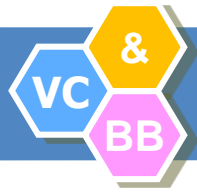


TỔNG QUAN VỀ LẬP TRÌNH VÀ NGÔN NGỮ LẬP TRÌNH C

Đặng Ngọc Hùng

Khoa: Công nghệ thông tin 1

Email: hungdn@ptit.edu.vn



Nội dung

1

Các khái niệm cơ bản

2

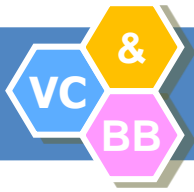
Các bước xây dựng chương trình

3

Biểu diễn thuật toán

4

Cài đặt thuật toán bằng NNLT



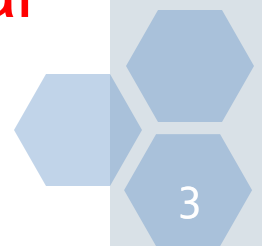
Các khái niệm cơ bản

❖ Lập trình máy tính

- Gọi tắt là **lập trình** (programming).
- Nghệ thuật **cài đặt** một hoặc nhiều **thuật toán** trừu tượng có liên quan với nhau bằng một **ngôn ngữ lập trình** để tạo ra một **chương trình máy tính**.

❖ Thuật toán

- Là **tập hợp** (dãy) **hữu hạn** các **chỉ thị** (hành động) được **định nghĩa rõ ràng** nhằm **giải quyết** một bài toán cụ thể nào đó.





Các khái niệm cơ bản

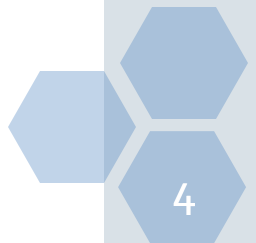
❖ Ví dụ

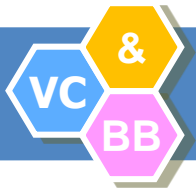
- Thuật toán giải PT bậc nhất: $ax + b = 0$ (a, b là các số thực).

Đầu vào: a, b thuộc \mathbb{R}

Đầu ra: nghiệm phương trình $ax + b = 0$

- Nếu $a = 0$
 - $b = 0$ thì phương trình có nghiệm bất kì.
 - $b \neq 0$ thì phương trình vô nghiệm.
- Nếu $a \neq 0$
 - Phương trình có nghiệm duy nhất $x = -b/a$





Các tính chất của thuật toán

❖ Bao gồm 5 tính chất sau:

- **Tính chính xác:** quá trình tính toán hay các thao tác máy tính thực hiện là chính xác.
- **Tính rõ ràng:** các câu lệnh minh bạch được sắp xếp theo thứ tự nhất định.
- **Tính khách quan:** được viết bởi nhiều người trên máy tính nhưng kết quả phải như nhau.
- **Tính phổ dụng:** có thể áp dụng cho một lớp các bài toán có đầu vào tương tự nhau.
- **Tính kết thúc:** hữu hạn các bước tính toán.

Các bước xây dựng chương trình





Sử dụng ngôn ngữ tự nhiên

Đầu vào: a, b thuộc \mathbb{R}

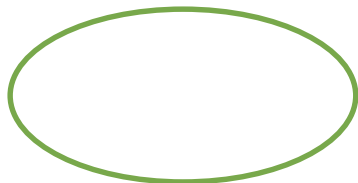
Đầu ra: nghiệm phương trình $ax + b = 0$

1. Nhập 2 số thực a và b .
2. Nếu $a = 0$ thì
 - 2.1. Nếu $b = 0$ thì
 - 2.1.1. Phương trình vô số nghiệm
 - 2.1.2. Kết thúc thuật toán.
 - 2.2. Ngược lại
 - 2.2.1. Phương trình vô nghiệm.
 - 2.2.2. Kết thúc thuật toán.
3. Ngược lại
 - 3.1. Phương trình có nghiệm.
 - 3.2. Giá trị của nghiệm đó là $x = -b/a$
 - 3.3. Kết thúc thuật toán.



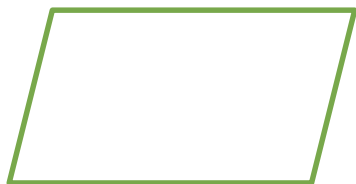


Sử dụng lưu đồ - sơ đồ khối



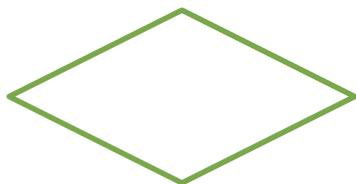
Khối giới hạn

Chỉ thị bắt đầu và kết thúc.



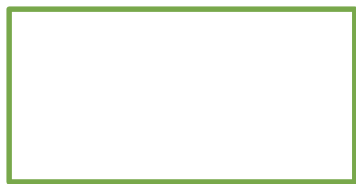
Khối vào ra

Nhập/Xuất dữ liệu.



Khối lựa chọn

Tùy điều kiện sẽ rẽ nhánh.



Khối thao tác

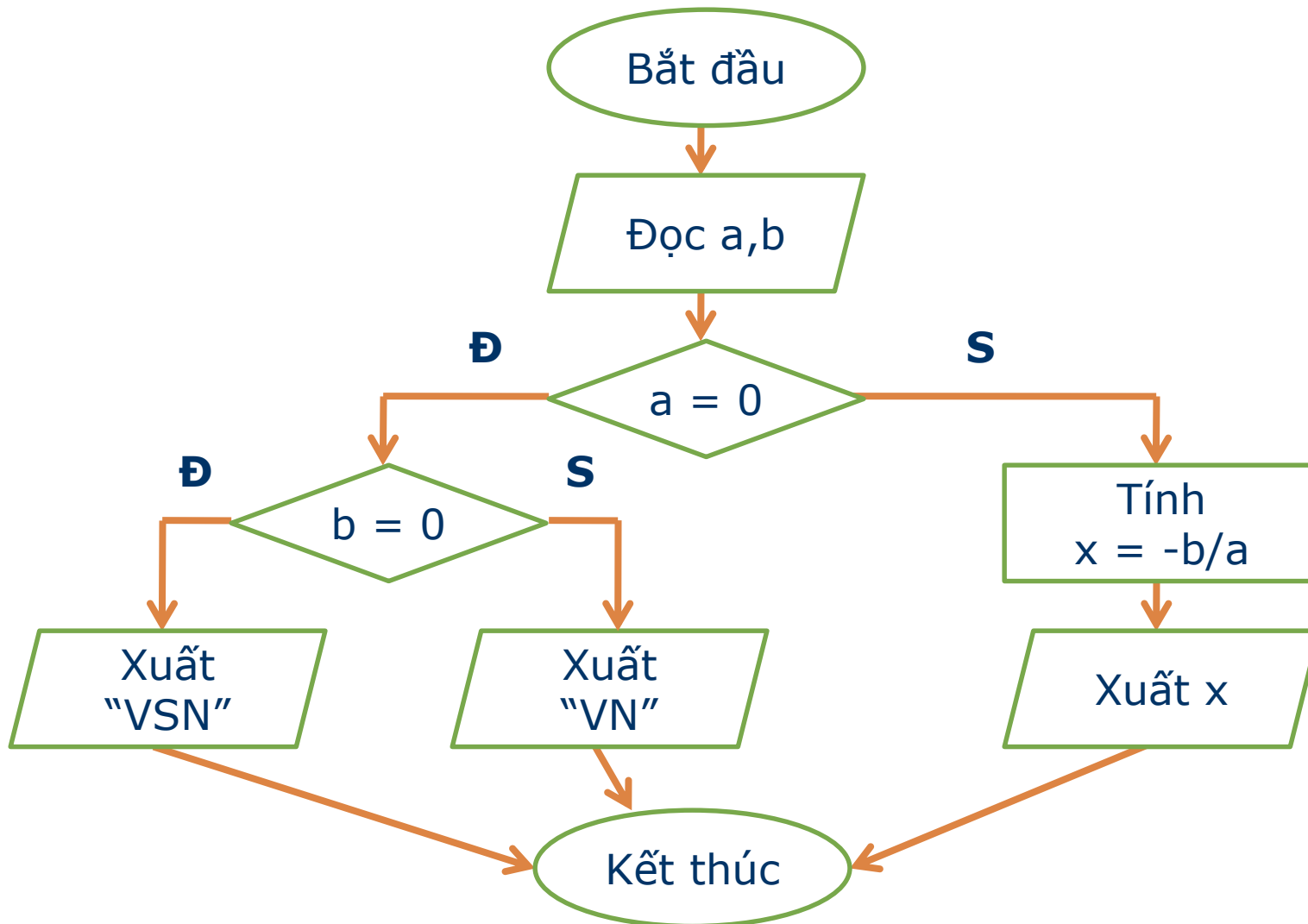
Ghi thao tác cần thực hiện.

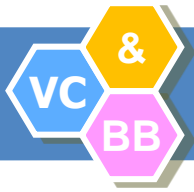


Đường đi

Chỉ hướng thao tác tiếp theo.

Sử dụng lưu đồ - sơ đồ khối





Sử dụng mã giả (pseudo code)

- ❖ Vay mượn ngôn ngữ nào đó (ví dụ Pascal) để biểu diễn thuật toán.

