



CẤU TRÚC ĐIỀU KHIỂN

NHẬP MÔN LẬP TRÌNH

GVHD: Trương Toàn Thịnh

NỘI DUNG

- Khối lệnh
- Cấu trúc rẽ nhánh
- Cấu trúc lặp
- Câu lệnh break & continue
- Điều kiện dừng vòng lặp
- Bài tập

KHỐI LỆNH

- Tập hợp các lệnh nằm trong ‘{’ và ‘}’
- Các khối lệnh có thể lồng nhau
- Biến nằm trong khối lệnh là biến cục bộ
- Biến không nằm trong bất kì khối lệnh nào là biến toàn cục

<pre>{ int x = 1, y = 0; y = ++x; //cout << z << endl; (error)</pre>	A
<pre>{ int z; z = x + y; printf("z = %d\n", z); }</pre>	B
<pre>// cout << z << endl; (error) }</pre>	

KHỐI LỆNH

- Biến trong khối lệnh không thể được sử dụng bên ngoài khối lệnh đó
- Hai biến trong cùng khối lệnh không được trùng tên
- Biến trong khối lệnh A được sử dụng trong A và khối lệnh được A lồng
- Nếu trùng tên biến giữa khối lệnh lồng và khối lệnh bị lồng, thì sẽ hiểu theo biến có phạm vi gần nhất

KHỎI LỆNH

Dòng	Mô tả
1	<code>#include <stdio.h></code>
2	<code>void main(){</code>
3	<code>int a = 1984, b = 1988;</code>
4	<code>printf("a(main block) = %d\n", a);</code>
5	<code>printf("b(main block) = %d\n", b);</code>
6	<code>{</code>
7	<code>int b = 1996;</code>
8	<code>a = 2001;</code>
9	<code>printf("a(of main block is changed) = %d\n", a);</code>
10	<code>printf("b(sub block) = %d\n", b);</code>
11	<code>}</code>
12	<code>printf("Now is main block:\n");</code>
13	<code>printf("a(changed) = %d\n", a);</code>
14	<code>printf("b(unchanged) = %d\n", b);</code>
15	<code>}</code>

KHỎI LỆNH

- Namespace trong C++ dùng để đặt tên nhằm tổ chức các lớp, đối tượng
- Ví dụ: biến cin và cout trong không gian tên std và thuộc lớp iostream

```
#include <iostream>
```

```
using namespace std;
```

- Cú pháp khai báo namespace

```
namespace <Tên>{  
    //Các thực thể  
}
```

- Cú pháp sử dụng namespace:

```
using namespace <tên>
```

KHỎI LỆNH

Dòng	Mô tả	
1	#include <iostream>	
2	using namespace std;	
3	namespace Data2D{	namespace Data3D{
4	int dX = 3, dY = 4;	float dX = 5, dY = 6, dZ = 7;
5	float Area;	float Volume;
6	}	namespace Base{ float Area, h;}
7		}
8	void main(){	
9	using namespace Data2D;	
10	Area = dX*dY;	
11	cout<<"Data2D::Area = "<<Area<<endl;	
12	Data3D::Base::Area = Data3D::dX*Data3D::dY;	
13	Data3D::Volume = Data3D::Base::Area*Data3D::dZ;	
14	Data3D::Base::h = (Data3D::dX*Data3D::dY)/2;	
15	cout<<"Volume="<<Data3D::Volume<<endl	
16	cout<<"h="<<Dat	
17	}	

KHỐI LỆNH

- Cách khai báo và dùng biến toàn cục
 - Khai báo/định nghĩa bên ngoài tất cả các khối lệnh (thường nằm ở đầu chương trình)
 - Dùng toán tử '::' nếu có biến cục bộ trùng tên

Dòng	Mô tả
1	<code>#include <stdio.h></code>
2	<code>int dX = 7;</code>
3	<code>void main(){</code>
4	<code>printf("dX (global) = %d\n", dX);</code>
5	<code>{</code>
6	<code>int dX = 9;</code>
7	<code>printf("dX (Sub block) = %d\n", dX);</code>
8	<code>printf("dX (global) is unchanged = %d\n", ::dX);</code>
9	<code>}</code>
10	<code>}</code>

