



Bài 1

CÁC KHÁI NIỆM CƠ BẢN VỀ LẬP TRÌNH



Nội dung

1. Cơ bản về lập trình
2. Thuật toán
3. Ngôn ngữ lập trình C



1. Cơ bản về lập trình

- Chương trình là gì?
- Ngôn ngữ lập trình
- Các bậc của ngôn ngữ lập trình
- Biên dịch và thông dịch



Chương trình là gì?

- Máy tính hoạt động theo chương trình (Program).
- Chương trình là một dãy các lệnh nhằm thực hiện một công việc nào đó.

```
#include <stdio.h>

main()
{
    printf("hello, world\n");
}
```

- Chương trình do con người tạo ra gọi là lập trình (programming).



Ngôn ngữ lập trình (NNLT)

- Là một ngôn ngữ dùng để diễn đạt các chương trình máy tính.
- Là ngôn ngữ giao tiếp giữa người và máy tính.
- Các tính chất chung của ngôn ngữ:
 - Bảng chữ cái (alphabet)
 - Từ khóa (keyword)
 - Cú pháp (syntax)



Ngôn ngữ lập trình

- Các đặc trưng riêng của ngôn ngữ cho máy tính:
 - Câu lệnh (statement)
 - Kiểu dữ liệu (data type)
 - Cấu trúc điều khiển (control structure)
 - Biến (variable), hằng (constant), biểu thức (expression)
 - Hàm (function)



Các mức của NNLT

- Ngôn ngữ máy (machine language):
 - Ngôn ngữ thể hiện trực tiếp trong hệ lệnh của máy.
 - Ngôn ngữ máy là ngôn ngữ ở mức các bit, nên cũng được gọi là ngôn ngữ nhị phân.
- Hợp ngữ (assembly):
 - Là ngôn ngữ gần với ngôn ngữ nhị phân,
 - Mỗi lệnh của ngôn ngữ máy có một lệnh tương ứng của hợp ngữ.
- Ngôn ngữ bậc cao – còn gọi là ngôn ngữ thuật toán (Algorithmic language) là ngôn ngữ biểu diễn thuật toán độc lập với hệ lệnh của máy.

Ngôn ngữ máy

- Chính là ngôn ngữ được viết bằng lệnh máy trong hệ nhị phân hoặc hệ 16.
- Ưu điểm, tận dụng được khả năng của máy, tối ưu được thời gian chạy.
- Nhược điểm: khó viết, khó sửa lỗi, phụ thuộc vào từng loại máy.

Mã máy nhị phân	Mã hexa	Ý nghĩa
1001 0001 0110 0000 0001 0000	A1 60 10	Nạp 1060 lên TG AX
0000 0011 0110 0110 0001 0000	03 66 10	Cộng AX với 1066 -> AX
1010 0011 0000 0000 0010 1011	A3 00 2B	Ghi từ AX về 2B00

Hợp ngữ

- Mỗi lệnh hợp ngữ tương tự với một lệnh máy nhưng dùng mã chữ nên dễ hiểu, dễ sửa.
- Phải dịch ra ngôn ngữ máy
- Ưu điểm: dễ lập trình dễ sửa lỗi hơn ngôn ngữ máy
- Nhược điểm: vẫn còn phức tạp và phụ thuộc vào máy

Hợp ngữ	Mã máy trong hệ hexa
MOV AX CHIEU_DAI	A1 64 10
ADD AX CHIEU_RONG	03 66 10
MOV NUA_CHU_VI AX	A3 00 2B



Ngôn ngữ lập trình bậc cao

- Ngôn ngữ bậc cao (high level language) dùng diễn đạt thuật toán giải bài toán cho dễ dàng.
- Hình thức gần giống với ngôn ngữ tự nhiên nhưng mang đặc thù của máy tính:
 - Đơn giản
 - Rõ ràng
 - Xác định

Ví dụ NNLT bậc cao

- Ví dụ giải phương trình bậc 2 trên PASCAL

```
DELTA := B*B - 4*A*C;  
IF DELTA >= 0 THEN  
    BEGIN  
        X1 := (- B + SQRT(DELTA))/(2*A);  
        X2 := (- B - SQRT(DELTA))/(2*A);  
        WRITE (X1,X2);  
    END  
ELSE  
    WRITE('Vô nghiệm');
```

- FORTRAN

```
DELTA = B*B - 4* A*C  
IF DELTA < 0 GOTO 10  
X1= (- B + SQRT(DELTA))/(2*A)  
X2 =(- B - SQRT(DELTA))/(2*A)  
WRITE (3,20) X1, X2  
20  FORMAT ('NGHIEM 1= ', F8.3,  
           'NGHIEM 2 = ', F8.3)  
    GOTO 30  
10  WRITE(3,40)  
40  FORMAT('VO NGHIEM')  
30  END
```