Đầu vào: a, b thuộc \mathbb{R}

Đầu ra: nghiệm phương trình $ax + b = 0$

```
If  $a = 0$  Then
```

```
Begin
```

```
    If  $b = 0$  Then
```

```
        Xuất "Phương trình vô số nghiệm"
```

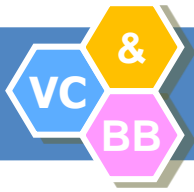
```
    Else
```

```
        Xuất "Phương trình vô nghiệm"
```

```
End
```

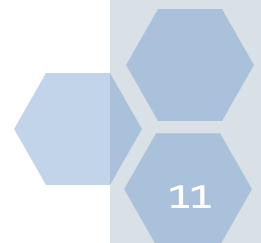
```
Else
```

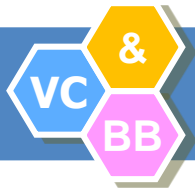
```
    Xuất "Phương trình có nghiệm  $x = -b/a$ "
```






Cài đặt thuật toán bằng C

```
#include <stdio.h>
void main()
{
    int a, b;
    printf("Nhap a, b: ");
    scanf("%d%d", &a, &b);
    if (a==0)
    {
        if (b==0)
            printf("Phuong trinh VSN");
        else
            printf("Phuong trinh VN");
    }
    else
        printf("x = %f", float(-b)/a);
}
```

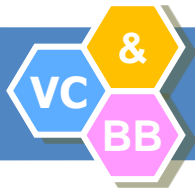








Bài tập

-  1. Nhập năm sinh của một người. Tính tuổi người đó.
-  2. Nhập 2 số a và b. Tính tổng, hiệu, tính và thương của hai số đó.
-  3. Nhập tên sản phẩm, số lượng và đơn giá. Tính tiền và thuế giá trị gia tăng phải trả, biết:
 - a. tiền = số lượng * đơn giá
 - b. thuế giá trị gia tăng = 10% tiền

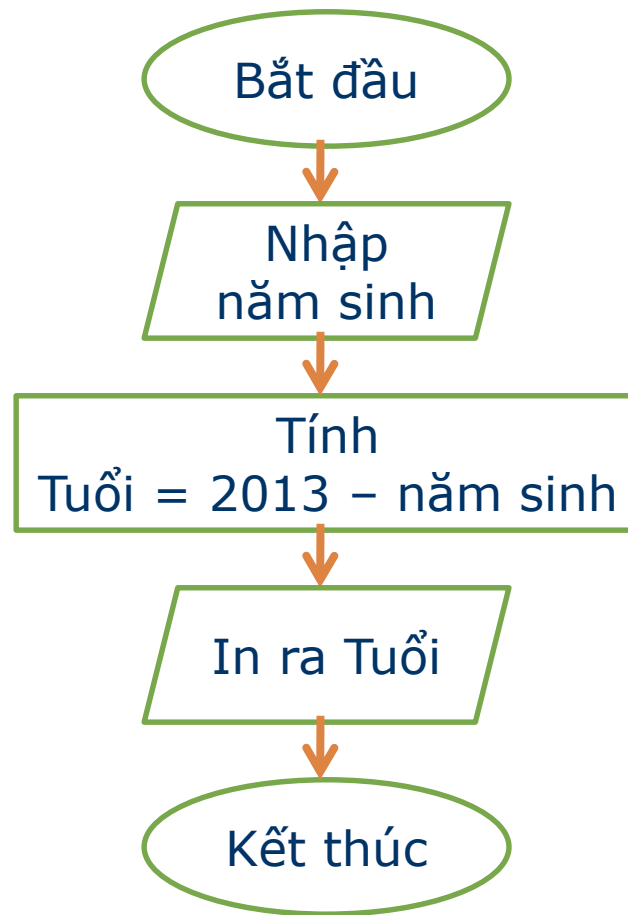


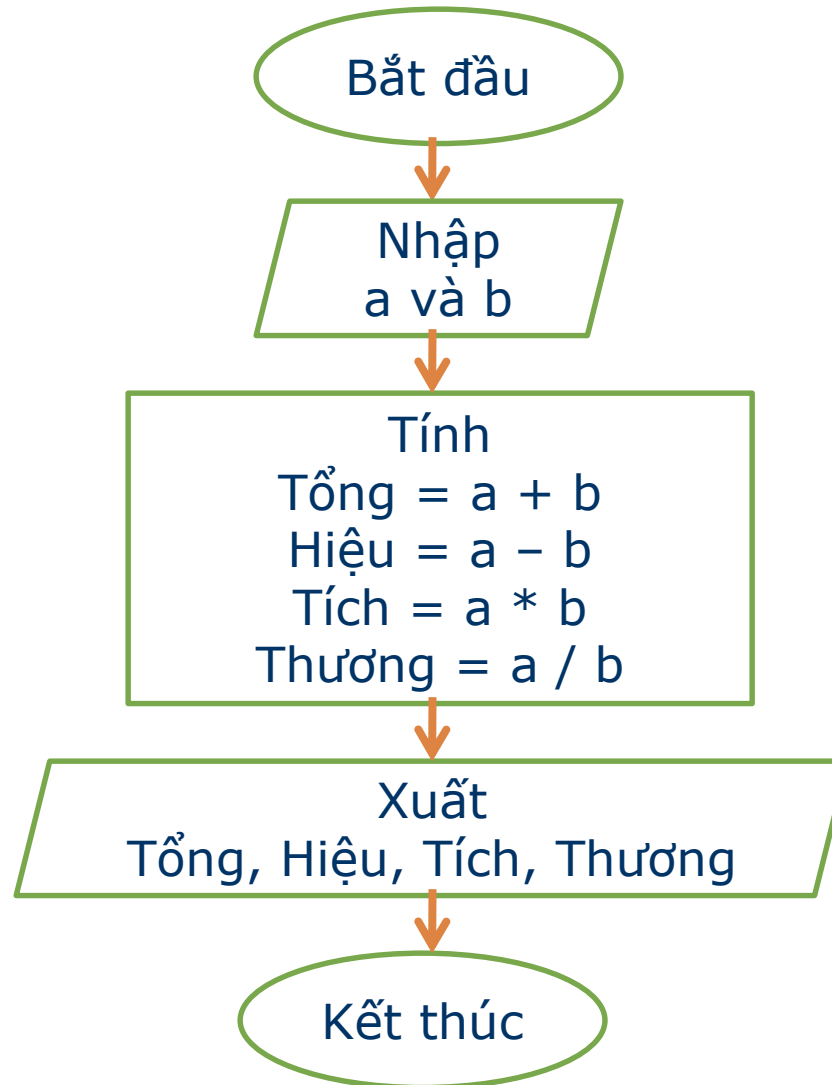


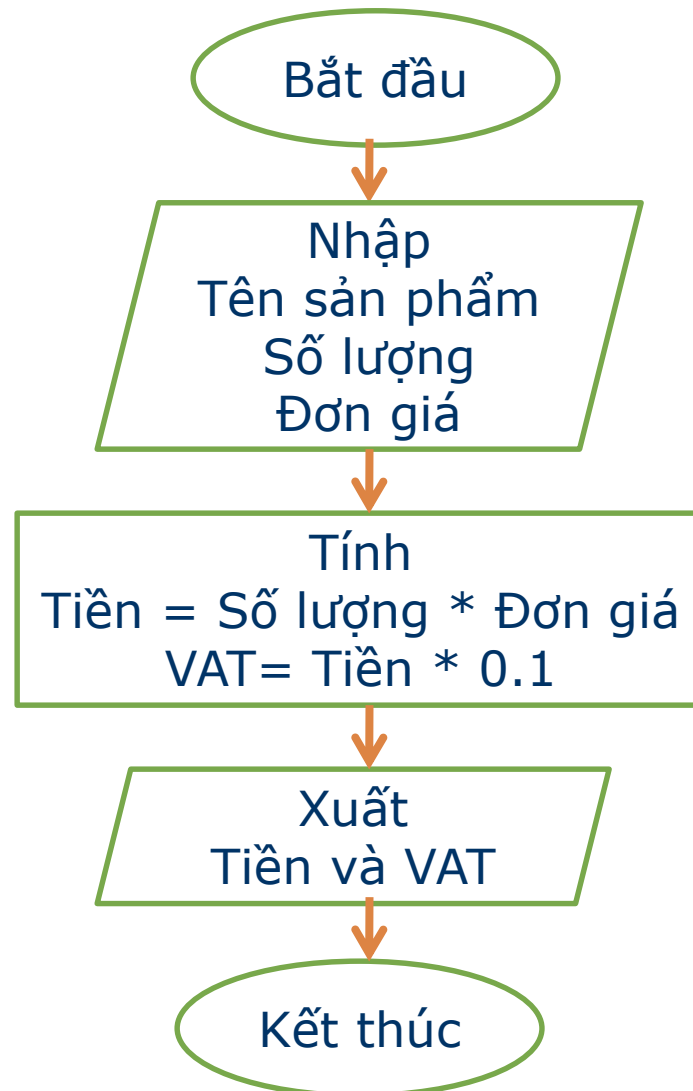
Bài tập thực hành

-  4. Nhập điểm thi và hệ số 3 môn Toán, Lý, Hóa của một sinh viên. Tính điểm trung bình của sinh viên đó.
-  5. Nhập bán kính của đường tròn. Tính chu vi và diện tích của hình tròn đó.
-  6. Nhập vào số xe (gồm 4 chữ số) của bạn. Cho biết số xe của bạn được mấy "nước"?
-  7. Nhập vào 2 số nguyên.
Tính min và max của hai số đó.

