

# Giáo Trình Bài Tập Kỹ Thuật Lập Trình



## LỜI MỞ ĐẦU



Khi bắt đầu làm quen với ngôn ngữ lập trình – Cụ thể là ngôn ngữ C – Sinh Viên thường gặp khó khăn trong việc chuyển vấn đề lý thuyết sang cài đặt cụ thể trên máy. Sách **“Giáo Trình Bài Tập Kỹ Thuật Lập Trình”** nhằm cung cấp cho các Học Sinh - Sinh Viên Trường CD Công Nghệ Thông Tin Tp. Hồ Chí Minh hệ thống các bài tập, những **kỹ năng thực hành cơ bản và nâng cao về ngôn ngữ lập trình C**. Cuốn sách này được xem như tài liệu hướng dẫn từng bước cho Học Sinh - Sinh Viên của Trường trong việc học và áp dụng kiến thức lý thuyết trên lớp một cách thành thạo và sâu rộng.

Giáo trình được chia thành **10 chương** theo từng nội dung kiến thức, kèm theo **Các đề thi mẫu** và **1 phụ lục** hướng dẫn viết chương trình, chuẩn đoán lỗi và sửa lỗi. Mỗi chương gồm 2 phần:

- ❖ **Phần lý thuyết:** được tóm tắt ngắn gọn với đầy đủ ví dụ minh họa kèm theo.
- ❖ **Phần bài tập:** với nhiều bài tập được chia làm hai mức độ cơ bản và luyện tập nâng cao, bài tập có đánh dấu \* là bài tập khó dành cho sinh viên luyện tập thêm.
- ❖ **Phần kết luận:** Tóm tắt nội dung và các thao tác mà sinh viên cần nắm hay những lưu ý của chương đó.

Trong quá trình biên soạn, chúng tôi đã cố gắng trích lọc những kiến thức rất cơ bản, những lỗi hay gặp đối với người mới lập trình. Bên cạnh đó chúng tôi cũng bổ sung thêm một số bài tập nâng cao để rèn luyện thêm kỹ năng lập trình.

Tuy nhiên, chủ đích chính của giáo trình này là phục vụ cho một môn học nên chắc chắn không thể tránh khỏi những thiếu sót, vì thế, rất mong nhận được những góp ý quý báu của các thầy cô, các đồng nghiệp và các bạn Học Sinh – Sinh Viên để giáo trình này ngày càng hoàn thiện hơn.

Chân thành cảm ơn.

## LỊCH TRÌNH THỰC HÀNH



**Tổng thời gian: 90 tiết.**

STT	NỘI DUNG	SỐ TIẾT
1	<b>Chương 1:</b> Lưu đồ thuật toán	<b>03</b>
2	<b>Chương 2:</b> Cấu trúc điều khiển	<b>06</b>
3	<b>Chương 3:</b> Hàm con	<b>12</b>
4	<b>Chương 4:</b> Mảng một chiều	<b>24</b>
5	<b>Chương 5:</b> Chuỗi ký tự	<b>06</b>
6	<b>Chương 6:</b> Mảng hai chiều	<b>12</b>
7	<b>Chương 7:</b> Kiểu dữ liệu có cấu trúc	<b>12</b>
8	<b>Chương 8:</b> Tập tin	<b>06</b>
9	<b>Chương 9:</b> Đệ qui	<b>06</b>
10	<b>Chương 10:</b> Hướng dẫn lập trình bằng phương pháp <b>Project</b>	<b>03</b>

# CHƯƠNG 1 LƯU ĐỒ THUẬT TOÁN (FLOWCHART)

Các ký hiệu biểu diễn lưu đồ thuật toán, cách biểu diễn các cấu trúc điều khiển rẽ nhánh, cấu trúc lặp và các kỹ thuật liên quan đến lưu đồ thuật toán.

## I. TÓM TẮT LÝ THUYẾT

### I.1. Khái niệm

Lưu đồ thuật toán là công cụ dùng để **biểu diễn thuật toán**, việc mô tả **nhập** (input), dữ liệu **xuất** (output) và luồng xử lý thông qua các **ký hiệu hình học**.

### I.2. Phương pháp duyệt

- Duyệt từ trên xuống.
- Duyệt từ trái sang phải.

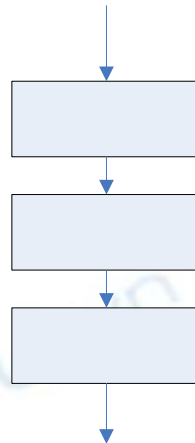
### I.3. Các ký hiệu

STT	KÝ HIỆU	DIỄN GIẢI
1		Bắt đầu chương trình
2		Kết thúc chương trình
3		Luồng xử lý
4		Điều khiển lựa chọn
5		Nhập
6		Xuất
7		Xử lý, tính toán hoặc gán
8		Trả về giá trị (return)
9		Điểm nối liên kết tiếp theo (Sử dụng khi lưu đồ vượt quá trang)

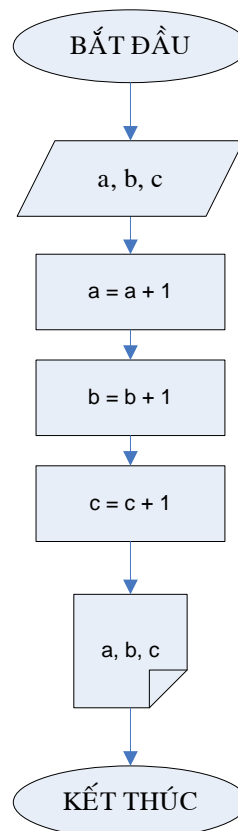
## I.4. Các cấu trúc điều khiển cơ bản

### a. Cấu trúc tuần tự

Tuần tự thực thi tiến trình. Mỗi lệnh được thực thi theo một chuỗi **từ trên xuống**, xong lệnh này rồi chuyển xuống lệnh kế tiếp.