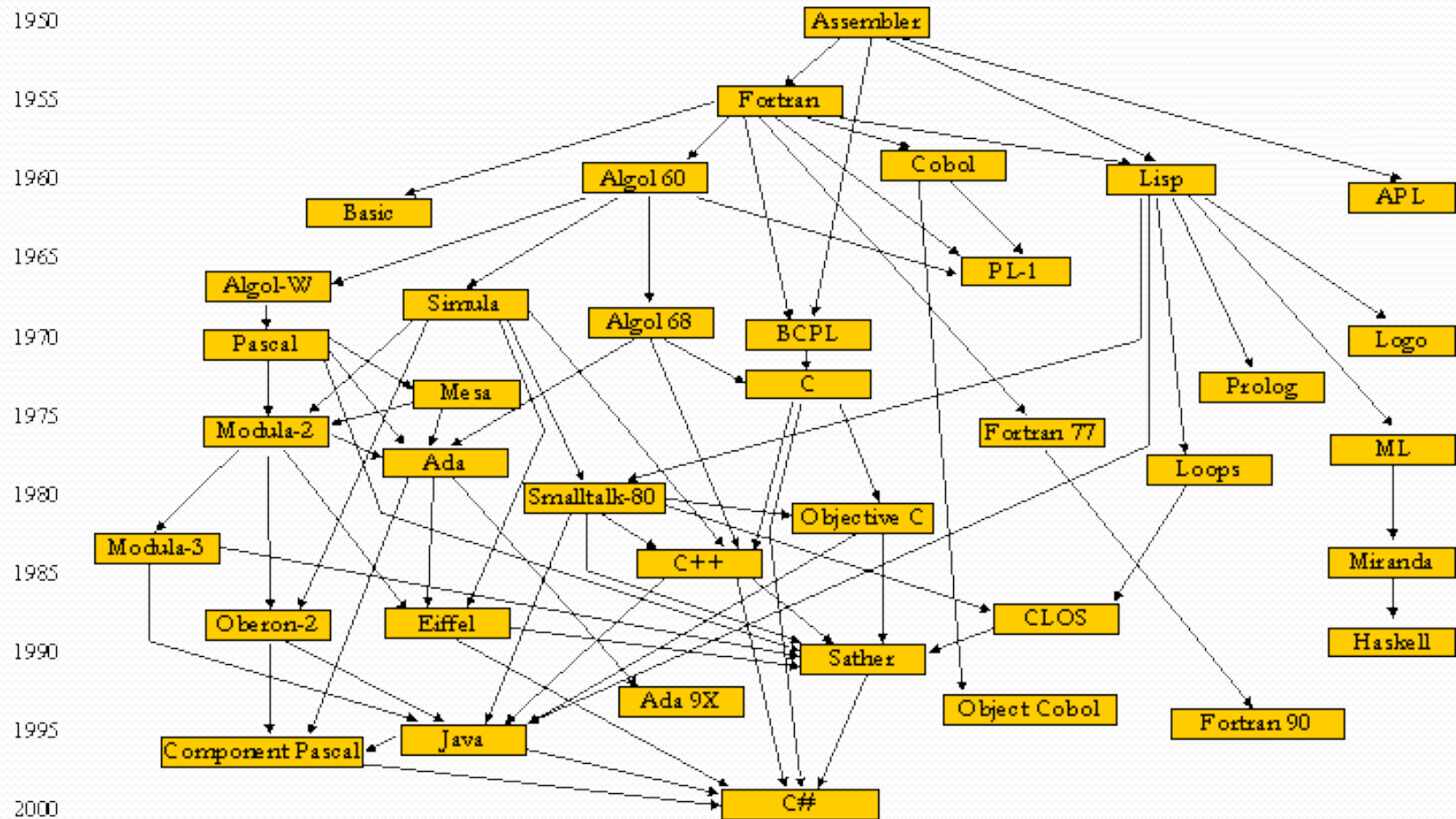
Thông dịch và biên dịch

- Để máy tính hiểu chương trình phải dịch sang ngôn ngữ máy.

Có hai cách dịch:

- Thông dịch (interpreter): dịch và thực thi từng lệnh của chương trình.
- Biên dịch (compiler): dịch chương trình trong ngôn ngữ thuật toán thành một chương trình ở ngôn ngữ máy.

