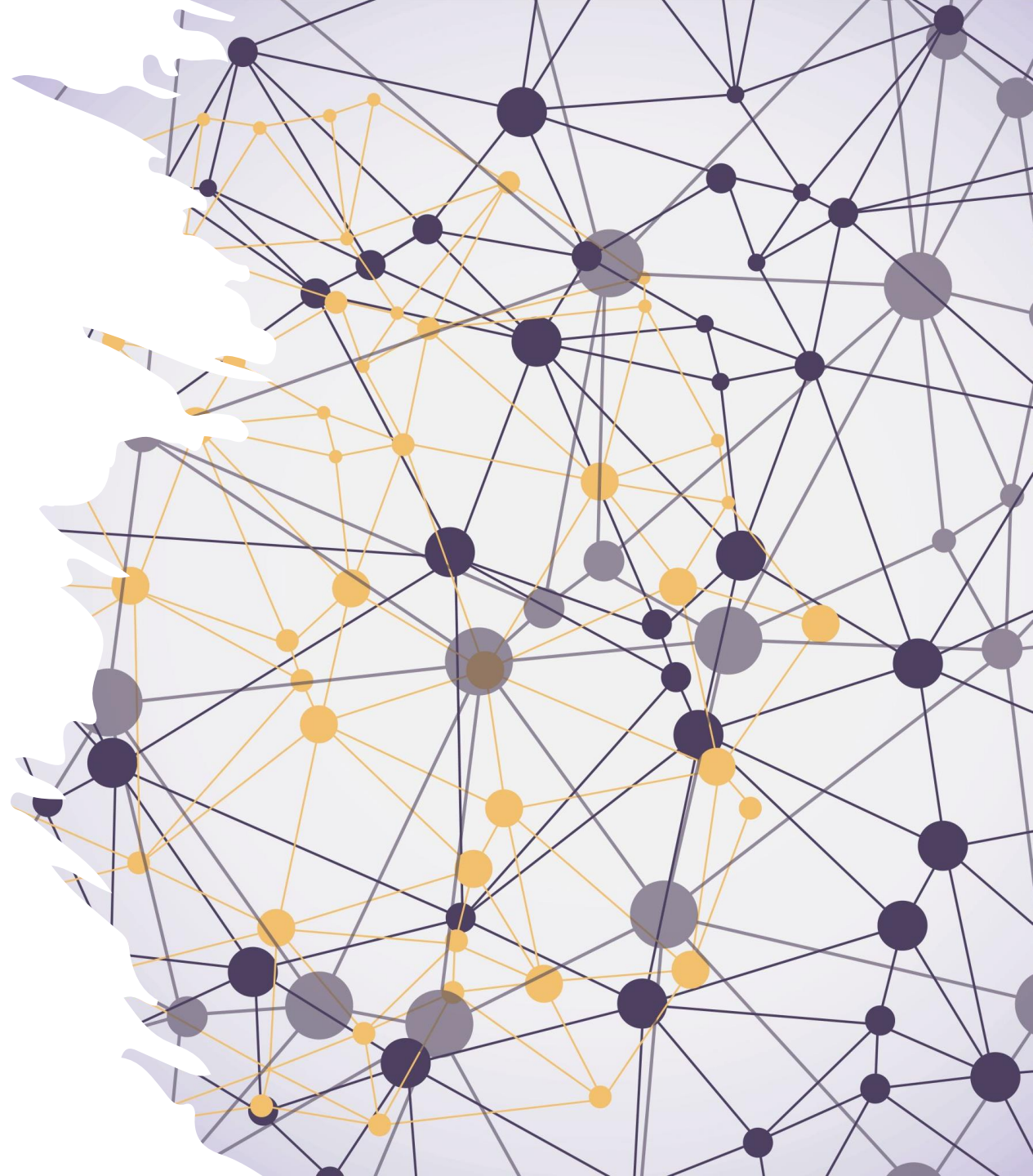


# KỸ THUẬT LẬP TRÌNH

## Chương 2. CÁC CẤU TRÚC ĐIỀU KHIỂN

Liên hệ gv: [thuong.bui@ut.edu.vn](mailto:thuong.bui@ut.edu.vn)



## 2.1 Cấu trúc tuần tự

Các lệnh được thực hiện tuần tự nối tiếp nhau từ trên xuống dưới.