Ví dụ: Nhập vào 3 số nguyên  $a, b, c$  và xuất ra màn hình với giá trị của mỗi số tăng lên 1.

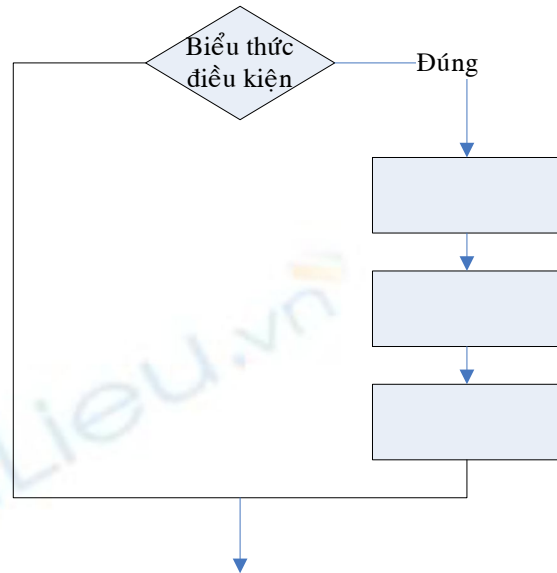


### b. Cấu trúc lựa chọn

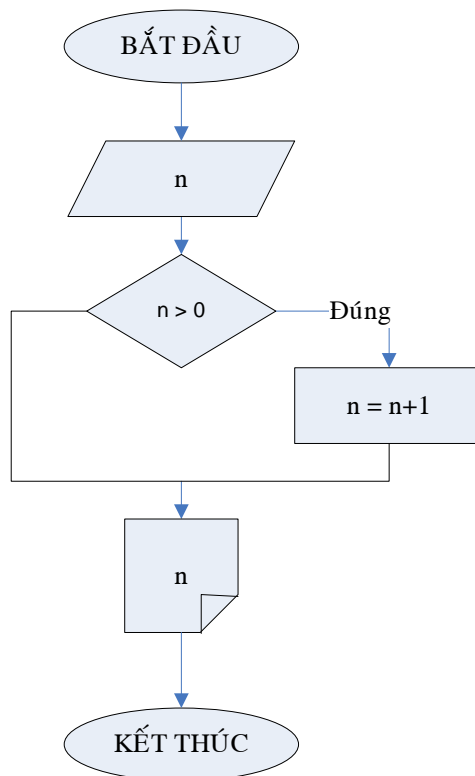
Điểm quyết định cho phép **chọn một trong hai trường hợp**.

- **if**

Chỉ xét trường hợp đúng.

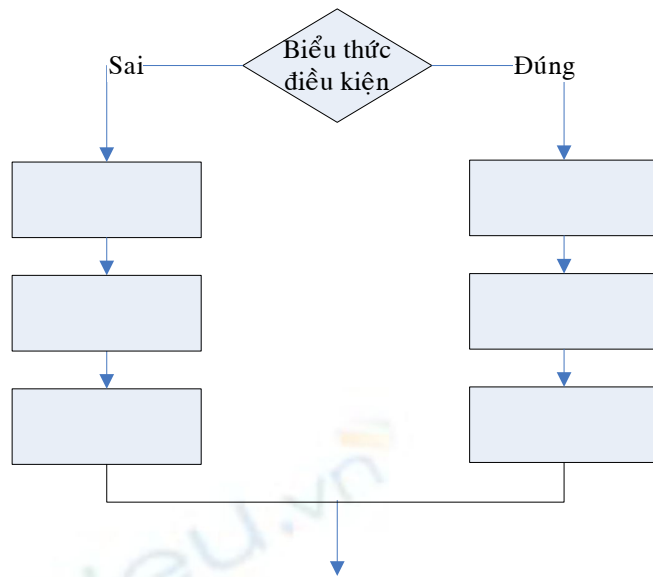


*Ví dụ: Nhập vào số nguyên  $n$ . Kiểm tra nếu  $n > 0$  tăng  $n$  lên 1 đơn vị. Xuất kết quả.*

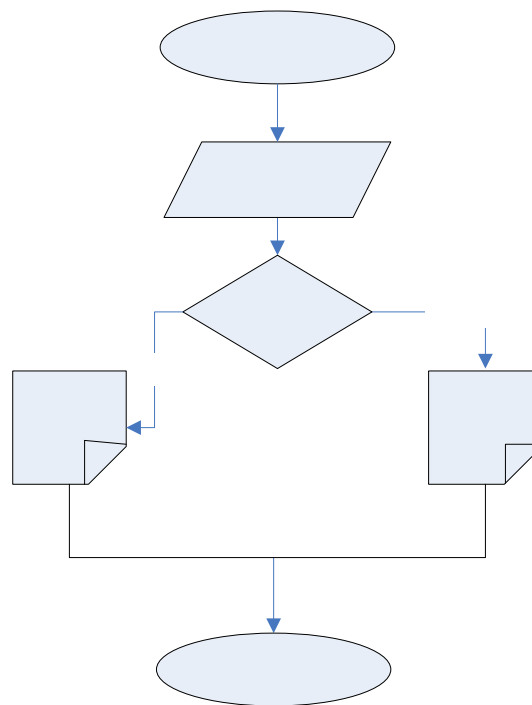


- **if...else**

Xét trường hợp đúng và trường hợp sai.