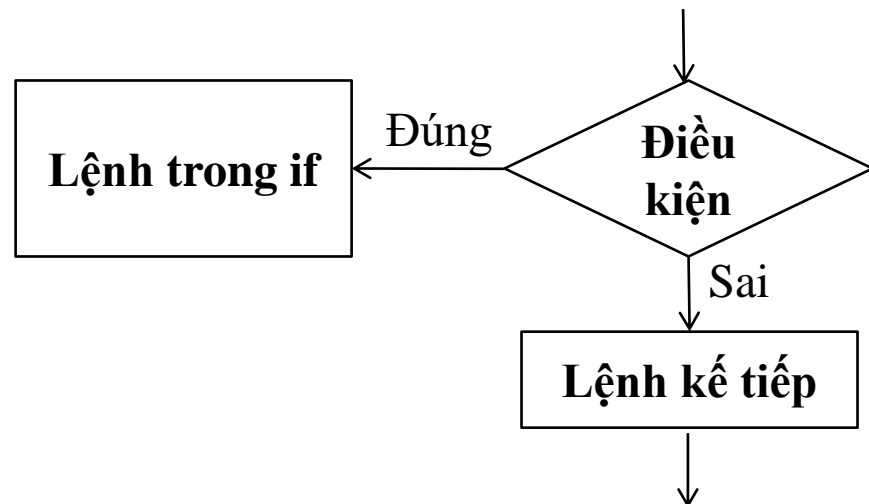
NỘI DUNG

- Khởi lệnh
- Cấu trúc rẽ nhánh
- Cấu trúc lặp
- Câu lệnh break & continue
- Điều kiện dừng vòng lặp
- Bài tập

CẤU TRÚC RỄ NHÁNH

- Sử dụng trong trường hợp việc tính toán phụ thuộc vào điều kiện luận lý
- C/C++ hỗ trợ hai cấu trúc rẽ nhánh là **if else** và **switch**
- Cấu trúc **if else** dạng đơn giản không có phần **else**

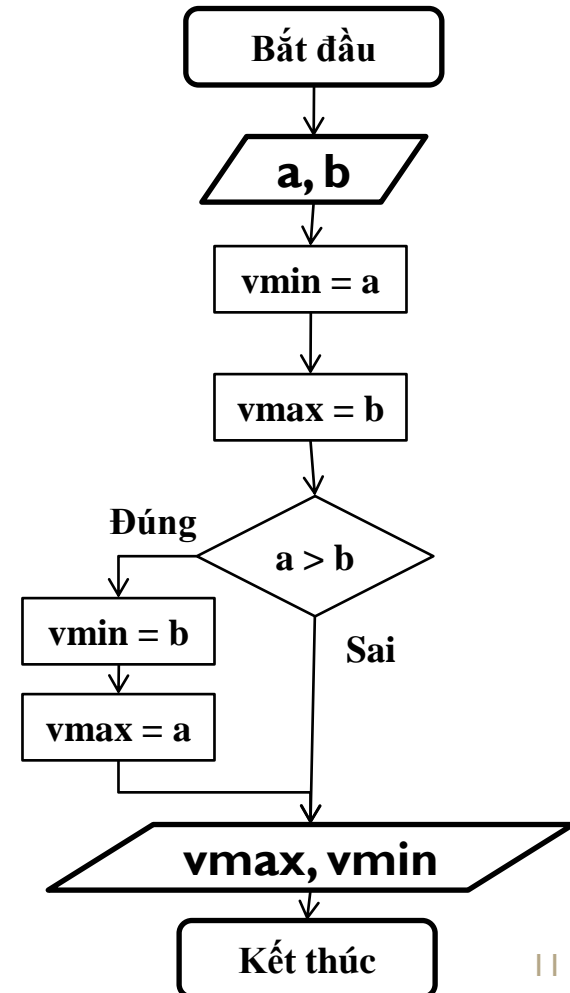
```
...  
if(<điều kiện>){  
    //Lệnh trong if  
}  
//Lệnh kế tiếp
```