Lịch sử phát triển các NNLT





2. Thuật toán

- Thuật toán là cách giải một bài toán bằng cách chỉ ra từng bước rõ ràng và chặt chẽ.
- Ví dụ: thuật toán tìm số lớn nhất trong hai số x, y .
 - Bước 1: Nếu x lớn hơn y thì số lớn là x
 - Bước 2: Ngược lại số lớn là y
 - Bước 3: Dừng

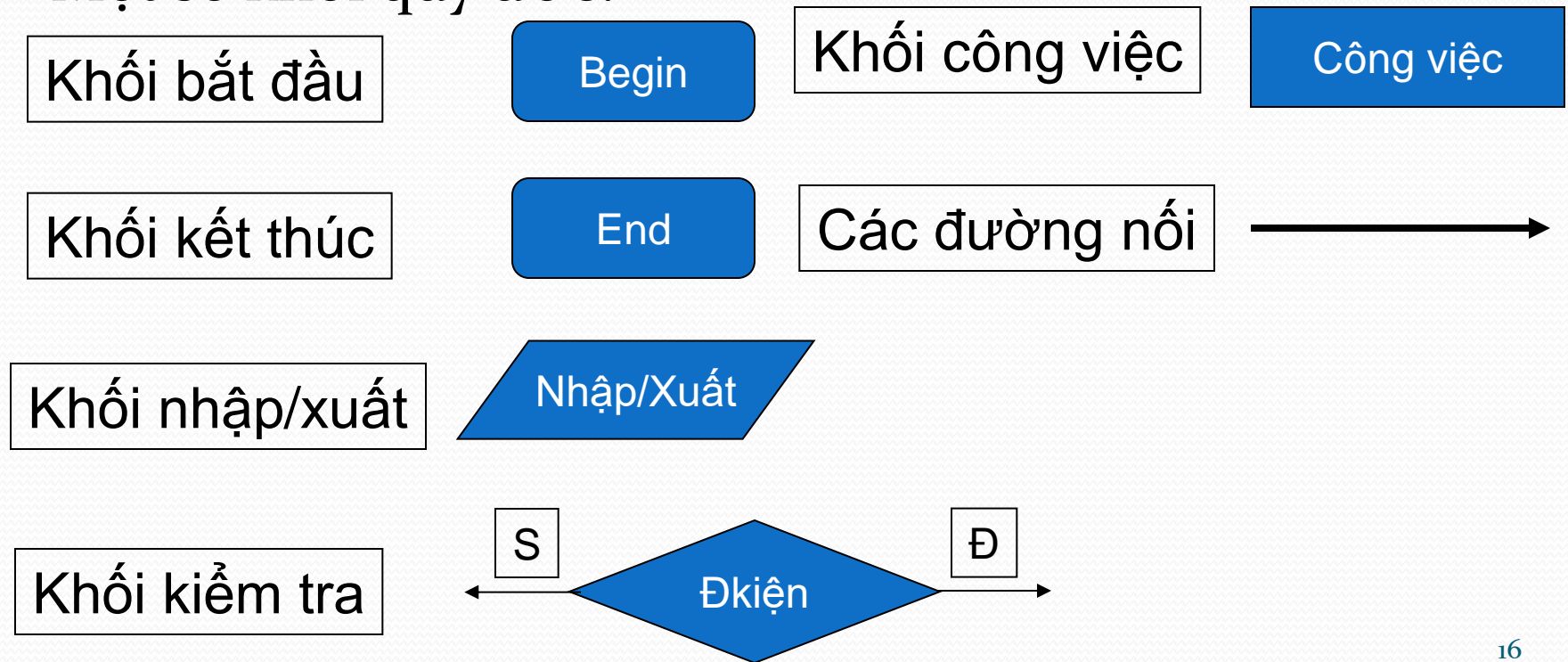


Biểu diễn thuật toán

- Các cách biểu diễn
 - Biểu diễn bằng sơ đồ khối
 - Biểu diễn bằng liệt kê từng bước
 - Biểu diễn bằng mã giả