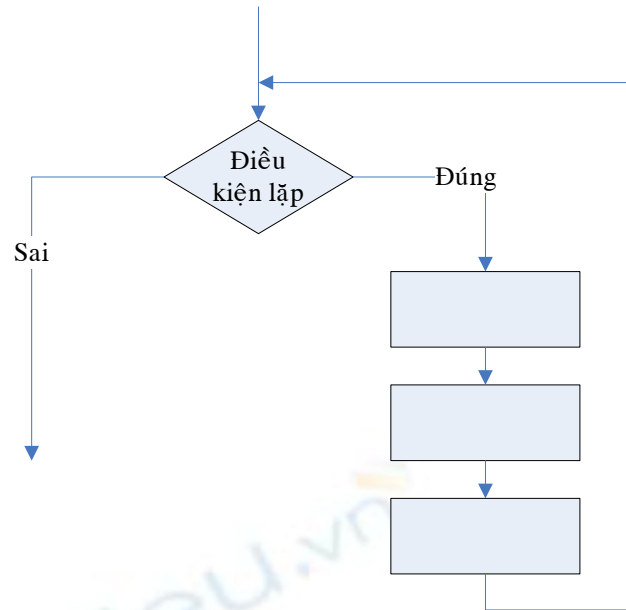
*Ví dụ: Nhập vào số nguyên  $n$ . Kiểm tra nếu  $n$  chẵn xuất ra màn hình " $n$  chẵn", ngược lại xuất " $n$  lẻ".*



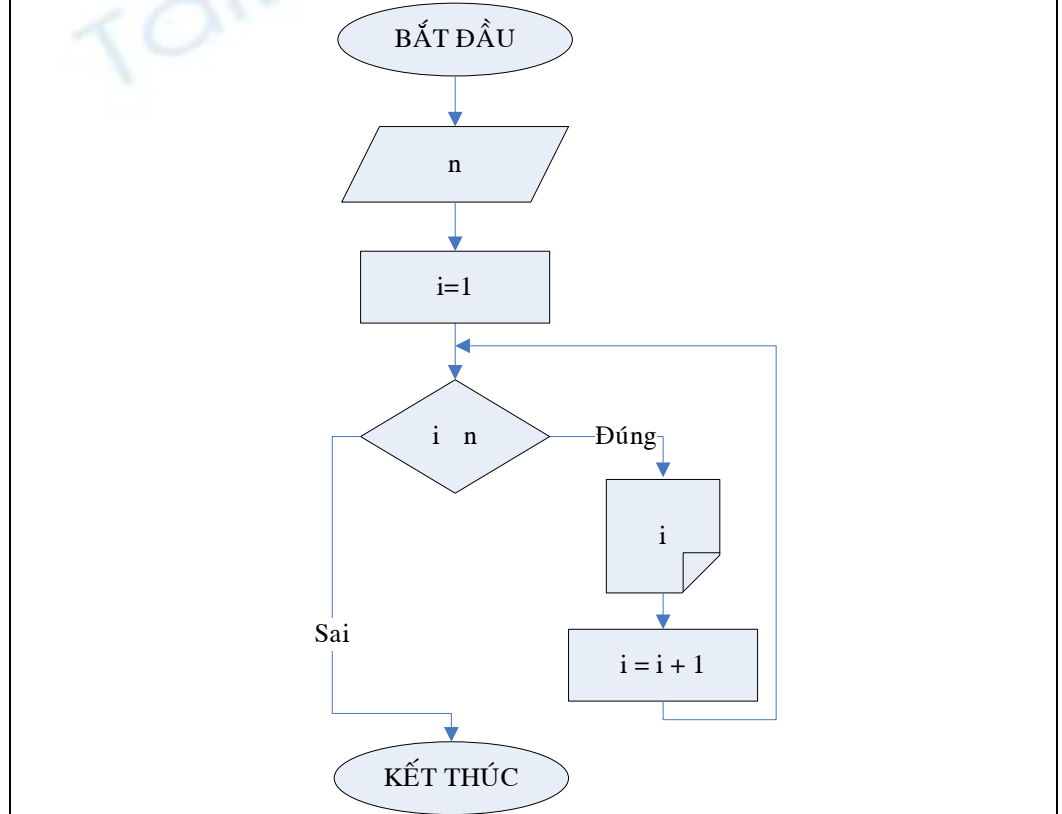
**c. Cấu trúc lặp**

Thực hiện liên tục 1 lệnh hay tập lệnh với số lần lặp dựa vào điều kiện.  
Lặp sẽ kết thúc khi điều kiện được thỏa.