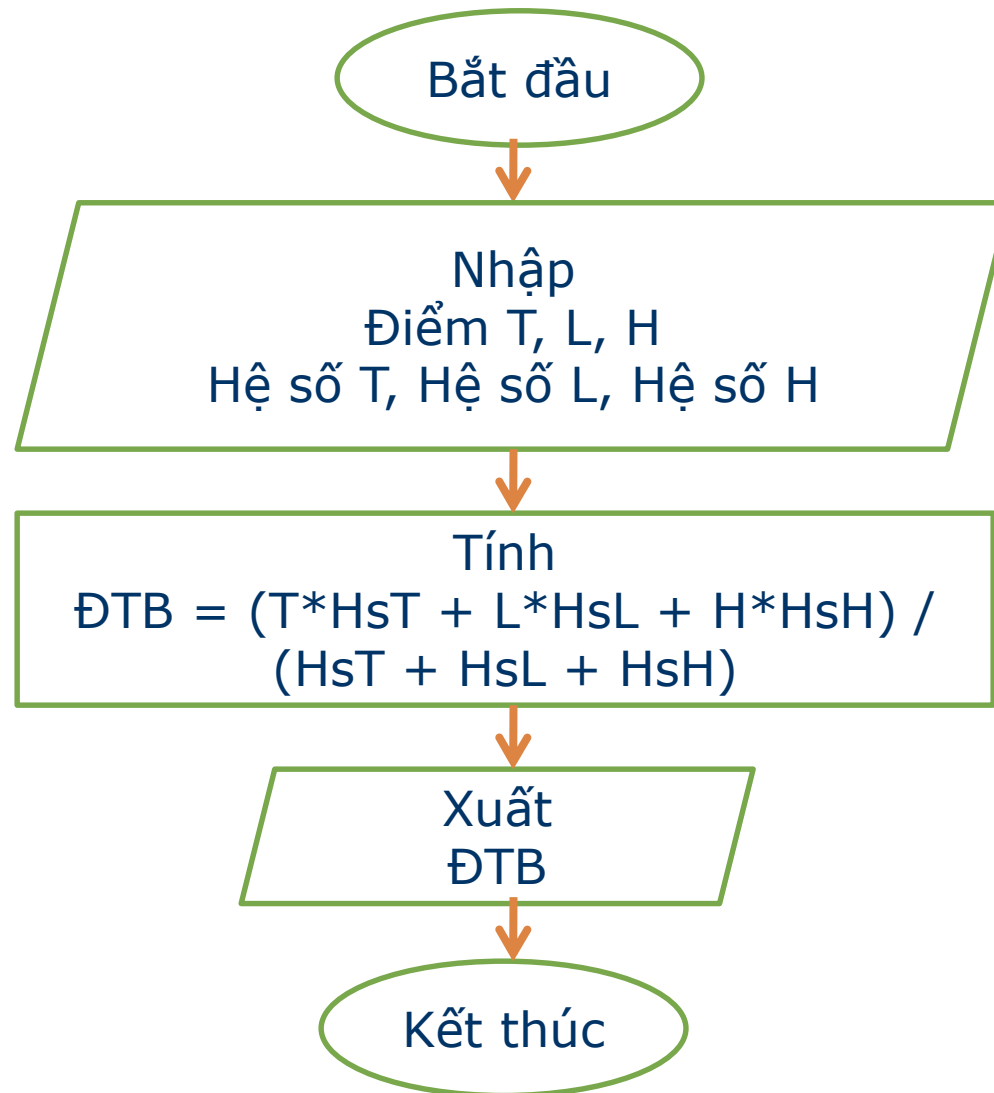


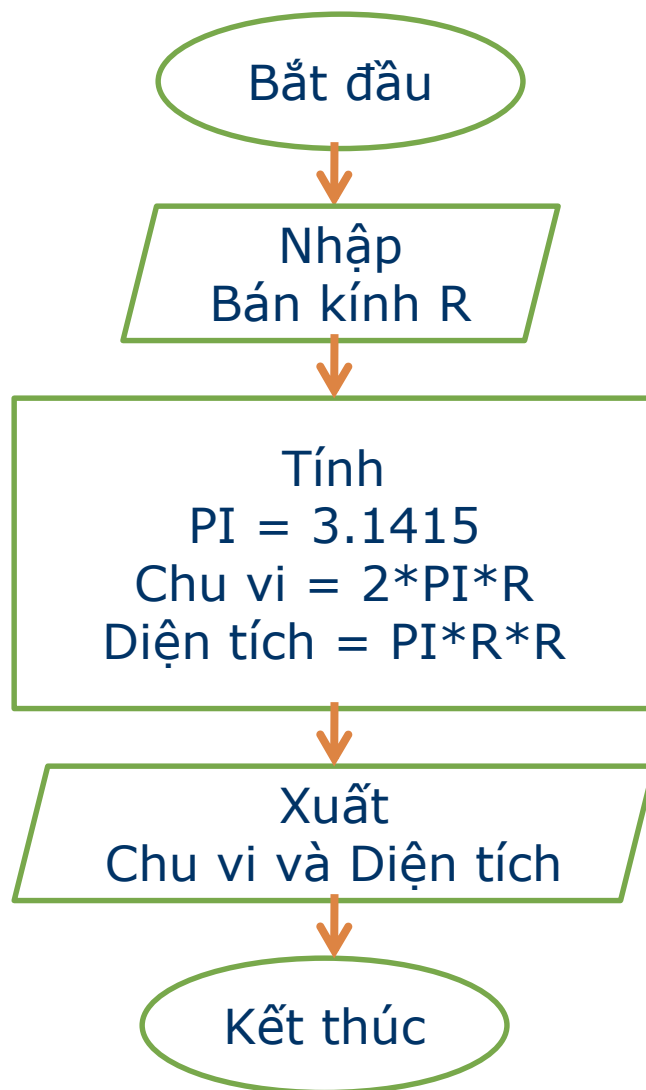




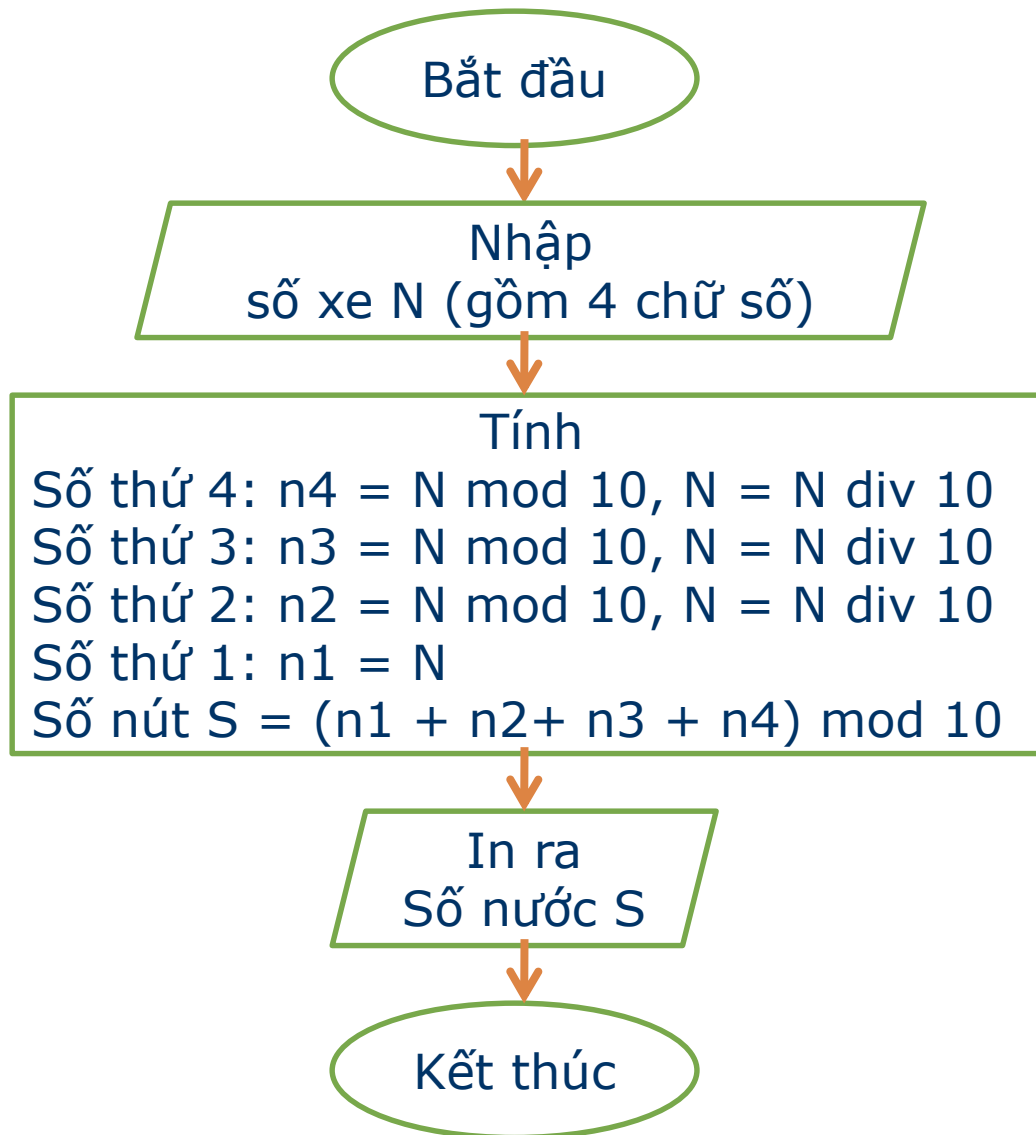


Bài tập 4

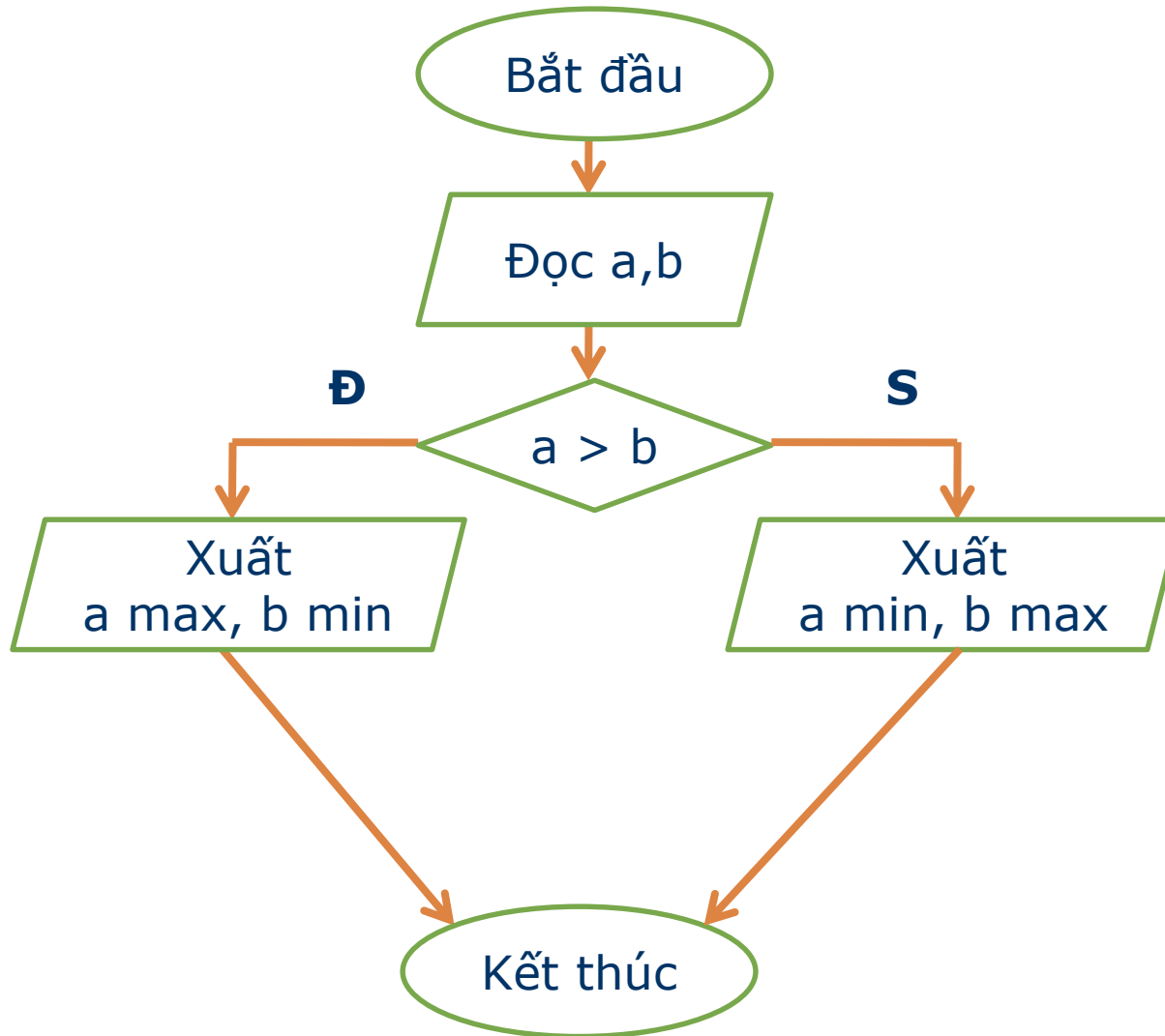




Bài tập 6

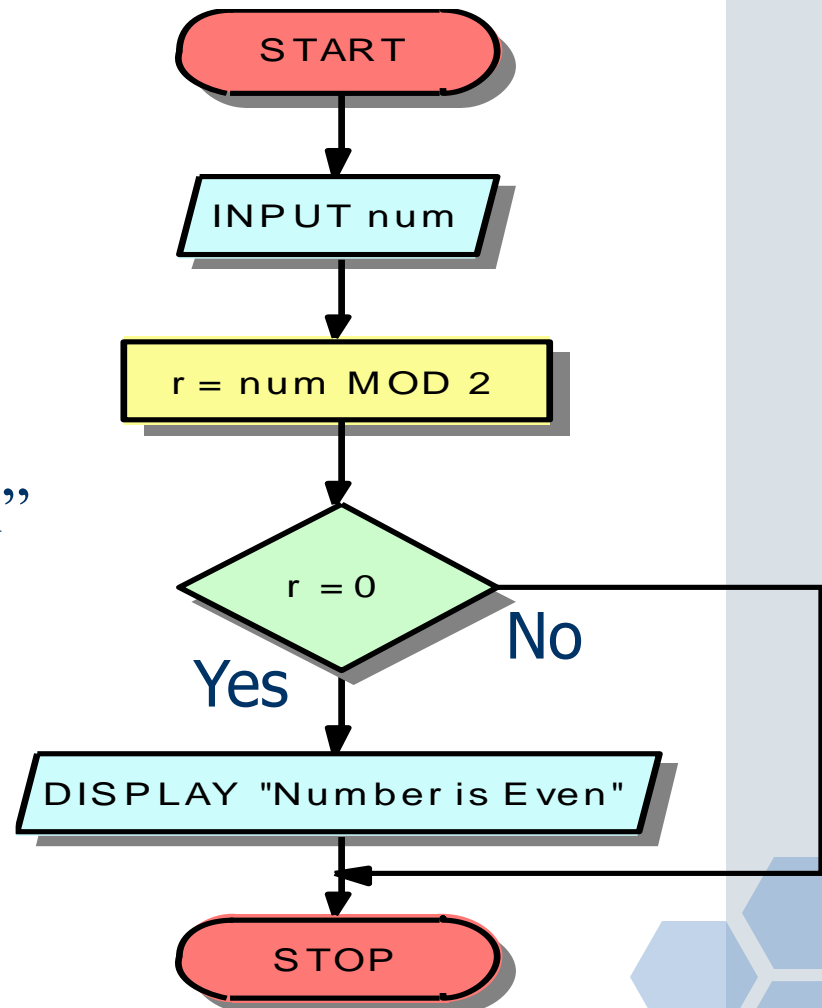


Bài tập 7

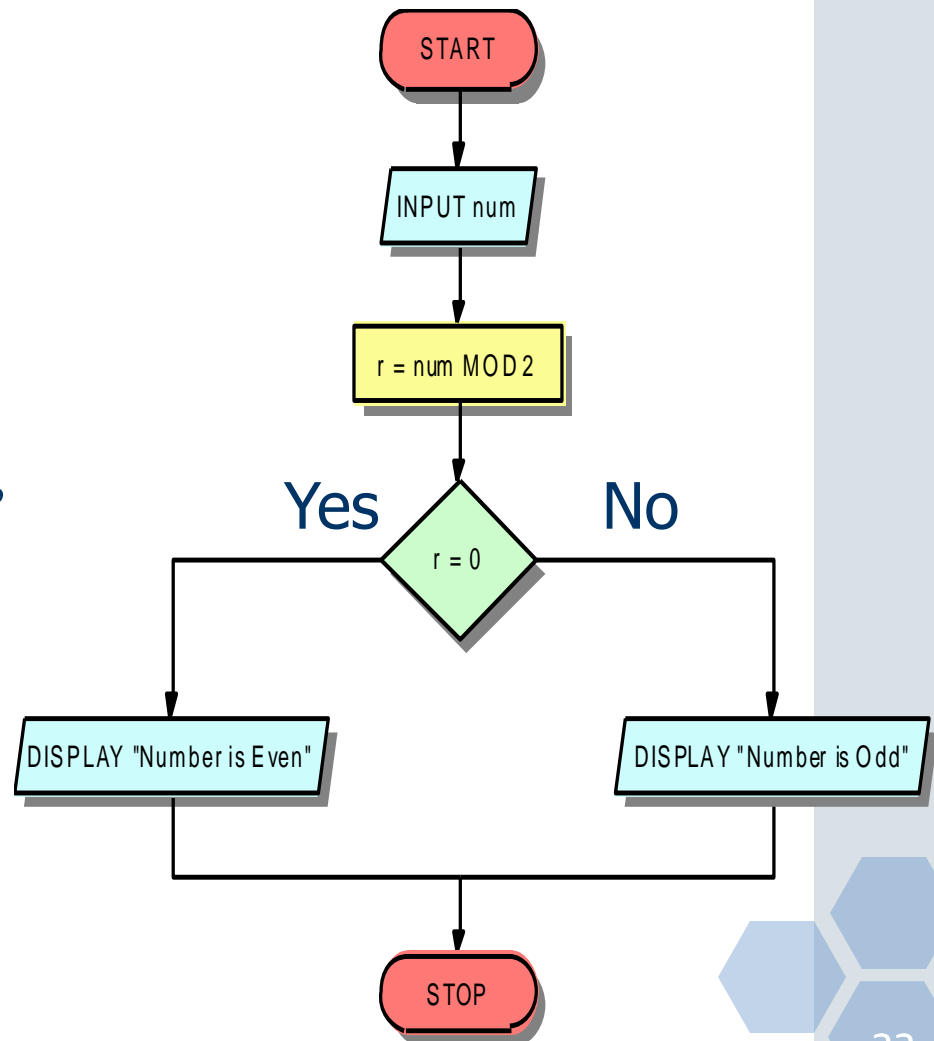


Cấu trúc IF

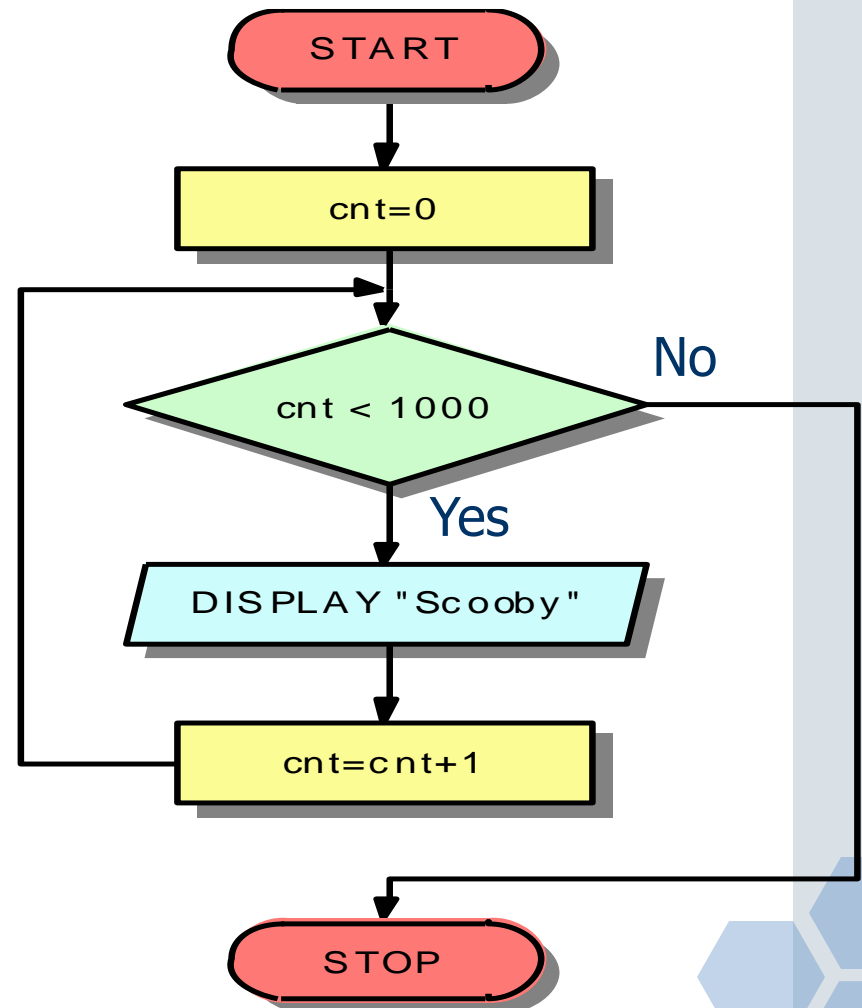
```
BEGIN  
INPUT num  
r = num MOD 2  
IF r=0  
    Display “Number is even”  
END IF  
END
```

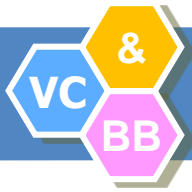


```
BEGIN  
INPUT num  
r=num MOD 2  
IF r=0  
    DISPLAY “Even Number”  
ELSE  
    DISPLAY “Odd Number”  
END IF  
END
```



```
BEGIN  
cnt=0  
WHILE (cnt < 1000)  
DO  
    DISPLAY "Scooby"  
    cnt=cnt+1  
END DO  
END
```

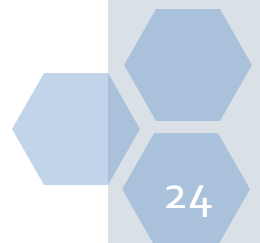


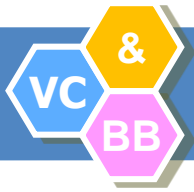


Độ phức tạp của thuật toán

❖ Phân tích thuật toán

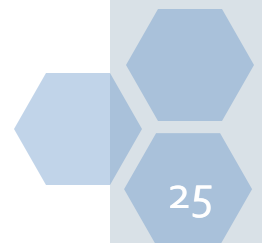
- Tính đúng
- Tính đơn giản