CẤU TRÚC RỄ NHÁNH

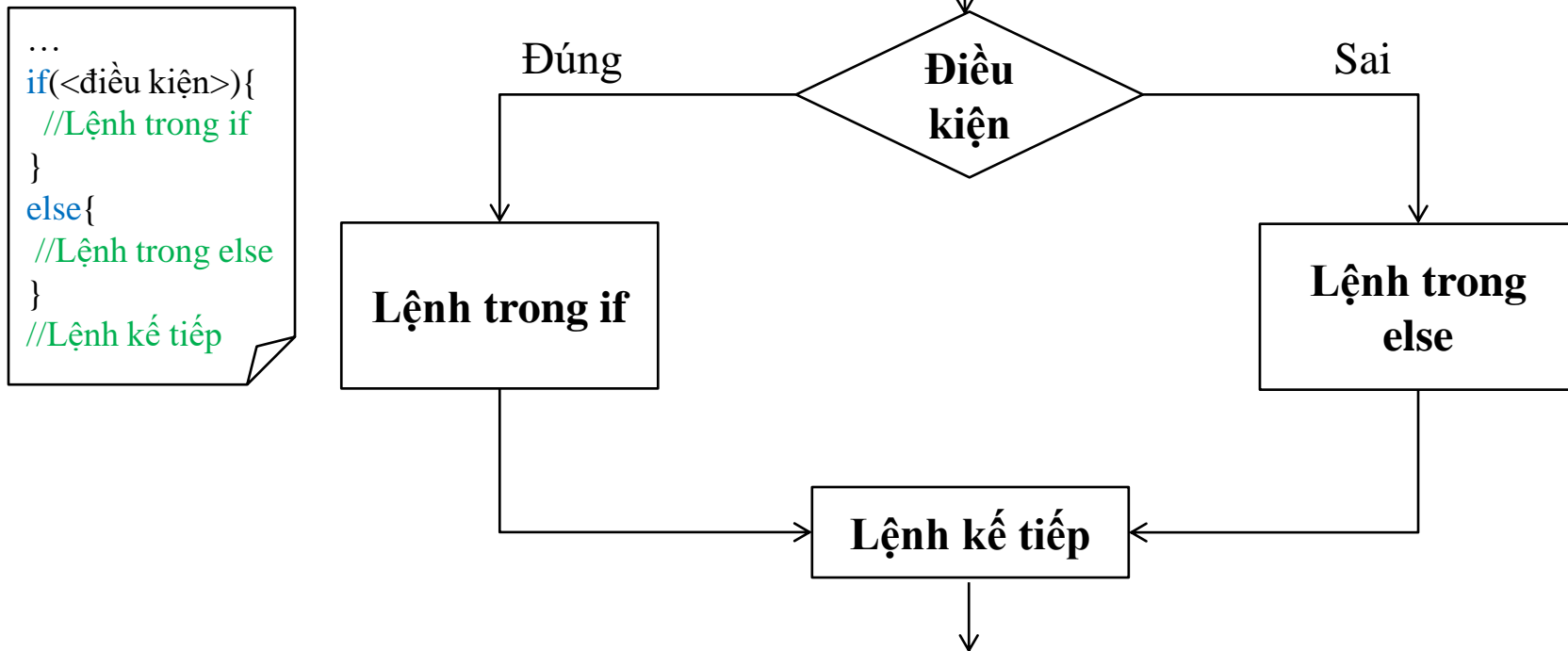
- Ví dụ cấu trúc rẽ nhánh if-else đơn giản

Dòng	Mô tả
1	<code>#include <stdio.h></code>
2	<code>void main(){</code>
3	<code>int a, b, vmax, vmin;</code>
4	<code>printf("a = ");</code>
5	<code>scanf("%d", &a);</code>
6	<code>printf("b = ");</code>
7	<code>scanf("%d", &b);</code>
8	<code>vmin = a; vmax = b;</code>
9	<code>if(a > b){</code>
10	<code> vmin = b; vmax = a;</code>
11	<code>}</code>
12	<code>printf("Max: %d, Min: %d\n", vmax, vmin);</code>
13	<code>}</code>



CẤU TRÚC RỄ NHÁNH

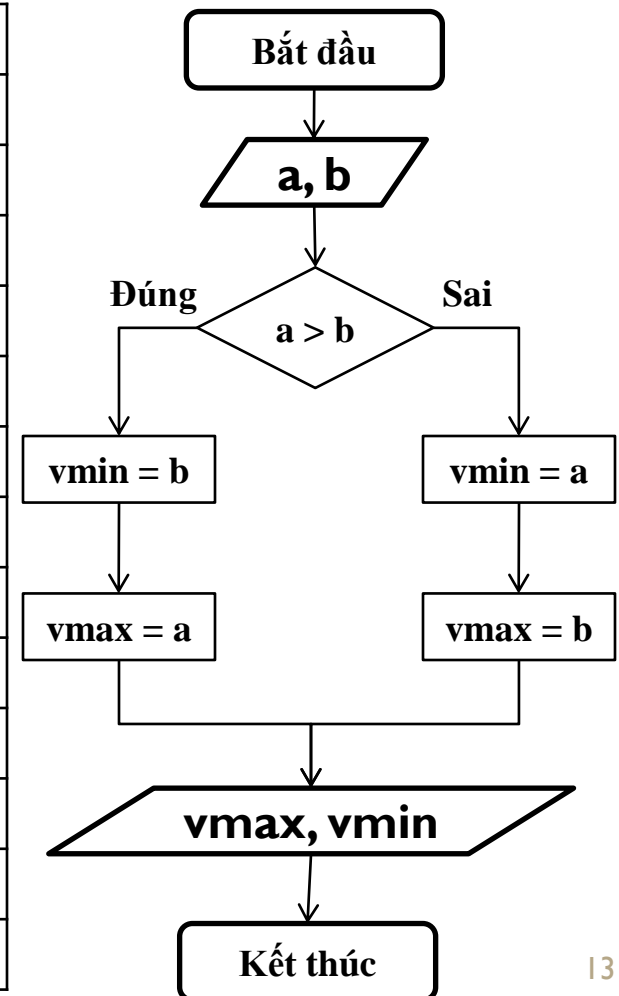
- Cấu trúc **if-else** dạng đầy đủ sẽ có cả phần **else**