Sơ đồ khối

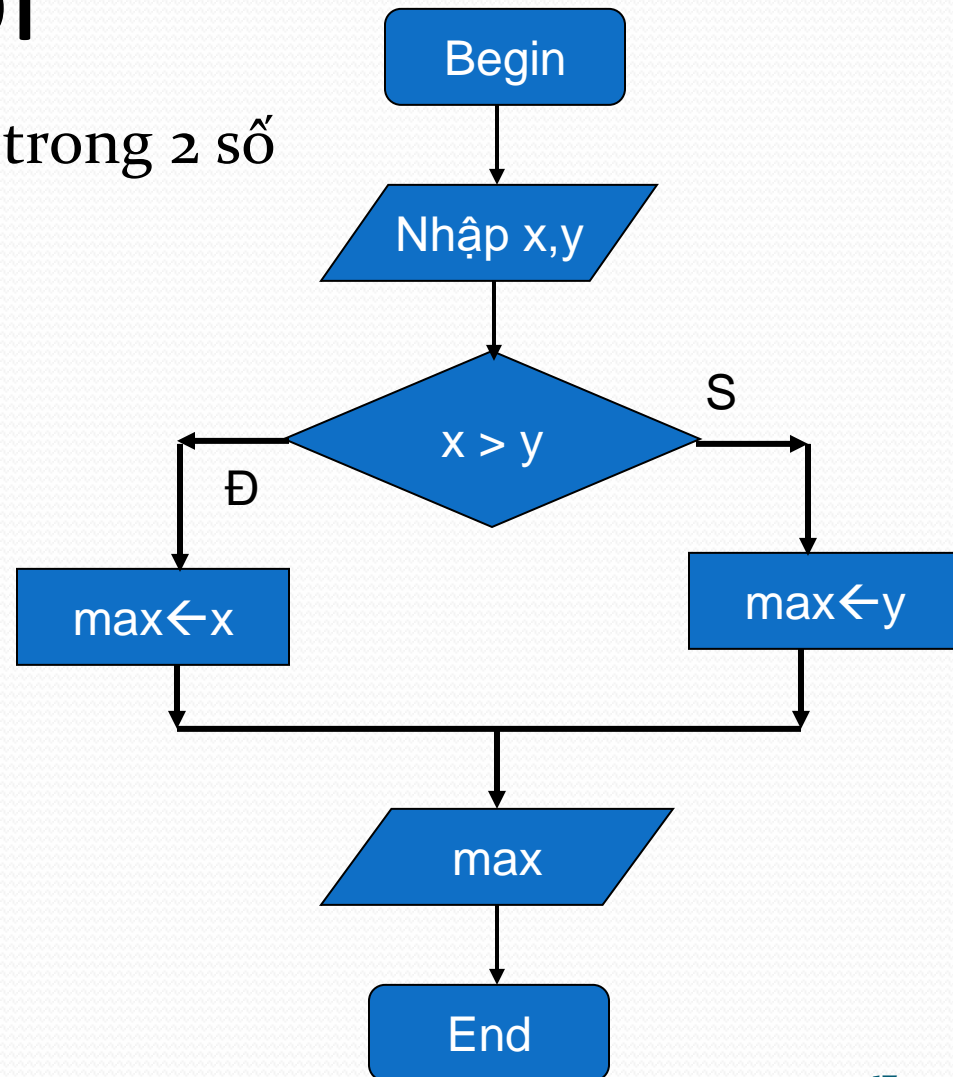
- Là các khối theo quy ước xác định cách thức thực hiện bằng hình ảnh.
- Một số khối quy ước:



Ví dụ sơ đồ khối

- Thuật toán tìm số lớn nhất trong 2 số

```
#include <stdio.h>
void main()
{
    int x, y, max;
    scanf("%d %d", &x, &y);
    if (x > y)
        max = x;
    else
        max = y;
    printf("%d", max);
}
```





Học lập trình như thế nào

- Học sử dụng 1 ngôn ngữ lập trình: C, C++, Java, Python,...
- Khái niệm cơ bản: kiểu dữ liệu, hằng, biến, hàm, biểu thức
- Cú pháp và ý nghĩa các lệnh: gán, nhập/xuất, cấu trúc điều khiển
- Tổ chức chương trình thành các hàm
- Các kiểu dữ liệu có cấu trúc: mảng, chuỗi, tệp, tự định nghĩa.

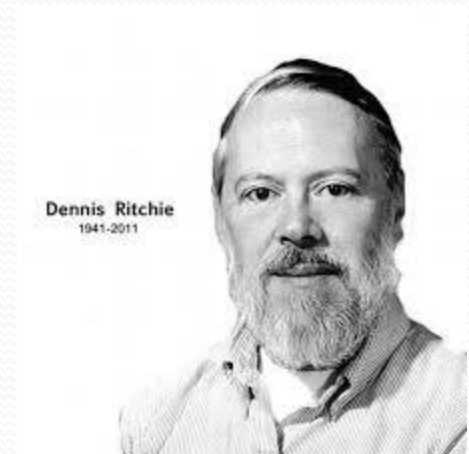


Học lập trình như thế nào

- Học các chương trình mẫu
 - Đọc hiểu lệnh, ý nghĩa
 - Chạy chương trình
 - Sửa đổi theo ý cá nhân
- Tự làm chương trình tương tự
- Học một số thuật toán cơ bản
- Chuyển các thuật toán thành chương trình
- Tự hình thành thuật toán từ các thuật toán cơ bản và viết chương trình

3. Ngôn ngữ lập trình C

- Được xây dựng bởi *Dennis Ritchie* năm 1970
- Là một ngôn ngữ lập trình được ưa chuộng
 - Ngắn gọn
 - Đơn giản
- Là ngôn ngữ chính để lập trình ra hệ điều hành Windows





Bảng chữ cái, từ khóa

- C sử dụng bảng chữ cái latin
- Từ khóa: 32 từ khóa

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while



Quy ước đặt tên

- Quy định đặt tên:
 - Bắt đầu bằng chữ cái, hoặc dấu _
 - Các ký tự tiếp theo có thể chữ cái, chữ số, dấu _
 - Tên không được trùng với từ khóa
 - Tên phân biệt chữ hoa và chữ thường
- Ví dụ tên đúng:
x1, x2, heSo, hoTen, que_quan
- Ví dụ tên sai:
1x, ho ten, he-so, hag#tag
- Nên nhất quán 1 cách đặt tên.