- Không gian nhớ
- **Thời gian chạy của thuật toán**

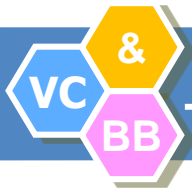




Độ phức tạp của thuật toán

- ❖ Thời gian chạy của thuật toán
 - Đánh giá như thế nào
 - Thực nghiệm
 - Xấp xỉ

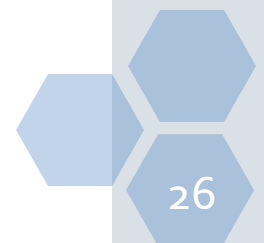


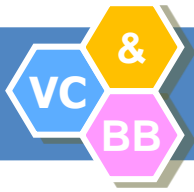


Độ phức tạp của thuật toán

❖ Thực nghiệm

- Chịu sự hạn chế của ngôn ngữ lập trình
- Ảnh hưởng bởi trình độ của người cài đặt
- Chọn được các bộ dữ liệu thử đặc trưng cho tất cả tập các dữ liệu vào của thuật toán: khó khăn và tốn nhiều chi phí
- Phụ thuộc nhiều vào phần cứng





Độ phức tạp của thuật toán

❖ Xấp xỉ tiệm cận

- Cách thông dụng nhất để đánh giá một thuật toán là ký hiệu tiệm cận gọi là Big-O

- *Định nghĩa toán học của Big-O:*

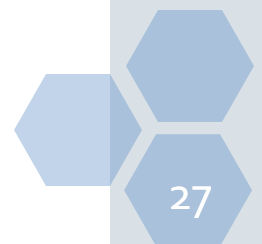
Cho f và g là hai hàm từ tập các số nguyên hoặc số thực đến số thực. Ta nói $f(x)$ là $O(g(x))$ nếu tồn tại hằng số C và k sao cho: $|f(x)| \leq C |g(x)|$ với mọi $x > k$