Ví dụ: Chương trình tính tổng 2 số nguyên

```
#include<iostream>
using namespace std;
int main(){
    int a, b, s;
    cout<<"Nhập vào hai số nguyên:\n";
    cin>>a>>b;
    s = a + b;
    cout<<"Tổng hai số vừa nhập là "<<s<<"\n";
}
```

Kết quả chạy chương trình

Nhập vào hai số nguyên:

5

6

Tổng hai số vừa nhập là 11



## 2.2 Cấu trúc rẽ nhánh

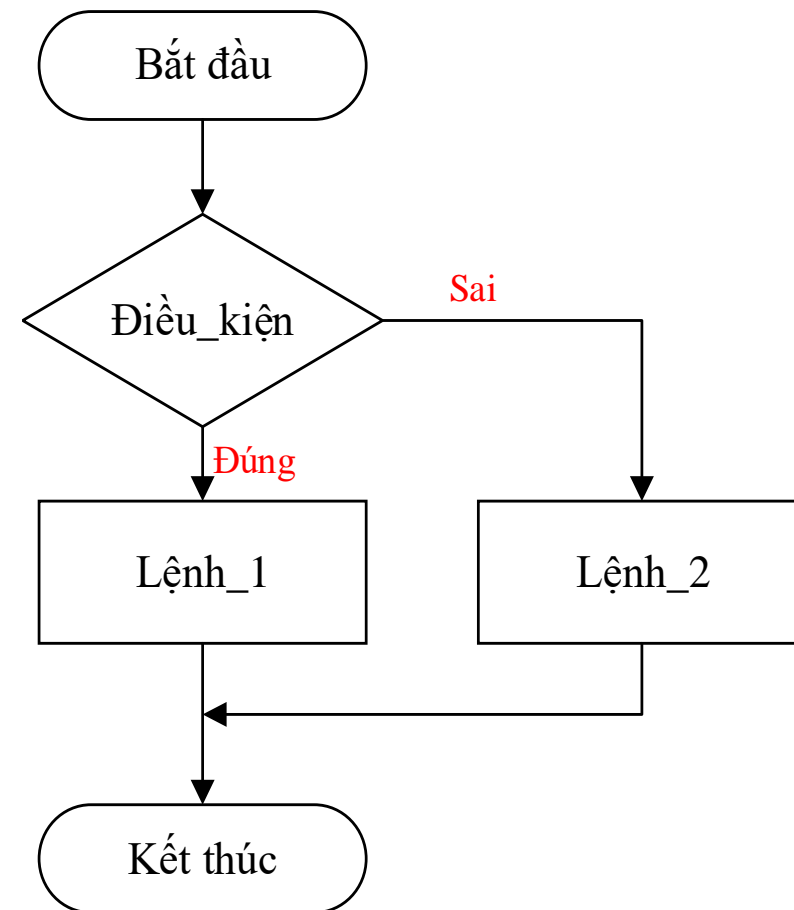
### Lệnh if

```
if(Điều_kiện){  
    //Các câu lệnh nếu điều kiện đúng  
    Lệnh_1;  
}  
else {  
    //Các câu lệnh nếu điều kiện sai  
    Lệnh_2;  
}
```

*Điều\_kiện: là một biểu thức, thông thường là biểu thức logic cho giá trị đúng (khác 0) hoặc sai (bằng 0).*

*Phần **else** không bắt buộc phải có - câu lệnh if không có phần **else** được gọi là câu lệnh “**if khuyết**”.*

*Các câu lệnh if có thể lồng trong nhau.*



## 2.2 Cấu trúc rẽ nhánh

Ví dụ: chương trình tính căn bậc hai của 1 số

```
#include<iostream>
#include<math.h>
using namespace std;
int main(){
    float x;
    cout<<"Nhap x = ";
    cin>>x;
    if(x<0){
        cout<<"Khong ton tai can bac hai cua "<<x;
    }
    else{
        cout<<"Can bac hai cua " << x << " bang "<< sqrt(x);
    }
}
```