CẤU TRÚC Rẽ NHÁNH

- Ví dụ cấu trúc rẽ nhánh if-else đầy đủ

Dòng	Mô tả
1	<code>#include <stdio.h></code>
2	<code>void main(){</code>
3	<code>int a, b, vmax, vmin;</code>
4	<code>printf("a = "); scanf("%d", &a);</code>
5	<code>printf("b = "); scanf("%d", &b);</code>
6	<code>if(a > b){</code>
7	<code> vmin = b; vmax = a;</code>
8	<code>}</code>
9	<code>else{</code>
10	<code> vmax = b; vmin = a;</code>
11	<code>}</code>
12	<code>printf("Max: %d, Min: %d\n", vmax, vmin);</code>
13	<code>}</code>



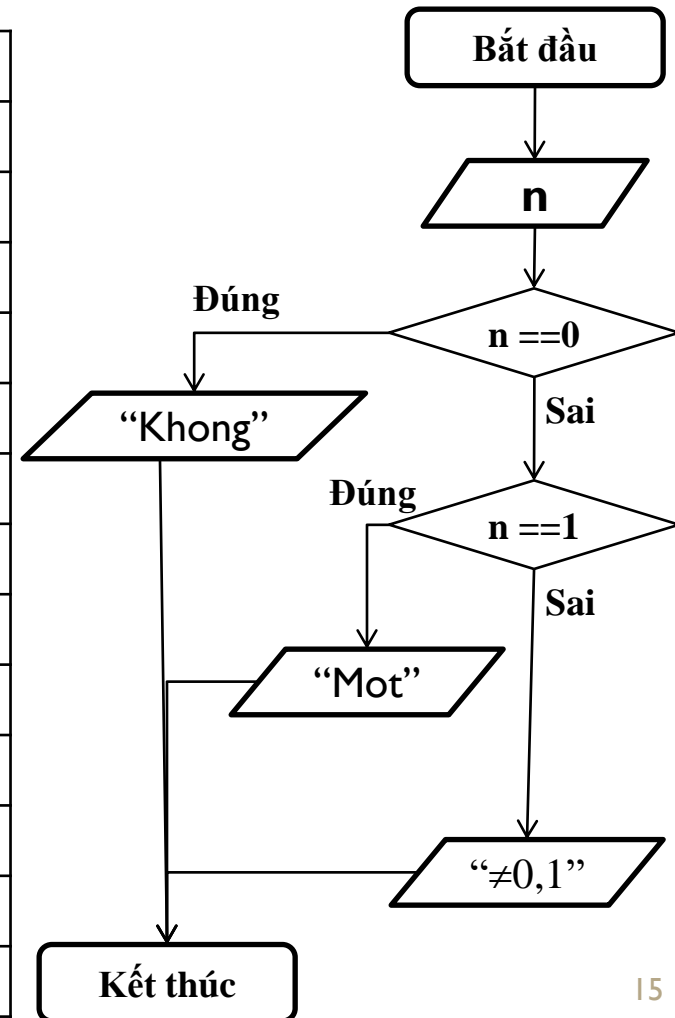
CẤU TRÚC RỄ NHÁNH

- Cấu trúc **switch** để so sánh nhiều trường hợp
- Chỉ so sánh kiểu nguyên
- Dùng câu lệnh **break** ở mỗi khối
- Nên luôn có khối **default**
- Các 'case' có thể trùng nhau

CẤU TRÚC RỄ NHÁNH

- Ví dụ cấu trúc rẽ nhánh switch

Dòng	Mô tả
1	<code>#include <stdio.h></code>
2	<code>void main(){</code>
3	<code>int n;</code>
4	<code>printf("n = "); scanf("%d", &n);</code>
6	<code>switch(n){</code>
7	<code>case 0: case 1: // gộp 2 case</code>
8	<code>printf("Khong\n");</code>
9	<code>break;</code>
10	<code>case 1:</code>
11	<code>printf("Mot\n");</code>
12	<code>break;</code>
13	<code>default: printf("Khac khong va mot\n");</code>
14	<code>}}</code>



NỘI DUNG

- Khối lệnh
- Cấu trúc rẽ nhánh
- Cấu trúc lặp
- Câu lệnh break & continue
- Điều kiện dừng vòng lặp
- Bài tập

