



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

- Xuất dữ liệu ra màn hình (thiết bị ra chuẩn):

`printf()`

- Nhập dữ liệu từ bàn phím (thiết bị vào chuẩn):

`scanf()`

- Cần khai báo tệp tiêu đề:

`#include <stdio.h>`

Hàm in dữ liệu (ra màn hình)

`printf()`

Mục đích

- Hiện thị ra màn hình các loại dữ liệu cơ bản:
 - Số nguyên
 - Số thực
 - Kí tự
 - Xâu kí tự
- Tạo một số hiệu ứng hiển thị đặc biệt:
tab, xuống dòng, sang trang,...

Cú pháp

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

- `xau_dinh_dang`: Là một chuỗi qui định cách thức hiển thị dữ liệu ra màn hình máy tính.
 - Bao gồm các ký tự hiển thị, ký tự điều khiển và các nhóm ký tự định dạng.
 - Nhóm ký tự định dạng thứ `k` xác định quy cách hiển thị tham số thứ `k` trong `DS_tham_so`
 - *Số lượng tham số trong `DS_tham_so` bằng số lượng nhóm các ký tự định dạng trong chuỗi định dạng.*
- `DS_tham_so`: Danh sách các biến/biểu thức sẽ được hiển thị giá trị theo cách thức được qui định trong `xau_dinh_dang`.

Ví dụ

```
#include <stdio.h>
int main()
{
    int a = 5;
    float x = 1.234;
    printf("Hien thi mot bieu thuc nguyen %d
va mot so thuc %f", 2*a, x);
    return 0;
}
```

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'), tab ('\t')
- Các nhóm kí tự định dạng:
 - Xác định quy cách hiển thị các tham số trong phần DS_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 số nguyên, %f dùng cho kiểu số 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 (không có 0 đằng trước)
%x %X	int, char	Số hệ hexa (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 (không có 0 đằng trước)
%lx, %LX	long	Số hệ hexa (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

Ký tự	Kiểu dữ liệu	Kết quả
%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**”, với:
 - 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:

<code>printf("\n%f", 17.346);</code>	→ 17.346000
<code>printf("\n%.2f", 17.346);</code>	→ 17.35
<code>printf("\n%.2f", 17.345);</code>	→ 17.34 ???
<code>printf("\n%8.2f", 17.346);</code>	→ □□□17.35
<code>printf("\n%8.2f", 17.344);</code>	→ ???

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:

<code>printf("%2d", 1234);</code>	\rightarrow 1234
<code>printf("%6.3f", 123.456);</code>	\rightarrow 123.456
<code>printf("%12.6e", 123.456);</code>	\rightarrow 1.234560e+02
<code>printf("%12.3e", 123.456);</code>	\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 quy 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 cấu trúc 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

Kí tự	Khuôn dạng dữ liệu nhập
%c	Đọc kí tự đơn lẻ
%d	Đọc số thập phân
%o	Đọc số hệ 8
%x	Đọc số hệ hexa
%u	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 <stdio.h>
int 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("\nNhap 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);  
return 0;  
}
```

Ví dụ => Kết quả thực hiện

```
C:\Users\lebavui\source\THDC\vd1.exe
Nhap vao mot so nguyen:1234

Nhap vao mot so thuc:56.7

Nhap vao mot ki tu:
Nhap vao mot xau ki tu:Ngon ngu lap trinh

Nhưng du lieu vua nhap vao:
So nguyen: 1234
So thuc: 56.70
Ky tu:

Xau ky tu: Ngon
-----
Process exited after 10.58 seconds with return value 0
Press any key to continue . . .
```

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.

=> Sử dụng lệnh fflush(stdin); để xóa bộ đệm bàn phím trước khi đọc ký tự hoặc xâu ký tự.

Các lệnh vào ra khác

- Hàm `gets()` :

Dùng để nhập vào từ bàn phím một chuỗi ký tự **bao gồm cả dấu cách**, điều mà hàm `scanf()` không làm được.

- Cú pháp :

`gets(biến_xâu_kí_tự);`

- Ví dụ:

```
char str[30];  
printf("Nhap vao mot xau ki tu: ");  
fflush(stdin); gets(str);
```


Các lệnh vào ra khác

- **Hàm 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("Nhập vào xâu kí tự: ");

- Tương đương với:

printf("%s\n", "Nhập vào xâu kí tự: ");

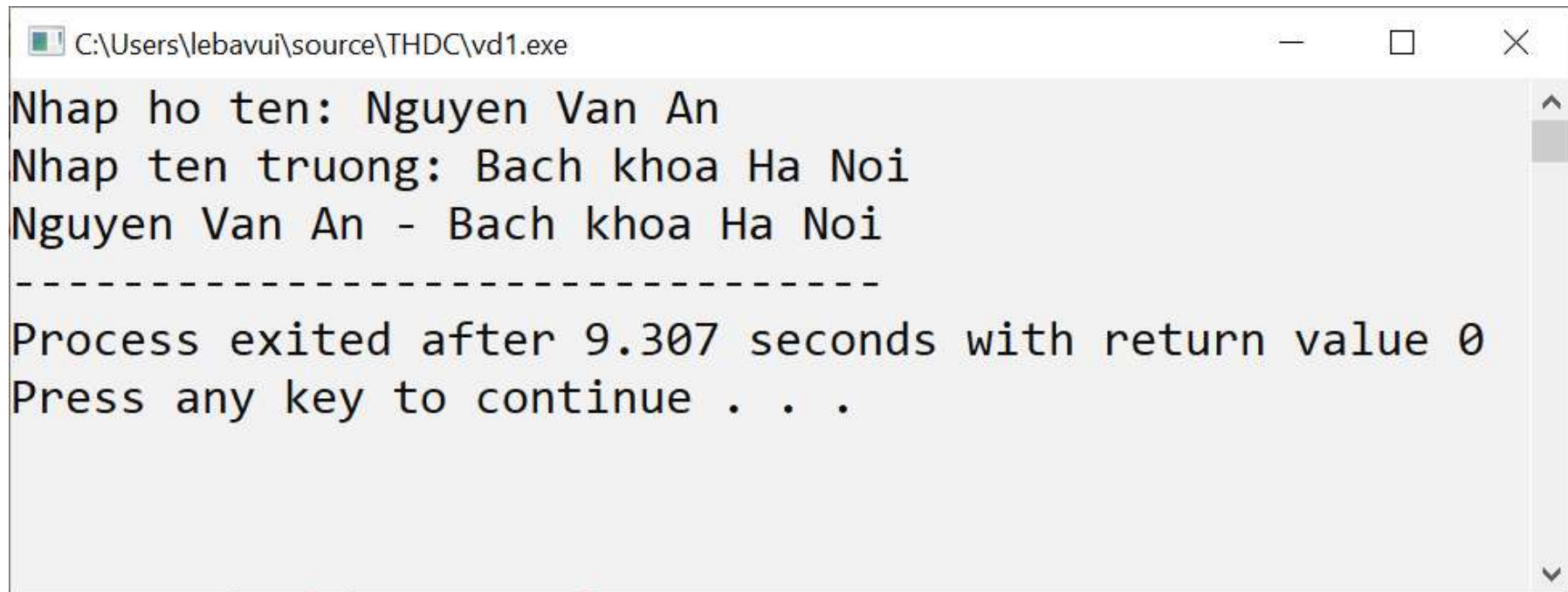
Hoặc

printf("Nhập vào xâu kí tự:\n");

Ví dụ

```
#include <stdio.h>
int main()
{
    char hoten[64], truong[64];
    printf("Nhap ho ten: ");
    gets(hoten);
    printf("Nhap ten truong: ");
    gets(truong);
    printf("%s - %s", hoten, truong);
    return 0;
}
```

Ví dụ => Kết quả thực hiện



A screenshot of a Windows command prompt window. The title bar shows the file path "C:\Users\lebavui\source\THDC\vd1.exe". The window contains the following text: "Nhap ho ten: Nguyen Van An", "Nhap ten truong: Bach khoa Ha Noi", "Nguyen Van An - Bach khoa Ha Noi", a line of dashes "-----", "Process exited after 9.307 seconds with return value 0", and "Press any key to continue . . .".

```
C:\Users\lebavui\source\THDC\vd1.exe
Nhap ho ten: Nguyen Van An
Nhap ten truong: Bach khoa Ha Noi
Nguyen Van An - Bach khoa Ha Noi
-----
Process exited after 9.307 seconds with return value 0
Press any key to continue . . .
```

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);
    return 0;
}
```

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

```
C:\Users\lebvui\source\THDC\vd1.exe
Nhap vao 2 so nguyen : 17 5

Tong 17 + 5 = 22
Hieu 17 - 5 = 12
Tich 17 x 5 = 85
Thuong 17 / 5 = 3.400
Chia nguyen 17 / 5 = 3
Chia du 17 % 5 = 2

-----
Process exited after 2.709 seconds with return value 0
Press any key to continue . . .
```

Ví dụ: Đọc tọa độ 3 điểm A,B,C và đưa ra diện tích ΔABC

```
#include <stdio.h>
#include <math.h>
int main(){
    float Ax, Ay, Bx, By, Cx, Cy, AB, BC, CA;
    float p, s;
    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);
```

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

```
// Tính độ dài các cạnh của tam giác
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));

// Tính diện tích tam giác
p = (AB + BC + CA)/2;
s = sqrt(p*(p-AB)*(p-BC)*(p-CA));
printf("Diện tích tam giác ABC là: %f", s);

return 0;
}
```

Ví dụ: Đọc tọa độ 3 điểm A,B,C và đưa ra d/tích ΔABC

```
C:\Users\lebavui\source\THDC\vd1.exe
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: 24.000000
-----
Process exited after 6.287 seconds with return value 0
Press any key to continue . . .
```


Bài tập tại lớp

1. 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
2. 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.
- π là hằng số được khai báo trong tệp tiêu đề math.h và có tên là M_PI

Bài tập

3. Viết chương trình nhập 2 số a và n . In ra màn hình giá trị bit thứ n của số a .
4. Viết chương trình nhập vào số n có 3 chữ số, in ra màn hình số n theo chiều ngược lại.
5. Viết chương trình nhập vào 1 ký tự, sau đó in ra mã ASCII của ký tự đó.
6. Viết chương trình thực hiện công việc sau:
 - Nhập vào 3 điểm Toán, Lý, Hóa
 - Tính và in ra điểm trung bình 3 môn (làm tròn đến 2 chữ số thập phân)
 - Tìm và in ra điểm lớn nhất (sử dụng biểu thức điều kiện)
 - Tìm và in ra điểm nhỏ nhất (sử dụng biểu thức điều kiện)