Chương 3: Giới thiệu về các cấu trúc điều khiển

Phần c: Cấu trúc điều khiển lặp

Nhập môn lập trình

Trình bày: Nguyễn Sơn Hoàng Quốc
Email: nshquoc@fit.hcmus.edu.vn

Nội dung

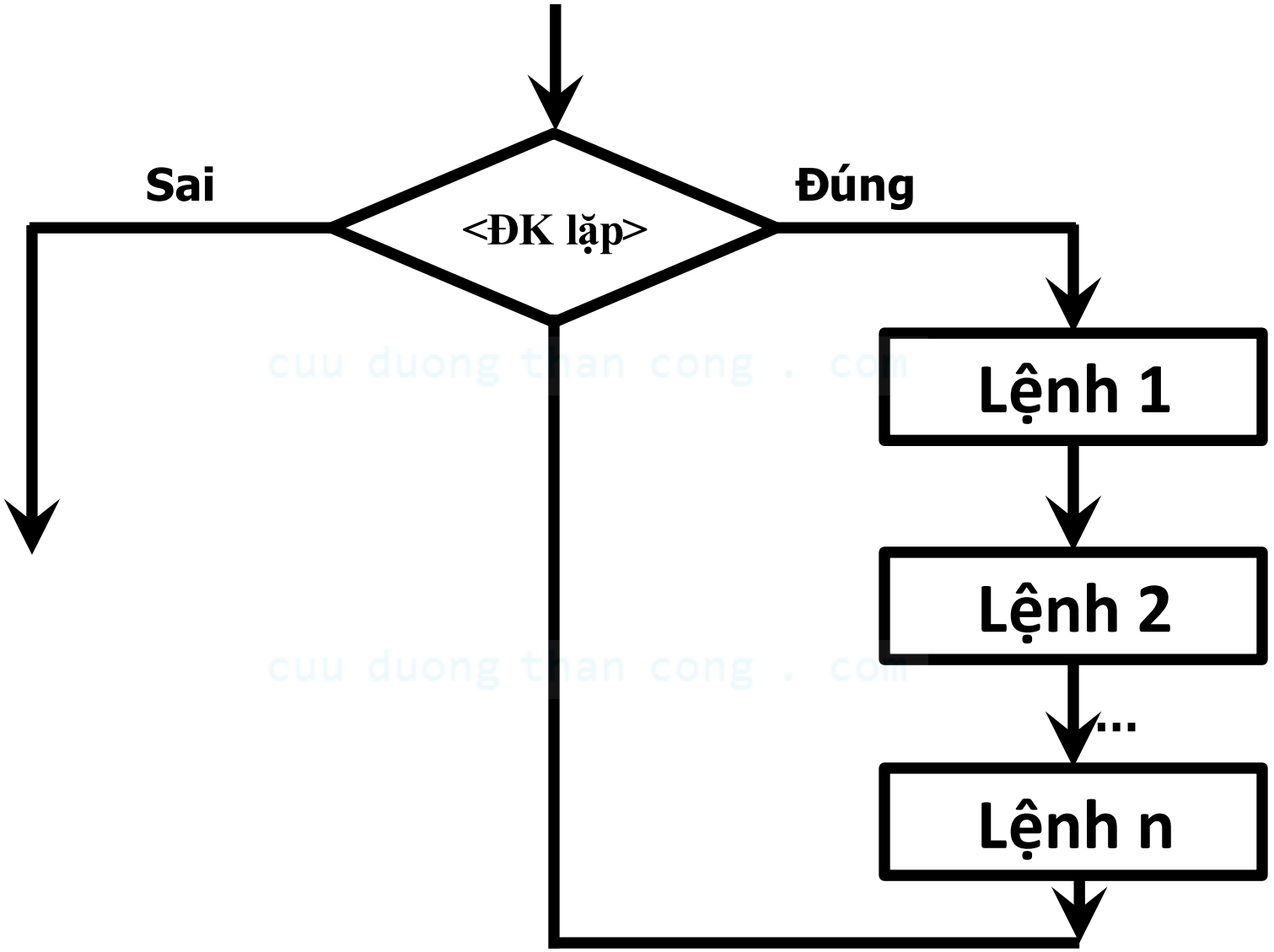
- Cấu trúc điều khiển lặp while
- Cấu trúc điều khiển lặp do while
- Cấu trúc điều khiển lặp for
- Các chỉ thị lệnh của vòng lặp (break, continue, return)
- Các vấn đề tìm hiểu mở rộng kiến thức nghề nghiệp
- Thuật ngữ và bài đọc thêm tiếng Anh