- **for / while (Kiểm tra điều kiện trước khi lặp)**

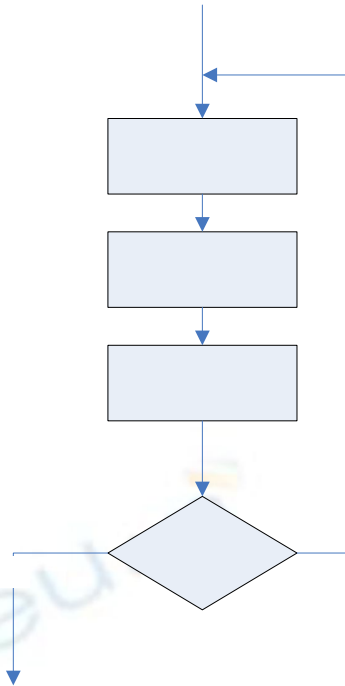


*Ví dụ: Nhập vào số nguyên  $n$ . Xuất ra màn hình từ 1 đến  $n$ .*

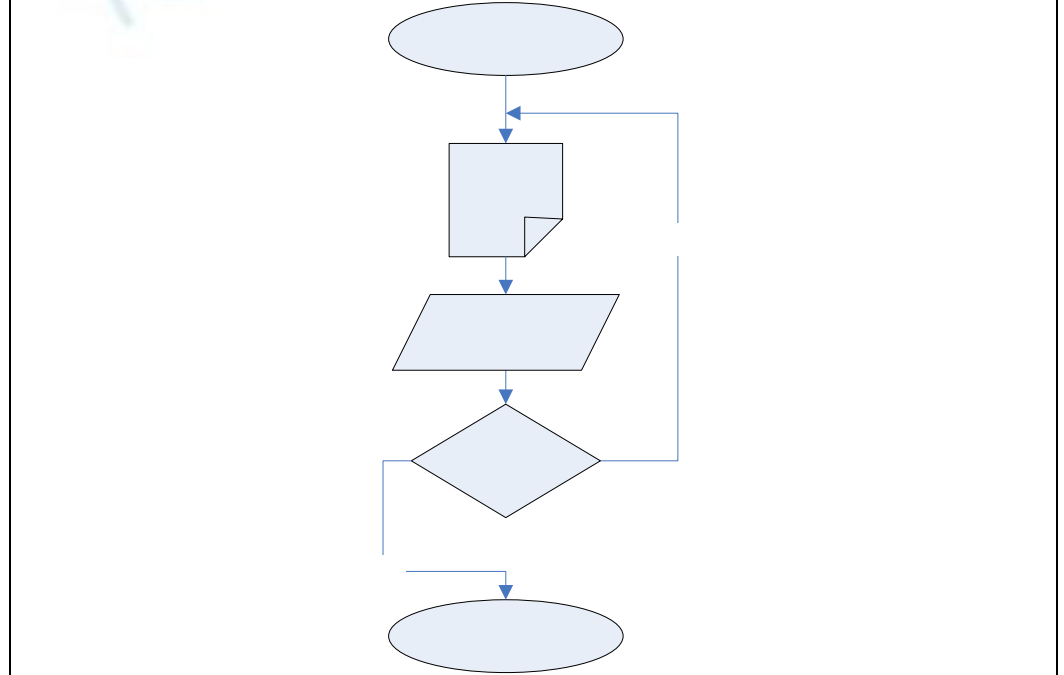




- **do ... while (Thực hiện lặp trước khi kiểm tra điều kiện)**