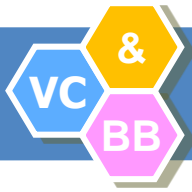
- Ví dụ, hàm $f(x) = x^2 + 3x + 2$ là $O(x^2)$

Thật vậy, khi $x > 2$ thì $x < x^2$ và $2 < 2x^2$

Do đó $x^2 + 3x + 2 < 6x^2$

Nghĩa là ta chọn được $C = 6$ và $k = 2$





Độ phức tạp của thuật toán

❖ Một số kết quả Big-O quan trọng:

■ Hàm đa thức:

- $f(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$
- Khi đó $f(x)$ là $O(x^n)$

■ Hàm giai thừa:

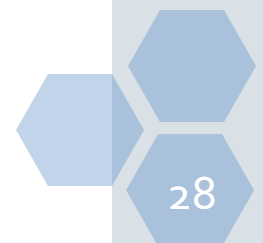
- $f(n) = n!$ là $O(n!)$

■ Logarit của hàm giai thừa:

- $f(n) = \log n!$ là $O(n \log n)$

■ Hàm điều hòa

- $H(n) = 1 + 1/2 + 1/3 + \dots + 1/n$ là $O(\log n)$





Độ phức tạp của thuật toán

❖ Một số lớp thuật toán

Độ phức tạp	Thuật ngữ/tên phân lớp
$O(1)$	Độ phức tạp hằng số
$O(\log n)$	Độ phức tạp logarit
$O(n)$	Độ phức tạp tuyến tính
$O(n \log n)$	Độ phức tạp $n \log n$
$O(n^a)$	Độ phức tạp đa thức
$O(a^n), a > 1$	Độ phức tạp hàm mũ
$O(n!)$	Độ phức tạp giai thừa