CẤU TRÚC ĐIỀU KHIỂN LẶP WHILE

Cấu trúc điều khiển lặp while

```
while (điều_kiện_lặp)
{
   Lệnh 1;
   Lệnh 2;
   …
   Lệnh n;
}
```

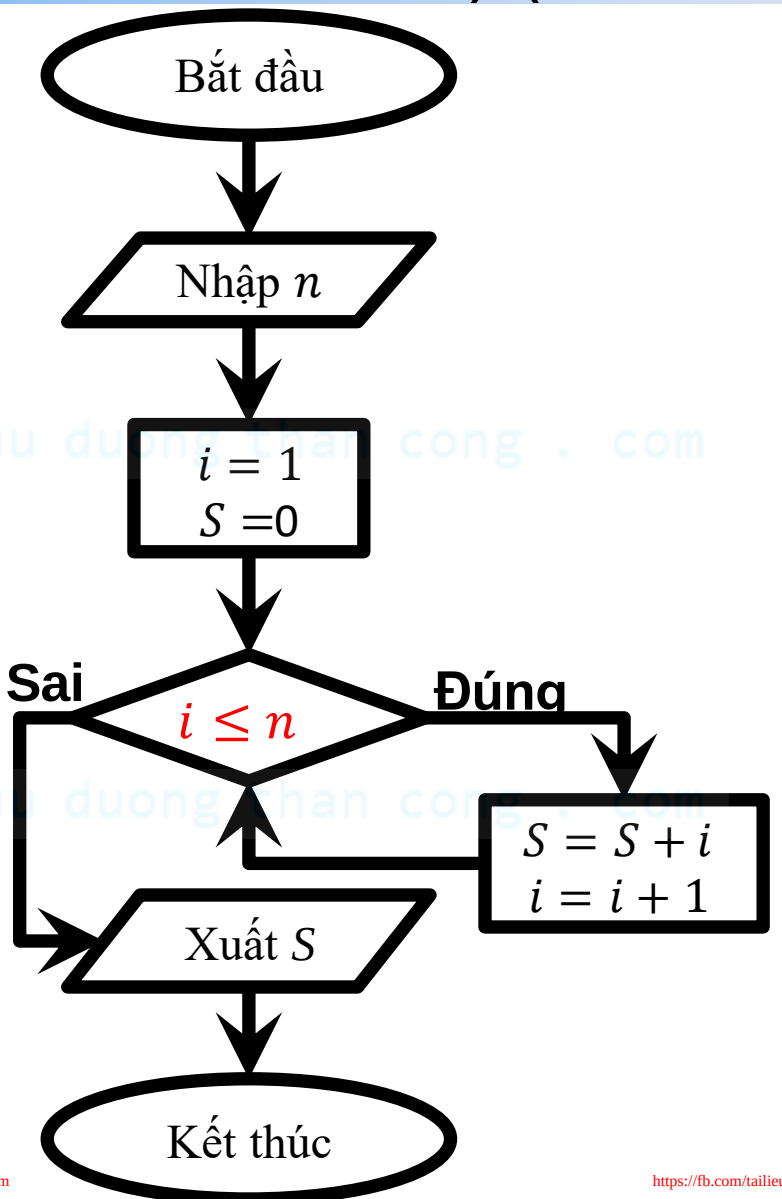
Lưu đồ thuật toán vòng lặp while



Ví dụ 1 (tính S = 1+2+…+n)