Kết quả chạy thử chương trình

Nhap x = 36

Can bac hai cua 36 bang 6

Nhap x = -16

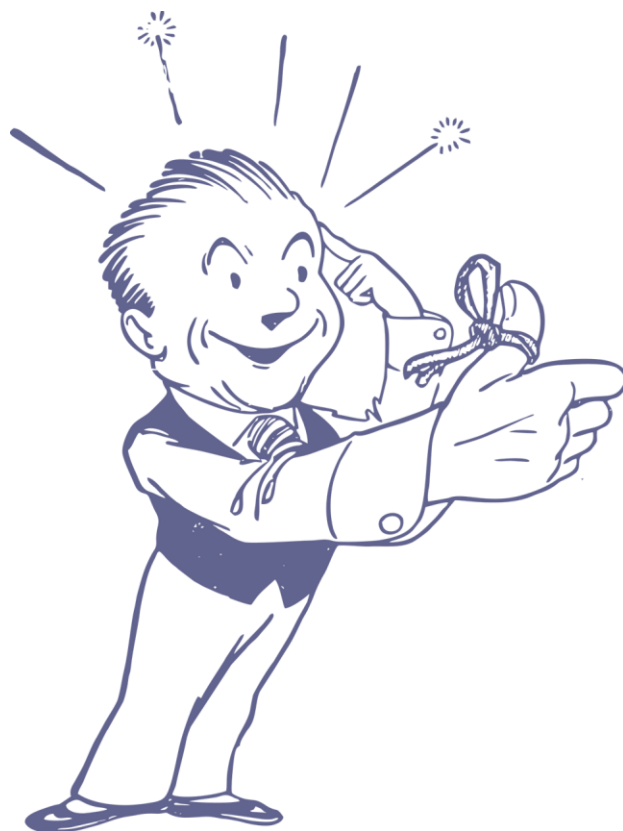
Khong ton tai can bac hai cua -16



## 2.2 Cấu trúc rẽ nhánh

### Lệnh switch

```
switch(biến/biểu_thức){  
    case giá_trị_1:  
        lệnh_1;  
        break;  
    case giá_trị_2:  
        lệnh_2;  
        break;  
    .....  
    default: lệnh_n ;  
}
```



Nếu *biến/biểu\_thức* có giá trị là *giá\_trị\_i* thì các lệnh bắt đầu từ *lệnh\_i* sẽ được thực hiện cho đến khi nào gặp lệnh **break** hoặc dấu **}** thì thoát khỏi **switch**. Ngược lại thì *lệnh\_n* sau từ khóa **default** sẽ được thực hiện.

**Chú ý:** biến/biểu\_thức trong switch chỉ áp dụng cho kiểu số **nguyên** hoặc **kí tự**.

## 2.2 Cấu trúc rẽ nhánh

### Ví dụ về lệnh switch

```
#include<iostream>
using namespace std;
int main(){
    int x;
    cout<<"Nhap so thu tu: ";
    cin>>x;
    switch(x){
        case 1: cout<<"Cao Van Hau\n"; break;
        case 2: cout<<"Ly Tuan Kiet\n"; break;
        case 3: cout<<"Van Tuan Anh\n"; break;
        default: cout<<"Khong co du lieu\n";
    }
}
```