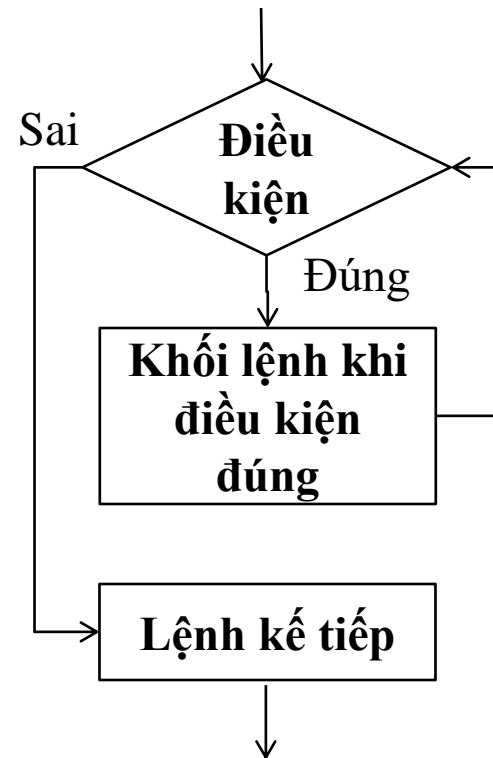
CẤU TRÚC LẶP

- Dùng để thực hiện lặp đi lặp lại nhiều lần một nhóm các chỉ thị nào đó
- C/C++ hỗ trợ các loại vòng lặp sau:
 - `while`
 - `do...while`
 - `for`
- Hầu hết các thuật toán từ đơn giản tới phức tạp đều cần cấu trúc lặp

CẤU TRÚC LẶP

- Cấu trúc **while**

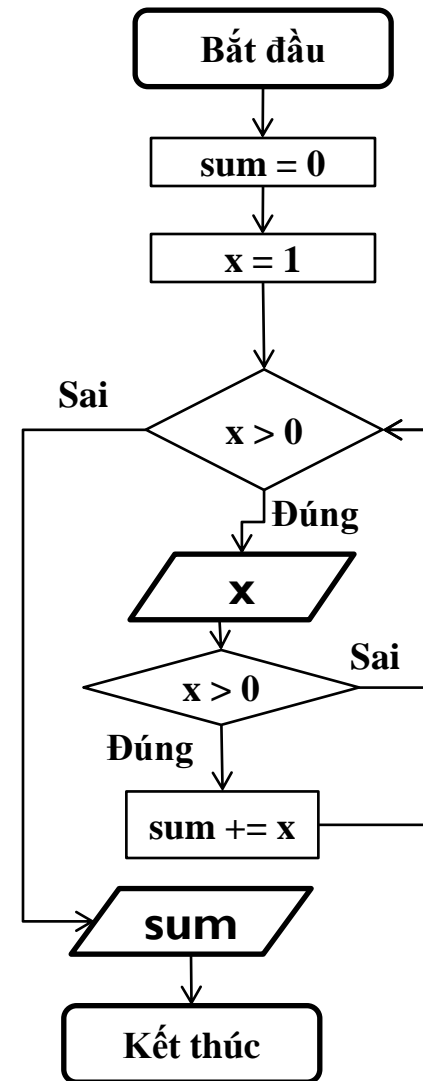
```
...  
while(<điều kiện>){  
    //Lệnh trong while  
}  
//Lệnh kế tiếp
```



CẤU TRÚC LẶP

- Ví dụ cấu trúc lặp **while**

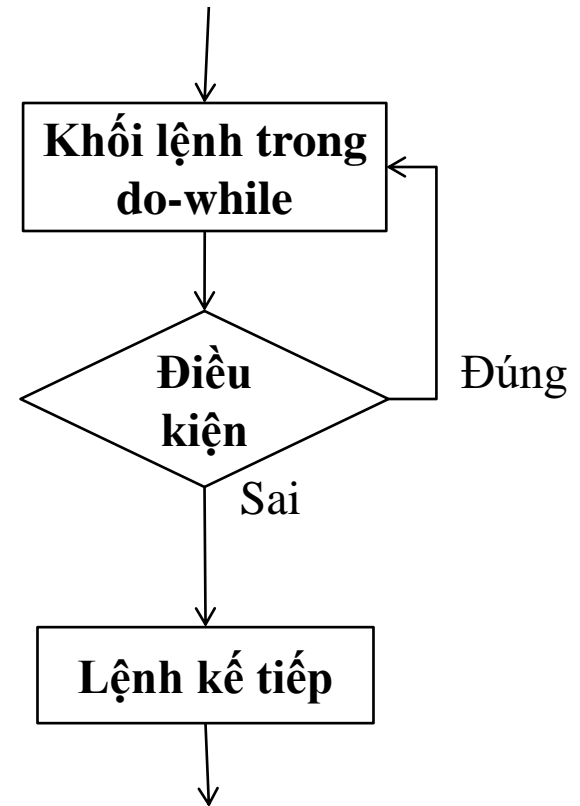
Dòng	Mô tả
1	<code>#include <stdio.h></code>
2	<code>void main(){</code>
3	<code>float sum = 0, x = 1;</code>
6	<code>while(x > 0){</code>
8	<code>printf("Nhập x: ");</code>
9	<code>scanf("%f", &x);</code>
10	<code>if(x > 0) sum+=x;</code>
12	<code>}</code>
13	<code>printf("Tong la: %f\n", sum);</code>
14	<code>}</code>



