



TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN TP.HCM  
KHOA CÔNG NGHỆ THÔNG TIN  
MÔN: **NHẬP MÔN LẬP TRÌNH**

# **HƯỚNG DẪN THỰC HÀNH**

## **TUẦN 01**

### **NHẬP XUẤT DỮ LIỆU**

TP.HCM, ngày 09 tháng 09 năm 2017

## MỤC LỤC

1	Khái niệm nhập xuất dữ liệu: .....	3
2	Các chương trình minh họa .....	3
2.1	Chương trình xuất một dòng chữ .....	3
2.2	Chương trình in ra hai dòng chữ .....	3
2.3	Chương trình tính diện tích hình tròn .....	4
2.4	Chương trình cộng hai số .....	4
2.5	Chương trình in hoa kí tự thường nhập từ bàn phím .....	5
2.6	Chương trình sử dụng thư viện toán học “math.h” .....	5
3	Các minh họa về hàm scanf và printf .....	6
3.1	Hàm printf .....	6
3.2	Hàm scanf .....	9
4	Bài tập .....	10

## 1 Khái niệm nhập xuất dữ liệu:

Đây là các thao tác cơ bản đầu tiên để giải quyết một vấn đề trong thực tế trên máy tính. Thao tác nhập có nghĩa là ta chuyển dữ liệu trong thế giới thực vào máy tính thông qua bộ phận tiếp nhận (ví dụ như máy quét trong siêu thị khi tính tiền một món hàng, hay bàn phím của một thiết bị tính toán nào đó...). Với dữ liệu được chuyển vào trong máy tính (dữ liệu nằm trong RAM, bộ nhớ tạm thời của máy tính) ta sẽ thực hiện tính toán xử lý để ra kết quả mong muốn, sau cùng là xuất kết quả ra bên ngoài thông qua các thiết bị xuất (màn hình, máy in, máy chiếu...).

## 2 Các chương trình minh họa

Phần này trình bày một số chương trình cơ bản bằng ngôn ngữ C minh họa quá trình nhập xuất dữ liệu.

### 2.1 Chương trình xuất một dòng chữ

Dòng	
1	<code>/*Hello.c*/</code>
2	<code>#include &lt;stdio.h&gt;</code>
3	<code>void main()</code>
4	<code>{</code>
5	<code>printf("Hello world!");</code>
6	<code>}</code>

Dòng 1 là comment của chương trình, dòng này chỉ phục vụ mô tả thêm thông tin để ta hiểu thêm về đoạn mã, máy tính hoàn toàn không xử lý các dòng comment. Dòng 2 minh họa việc ta “include” thêm các thư viện ngoài hỗ trợ quá trình chúng ta lập trình, ở đây ta dùng thư viện “stdio.h” của ngôn ngữ C. Dòng 3 là đầu vào của chương trình, đây được gọi là hàm main (nơi bắt đầu chạy chương trình), hàm này tên là main, không có giá trị trả về (kiểu void) và không có tham số. Dòng 4 và 6 là cú pháp bắt đầu và kết thúc một hàm. Dòng 5 ta dùng hàm “printf” trong thư viện “stdio.h” để in ra một chuỗi có giá trị là “Hello world”.

### 2.2 Chương trình in ra hai dòng chữ

Dòng	
1	<code>/*Hello_version_2.c*/</code>
2	<code>#include &lt;stdio.h&gt;</code>
3	<code>void main()</code>
4	<code>{</code>
5	<code>printf("Hello world!");</code>

6	printf("This is a second program");
7	}

Chương trình này hoàn toàn giống như chương trình đầu tiên, tuy nhiên ta sử dụng thêm một hàm “printf” để in thêm một dòng chữ ra màn hình. Như vậy, tùy vào nhu cầu mà ta muốn, ta sẽ linh hoạt sử dụng các hàm để in ra những giá trị mình mong muốn.