### Kết quả chạy chương trình

Nhap so thu tu: 2  
Ly Tuan Kiet

Nhap so thu tu: 4  
Khong co du lieu



## 2.3 Cấu trúc lặp

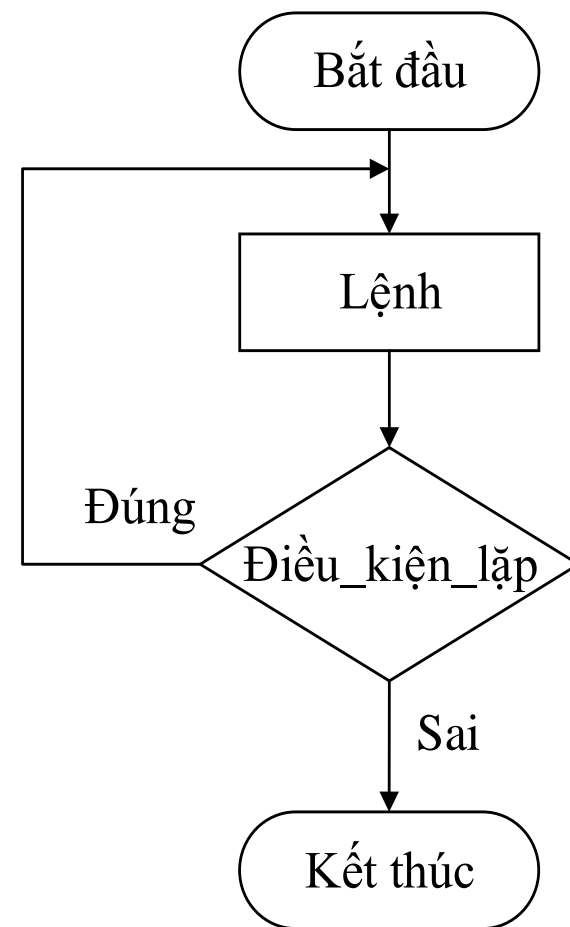
### Vòng lặp do...while

```
do {  
    /*thân vòng lặp*/  
    Lệnh;  
}  
while(điều_kiện_lặp);
```

Cách hoạt động của vòng lặp do...while() là thực hiện các lệnh trong thân vòng lặp cho đến khi nào điều kiện lặp mang giá trị sai hoặc bằng 0.

Lệnh thực hiện ít nhất được 1 lần, vì thực hiện lệnh trước rồi mới kiểm tra điều kiện lặp sau.

Các vòng lặp do...while có thể lồng trong nhau.





## 2.3 Cấu trúc lặp

Ví dụ: Nhập/xuất ký tự liên tục cho tới khi nhập khoảng trống

```
#include<iostream>
using namespace std;
int main() {
    char kt;
    do {
        cout << "Nhap vao mot ki tu: ";
        cin.get(kt);
        cout << "Ban vua nhap: " << kt << endl;
        cin.ignore();
    }
    while(kt!=' ');
    cout<<"===== END =====\n";
    getchar(); //Dừng màn hình xem kết quả
}
```

Kết quả chạy chương trình

```
Nhap vao mot ki tu: t
Ban vua nhap: t
Nhap vao mot ki tu: K
Ban vua nhap: K
Nhap vao mot ki tu: 4
Ban vua nhap: 4
Nhap vao mot ki tu:
Ban vua nhap:
===== END =====
```





## 2.3 Cấu trúc lặp

### Vòng lặp while

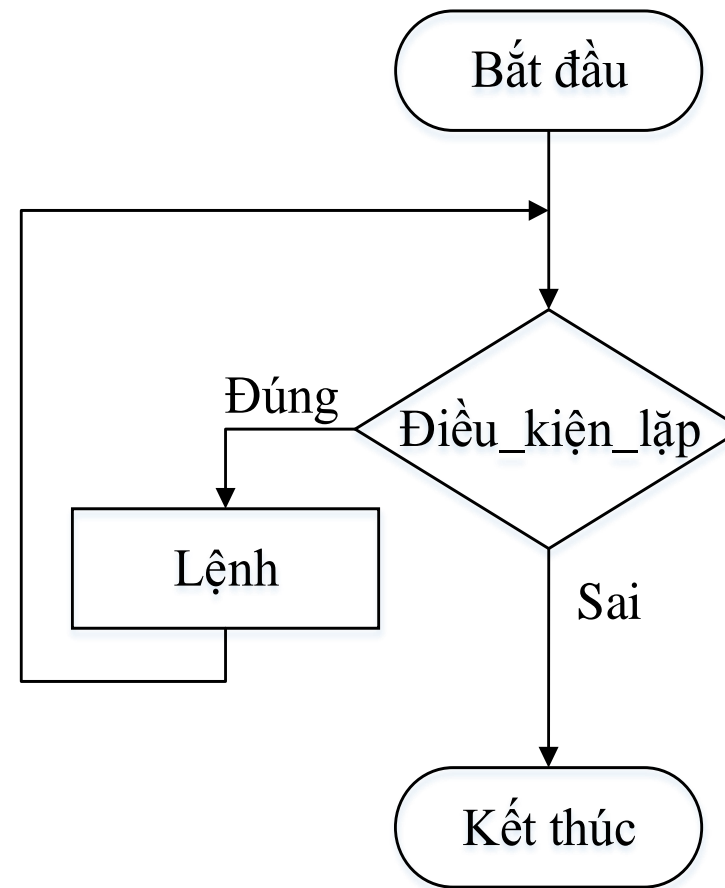
```
while(điều_kiện_lặp){  
    /*thân vòng lặp*/  
    Lệnh;  
}
```