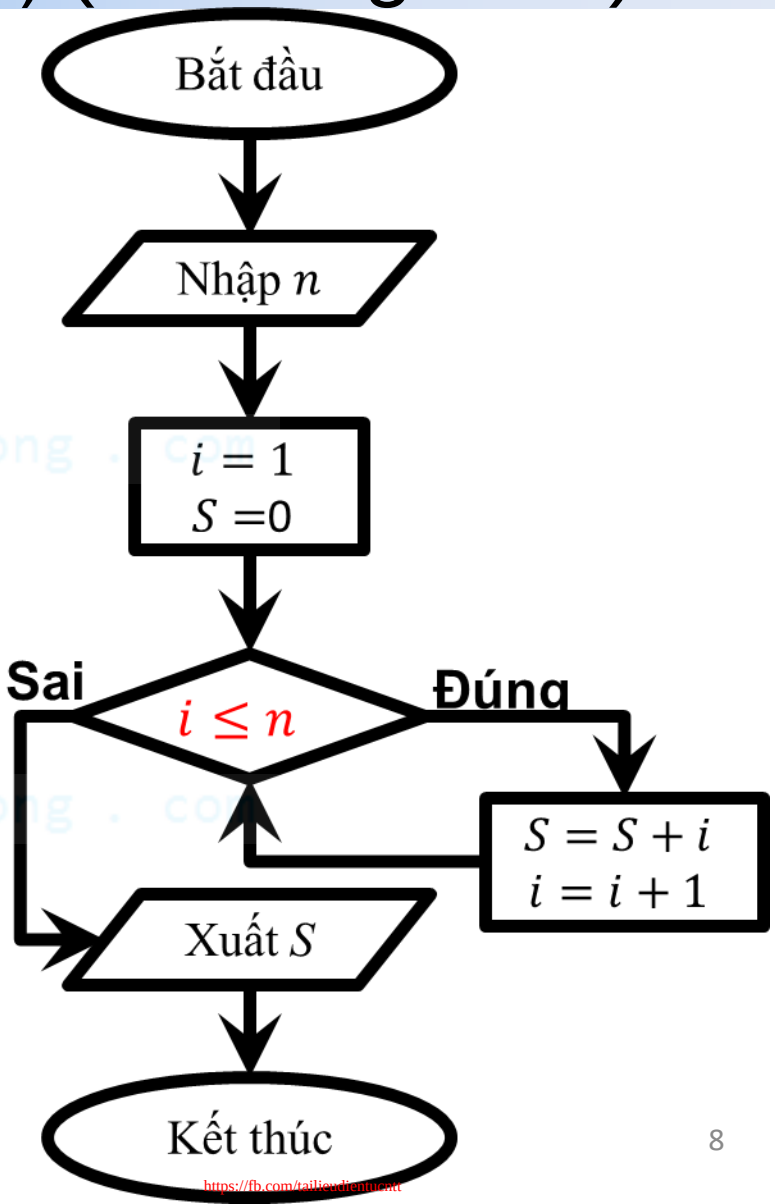
- Viết chương trình nhập vào một số n, tính tổng S = 1 + 2 + … + n

Ví dụ 1 (tính S = 1+2+…+n) (Lưu đồ)



Ví dụ 1 (tính S = 1+2+…+n) (Chương trình)

```
1. #include <stdio.h>
2.
3. void main() {
4.      int n, i, S;
5.
6.      printf("Nhap n: ");
7.      scanf("%d", &n);
8.
9.      i = 1;
10.     S = 0;
11.
12.     while (i <= n)
13.     {
14.         S = S + i;
15.         i = i + 1;
16.     }
17.
18.     printf("Tong S la %d\n", S);
19.}
```



Ví dụ 2 : Tìm n nhỏ nhất với 1+2+…+n>10000

- Vẽ lưu đồ và viết chương trình tìm số nguyên dương n nhỏ nhất sao cho $1 + 2 + \cdots + n > 10000$.

Ví dụ 2 : Tìm n nhỏ nhất với 1+2+…+n>10000

```
1. #include <stdio.h>
2.
3. void main() {
4.
5.     int S = 0, n = 0;
6.
7.     while (S <= 10000)
8.     {
9.         n = n + 1;
10.        S = S + n;
11.    }
12.
13.    printf("So n la %d\n", n);
14.}
```