CẤU TRÚC LẶP

- Cấu trúc **do-while**

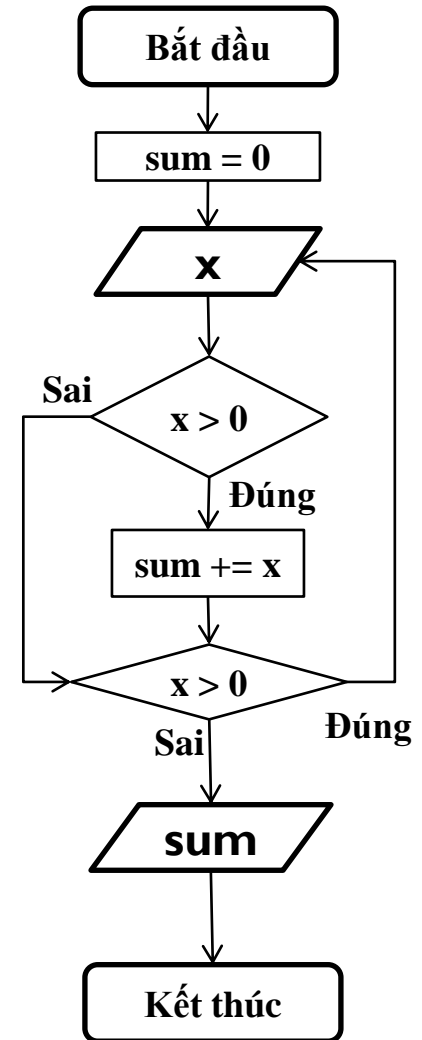
```
...  
do{  
    //Lệnh trong do-while  
}while(<điều kiện>;  
//Lệnh kế tiếp
```



CẤU TRÚC LẶP

- Ví dụ cấu trúc lặp **do-while**

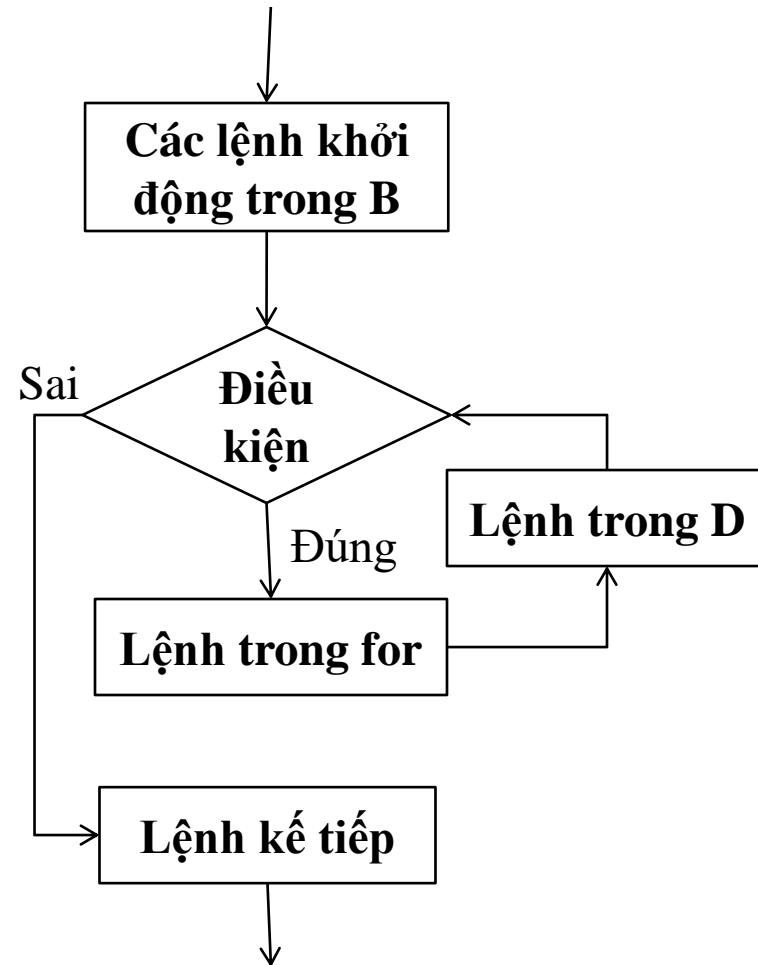
Dòng	Mô tả
1	<code>#include <stdio.h></code>
2	<code>void main(){</code>
3	<code>float sum = 0, x;</code>
6	<code>do{</code>
8	<code>printf("Nhập x: ");</code>
9	<code>scanf("%f", &x);</code>
10	<code>if(x > 0) sum+=x;</code>
12	<code>}while(x > 0);</code>
13	<code>printf("Tổng là: %f\n", sum);</code>
14	<code>}</code>