Câu lệnh

- Câu lệnh đơn: kết thúc bằng dấu ;
- Ví dụ: `max = x;`
- Câu lệnh điều khiển: không có dấu ;
- Ví dụ: `if (x > y)`
- Khối lệnh: nhóm các lệnh, đặt trong { ... }
- Ví dụ:
`if (x > y)`
`{`
 `max = x;`
 `min = y;`
`}`



Chú thích

- Chú thích:
 - Trên 1 dòng: dùng // chú thích
 - Trên nhiều dòng
/* Chú thích dòng 1
Chú thích dòng 2 */



Cấu trúc chương trình

- Cấu trúc chương trình C đơn giản:

#include <thư viện>

Khai báo biến, hằng, kiểu, hàm

void main()

{

 các lệnh;

}

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

```
main()
```

```
{
```

```
    printf("hello, world\n");
```

```
}
```



Kiểu dữ liệu

- Số nguyên: int, long
 - Kích thước: 4 bytes
 - Phạm vi: $-2.147.483.648 \rightarrow 2.147.483.647$
 - Phép toán:
 - $+, -, *, /, \%, ++, --$
- Ký tự: char
 - Kích thước: 1byte
 - Hằng ký tự: đặt giữa “
 - Một số hằng: `‘\n’, ‘\t’, ‘\’, ‘\\’`



Kiểu dữ liệu

- Kiểu số thực:
 - float (4 bytes), double (8 bytes)
- Kiểu logic: dùng kiểu số nguyên
 - Quy ước: khác 0 là đúng, 0 là sai.
 - Phép toán: !, &&, ||



- Các phép toán chung:
 - Phép gán: =
 - Phép gán mở rộng: +=, -=, *=, /=, %=
 - Phép so sánh bằng: ==
 - So sánh khác: !=
 - Thứ tự: >, <, >=, <=
 - Phép toán ?: đk ? GT₁: GT₂
 - Ví dụ: $x > 5$? 1: 2



Hằng, biến

- Hằng (constant): là giá trị không thay đổi trong chương trình.
- `#define TEN_HANG giaTri`
Ví dụ: `#define PI 3.1415`
- Biến (variable): là đại lượng thay đổi trong chương trình.
- Khai báo biến: Kiểu dữ liệu `tenBien`;
- Ví dụ: `int x, y;`



Hàm

- Hàm (function):
trong C cung cấp sẵn
một số hàm thông
dụng trong tính
toán.

*Các hàm lượng giác
dùng độ đo radian.*

Tên hàm	Ý nghĩa
<code>exp(x)</code>	e^x
<code>log(x)</code>	logaric cơ số e của x
<code>log₁₀(x)</code>	logaric cơ số 10 của x
<code>abs(x)</code>	trị tuyệt đối của x (số nguyên)
<code>fabs(x)</code>	trị tuyệt đối của x (số thực)
<code>pow(a,x)</code>	a^x
<code>sqrt(x)</code>	căn bậc 2 của x
<code>sin(x), cos(x)</code>	$\sin(x)$, $\cos(x)$
<code>tan(x)</code>	$\tan(x)$
<code>acos(x)</code>	$\arccos(x)$
<code>asin(x)</code>	$\arcsin(x)$
<code>ceil(x)</code>	số nguyên nhỏ nhất $\geq x$
<code>floor(x)</code>	số nguyên lớn nhất $\leq x$



Biểu thức

- Biểu thức là kết hợp của hằng, biến, hàm.
- Ví dụ: $x_1 = (-b + \sqrt{d}) / (a * a)$
- Thứ tự ưu tiên phép toán

Ép kiểu: chuyển đổi kiểu dữ liệu của biểu thức.