## 2.3 Chương trình tính diện tích hình tròn

Dòng	
1	<code>/*Tinh_dien_tich_hinh_tron.c*/</code>
2	<code>#include &lt;stdio.h&gt;</code>
3	<code>#define Pi 3.14159</code>
4	<code>void main()</code>
5	<code>{</code>
6	<code>float R = 1.25;</code>
7	<code>float DienTich;</code>
8	<code>DienTich = Pi*R*R;</code>
9	<code>printf("Hình tròn, bán kính = %f\n", R);</code>
10	<code>printf("Diện tích = %f", DienTich);</code>
11	<code>}</code>

Ở chương trình này, ta thấy có nhiều điểm khác biệt hơn. Không như hai chương trình trước chỉ in ra chuỗi giá trị, ở đây ta cần tính diện tích của hình tròn. Như vậy theo công thức toán học để tính diện tích hình tròn ta cần bán kính R và hằng số Pi. Ở dòng số 3 là cú pháp định nghĩa hằng số Pi (hằng số là một biến số có **giá trị không thể thay đổi**), ta thấy Pi là 3.14159, như vậy trong hàm main, chỗ nào dùng Pi có nghĩa là dùng 3.14159. Ở dòng số 6 ta khai báo một biến tên là R kiểu số thực có giá trị là 1.25 (Lưu ý giá trị này là ta gán, chứ không phải người dùng nhập từ bàn phím). Dòng 7 là biến DienTich nhưng không có giá trị (Trong C nó sẽ có giá trị rác). Dòng 8 là dòng xử lý theo yêu cầu của chương trình, vì đây là chương trình tính diện tích hình tròn, nên ta sẽ dùng công thức tính, sau đó giá trị kết quả được gán lại cho biến DienTich giữ. Dòng 9 và dòng 10 là nơi ta in kết quả ra cho người dùng thấy (Lưu ý: dùng %f tương ứng với số thực).

## 2.4 Chương trình cộng hai số

Dòng	
1	<code>/*Cong_hai_so.c*/</code>
2	<code>#include &lt;stdio.h&gt;</code>
3	<code>void main()</code>
4	<code>{</code>
5	<code>int A, B, Sum;</code>
6	<code>printf("Nhập giá trị của A = ");</code>
7	<code>scanf("%d", &amp;A);</code>

8	printf(“Nhập giá trị của B = ”);
9	scanf(“%d”, &B);
10	Sum = A + B;
11	printf(“%d + %d = %d”, A, B, Sum);
12	}

Trong chương trình này, ta thấy đã minh họa ba thao tác rất cơ bản đó là **nhập** vào hai số, sau đó **xử lý** tính toán tổng và **xuất** ra kết quả. Ta thấy giá trị của A và B không được gán trực tiếp như trong chương trình tính diện tích mà ta đã làm với biến R. Thay vào đó người dùng đã nhập từ bàn phím giá trị họ cần (đĩ nhiên phải là giá trị hợp lệ). Sau đó ở dòng 10 ta tính toán và xuất ra kết quả ở dòng 11.

## 2.5 Chương trình in hoa kí tự thường nhập từ bàn phím

Dòng	
1	/*In_hoa_ky_tu.c*/
2	#include <stdio.h>
3	void main()
4	{
5	char KyTu, KyTuInHoa;
6	printf(“Nhập giá trị của KyTu = ”);
7	scanf(“%c”, &KyTu);
8	KyTuInHoa = KyTu – 32;
9	printf(“Giá trị in hoa của %c là %c ”, KyTu, KyTuInHoa);
10	}

Chương trình này cũng minh họa ba thao tác chính là nhập, xử lý và xuất. Tuy nhiên ở đây minh họa cách sử dụng kiểu dữ liệu char. Trong C thì ta có thể dùng các phép tính toán của số nguyên (kiểu int, short, long) cho kiểu char. Ta thấy ở dòng 8, biến KyTu sẽ chứa một giá trị nào đó người dùng nhập vào (Lưu ý bắt buộc người dùng nhập kí tự thường), sau đó ta trừ cho 32 sẽ cho ra kí tự hoa (Xem bảng ASCII để hiểu rõ).