❖ Một số lớp thuật toán

$n \setminus \text{Hàm}$	n	$\lg n$	$N \lg n$	n^2	n^3	2^n
1	1	0	0	1	1	2
2	2	1	2	4	8	4
4	n	2	8	16	64	16
8	8	3	24	64	512	256
16	16	4	64	256	4096	65536
32	32	5	160	1024	32768	2,147,483,648



Giới thiệu ngôn ngữ lập trình C

1

Giới thiệu

2

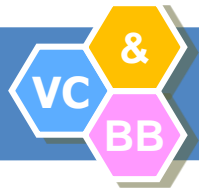
Bộ từ vựng của C

3

Cấu trúc chương trình C

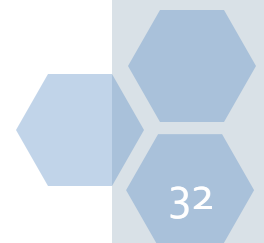
4

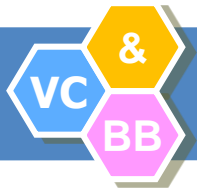
Một số ví dụ minh họa



Lịch sử của ngôn ngữ lập trình C

- ❖ Ngôn ngữ lập trình C ra đời năm 1972, tác giả: Dennis Ritchie.
- ❖ C được tạo ra để sử dụng như một phần căn bản của hệ điều hành UNIX (Ken Thompson, Dennis Ritchie và Douglas McIlroy, 1969)
- ❖ C được sử dụng rộng rãi và có ảnh hưởng lớn đến nhiều ngôn ngữ lập trình hiện đại, trong đó có C++, được xem là mở rộng của C.
- ❖ Là ngôn ngữ lập trình có cấu trúc và phân biệt chữ Hoa - thường (**case sensitive**)





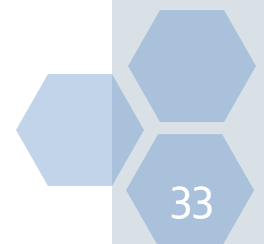
Môi trường lập trình C/C++

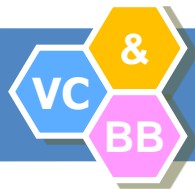
❖ Môi trường C:

- Borland C (còn gọi là Turbo C)
- Borland C++ 3.1 for DOS.

❖ Môi trường C/C++:

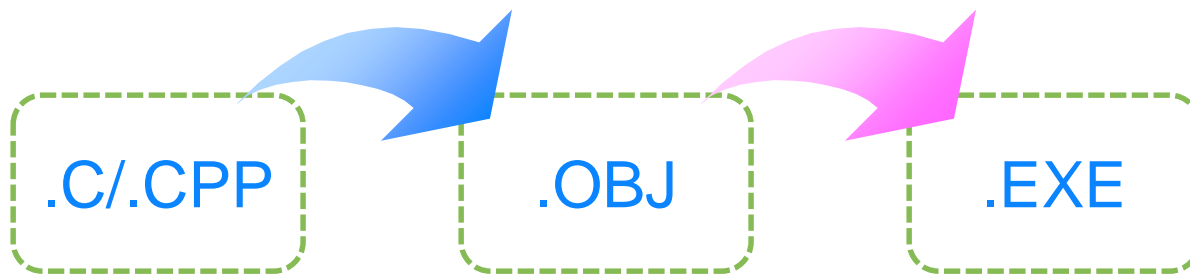
- Dev-C++
- Visual C++ của Microsoft
- Visual Studio
- Code Block

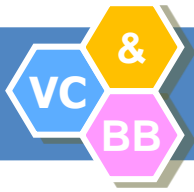




Môi trường lập trình C/C++

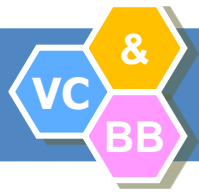
- ❖ Môi trường phát triển tích hợp IDE (Integrated Development Environment)
 - Biên tập chương trình nguồn (Trình EDIT).
 - Biên dịch chương trình (Trình COMPILE).
 - Chạy chương trình nguồn (Trình RUNTIME).
 - Sửa lỗi chương trình nguồn (Trình DEBUG).





Các lĩnh vực ứng dụng của C

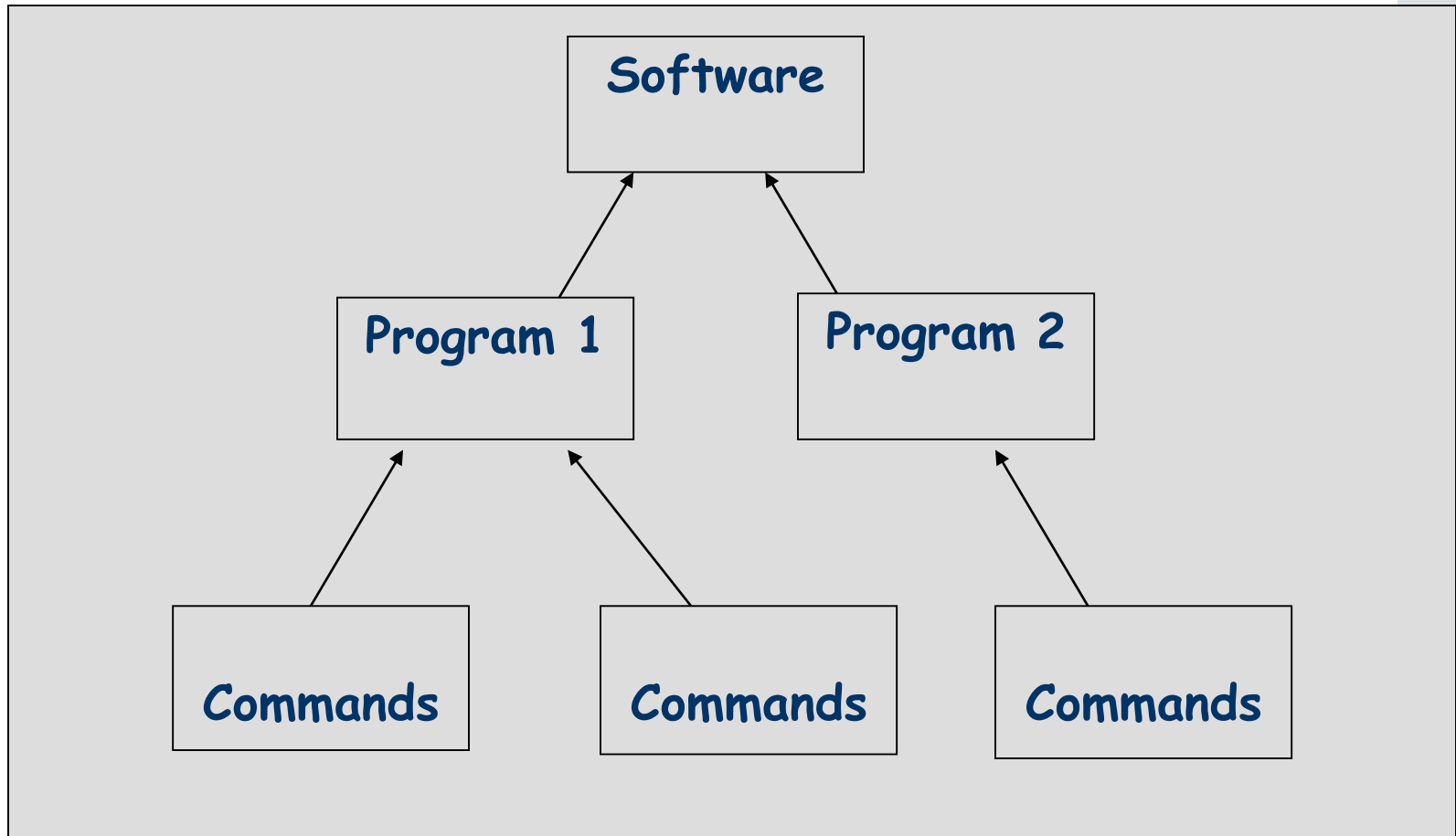
- ❖ C được dùng để lập trình hệ thống
- ❖ Một chương trình hệ thống làm thành một phần hệ điều hành hoặc các tiện ích hỗ trợ của hệ điều hành
- ❖ Hệ điều hành (Operating Systems), trình thông dịch (Interpreters), trình soạn thảo (Editors), trình Hợp Ngữ (Assembly) được gọi là chương trình hệ thống
- ❖ Hệ điều hành UNIX được phát triển dựa vào C
- ❖ Có các trình biên dịch dành cho hầu hết các loại hệ thống PC

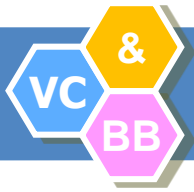


Đặc điểm của C

- C có 32 từ khóa
- Những từ khóa này kết hợp với cú pháp của C hình thành ngôn ngữ C
- Các quy tắc được áp dụng cho các chương trình C
 - Tất cả từ khóa là chữ thường
 - Đoạn mã trong chương trình C có phân biệt chữ thường, chữ hoa, do while khác DO WHILE
 - Từ khóa không thể dùng đặt tên biến (variable name) hoặc tên hàm (function name)

```
main()  
{  
  /* This is a sample Program*/  
    int i,j;  
    i=100;  
    j=200;  
    :  
}
```

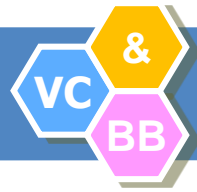




Cấu trúc chương trình C

main():

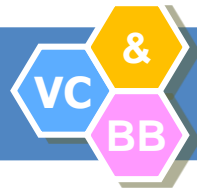
- ❖ Chương trình C được chia nhỏ thành những đơn vị gọi là hàm
- ❖ Cho dù có bao nhiêu hàm trong chương trình, Hệ điều hành luôn trao quyền điều khiển cho hàm main() khi một chương trình C được thực thi.
- ❖ Theo sau tên hàm là dấu ngoặc đơn
- ❖ Dấu ngoặc đơn có thể có chứa hay không chứa những tham số



Cấu trúc chương trình C (tt.)