Nếu điều kiện lặp có giá trị đúng hoặc khác 0 thì các lệnh trong thân vòng lặp sẽ thực hiện cho đến khi nào điều kiện lặp sai hoặc bằng 0 thì mới dừng.

Lệnh có thể không thực hiện được lần nào vì nếu lần đầu kiểm tra điều kiện có giá trị sai hoặc bằng 0.

Trường hợp vòng lặp luôn đúng thì số lần lặp khi đó là vô hạn.

Các vòng lặp while có thể lồng nhau.



## 2.3 Cấu trúc lặp

Ví dụ: Xuất ra màn hình các số dương chẵn nhỏ hơn n

```
#include<iostream>
using namespace std;
int main() {
    int n, x = 0;
    cout << "Nhap n = ";
    cin >> n;
    cout<<"Cac so chan nho hon "<<n<<" la:\n";
    while (x < n){
        cout << x << '\t';
        x += 2;
    }
    cout<<"\n===== END =====\n";
    system("pause");
}
```

Kết quả chạy chương trình

```
Nhap n = 7
Cac so chan nho hon 7 la:
0      2      4      6
===== END =====
```



## 2.3 Cấu trúc lặp

### Vòng lặp for

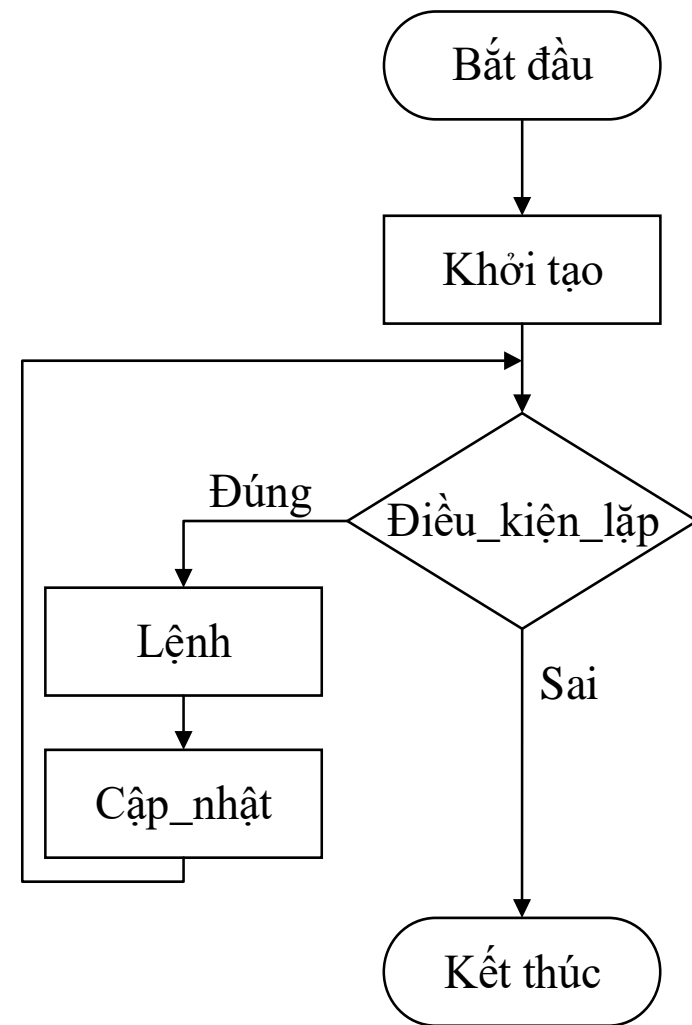
```
for(khởi_tạo; điều_kiện_lặp; cập_nhật) {  
    Lệnh; // thân vòng lặp  
}
```

**Ý nghĩa:** đầu tiên khởi tạo giá trị, kiểm tra điều kiện lặp, nếu đúng thì thực hiện lệnh, cập nhật lại giá trị của biến đã khởi tạo, tiếp tục kiểm tra điều kiện lặp, nếu đúng thực hiện lệnh, lặp lại đến khi điều kiện lặp có giá trị sai thì thoát vòng lặp.