## 2.6 Chương trình sử dụng thư viện toán học “math.h”

Giả sử trong chương trình này ta cần tính giá trị y sau:

$$y = \sqrt[3]{\sqrt{p^6 + q^2} - q} - \sqrt[3]{\sqrt{p^6 + q^2} + q}$$

Dòng	
1	/*Tinh_can_so.c*/
2	#include <stdio.h>
3	#include <math.h>
4	void main()
5	{

6	<code>double p, q, y, delta, test;</code>
7	<code>printf("Nhập p: ");</code>
8	<code>scanf("%lf", &amp;p);</code>
9	<code>printf("Nhập q: ");</code>
10	<code>scanf("%lf", &amp;q);</code>
11	<code>delta = pow(p, 6.0) + pow(p, 2.0);</code>
12	<code>delta = sqrt(delta);</code>
13	<code>y = pow(delta - q, 1.0/3) - pow(delta + q, 1.0/3);</code>
14	<code>printf("Nghiem y = %lf\n", y);</code>
15	<code>}</code>

Ta thấy trong chương trình có sử dụng các hàm số thông dụng như *sqrt* đại diện cho phép căn số bậc 2  $\sqrt{\quad}$  còn *pow* đại diện cho phép mũ  $a^b$ . Để sử dụng các hàm số này ta cần "include" thư viện "math.h" của C.

### 3 Các minh họa về hàm scanf và printf

Phần này sẽ hướng dẫn cách dùng hàm scanf và printf. Hiểu rõ cách sử dụng những hàm này ta có thể tránh được các trường hợp trôi hàm cũng như xuất ra màn hình các kết quả theo định dạng mong muốn.

#### 3.1 Hàm printf

Cần khai báo <stdio.h> để sử dụng hàm. Cú pháp: printf("chuỗi điều khiển", danh sách tham số).

- Trong đó danh sách tham số bao gồm:

- Hằng
- Biến
- Biểu thức
- Hàm

Các thành phần này phải ngăn cách nhau bằng dấu phẩy ','. Ứng với mỗi thành phần trong danh sách tham số, sẽ có một mã định dạng kèm theo. Các mã định dạng phải khớp với các thành phần trong danh sách tham số về kiểu dữ liệu và thứ tự. Sau đây là các ví dụ được xem là **sai cú pháp** hoặc **chạy không ra kết quả như mong muốn**.

<code>int a = 32000;</code> <code>printf("%c", a);</code>	<code>int a = 5;</code> <code>printf(" ", a);</code>
<code>int a = 5;</code> <code>printf("%d", a);</code>	<code>int a = 5;</code> <code>printf("%d %d", a);</code>

Sau đây là một số ví dụ đúng về hàm printf

1	<code>int a = 5;</code>	
2	<code>int b = 6;</code>	
3	<code>printf("%d", 10);</code>	// in ra 10
4	<code>printf("%c", 'T');</code>	// in ra T
5	<code>printf("%d", a);</code>	// in ra 5
6	<code>printf("%d %d", a, b);</code>	// in ra 5 6
7	<code>printf("%d", a + b);</code>	// in ra 11

Về phần chuỗi điều khiển chỉ chứa ba loại dữ liệu chính sau:

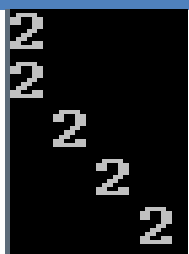
- Các kí tự không in: như dấu xuống dòng, tab, hoặc khoảng trắng
- Các kí tự in: như các kí tự chữ a, b, c...
- Các mã định dạng: như %d, %u...

Sau đây là một số ví dụ về việc dùng các mã định dạng

1	<code>printf("%g\n", 2.3);</code>	//in ra 2.3
2	<code>printf("%x\n", 2458111);</code>	//in ra 2581ff
3	<code>printf("%x\n", 0x2458111);</code>	//in ra 2458111
4	<code>printf("%d\n", 3);</code>	//in ra 3
5	<code>printf("%s\n", "Hello World");</code>	//in ra "Hello World"
6	<code>printf("%c\n", 'A');</code>	//in ra 'A'
7	<code>printf("%e\n", 2787.8);</code>	//in ra 2.787800e+003
8	<code>printf("%f\n", 12.3453453);</code>	//in ra 12.345345

Tiếp theo ta sẽ học cách dùng các tham số bổ trợ cho hàm printf.