Nhận xét

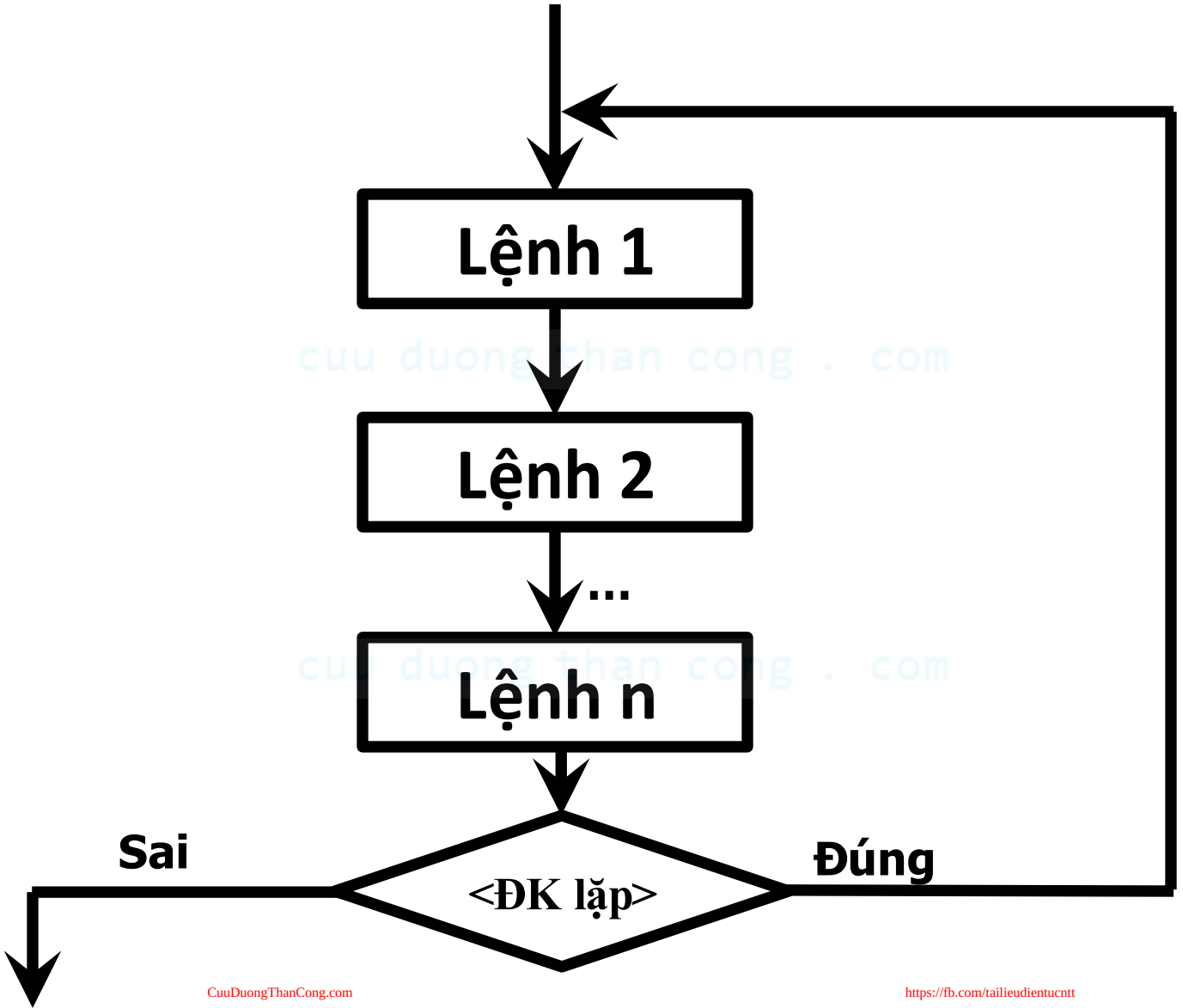
- Các lệnh trong khối lệnh của vòng lặp while có thể sẽ không thực hiện (lặp ít nhất 0 lần)
- Điều kiện lặp của vòng lặp while thường được cập nhật sau mỗi lần thực hiện khối lệnh hay có một biến cố nào thuận lợi xảy ra.