Một câu lệnh **for** có thể lồng bên trong nó một lệnh **for** khác.

Ba thành phần trong lệnh **for** có thể vắng, nhưng vẫn phải có đủ hai dấu chấm phẩy.

Các thành phần trong lệnh **for** có thể là các biểu thức ghép. Biểu thức ghép phân cách nhau bởi toán tử dấu phẩy và được thực hiện từ trái sang phải.



## 2.3 Cấu trúc lặp

Ví dụ: Xuất ra màn hình các số dương lẻ nhỏ hơn n

```
#include<iostream>
using namespace std;
int main() {
    int n;
    cout << "Nhap n = ";
    cin >> n;
    cout<<"Cac so le tu 1 den "<<n<<" la:\n";
    for(int i = 1; i <= n; i+=2) {
        cout << i << "\t";
    }
    cout<<"\n===== END =====\n";
    system("pause");
}
```

Kết quả chạy chương trình

```
Nhap n = 8
Cac so le tu 1 den 8 la:
1          3          5          7
===== END =====
```



## 2.4 Lệnh nhảy vô điều kiện

### Lệnh nhảy break

Lệnh nhảy break - Lập tức thoát khỏi một cấu trúc lặp hoặc cấu trúc switch chứa nó.

```
int i, a, b, c;
cout<<"Nhap vao a,b,c:\n"; cin>>a>>b>>c;
cout<<"Trong khoang ("<<a<<', '<<b<<')';
if(c<b){
    for(i = b-1; i>a; i--){
        if (i%c == 0) break;
        if(i!=a) cout<<" so lon nhat chia het cho "<<c<<" la "<<i<<'\n';
        else cout<<" khong co so nao chia het cho "<<c<<'\n';
    }
else cout<<" khong co so nao chia het cho "<<c<<'\n';
system("pause");
```

Nhap vao a,b,c:

1 100 9

Trong khoang (1,100) so lon nhat chia het cho 9 la 99

## 2.4 Lệnh nhảy vô điều kiện

### Lệnh continue

Lệnh continue – Dùng trong các cấu trúc lặp để kết thúc lần lặp hiện hành và chuyển sang lần lặp kế tiếp.

```
#include<iostream>
using namespace std;
int main(){
    for(int i = 10; i <= 20; i++){
        if(i == 13 || i == 17) continue;
        cout<< i << (i != 20 ? ", " : "");
    }
    cout << '\n';
    system("pause");
}
```

Kết quả chạy chương trình trên sẽ in ra màn hình các số từ 10 đến 20 nhưng bỏ qua số 13 và 17.

```
10, 11, 12, 14, 15, 16, 18, 19, 20
```

