A, B;
    printf("Nhap vao 2 so nguyen : "); scanf("%d %d",&A,&B);
    printf("\n");
    printf("Tong %d + %d = %d \n", A, B, A + B);
    printf("Hieu %d - %d = %d\n", A, B, A - B);
    printf("Tich %d x %d = %d\n", A, B, A * B);
    printf("Thuong %d / %d = %.3f\n", A, B, (float)A / B);
    printf("Chia nguyen %d / %d = %d\n", A, B, A / B);
    printf("Chia du %d %% %d = %d\n", A, B, A % B);
    printf("\n");
    return 0;
}
```



# Ví dụ: Đọc 2 số nguyên, đưa ra tổng, hiệu, tích...

Nhap vao 2 so nguyen : 17 5

Tong  $17 + 5 = 22$

Hieu  $17 - 5 = 12$

Tich  $17 \times 5 = 85$

Thuong  $17 / 5 = 3.400$

Chia nguyen  $17 / 5 = 3$

Chia du  $17 \% 5 = 2$

# Bài tập

- Viết chương trình nhập vào từ bàn phím chiều dài 3 cạnh của một tam giác, rồi đưa ra diện tích và các đường cao của tam giác
- Nhập vào từ bàn phím tọa độ 3 điểm A, B, C rồi đưa ra độ dài các cạnh của tam giác ABC và của đường trung tuyến AM
- Cho hàm số:  $f(x) = x^7 + 5\sqrt[3]{x^5 + 3x^3 + 2} + 12$   
Viết chương trình nhập vào 3 số thực a,b,c và đưa ra trung bình cộng của f(a), f(b), f(c)
- Nhập x vào từ bàn phím và tính giá trị của biểu thức

$$A = \frac{\cos 3a + \sqrt[5]{2x^3 + x + 1}}{\log_7(3^{x^2} + 2.14b)} \text{ trong đó } a = \sqrt{2^x + \pi} \text{ và } b = \ln(e^{x+1.23} + 1)$$

# Ví dụ: Đọc tọa độ 3 điểm A,B,C và đưa ra diện tích $\Delta ABC$

```
#include <stdio.h>
#include <math.h>
int main(){
    float Ax,Ay, Bx, By, Cx, Cy, AB, BC, CA,p;
    printf("Nhap vao toa do diem A  : "); scanf("%f %f",&Ax,&Ay);
    printf("Nhap vao toa do diem B  : "); scanf("%f %f",&Bx,&By);
    printf("Nhap vao toa do diem C  : "); scanf("%f %f",&Cx,&Cy);
    //Tinh do dai cac canh cua tam giac
    AB = sqrt((Ax-Bx)*(Ax-Bx)+(Ay-By)*(Ay-By));
    BC = sqrt((Bx-Cx)*(Bx-Cx)+(By-Cy)*(By-Cy));
    CA = sqrt((Cx-Ax)*(Cx-Ax)+(Cy-Ay)*(Cy-Ay));
    p = (AB + BC + CA)/2;
    printf("Dien tich tam giac ABC la: %f",sqrt(p*(p-AB)*(p-BC)*(p-CA)));
    printf("\n");
    return 0;
}
```

# Ví dụ: Đọc tọa độ 3 điểm A,B,C và đưa ra d/tích $\Delta ABC$

```
Nhap vao toa do diem A   : 0 0
Nhap vao toa do diem B   : 6 0
Nhap vao toa do diem C   : 0 8
Dien tich tam giac ABC la: 24.0000000
```

```
Nhap vao toa do diem A   : -1 -2
Nhap vao toa do diem B   : 5 -2
Nhap vao toa do diem C   : 6 6
Dien tich tam giac ABC la: 24.0000000
```

```
Nhap vao toa do diem A   : 1 1
Nhap vao toa do diem B   : 2 2
Nhap vao toa do diem C   : 4 4
Dien tich tam giac ABC la: 0.0000000
```

# Bài tập tại lớp

- Viết chương trình nhập vào từ bàn phím bán kính một đường tròn và đưa ra màn hình diện tích và chu vi đường tròn
- Viết chương trình nhập vào từ bàn phím một giá trị thực. Hãy đưa ra diện tích của các hình tròn, vuông, tam giác đều có chu vi bằng giá trị vừa nhập.
- Ghi chú:
  - Giả thiết  $\pi = 3.1416$ . Cần khai báo hằng PI trong chương trình.
  - $\pi$  là hằng số được khai báo trong tệp tiêu đề math.h và có tên là M\_PI