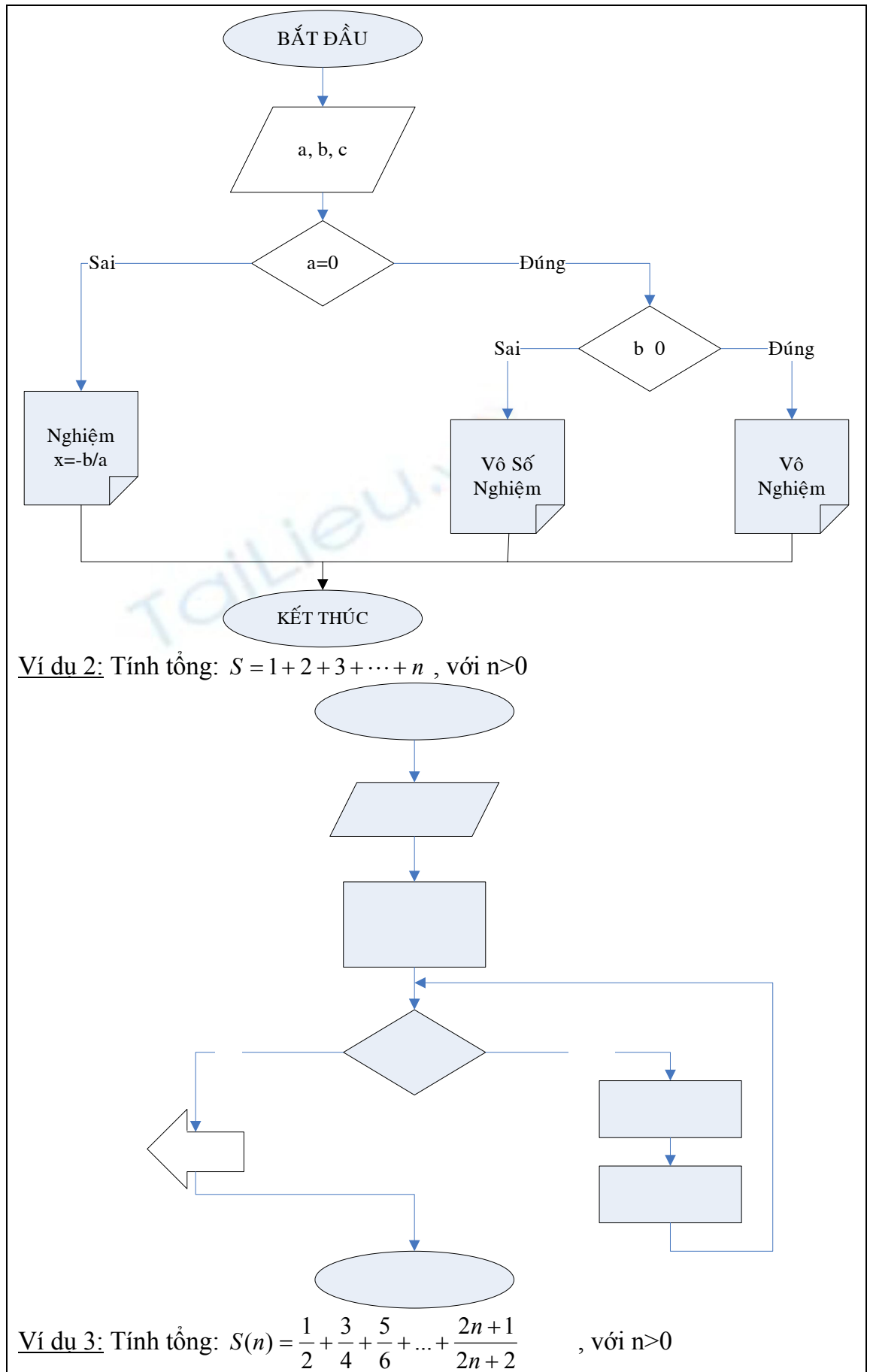


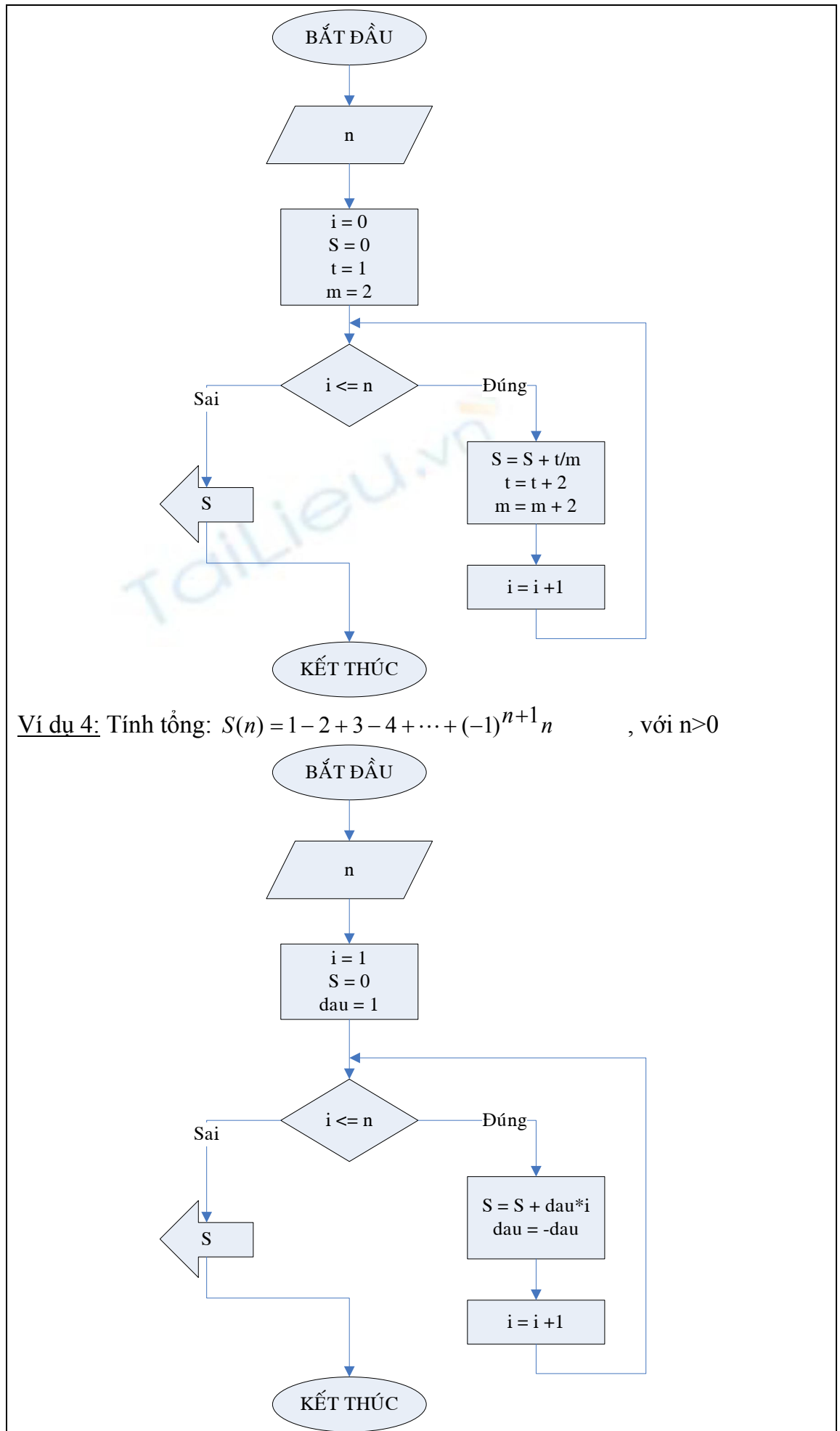
Ví dụ: Nhập vào số nguyên dương  $n$ . Nếu nhập sai yêu cầu nhập lại.



**d. Các ví dụ**

Ví dụ 1: Giải và biện luận phương trình:  $ax+b=0$ .