- **Bổ từ về trường độ rộng**: Có thể sử dụng với các kiểu dữ liệu float, double hoặc chuỗi. Bổ từ này là một con số nguyên đứng phía trước mã định dạng nhằm xác định độ rộng cho trường dữ liệu

Dòng	Câu lệnh	Kết quả đoạn mã
1	<code>printf("%d\n", 2)</code>	
2	<code>printf("%1d\n", 2);</code>	
3	<code>printf("%2d\n", 2);</code>	
4	<code>printf("%3d\n", 2);</code>	
5	<code>printf("%4d\n", 2);</code>	

- **Bổ từ ‘-’**: Bổ từ này sẽ canh trái cho dữ liệu đầu ra, bất chấp có đặc tả bổ từ độ rộng bên trên

Dòng	Câu lệnh	Kết quả
1	<code>printf("%10d\n", 2)</code>	2
2	<code>printf("%-10d", 2)</code>	2

- **Bổ từ độ chính xác**: Bổ từ này được dùng cho kiểu float, double, và chuỗi. Nếu được sử dụng với kiểu dữ liệu float và double, thì chuỗi kí số sẽ chỉ định số lượng lớn nhất các kí số sẽ được in ra phía bên phải dấu chấm ‘.’

Dòng	Câu lệnh	Kết quả
1	<code>printf("%f\n", 2.39874);</code>	2.398740
2	<code>printf("%.1f\n", 2.39874);</code>	2.4
3	<code>printf("%.2f\n", 2.39874);</code>	2.40
4	<code>printf("%.3f\n", 2.39874);</code>	2.399
5	<code>printf("%.4f\n", 2.39874);</code>	2.3987

- **Bổ từ ‘0’**: Việc lấp các khoảng trống độ rộng là các kí tự khoảng trắng. Ta có thể dùng bổ từ này để thay thế.

Dòng	Câu lệnh	Kết quả
1	<code>printf("%10d\n", 2);</code>	2
2	<code>printf("%010d\n", 2);</code>	0000000002
3	<code>printf("%-10d\n", 2);</code>	2
4	<code>printf("%-010d\n", 2);</code>	2

- **Bổ từ ‘\*’**: Nếu người lập trình không muốn tự xác định độ rộng, mà muốn chương trình sẽ tự xác định theo ý của mình, ta phải dùng bổ từ này

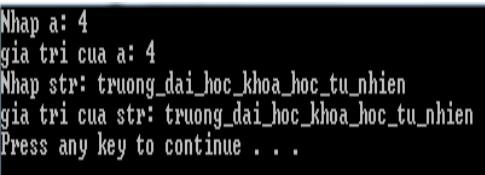
Dòng	Câu lệnh	Kết quả
1	<code>int n = 8;</code>	25.600000
2	<code>float x = 25.6, y = -47.335;</code>	-47.33
3	<code>printf("%f\n%*.2f", x, n, y);</code>	



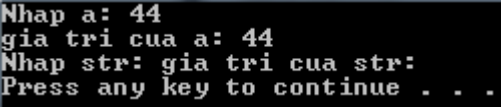
## 3.2 Hàm scanf

Về mặt cú pháp sử dụng, hàm này hoàn toàn tương tự như hàm printf. Nhưng có một số lưu ý nhỏ khi sử dụng hàm này ta cần phải biết.

- Nếu ta muốn nhập liệu vào một biến thuộc kiểu dữ liệu cơ bản như int, float, double, char, ta cần thêm toán tử có kí hiệu ‘&’ trước tên biến để lấy địa chỉ của tên biến đó.
- Nếu ta muốn nhập liệu vào một biến thuộc kiểu dữ liệu mảng hoặc con trỏ, ta không dùng toán tử có kí hiệu ‘&’, vì bản thân tên biến lúc này đã là địa chỉ.

Dòng	Câu lệnh	Kết quả
1	<code>int a;</code>	
2	<code>char str[200];</code>	
3	<code>printf("Nhap a: ");</code>	
4	<code>scanf("%d", &amp;a);</code>	
5	<code>printf("gia tri cua a: %d\n",a);</code>	
6	<code>printf("Nhap str: ");</code>	
7	<code>scanf("%s", str);</code>	
8	<code>printf("gia tri cua str: %s\n",str);</code>	

Trong phần tiếp theo ta sẽ xem xét hiện tượng trôi hàm nhập gets(). Nguyên nhân chính của hiện tượng này là do vẫn còn tồn tại kí tự ‘enter’ trong ‘đường ống’ stdin. Kí tự còn sót lại này sẽ tự động gán vào cho biến và dẫn tới việc biến không nhận dữ liệu từ bàn phím nữa mà đã tự động được gán kí tự ‘enter’. Bên dưới là một ví dụ trôi hàm, và các khắc phục.