CẤU TRÚC ĐIỀU KHIỂN LẶP DO WHILE

Cấu trúc điều khiển lặp do while

```
do
{
    Lệnh 1;
    Lệnh 2;
    …
    Lệnh n;
}
while (điều_kiện_lặp);
```

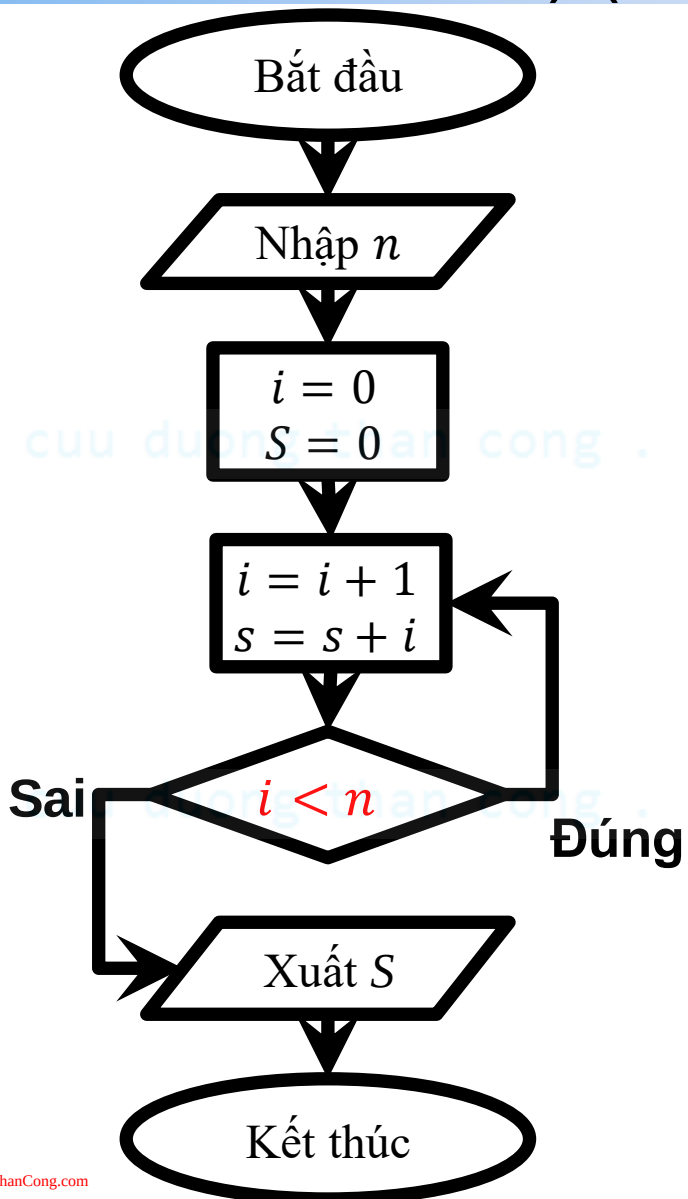
Lưu đồ thuật toán vòng lặp do while



Ví dụ 1 (tính S = 1+2+…+n)

- Viết chương trình nhập vào một số n, tính tổng S = 1 + 2 + … + n

Ví dụ 1 (tính S = 1+2+…+n) (Lưu đồ)



Ví dụ 1 vòng lặp do while

```
1. #include <stdio.h>
2.
3. void main() {
4.     int n, i, s;
5.
6.     printf("Nhap n: ");
7.     scanf("%d", &n);
8.
9.     i = 0;
10.    s = 0;
11.
12.    do
13.    {
14.        i = i + 1;
15.        s = s + i;
16.    } while (i < n);
17.
18.    printf("Tong s la %d\n", s);
19.}
```

Ví dụ 2 : Tìm n lớn nhất với 1+2+…+n<10000

- Vẽ lưu đồ và viết chương trình tìm số nguyên dương n lớn nhất sao cho $1 + 2 + \cdots + n < 10000$.

Ví dụ 2 : Tìm n lớn nhất với 1+2+…+n<10000

```
1. #include <stdio.h>
2.
3. void main()
4. {
5.
6.         int n = 0;
7.         int S = 0;
8.
9.         do
10.        {
11.                n++;
12.                S = S + n;
13.        } while (S + n + 1 < 10000);
14.
15.        printf("So n la %d\n", n);
16.}
```

Phân tích hoạt động của cấu trúc lặp do/while

- **Bước 1:** Thực hiện các câu lệnh trong khối lệnh lặp do while.
- **Bước 2:** Khi gặp đến cuối khối lệnh lặp do while, chương trình sẽ xác định giá trị của điều kiện lặp sau từ khóa while.
- **Bước 3:** Chương trình sẽ thực thi một trong 2 nhánh sau tùy theo giá trị của biểu thức vừa nhận được.