Ví dụ:

```
int x;
```

```
x = (int) 1.5/2;
```

`() [] -> .`

`- ! ~ ++ -- (type)`

`* & sizeof`

`* / %`

`+ -`

`<< >>`

`< <= > >=`

`== !=`

`&`

`^`

`|`

`&&`

`||`

`? :`

`= += -= *= /= %=`



Nhập/Xuất

- Hàm in dữ liệu ra màn hình:

`printf("Chuỗi định dạng", ds biểu thức);`

- Mã định dạng:

Ví dụ:

```
printf("Xin chào \n mọi người.");  
printf("Gia tri cua m la: %d", m);  
printf("Phuong trinh co nghiem x =  
%f", -b/a);  
printf("sin(%f)=%f", x, sin(x));
```

Mã	Kiểu dữ liệu
%c	char
%d	int
%ld	long int
%f	float
%lf	double
%s	Xâu ký tự



- Định dạng độ rộng: x.y
 - .y%d: y là số chữ số
 - x.y%f: y là số chữ số thập phân
 - x.y%s: x là số ký tự tối thiểu, y là tối đa

```
#include <stdio.h>
main()
{
    printf("%.4f\n", 123.1234567);
    printf("%.8d\n", 1000);
    printf("%10.15s\n", "This is a simple test.");
}
```

```
123.1235
00001000
This is a simpl
```



Nhập/Xuất

- Hàm nhập dữ liệu từ bàn phím:
`scanf("chuỗi định dạng", dsách các biến);`
- Ví dụ:
- `scanf("%d", &n);`
- `scanf("%d %f", &n, &x);`



Ví dụ minh họa

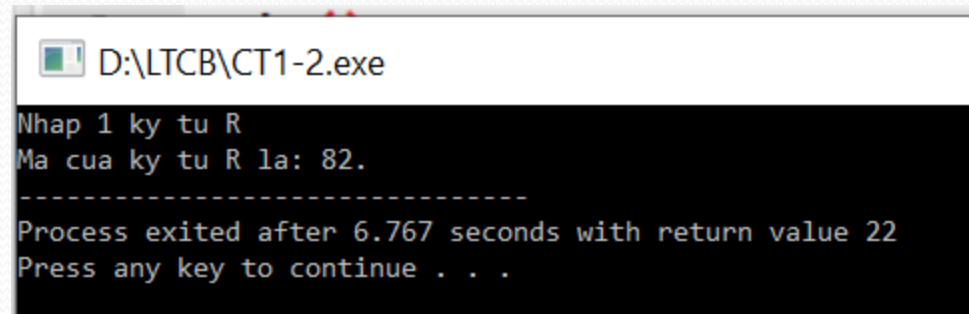
- Ví dụ 1. Chương trình in lên màn hình dòng chữ Hello world.

```
#include <stdio.h>
main()
{
    printf("Hello world.");
}
```

- Ví dụ 2. Chương trình nhập 1 ký tự và in lên màn hình mã ASCII của ký tự đó.



```
#include <stdio.h>
main()
{
    char c;
    printf("Nhap 1 ky tu "); scanf("%c",&c);
    printf("Ma cua ky tu %c la: %d.",c,c);
}
```



The screenshot shows a Windows command prompt window titled "D:\LTCB\CT1-2.exe". The output of the program is displayed as follows:

```
Nhap 1 ky tu R
Ma cua ky tu R la: 82.
-----
Process exited after 6.767 seconds with return value 22
Press any key to continue . . .
```



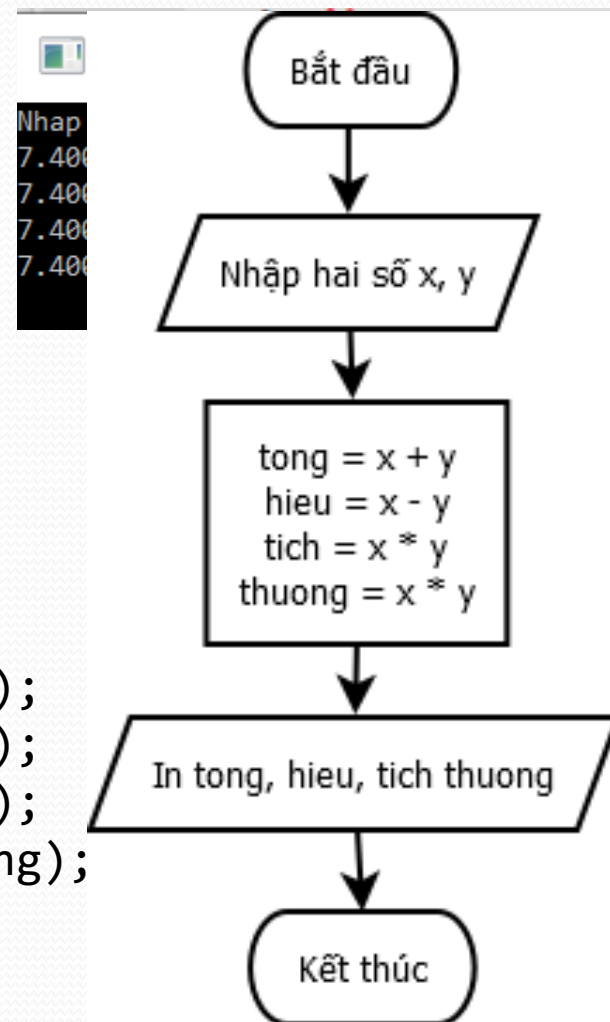
Ghi nhớ 1:

Hình dung các bước trước khi viết chương trình.