CẤU TRÚC LẶP

- Cấu trúc **for**

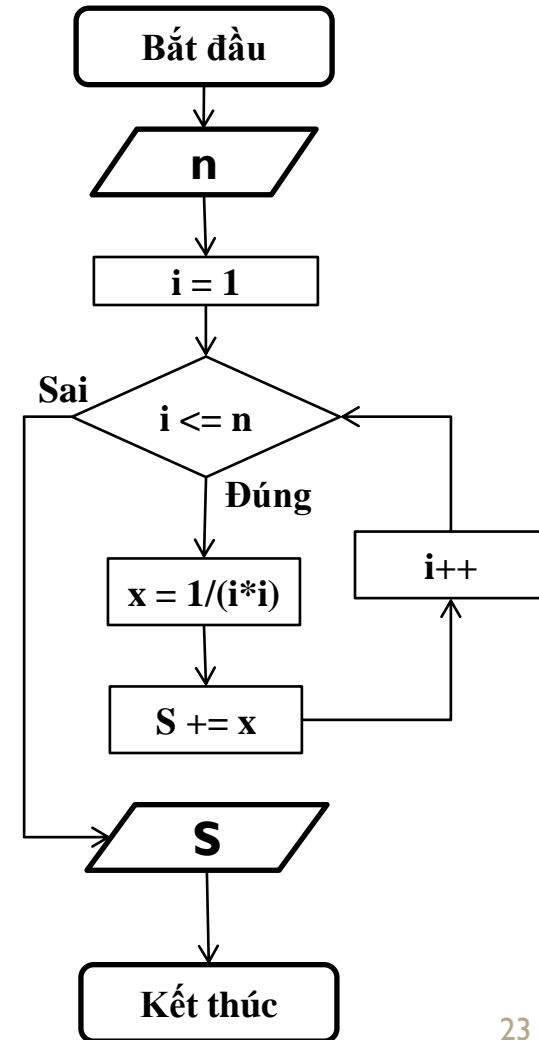
```
...  
for(B; <điều kiện>; D)  
  //Lệnh trong for  
}  
//Lệnh kế tiếp
```



CÂU TRÚC LẶP

- Ví dụ cấu trúc lặp **for**

Dòng	Mô tả
1	<code>#include <stdio.h></code>
2	<code>void main(){</code>
3	<code> long n; double S, x, i;</code>
4	<code> printf("Input n: ");</code>
5	<code> scanf("%ld", &n);</code>
6	<code> for(i = 1; i <= n; i++){</code>
7	<code> x = 1/(i*i);</code>
8	<code> S+=x;</code>
9	<code> }</code>
10	<code> printf("Result = %lf\n", S);</code>
11	<code>}</code>



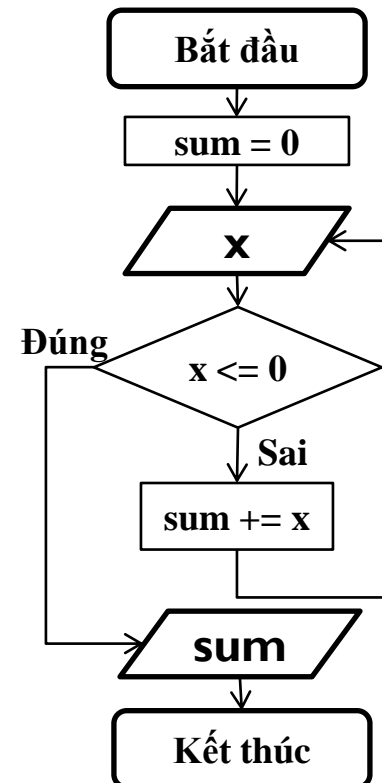
NỘI DUNG

- Khởi lệnh
- Cấu trúc rẽ nhánh
- Cấu trúc lặp
- Câu lệnh break & continue
- Điều kiện dừng vòng lặp
- Bài tập