Phân tích hoạt động của cấu trúc lặp do/while

- **Bước 3.1:** Nếu biểu thức có giá trị đúng (khác 0), chương trình sẽ quay trở lại bước 1 để tiếp tục thực hiện vòng lặp mới.
- **Bước 3.2:** Nếu biểu thức có giá trị sai (bằng 0), chương trình sẽ ra khỏi chu trình và chuyển tới câu lệnh đúng sau dấu chấm phẩy đặt cuối từ khóa do while.

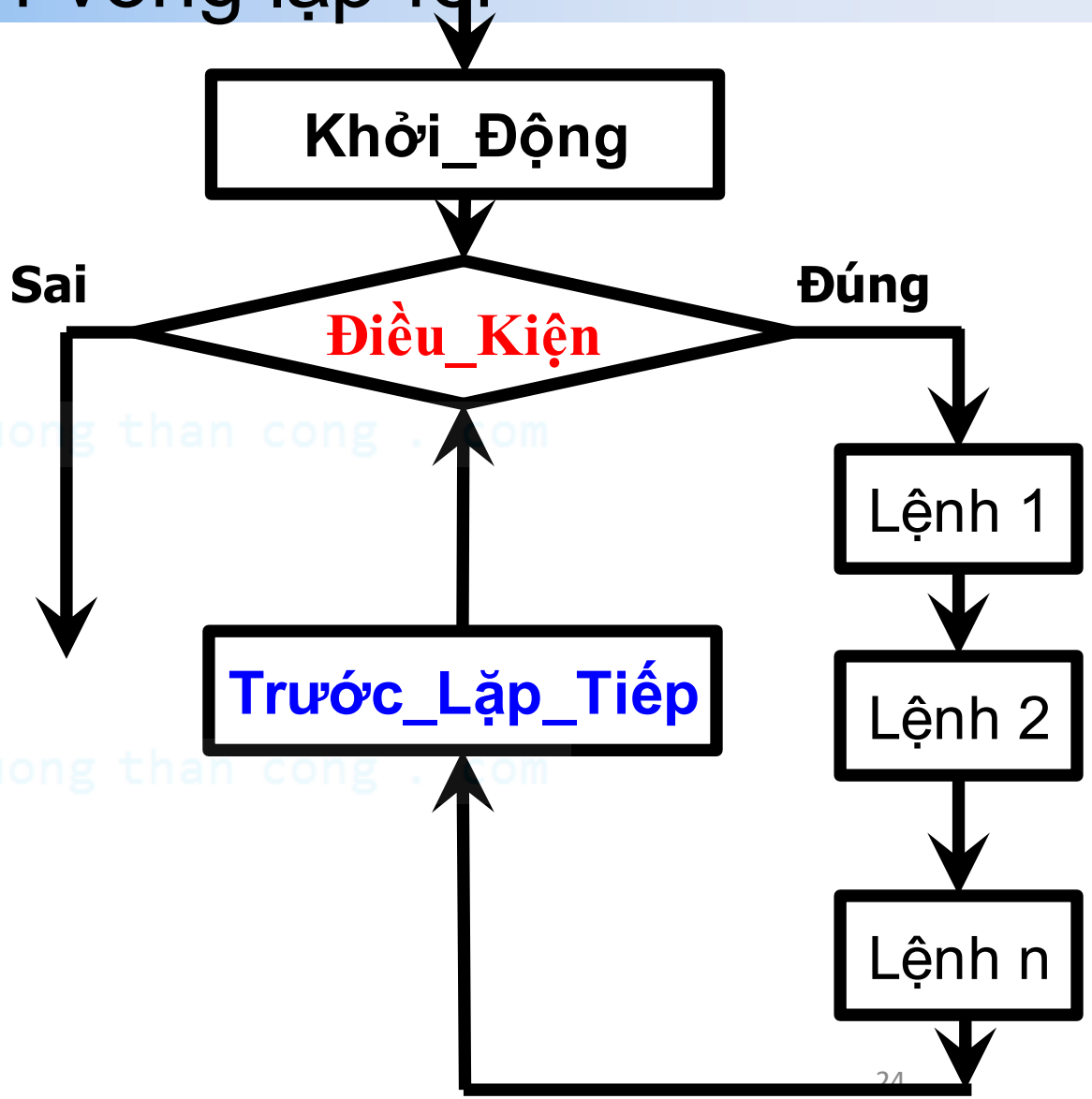
CẤU TRÚC LẶP FOR

Cấu trúc điều khiển lặp for

```
for (Khởi_Động;Điều_Kiện;Trước_Lặp_Tiếp)
{
      Lệnh 1;
      Lệnh 2;
      …
      Lệnh n;
}
```