## II. BÀI TẬP

### Vẽ lưu đồ thuật toán sau

#### II.1. Bài tập cơ bản

1. Nhập vào hai số  $x, y$ . Xuất ra màn hình tổng, hiệu, tích, thương của hai số trên.
2. Nhập vào số nguyên  $n$ , kiểm tra xem  $n$  chẵn hay lẻ và xuất ra màn hình.
3. Nhập vào ba cạnh  $a, b, c$  của tam giác. Xuất ra màn hình tam giác đó thuộc loại tam giác gì? (Thường, cân, vuông, đều hay vuông cân).
4. Nhập vào số nguyên  $n$ . Xuất ra  $n$  màn hình (Nếu  $n$  chẵn thì gấp đôi giá trị).
5. Nhập vào số nguyên  $n$ . Nếu  $n > 5$  thì tăng  $n$  lên 2 đơn vị và trả về giá trị  $n$ , ngược lại trả về giá trị 0.
6. Tính  $n!$ , với  $n \geq 0$
7. Tính  $P(n) = 1.3.5 \dots (2n+1)$ , với  $n \geq 0$
8. Tính  $S(n) = 1 + 3 + 5 + \dots + (2 \times n + 1)$ , với  $n \geq 0$
9. Tính  $S(n) = 1 - 2 + 3 - 4 + \dots + (-1)^{n+1} n$ , với  $n > 0$
10. Tính  $S(n) = 1 + 1.2 + 1.2.3 + \dots + 1.2.3 \dots n$ , với  $n > 0$
11. Tính  $S(n) = 1^2 + 2^2 + 3^2 + \dots + n^2$ , với  $n > 0$
12. Tính  $S(n) = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}$ , với  $n > 0$
13. (\*) Tính  $S(n) = 1 + \frac{1}{1+2} + \frac{1}{1+2+3} + \dots + \frac{1}{1+2+3+\dots+n}$ , với  $n > 0$
14. Tính  $P(x, y) = x^y$ .
15. Tính  $S(n) = 1 + (1+2) + (1+2+3) + \dots + (1+2+3+\dots+n)$ , với  $n > 0$
16. Cho số nguyên  $n$ . Tính trị tuyệt đối của  $n$ .
17. Cho số nguyên dương  $n$  gồm  $k$  chữ số. Tìm chữ số có giá trị lớn nhất.
18. Đếm số lượng ước số chẵn của số nguyên dương  $n$ .
19. In ra chữ số đầu tiên của số nguyên dương  $n$  gồm  $k$  chữ số.
20. Cho 2 số nguyên dương  $a, b$ . Tìm USCLN của  $a$  và  $b$ .
21. Cho 2 số nguyên dương  $a, b$ . Tìm BSCNN của  $a$  và  $b$ .
22. Cho số nguyên dương  $x$ . Kiểm tra xem  $x$  có phải là số nguyên tố không?
23. Cho số nguyên dương  $x$ . Kiểm tra  $x$  có phải là số chính phương không?
24. Cho số nguyên dương  $x$ . Kiểm tra xem  $x$  có phải là số hoàn thiện không?

## II.2. Bài tập luyện tập và nâng cao

25. Tính  $S(n) = 1 + 2^2 + 3^3 + \dots + n^n$ , với  $n \geq 0$

26. Tính  $S(n) = \frac{1}{2} + \frac{2}{3} + \frac{3}{4} + \dots + \frac{n}{n+1}$ , với  $n > 0$

27. Tính  $S(n) = 1 + \frac{1}{2!} + \frac{1}{3!} + \dots + \frac{1}{n!}$ , với  $n > 0$

28. Tính  $S(n) = 1 + \frac{1+2}{2!} + \frac{1+2+3}{3!} + \dots + \frac{1+2+3+\dots+n}{n!}$ , với  $n > 0$

29. Giải và biện luận phương trình:  $ax^2 + bx + c = 0$

30. Giải và biện luận phương trình:  $ax^4 + bx^2 + c = 0$

31. (\*) Tính  $S(n) = \sqrt{n + \sqrt{(n-1) + \sqrt{(n-2) + \dots + \sqrt{1}}}}$ , với  $n > 0$

32. (\*\*) Tính  $S(n) = \sqrt{1 + \sqrt{2 + \sqrt{3 + \dots + \sqrt{n}}}}$ , với  $n > 0$

## III. KẾT LUẬN

Lưu đồ thuật toán rất hữu ích trong việc mô tả cách giải quyết của một bài toán. Việc mô tả này rất trực quan thông qua các ký hiệu hình học, đây là giai đoạn đầu tiên trước khi bắt tay vào lập trình trên một ngôn ngữ lập trình cụ thể.

Khi xây dựng lưu đồ thuật toán, chúng ta cần chú ý một vài điểm sau:

- ❖ Một lưu đồ phải có điểm **bắt đầu** và điểm **kết thúc** (điều kiện kết thúc).
- ❖ Phải có **dữ liệu vào**, **dữ liệu ra** sau khi xử lý tính toán.
- ❖ Tại mỗi vị trí quyết định lựa chọn rẽ nhánh phải ghi rõ điều kiện **đúng hoặc sai** thì đi theo nhánh nào.

## CHƯƠNG 2    **CẤU TRÚC ĐIỀU KHIỂN**

*Tìm hiểu và cài đặt các cấu trúc rẽ nhánh, lựa chọn, lặp và các ký hiệu phép toán trong ngôn ngữ C. Mô tả cách hoạt động và hướng dẫn chạy từng bước chương trình.*

### **I. TÓM TẮT LÝ THUYẾT**

#### **I.1. Các ký hiệu**

STT	KÝ HIỆU	DIỄN GIẢI	VÍ DỤ
1	{ }	Bắt đầu và kết thúc hàm hay khối lệnh.	void main() {  }
2	;	Kết thúc khai báo biến, một lệnh, một lời gọi hàm, hay khai báo nguyên mẫu hàm.	int x; void NhapMang(int a[], int &n);
3	//	Chú thích (ghi chú) cho một dòng. Chỉ có tác dụng đối với người đọc chương trình.	//Ham nay dung de nhap mang void NhapMang(int a[], int &n);
4	/* */	Tương tự như ký hiệu //, nhưng cho trường hợp nhiều dòng.	/* Dau tien nhap vao n. Sau do nhap cho tung phan tu */ void NhapMang(int a[], int &n);