## 2.4 Lệnh nhảy vô điều kiện

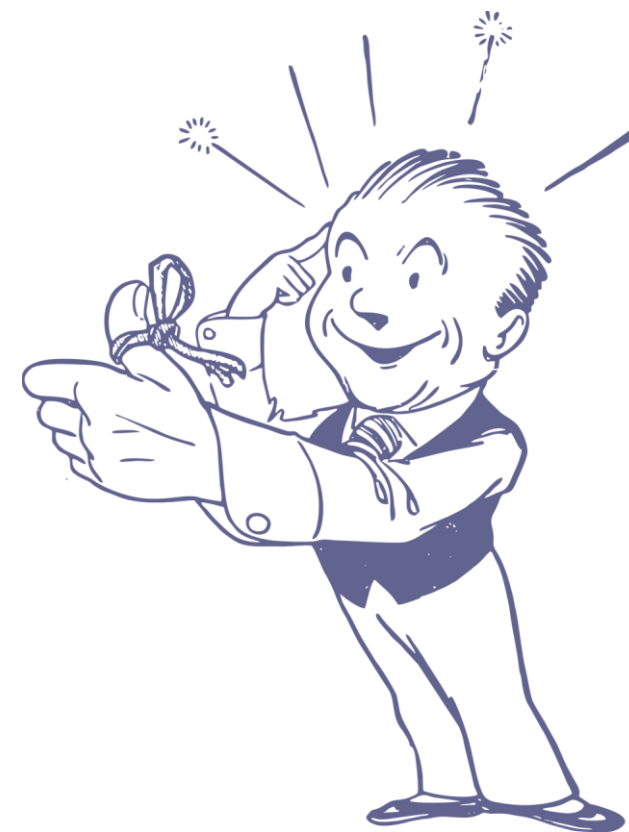
### Lệnh goto

Lệnh nhảy goto - Nhảy chương trình về thực hiện lệnh từ vị trí được gán nhãn.

```
#include<iostream>
using namespace std;
int main() {
    int n = 1;
    Lap: //Đây là nhãn
    cout<<n<<" "; n++;
    if(n<10) goto Lap; //Nhảy về nhãn Lap
    cout<<"\n";
    system("pause");
}
```

Kết quả chạy chương trình trên sẽ in ra màn hình các số từ 1 đến 9.

1 2 3 4 5 6 7 8 9





# Bài tập luyện tập

**Bài 1.** Viết chương trình giải biện luận phương trình bậc nhất  $ax + b = 0$ , theo các hệ số  $a, b$  bất kì.

**Bài 2.** Viết chương trình giải và biện luận phương trình bậc hai:  $ax^2 + bx + c = 0$ , với mọi  $a, b, c$ .

**Bài 3.** Nhập vào một kí tự. Cho biết kí tự đó có phải chữ cái Latinh không? ( $a \rightarrow z, A \rightarrow Z$ )

**Bài 4.** Viết chương trình nhập vào tháng, xuất ra mùa tương ứng với tháng đó (dùng switch).

**Bài 5.** Nhập vào 4 số nguyên, xuất ra số lớn nhất trong 4 số đó.

**Bài 6.** Gõ 1 phím, nếu là chữ hoa đổi ra chữ thường và ngược lại. Nếu là ký tự khác thì giữ nguyên.

**Bài 7.** Nhập tháng, năm cho biết tháng vừa nhập có bao nhiêu ngày (lưu ý năm nhuận).

(Biết rằng năm nhuận là năm (chia hết cho 400) hoặc (chia hết cho 4 và không chia hết cho 100))

**Bài 8.** Nhập  $n$  nguyên dương. Tính:

a.  $S = 1 + 2 + 3 + \dots + n$

b.  $S = 1/1 + 1/2 + \dots + 1/n$

**Bài 9.** Viết chương trình tính tổng các ước của 1 số nguyên dương nhập từ bàn phím.

# Bài tập luyện tập

**Bài 10.** Tìm tất cả các số dương có 4 chữ số sao cho tổng các chữ số của nó chia hết cho 6.

**Bài 11.** Viết chương trình phân tích 1 số nguyên dương  $n$  thành tích của 1 và các thừa số nguyên tố (ví dụ với  $n = 24$  thì ta phân tích:  $n = 1 * 2 * 2 * 2 * 3$ ).

**Bài 12.** Nhập 3 số nguyên dương  $a, b, n$  với  $a < n$  và  $b < n$  (có kiểm tra). Tìm các cặp số nguyên dương nhỏ hơn  $n$  sao cho tổng của chúng chia hết cho  $a$  và không chia hết cho  $b$ .

**Bài 13.** Viết chương trình đếm xem một số nguyên bất kì nhập từ bàn phím gồm mấy chữ số (ví dụ số 32145 gồm 5 chữ số là 3, 2, 1, 4, 5).

**Bài 14.** In ra màn hình tất cả các số nguyên trong phạm vi từ 10->99 sao cho tích của 2 chữ số bằng 2 lần tổng của chúng.

**Bài 15.** Viết chương trình in ra màn hình các ước số của một số nguyên dương  $n$ .

**Bài 16.** Tìm bội chung nhỏ nhất của 2 số nguyên dương bất kỳ.

**Bài 17.** Vẽ hình chữ nhật và hình tam giác vuông rỗng bằng dấu \*