CÂU LỆNH BREAK & CONTINUE

- ‘**break**’ dùng để thoát ra khỏi vòng lặp gần nhất trực tiếp chứa nó
- Thường dùng **break** với **if** hay **switch**

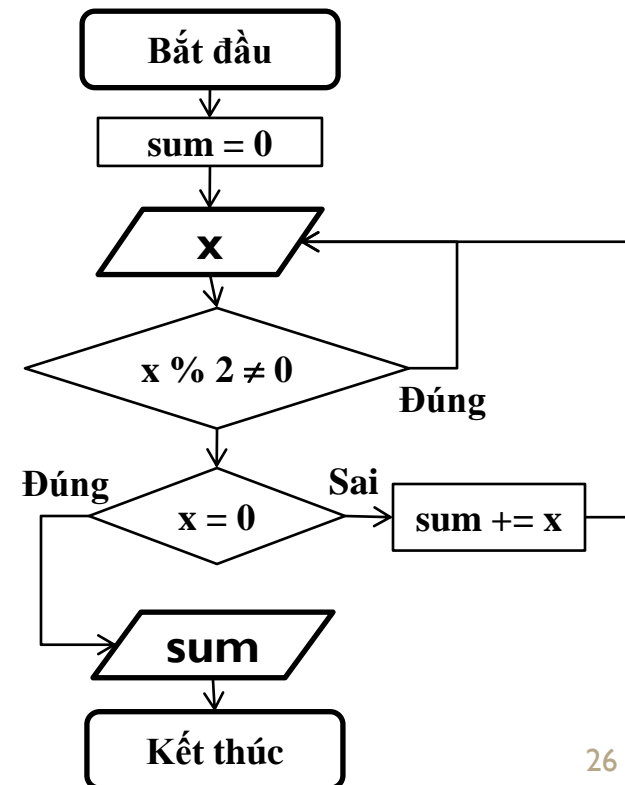
Dòng	Mô tả
1	<code>#include <stdio.h></code>
2	<code>void main(){</code>
3	<code>float sum = 0, x;</code>
4	<code>do{</code>
5	<code>printf("Nhập x: ");</code>
6	<code>scanf("%f", &x);</code>
7	<code>if(x <= 0) break;</code>
8	<code>sum+=x;</code>
9	<code>}while(1);</code>
10	<code>printf("Tổng là: %f\n", sum);}</code>



CÂU LỆNH BREAK & CONTINUE

- ‘**continue**’ dùng để quay lại vòng lặp và bỏ qua các lệnh phía dưới **continue**
- Thường dùng **continue** với **if**

Dòng	Mô tả
1	<code>void main(){</code>
2	<code>int sum = 0, x;</code>
3	<code>do{</code>
4	<code>printf("Nhập x: ");</code>
5	<code>scanf("%d", &x);</code>
6	<code>if(x % 2 != 0) continue;</code>
7	<code>if(x == 0) break;</code>
8	<code>sum+=x;</code>
9	<code>}while(1);</code>
10	<code>printf("Tổng là: %d\n", sum);}</code>



NỘI DUNG

- Khởi lệnh
- Cấu trúc rẽ nhánh
- Cấu trúc lặp
- Câu lệnh break & continue
- Điều kiện dừng vòng lặp
- Bài tập

ĐIỀU KIỆN DỪNG VÒNG LẶP

- Khi dừng vòng lặp phải đảm bảo có điều kiện thoát ra khỏi vòng lặp
- Nếu không có sẽ dẫn tới tình trạng lặp vô tận
- Cần đảm bảo một trong các điều kiện sau:
 - Có ít nhất một lần các lệnh trong vòng lặp làm điều kiện lặp không thỏa
 - Có ít nhất một lần điều khiển chương trình chạy tới chỉ thị **break**

NỘI DUNG

- Khởi lệnh
- Cấu trúc rẽ nhánh
- Cấu trúc lặp
- Câu lệnh break & continue
- Điều kiện dừng vòng lặp
- Bài tập

BÀI TẬP

- Thực hiện các bài tập sau:
 - Tính $S(n) = 1 + 2 + \dots + n$
 - Tính $S(n) = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}$
 - Tính $S(n) = 1 + 1 \times 2 + 1 \times 2 \times 3 + \dots + 1 \times \dots \times n$
 - Tính $S(x, n) = x + x^2 + x^3 + \dots + x^n$
 - Tính $S(n) = \sqrt{2 + \sqrt{2 + \sqrt{2 + \dots \sqrt{2 + \sqrt{2}}}}}$
 - Đếm số lượng ký số của số nguyên dương n:
ví dụ $n = 2938$ thì kết quả sẽ ra 4
 - Tính tích của các ký số của số nguyên dương n: ví dụ $n = 1234$ thì kết quả là 24