Dòng	Câu lệnh	Kết quả
1	<code>int a;</code>	
2	<code>char str[200];</code>	
3	<code>printf("Nhap a: ");</code>	
4	<code>scanf("%d", &amp;a);</code>	
5	<code>printf("gia tri cua a: %d\n",a);</code>	
6	<code>printf("Nhap str: ");</code>	
7	<code>gets(str);</code>	
8	<code>printf("gia tri cua str: %s\n",str);</code>	

Do trong ‘đường ống stdin’ còn lưu kí tự ‘enter’ nên nó là trôi hàm gets bên dưới, để xử lí vấn đề này ta dùng “%\*c” để khử kí tự ‘enter’

Dòng	Câu lệnh	Kết quả
1	<code>int a;</code>	
2	<code>char str[200];</code>	
3	<code>printf("Nhap a: ");</code>	
4	<code>scanf("%d%c", &amp;a);</code>	
5	<code>printf("gia tri cua a: %d\n",a);</code>	
6	<code>printf("Nhap str: ");</code>	
7	<code>gets(str);</code>	
8	<code>printf("gia tri cua str: %s\n",str);</code>	

```
Nhap a: 4
gia tri cua a: 4
Nhap str: truong_dai_hoc_khoa_hoc_tu_nhien
gia tri cua str: truong_dai_hoc_khoa_hoc_tu_nhien
Press any key to continue . . .
```

## 4 Bài tập.

Sinh viên làm các bài tập sau:

4.1 Viết chương trình nhập vào hai số nguyên a và b. Tính thương giữa chúng và xuất kết quả. Nếu mẫu nhập vào bằng 0 thì xuất ra câu thông báo “Khong tinh duoc do mau bang 0”.

Ví dụ 1:

```
Nhap a = 3
Nhap b = 2
Thuong = 1.5
```

Ví dụ 2:

```
Nhap a = 3
Nhap b = 0
Khong tinh duoc do mau bang 0
```

4.2 Viết chương trình nhập vào số có hai chữ số. Tính tổng của hai chữ số đó và xuất kết quả.

Ví dụ:

```
Nhap x = 45
Tong cac chu so = 9
```

4.3 Viết chương trình nhập vào năm sinh, in ra tuổi, ví dụ nhập 1984 in ra:

Ban sinh năm 1984, vay ban 19 tuoi.

4.4 Viết chương trình, thực hiện các yêu cầu sau (không dùng hàm chuyển đổi):

Nhập vào một ký tự và in ra mã ASCII tương ứng với ký tự đó.

Nhập vào một số nguyên (1  $\rightarrow$  255) và in ra ký tự có mã ASCII tương ứng.

4.5 Nhập vào bán kính của hình tròn, tính và in ra chu vi, diện tích của hình tròn đó.

4.6 Nhập vào bán kính đáy R và chiều cao h của hình trụ tròn, tính diện tích đáy, diện tích xung quanh và thể tích của hình trụ tròn theo công thức (các số liệu là số thực, giá trị  $\pi$  đã được C định nghĩa sẵn bằng hằng số M\_PI):

$$SDAY = \pi R^2 \quad SXQ = 2\pi Rh \quad V = SDAY * h$$

4.7 Nhập vào số thực x, tính và in ra các giá trị y1, y2, lấy 2 số lẻ:

$$y1 = 4(x^2 + 10x\sqrt{x} + 3x + 1) \quad y2 = \frac{\sin(\pi x^2) + \sqrt{x^2 + 1}}{e^{2x} + \cos(\frac{\pi}{4}x)}$$

4.8 Nhập số tiền nguyên N đồng, đổi ra xem được bao nhiêu tờ 10 đồng, 5 đồng, 2 đồng và 1 đồng. Ví dụ: N = 543đ = 54 tờ 10đ + 0 tờ 5đ + 1 tờ 2đ + 1 tờ 1đ