#### **I.2. Các kiểu dữ liệu cơ bản trong C**

STT	Kiểu	GHI CHÚ	KÍCH THƯỚC	ĐỊNH DẠNG
<b>Kiểu liên tục (số thực)</b>				
1	float		4 bytes	%f
2	double		8 bytes	%lf
3	long double		10 bytes	%lf
<b>Kiểu rời rạc (số nguyên)</b>				
1	char	Ký tự	1 byte	%c
		Số nguyên	1 byte	%d
2	unsigned char	Số nguyên dương	1 byte	%d
3	int	Số nguyên	2 bytes	%d
4	unsigned int	Số nguyên dương	2 bytes	%u
5	long	Số nguyên	4 bytes	%ld
6	unsigned long	Số nguyên dương	4 bytes	%lu
7	char *	Chuỗi		%s

### I.3. Bảng ký hiệu các phép toán

STT	PHÉP TOÁN	Ý NGHĨA	GHI CHÚ
PHÉP TOÁN SỐ HỌC			
1	+	Cộng	
2	-	Trừ	
3	*	Nhân	
4	/	Chia lấy phần nguyên	
5	%	Chia lấy phần dư	
PHÉP TOÁN QUAN HỆ			
1	>	Lớn hơn	
2	<	Nhỏ hơn	
3	>=	Lớn hơn hoặc bằng	
4	<=	Nhỏ hơn hoặc bằng	
5	==	Bằng nhau	
6	!=	Khác nhau	
PHÉP TOÁN LOGIC			
1	!	NOT	
2	&&	AND	
3		OR	
TOÁN TỬ TĂNG GIẢM			
1	++	Tăng 1	Nếu toán tử tăng giảm đặt trước thì tăng giảm trước rồi tính biểu thức hoặc ngược lại.
2	--	Giảm 1	
PHÉP TOÁN THAO TÁC TRÊN BIT			
1	&	AND	
2		OR	
3	^	XOR	
4	<<	Dịch trái	
5	>>	Dịch phải	
6	~	Lấy phần bù theo bit	

#### I.4. Các hàm cơ bản

STT	TÊN HÀM	THƯ VIỆN	DIỄN GIẢI	VÍ DỤ
1	printf	#include<stdio.h>	Xuất ra màn hình.	<pre>#include&lt;stdio.h&gt; #include&lt;conio.h&gt; #include&lt;dos.h&gt; void main() {     int c = 1, n;      clrscr();     printf("Nhap n:");     scanf("%d", &amp;n);     do{         textcolor(c);         gotoxy(20, 10);         cprintf("%d", n);         c++;         if (c&gt;15)             c = 1;         delay(200);     } while(!kbhit()); }</pre>
2	scanf	#include<stdio.h>	Lấy dữ liệu từ bàn phím.	
3	gotoxy	#include<conio.h>	Di chuyển dấu nháy đến tọa độ (x, y) trên màn hình văn bản.	
4	textcolor	#include<conio.h>	Đặt màu cho chữ (có giá trị từ 0 đến 15).	
5	cprintf	#include<stdio.h>	Xuất ra màn hình với màu chữ đã định liền trước đó.	
6	delay	#include<dos.h>	Dừng thực hiện lệnh tiếp sau một khoảng thời gian.	
7	kbhit	#include<conio.h>	Kiểm tra xem có nhấn phím.	

#### I.5. Cấu trúc rẽ nhánh

##### a. Cấu trúc if

```
if (biểu thức điều kiện)
{
    <khối lệnh> ;
}
```

Nếu biểu thức điều kiện cho kết quả khác không thì thực hiện khối lệnh.

Ví dụ:

```
#include <conio.h>
#include <stdio.h>
void main ()
{
    float number ;

    printf ( "Nhap mot so trong khoang tu 1 den 10 => " );
    scanf ( "%f", &number );
    if (number >5)
        printf ( "So ban nhap lon hon 5. \n" );
    printf ( "%f la so ban nhap. ", number );
}
```



**b. Cấu trúc if ... else**

```
if (biểu thức điều kiện)
{
    <khối lệnh 1>;
}
else
{
    <khối lệnh 2>;
}
```

Nếu biểu thức điều kiện cho kết quả khác không thì thực hiện khối lệnh 1, ngược lại thì cho thực hiện khối lệnh thứ 2. Biểu thức điều kiện phải đặt trong cặp dấu ngoặc tròn.

Ví dụ: Giải và biện luận phương trình:  $ax+b=0$

```
#include <conio.h>
#include <stdio.h>
void main ()
{
    float a, b;
    printf ( "\n Nhap vao a:");
    scanf ( "%f", &a);
    printf ( " Nhap vao b:");
    scanf ( "%f", &b) ;
    if (a== 0)
        if (b== 0)
            printf ( " \n PTVSN");
        else
            printf ( " \n PTVN");
    else
        printf ( " \n Nghiem x=%f", -b/a);
    getch ();
}
```

**I.6. Cấu trúc lựa chọn switch**

```
switch (biểu thức)
{
    case n1:
        các câu lệnh ;
        break ;
    case n2:
        các câu lệnh ;
        break ;
    .....
    case nk:
        <các câu lệnh> ;
        break ;
}
```

[**default:** các câu lệnh]

}

- $n_i$  là các **hằng số nguyên hoặc ký tự**.
- Phụ thuộc vào giá trị của biểu thức viết sau **switch**, nếu:
  - Giá trị này =  $n_i$  thì thực hiện câu lệnh sau case  $n_i$ .
  - Khi giá trị biểu thức không thỏa tất cả các  $n_i$  thì thực hiện câu lệnh sau **default** nếu có, hoặc thoát khỏi câu lệnh **switch**.
  - Khi chương trình đã thực hiện xong câu lệnh của **case**  $n_i$  nào đó thì nó sẽ thực hiện luôn các lệnh thuộc **case** bên dưới nó mà không xét lại điều kiện (do các  $n_i$  được xem như các nhãn) → Vì vậy, để chương trình thoát khỏi lệnh **switch** sau khi thực hiện xong một trường hợp, ta dùng lệnh **break**.