Dấu phân cách {...}

- ❖ Sau phần đầu hàm là dấu ngoặc xoắn mở { , nó cho biết việc thi hành lệnh trong hàm bắt đầu
- ❖ Tương tự, dấu ngoặc xoắn đóng } sau câu lệnh cuối cùng trong hàm chỉ ra điểm kết thúc của hàm



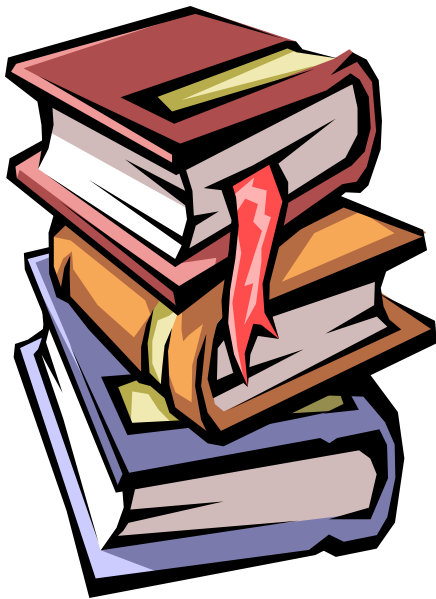
Cấu trúc chương trình C (tt.)

Dấu kết thúc câu lệnh ... ;

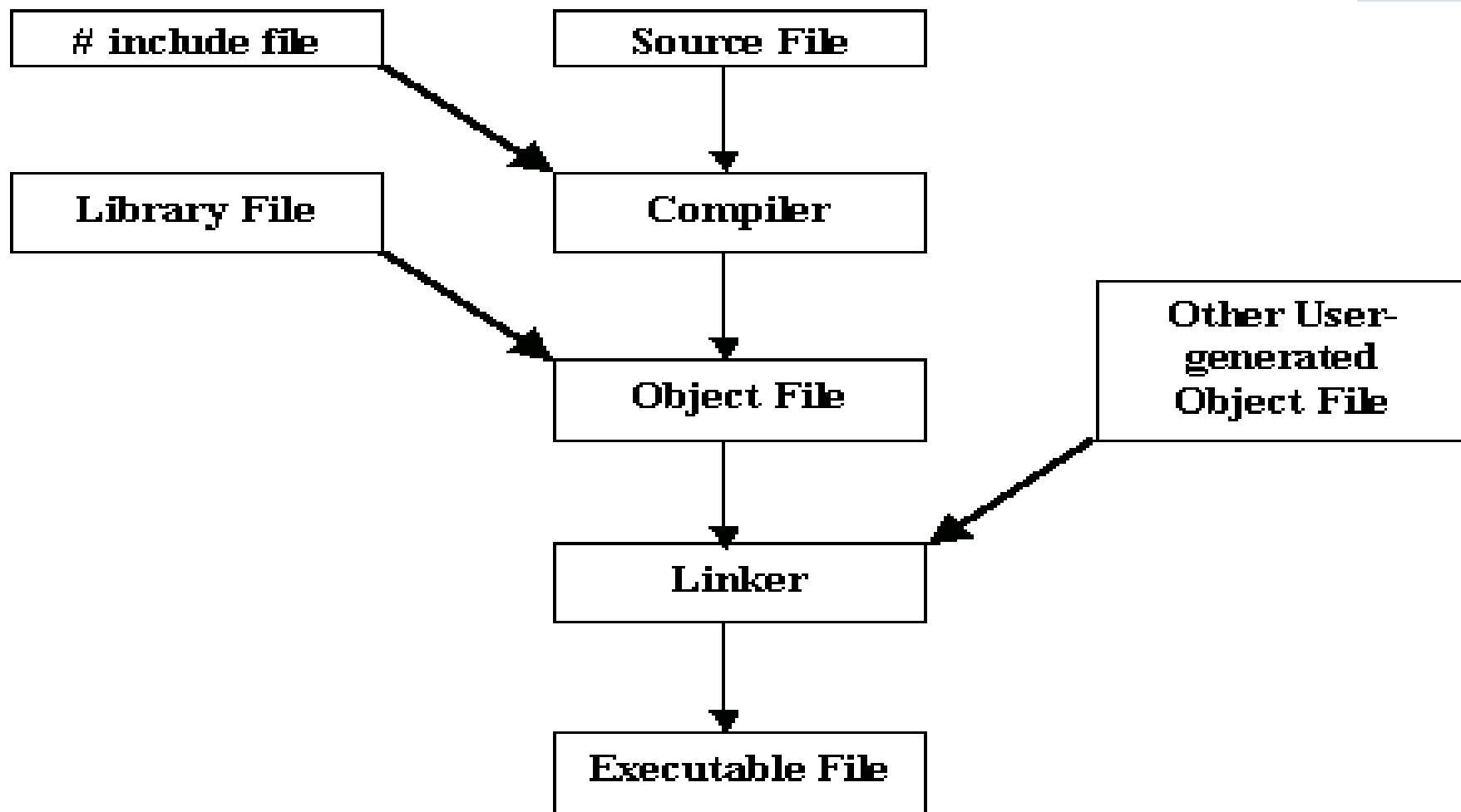
- ❖ Một câu lệnh trong C được kết thúc bằng dấu chấm phẩy ;
- ❖ Trình biên dịch C không hiểu việc xuống dòng, khoảng trắng hay tab
- ❖ Một câu lệnh không kết thúc bằng dấu chấm phẩy sẽ được xem như dòng lệnh lỗi trong C

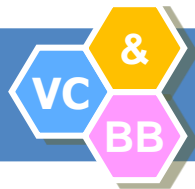
// Dòng chú thích
/*Dòng chú thích*/

- ❖ Những chú thích thường được viết để mô tả công việc của một lệnh đặc biệt, một hàm hay toàn bộ chương trình
- ❖ Trình biên dịch sẽ bỏ qua phần chú thích
- ❖ Dòng chú thích bắt đầu bằng //
- ❖ Trong trường hợp chú thích nhiều dòng, nó sẽ bắt đầu bằng ký hiệu /* và kết thúc là */



- Tất cả trình biên dịch C đều chứa một thư viện hàm chuẩn
- Một hàm được viết bởi lập trình viên có thể được đặt trong thư viện và được dùng khi cần thiết
- Một số trình biên dịch cho phép thêm hàm vào thư viện chuẩn
- Một số trình biên dịch yêu cầu tạo một thư viện riêng

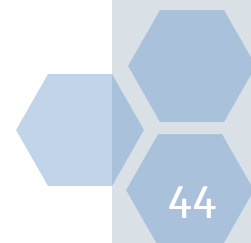


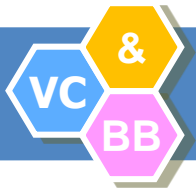


Bộ từ vựng của C

❖ Các ký tự được sử dụng

- Bộ chữ cái 26 ký tự Latinh **A, B, C, ..., Z, a, b, c, ..., z**
- Bộ chữ số thập phân : **0, 1, 2, ..., 9**
- Các ký hiệu toán học : **+ - * / = < > ()**
- Các ký tự đặc biệt : **. , : ; [] % \ # \$ '**
- Ký tự gạch nối **_** và khoảng trắng **' '**

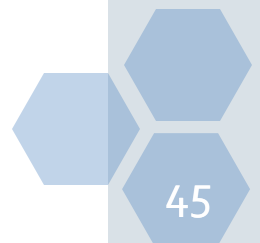


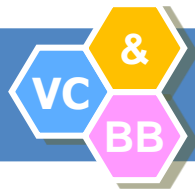


Bộ từ vựng của C

❖ Từ khóa (keyword)

- Các từ **dành riêng** trong ngôn ngữ.
- **Không** thể sử dụng từ khóa để đặt tên cho biến, hàm, tên chương trình con.
- Một số từ khóa thông dụng:
 - const, enum, signed, struct, typedef, unsigned...
 - char, double, float, int, long, short, void
 - case, default, else, if, switch
 - do, for, while
 - break, continue, goto, return





Bộ từ vựng của C

❖ Tên/Định danh (Identifier)

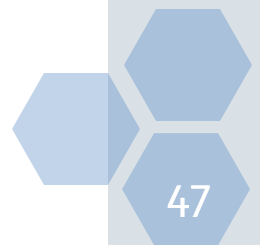
- Một dãy ký tự dùng để chỉ tên một hằng số, hằng ký tự, tên một biến, một kiểu dữ liệu, một hàm một hay thủ tục.
- Không được trùng với các từ khóa và được tạo thành từ các chữ cái và các chữ số nhưng bắt buộc chữ đầu phải là chữ cái hoặc _.
- Số ký tự tối đa trong một tên là 255 ký tự và được dùng ký tự _ chen trong tên nhưng không cho phép chen giữa các khoảng trắng.