- Ví dụ 3. Chương trình nhập 2 số thực x, y ($y \neq 0$). Tính và in lên màn hình $x+y, x-y, x*y, x/y$.

```
#include <stdio.h>
main()
{
    float x, y, tong, hieu, tich, thuong;
    printf("Nhap hai so thuc ");
    scanf("%f %f", &x, &y);
    tong = x + y;
    hieu = x - y;
    tich = x * y;
    thuong = x / y;
    printf("%f + %f = %10.4f \n", x, y, tong);
    printf("%f - %f = %10.4f \n", x, y, hieu);
    printf("%f * %f = %10.4f \n", x, y, tich);
    printf("%f / %f = %10.4f \n", x, y, thuong);
}
```





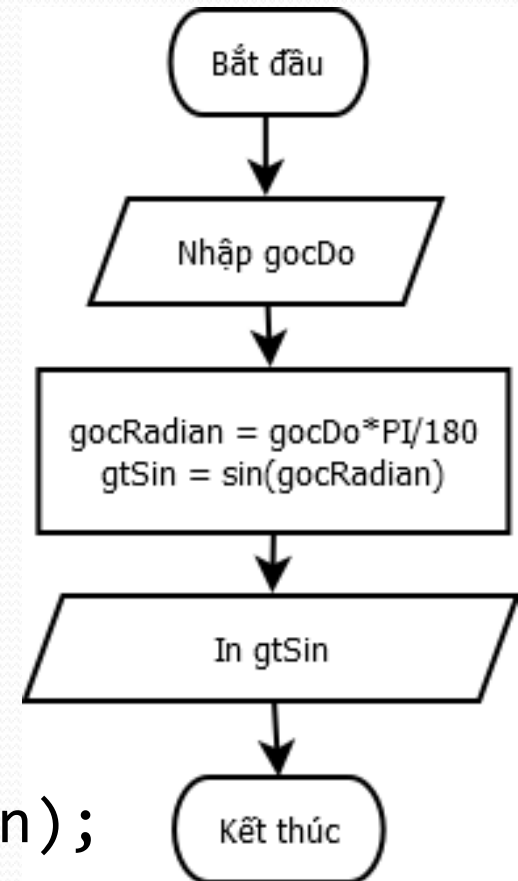
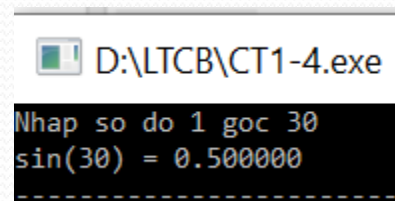
Ghi nhớ 2:

Các giá trị tính toán nên lưu ra biển.



- Ví dụ 4. Chương trình nhập số đo một góc bằng độ. Tính và in lên màn hình sin của góc đó.

```
#include <stdio.h>
#include <math.h>
main()
{
    int gocDo;
    float gocRadian, gtSin;
    printf("Nhap so do 1 goc ");
    scanf("%d", &gocDo);
    gocRadian = gocDo*M_PI/180;
    gtSin = sin(gocRadian);
    printf("sin(%d) = %f",gocDo, gtSin);
}
```





Ghi nhớ 3:

Sử dụng hàm phải đúng đối số.



- Ví dụ 5. Chương trình đổi giá trị hai biến số nguyên x và y.

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

```
main()
```

```
{
```

```
    int x, y, tg;
```

```
    printf("Nhap hai so nguyen ");
```

```
    scanf("%d %d", &x, &y);
```

```
    printf("Truoc khi doi x = %d, y = %d \n",x,y);
```