Lưu đồ thuật toán vòng lặp for

```
for (Khởi_Động;
Điều_Kiện;
Trước_Lặp_Tiếp)
{
      Lệnh 1;
      Lệnh 2;
      …
      Lệnh n;
}
```



Ví dụ 1 của vòng lặp for

- Xuất các ký tự từ ‘A’ đến ‘Z’.

```
1.#include <stdio.h>
2.
3.void main()
4.{
5.    char kytu;
6.
7.    for (kytu = ‘A’; kytu <= ‘Z’; kytu++)
8.    {
9.       printf("%c ", kytu);
10.   }
11.}
```

Ví dụ 2 của vòng lặp for

- Tính tổng các số lẻ của số nguyên dương n.

```
1. #include <stdio.h>
2.
3. void main()
4. {
5.     int n, i, s;
6.
7.     printf("Nhap n: ");
8.     scanf("%d", &n);
9.
10.    s = 0;
11.
12.    for (i = 1; i < n; i = i + 2)
13.    {
14.        s = s + i;
15.    }
16.
17.    printf("Tong cac so duong le nho hon %d la %d\n", n, s);
18.}
```

Ví dụ 2 của vòng lặp for (cách khác)

- Tính tổng các số lẻ của số nguyên dương n.

```
1. #include <stdio.h>
2. void main()
3. {
4.      int n, i, s;
5.      printf("Nhap n: ");
6.      scanf("%d", &n);
7.      s = 0;
8.      i = 1;
9.      for (;;)
10.     {
11.         s = s + i;
12.         i = i + 2;
13.         if (i >= n)
14.             break;
15.     }
16.     printf("Tong cac so duong le nho hon %d la %d\n", n, s);
17.}
```

Ý nghĩa của các biểu thức trong vòng lặp for

- **Biểu thức 1 (Khởi_Động):** thường dùng để **khởi tạo** biến đếm của vòng lặp.
- **Biểu thức 2 (Điều_Kiện):** thường dùng để kiểm tra điều kiện của vòng lặp. Nếu không có biểu thức này phải dùng lệnh break để kết thúc
- **Biểu thức 3 (Trước_Lặp_Tiếp):** thường dùng để điều khiển biến đếm của vòng lặp.

Phân tích hoạt động của cấu trúc lặp for

- **Bước 1:** Xác định biểu thức 1 (Khởi_Động)
- **Bước 2:** Xác định biểu thức 2 (Điều_Kiện)
- **Bước 3:** Thực thi hai bước sau tùy vào giá trị của biểu thức 2.
  - **Bước 3.1:** Nếu biểu thức 2 có giá trị 0 (sai) ➔ thoát khỏi for và chuyển tới câu lệnh sau khối lệnh của for.
  - **Bước 3.2:** Nếu biểu thức 2 có giá trị khác 0 (đúng) ➔ thực hiện các câu lệnh trong khối lệnh for
- **Bước 4:** Thực hiện biểu thức 3 (Trước_Lặp_Tiếp) ➔ quay lại **bước 2** để bắt đầu một vòng lặp mới.

Điều kiện dừng của vòng lặp

- Điều kiện dừng của vòng lặp sẽ trả về **true** hoặc **false**, nếu **true** vòng lặp chạy **tiếp** và **false** sẽ **thoát**.

CÁC CHỈ THỊ LỆNH CỦA VÒNG LẶP

Các chỉ thị can thiệp vào vòng lặp

- Lệnh break
- Lệnh continue
- Lệnh return

Câu lệnh break

- Câu lệnh break cho phép ra khỏi cấu trúc điều khiển lặp (vòng for, while, do while) và cấu trúc chọn switch. Khi có nhiều vòng lặp lồng nhau, câu lệnh break sẽ thoát khỏi vòng lặp chứa nó bên trong khối lệnh lặp.
- Như vậy break cho ta khả năng ra khỏi một cấu trúc điều khiển lặp mà không dùng điều kiện kết thúc chương trình.