Bộ từ vựng của C

❖ Ví dụ Tên/Định danh (Identifier)

- Các tên hợp lệ: GiaiPhuongTrinh, Bai_Tap1
- Các tên không hợp lệ: 1A, Giai Phuong Trinh
- **Phân biệt chữ hoa chữ thường**, do đó các tên sau đây khác nhau:
 - A, a
 - BaiTap, baitap, BAITAP, bAltaP...





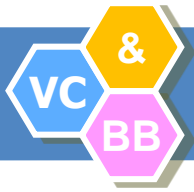
Bộ từ vựng của C

❖ Dấu chấm phẩy ;

- Dùng để phân cách các câu lệnh.
- Ví dụ: `printf("Hello World!"); printf("\n");`

❖ Câu chú thích

- Từ vị trí cụ thể đến cuối dòng: dùng `//` ngay tại vị trí đó.
- Ghi chú gồm nhiều dòng: dùng cặp ký tự `/*` ý ngay trước dòng đầu tiên và cặp ký tự `*/` ngay sau dòng cuối cùng
- Ví dụ: `/*Ho & Ten: NVA*/, // MSSV: 0712078`



Các khái niệm

❖ Hằng ký tự và hằng chuỗi

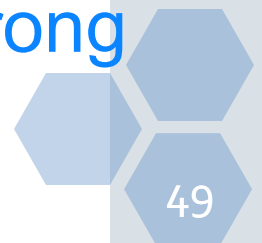
- Hằng ký tự: 'A', 'a', ...
- Hằng chuỗi: "Hello World!", "Nguyen Van A"

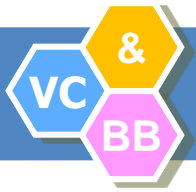
❖ Lệnh:

- Lệnh thực hiện một chức năng nào đó (khai báo, gán, xuất, nhập,...) và được kết thúc bằng dấu chấm phẩy (;)

❖ Khối lệnh:

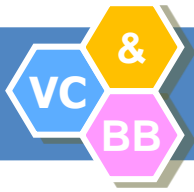
- Khối lệnh gồm nhiều lệnh và được đặt trong cặp dấu ngoặc { }





Qui ước viết lệnh trong C

- ❖ Mỗi lệnh nằm trên một dòng. Cuối dòng lệnh PHẢI có dấu chấm phẩy (;)
- ❖ Không nên đặt nhiều lệnh trên cùng một dòng, ngay cả các khai báo biến. Trừ cặp lệnh nhập xuất có thể viết trên cùng một dòng.
- ❖ Phải sử dụng Tab để trình bày lệnh.
- ❖ Phải khai báo biến trước khi sử dụng.

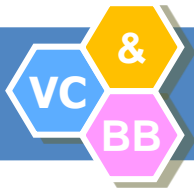


Cấu trúc chương trình C

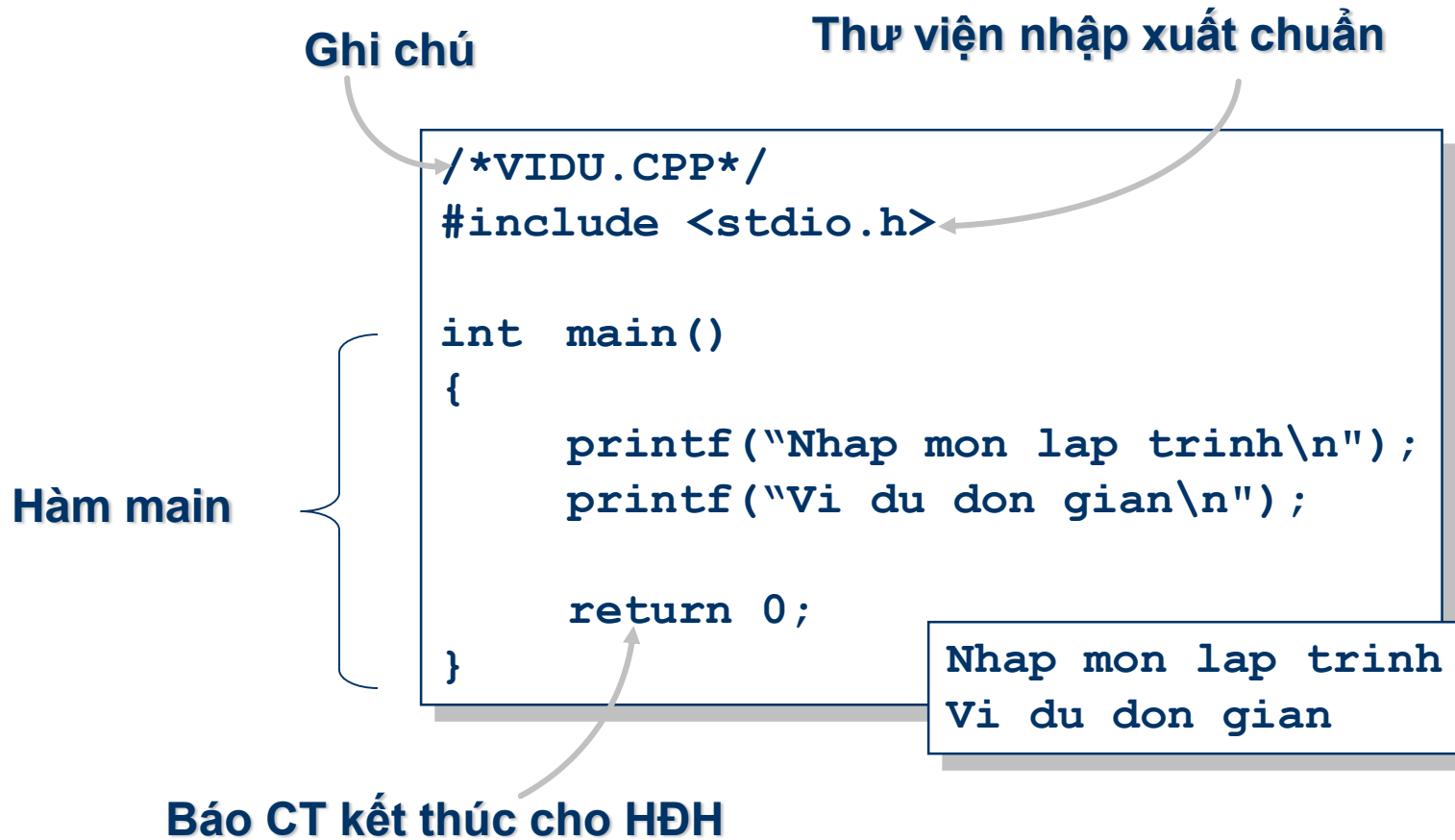
```
#include "...";    // Khai báo file tiêu đề

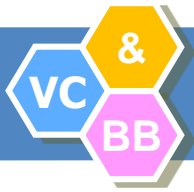
int x;             // Khai báo biến hàm
void Nhap();       // Khai báo hàm

void main()        // Hàm chính
{
    // Các lệnh và thủ tục
}
```



Ví dụ 1

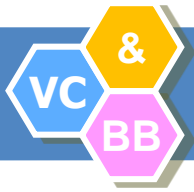




Ví dụ 2

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

void main()
{
    int x, y, tong;
    printf("Nhap hai so nguyen: ");
    scanf("%d%d", &x, &y);
    tong = x + y;
    printf("Tong hai so la %d", tong);
    getch();
}
```



Ví dụ 3

Khai báo 2 biến số nguyên, “a” và “b”

Nhập 2 số nguyên vào a và b

Viết các biểu thức “a”, “b” và “a-b” theo định dạng %i

```
#include <stdio.h>

int main(void)
{
    int a, b;

    printf("Nhap 2 so ngguyen: ");
    scanf("%i %i", &a, &b);

    printf("%i - %i = %i\n", a, b, a - b);

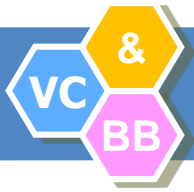
    return 0;
}
```

```
Nhap 2 so nguyen: 21 17
21 - 17 = 4
```



Một số lưu ý từ ví dụ

- ❖ Phần ghi chú được trình biên dịch bỏ qua
- ❖ Phân biệt chữ in hoa và chữ in thường
- ❖ Câu lệnh luôn được kết thúc bằng dấu ;
- ❖ Chuỗi ký tự phải ghi giữa cặp nháy kép "
- ❖ In xuống dòng dùng ký tự \n
- ❖ Chương trình có một hàm main



❖ Tên/Định danh (Identifier)

- Một dãy ký tự dùng để chỉ tên một hằng số, hằng ký tự, tên một biến, một kiểu dữ liệu, một hàm một hay thủ tục.
- Không được trùng với các từ khóa và được tạo thành từ các chữ cái và các chữ số nhưng bắt buộc chữ đầu phải là chữ cái hoặc _.
- Số ký tự tối đa trong một tên là 255 ký tự và được dùng ký tự _ chen trong tên nhưng không cho phép chen giữa các khoảng trắng.