Ví dụ: Tạo menu cấp 1 cho phép chọn menu bằng số nhập từ bàn phím.

```
#include<stdio.h>
#include<conio.h>
int ChonTD ()
{
    int chon ;

    printf ("Thuc Don") ;
    printf ("\n1. Lau thai!") ;
    printf ("\n2. Nuoc ngot!") ;
    printf ("\n3. Ca loc hap bau!") ;
    printf ("\n4. Chuot dong!") ;
    printf ("\n Xin moi ban chon mon an!") ;
    scanf ("%d",&chon) ;
    return chon ;
}

void TDchon(int chon)
{
    switch (chon)
    {
        case 1:
            printf ("\nBan chon lau thai!") ;
            break ;
        case 2:
            printf ("\nBan chon nuoc ngot!") ;
            break ;
        case 3:
            printf ("\nBan chon ca loc hap bau!") ;
            break ;
        case 4:
            printf ("\nBan chon chuot dong!") ;
```

```
        break ;
    default:
        printf ("\nBan chon khong dung!") ;
    }
}

void main()
{
    clrscr() ;
    int c ;
    c=ChonTD() ;
    TDchon(c) ;
    getch() ;
}
```

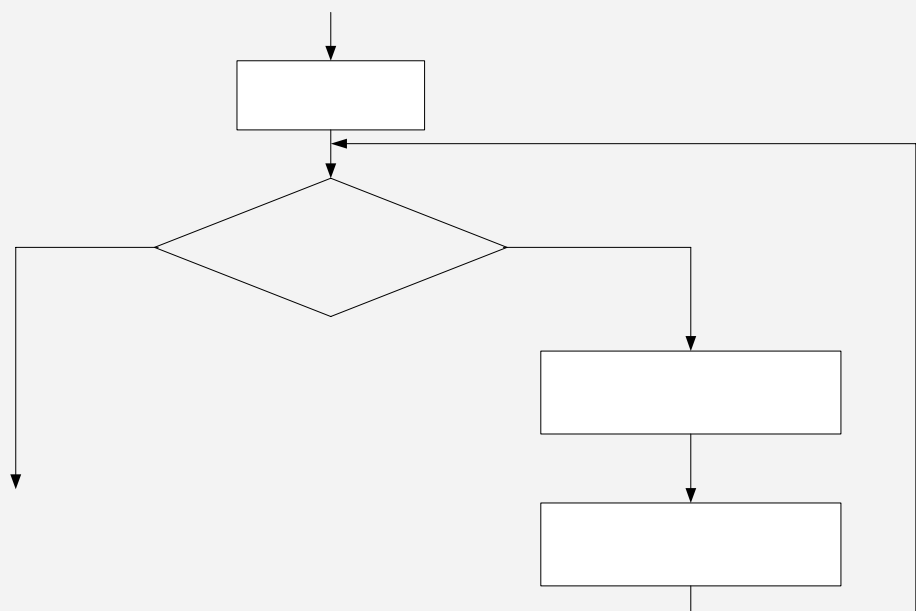
## I.7. Cấu trúc lặp

### a. *for*

```
for (<biểu thức khởi gán>; <biểu thức điều kiện>; <biểu thức tăng/giảm>)
{
    <khối lệnh>;
}
```

Bất kỳ biểu thức nào trong 3 biểu thức nói trên đều có thể vắng nhưng phải giữ dấu chấm phẩy (;).

#### Hoạt động của cấu trúc điều khiển for:



**Bước 1:** Khởi gán cho biểu thức 1

**Bước 2:** Kiểm tra điều kiện của biểu thức 2.

- Nếu **biểu thức 2**  $\neq 0$  thì cho thực hiện các lệnh của vòng lặp, thực hiện biểu thức 3. Quay trở lại bước 2.
- Ngược lại thoát khỏi lặp.

Ví dụ: In ra màn hình bảng mã ASCII từ ký tự số 33 đến 255.

```
#include<conio.h>
#include<stdio.h>
void main()
{
    for (int i=33;i<=255;i++)
        printf("Ma ASCII cua %c: %d\t", i, i) ;
    getch () ;
}
```

**b. while**

```
< Khởi gán>
while ( <biểu thức điều kiện>)
{
    lệnh/ khối lệnh;
    < tăng/giảm chỉ số lặp>;
}
```

🔗 **Lưu ý:** Cách hoạt động của **while** giống **for**

Ví dụ: Tính giá trị trung bình các chữ số của số nguyên n gồm k chữ số.

```
#include<stdio.h>
#include<conio.h>
void main()
{
    long n, tong=0;
    int sochuso=0;
    float tb;

    printf ("Nhap vao gia tri n gom k chu so") ;
    scanf ("%ld",&n) ;
    while(n>0)
    {
        tong=tong+n%10 ;
        sochuso++ ;
        n=n/10 ;
    }

    tb=1.0*tong/sochuso ;
    printf ("Gia tri trung binh la: %f", tb) ;
}
```