Câu lệnh break (Ví dụ)

- Tính tổng các số lẻ của số nguyên dương n.

```
1. #include <stdio.h>
2. void main()
3. {
4.      int n, i, s;
5.      printf("Nhap n: ");
6.      scanf("%d", &n);
7.      s = 0;
8.      i = 1;
9.      for (;;)
10.     {
11.         s = s + i;
12.         i = i + 2;
13.         if (i >= n)
14.             break;
15.     }
16.     printf("Tong cac so duong le nho hon %d la %d\n", n, s);
17.}
```

Câu lệnh continue

- Ngược lại với câu lệnh break, câu lệnh continue dùng để bắt đầu một vòng mới của cấu trúc điều khiển lặp chứa nó.
- Khi gặp câu lệnh continue bên trong thân của một toán tử for, chương trình sẽ thực hiện bước 4 trong phần "phân tích sự hoạt động của cấu trúc lặp for"

Câu lệnh continue

- Khi gặp câu lệnh continue bên trong thân của while hoặc do while, chương trình thực hiện **bước 1** trong phần "phân tích sự hoạt động của cấu trúc lặp while"
- **Ghi chú**: Câu lệnh continue chỉ áp dụng cho các cấu trúc điều khiển lặp chứ không áp dụng cho cấu trúc điều khiển chọn switch.

Ví dụ lệnh continue

- In ra các số lẻ nhỏ hơn 100, trừ các số 5, 7, 93.

```
1. #include <stdio.h>
2.
3. void main()
4. {
5.     int i;
6.
7.     for (i = 1; i < 100; i += 2)
8.     {
9.         if (i == 5 || i == 7 || i == 93)
10.            continue;
11.        printf("%5d", i);
12.    }
13.}
```

Câu lệnh return

- Trả về dòng điều khiển mà nơi nó gọi, khi lệnh return được theo sau bởi một biểu thức thì biểu thức đó sẽ được đánh giá và giá trị này sẽ được trả về cho nơi đã gọi hàm. Khi return được gọi mà không có biểu thức đi kèm thì giá trị trả về là không xác định.
- Câu lệnh return không chỉ thoát khỏi vòng lặp mà nó còn thoát luôn khỏi hàm mà đang chứa nó.

Câu lệnh return(Ví dụ)

- Tính tổng số lẻ của số nguyên dương n.

```
1. #include <stdio.h>
2. void main()
3. {
4.      int n, i, s;
5.      printf("Nhap n: ");
6.      scanf("%d", &n);
7.
8.      s = 0;
9.      if (n < 0)
10.     {
11.         printf("n khong phai so nguyen duong");
12.         return;
13.     }
14.     for (i = 1; i < n; i = i + 2)
15.         s = s + i;
16.     printf("Tong cac so duong le nho hon %d la %d\n", n, s);
17.}
```

CÁC VẤN ĐỀ TÌM HIỂU MỞ RỘNG KIẾN THỨC NGHỀ NGHIỆP

Tìm hiểu thêm

- Tránh sự nhập nhằng và khó hiểu trong mã nguồn
- Các chỉ thị đặc biệt bao hàm cấu trúc điều khiển
- Cấu trúc điều khiển cấp cao trong các NNLT
- Sự khác biệt, tương đồng giữa các NNLT

THUẬT NGỮ VÀ BÀI ĐỌC THÊM TIẾNG ANH

Thuật ngữ tiếng Anh

- ***block***: khối lệnh
- ***branching***: rẽ nhánh, phân nhánh.
- ***control structures***: các cấu trúc điều khiển.
- ***global variables***: biến toàn cục
- ***instruction***: lệnh.
- ***local variables***: biến cục bộ
- ***loop***: vòng lặp.
- ***program***: chương trình.
- ***variable***: biến.

Bài đọc thêm tiếng Anh

- **Thinking in C**, Bruce Eckel, E-book, 2006.
- **Theory and Problems of Fundamentals of Computing with C++**, John R.Hubbard, Schaum's Outlines Series, McGraw-Hill, 1998.