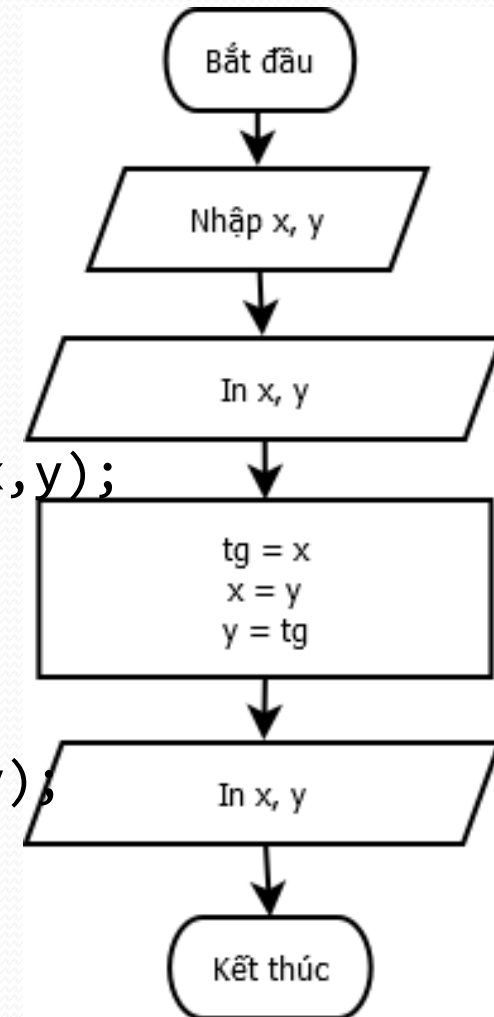
```
    tg = x;
```

```
    x = y;
```

```
    y = tg;
```

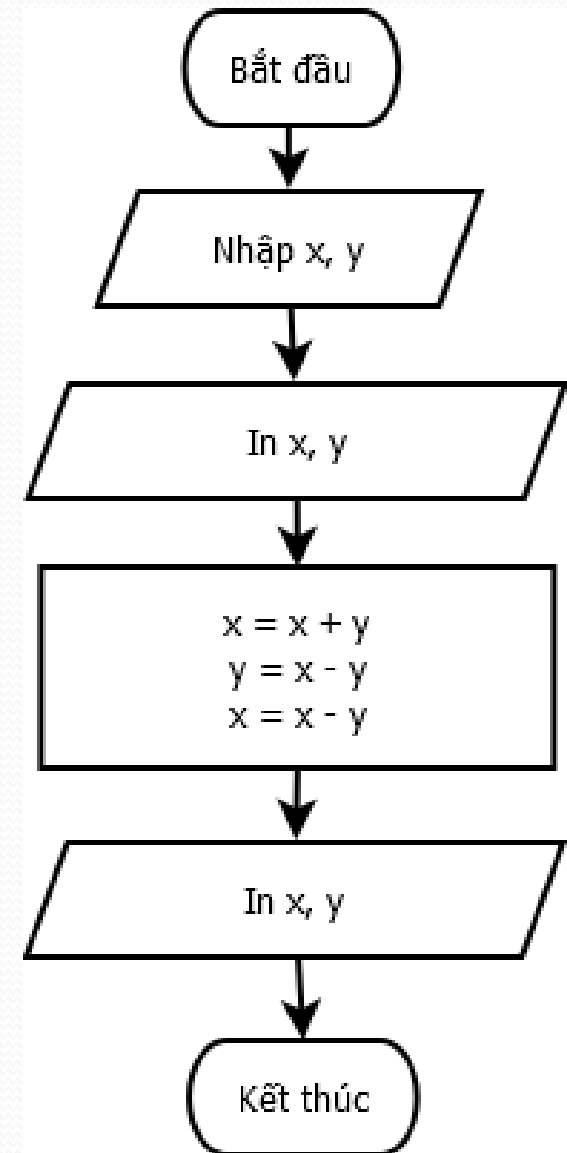
```
    printf("Sau khi doi x = %d, y = %d \n",x,y);
```

```
}
```



Cách khác

x	y
5	7
12	7
12	5
7	5





Ghi nhớ 4:

Thuật toán càng đơn giản càng tốt.



Luyện tập

- Viết chương trình nhập vào bán kính của một hình tròn là một số nguyên. Tính và in lên màn hình chu vi và diện tích của hình tròn.



Tổng kết

- Ngôn ngữ lập trình là ngôn ngữ giao tiếp giữa người lập trình và máy tính.
- Để viết chương trình cần có thuật toán.
- Các yếu tố cơ bản của ngôn ngữ lập trình:
 - Bảng chữ cái, Từ khóa
 - Kiểu dữ liệu
 - Biến, hằng, hàm, biểu thức



Bài tập

- Viết các chương trình thực hiện:
 1. Nhập vào 1 ký tự là chữ thường. In lên màn hình chữ hoa tương ứng với chữ thường đó.
 2. Nhập vào chiều dài 2 cạnh của một hình chữ nhật. Tính chu vi và diện tích của hình chữ nhật đó.
 3. Nhập vào một số thực dương. Tính và in lên màn hình căn bậc 4 của số đó.
 4. Nhập vào số nguyên dương n . Tính và in lên màn hình tổng $S = 1+2+\dots+n$.
 5. Nhập vào độ dài 3 cạnh của một hình tam giác. Tính diện tích của tam giác đó.
 6. Nhập vào 2 số thực, tìm số lớn trong 2 số đó.