



THỰC HỌC – THỰC NGHIỆP

NHẬP MÔN LẬP TRÌNH

ĐẠI CƯƠNG LẬP TRÌNH

BÀI 4: LỆNH LẶP



- ⊙ Hiểu về các cấu trúc trong lập trình
- ⊙ Biết cách sử dụng lệnh lặp while
- ⊙ Biết cách sử dụng lệnh lặp do while
- ⊙ Biết cách sử dụng lệnh lặp for
- ⊙ Biết cách sử dụng lệnh ngắt (break, continue, goto)



⊙ Lệnh lặp

❖ while

❖ do...While

⊙ Lệnh lặp & lệnh ngắt

❖ for(;điều kiện;)

❖ break

❖ continue

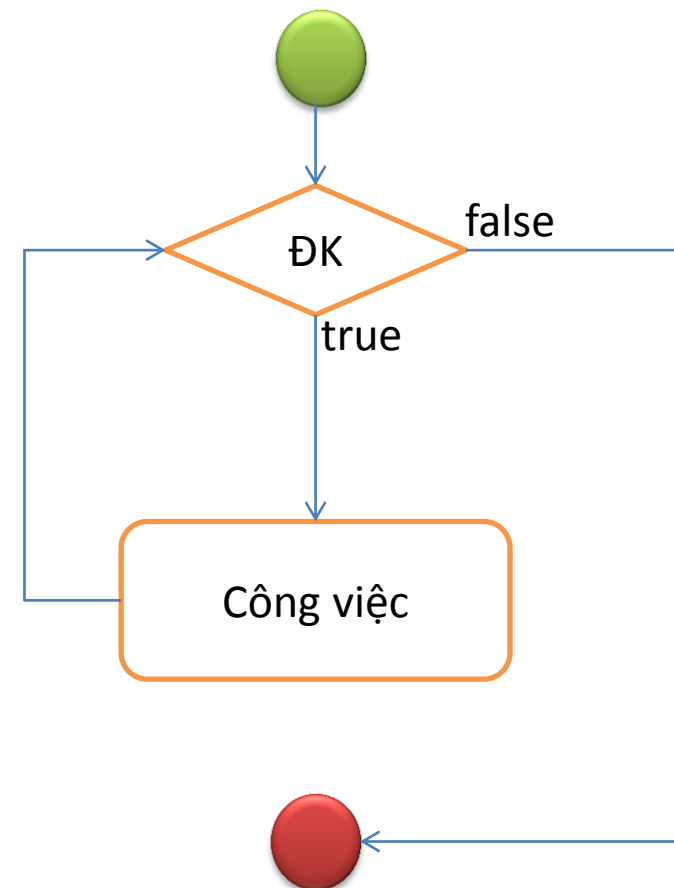
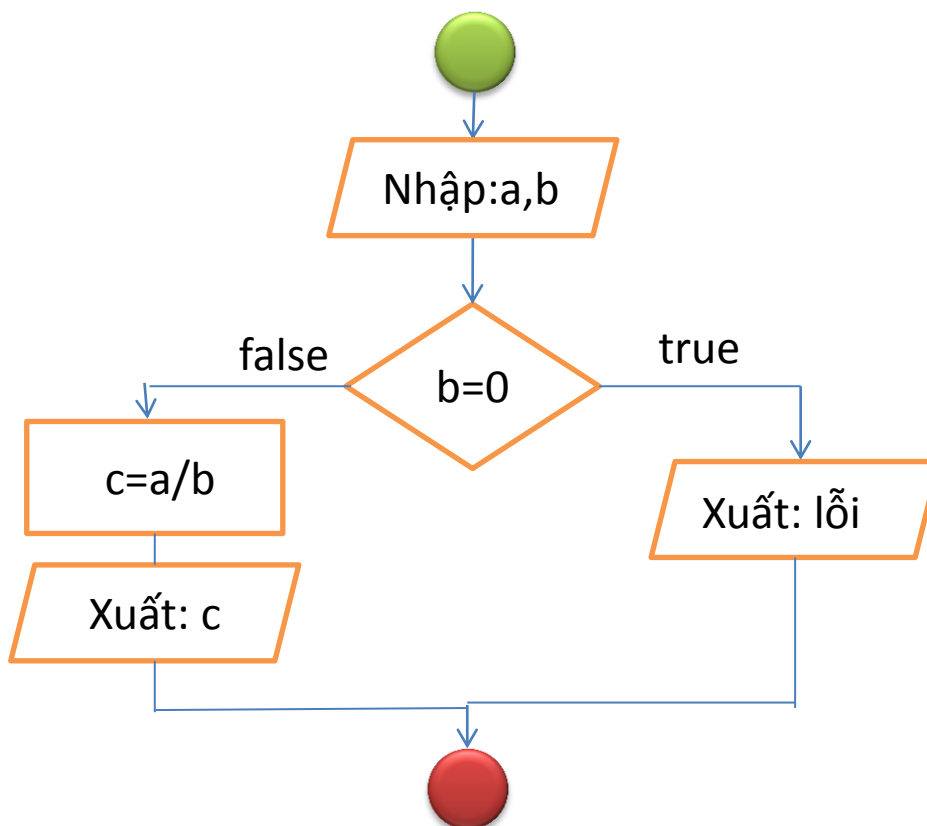
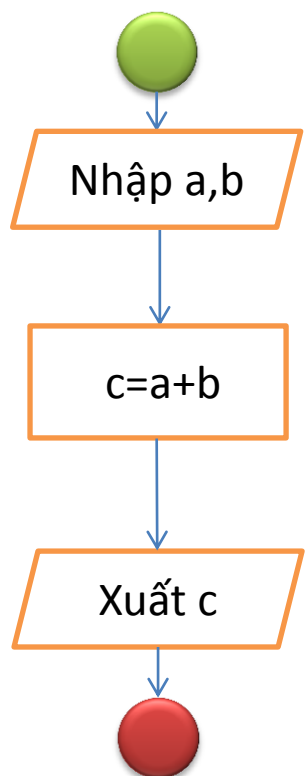
❖ goto



CẤU TRÚC LẬP TRÌNH

❑ Trong các ngôn ngữ lập trình, thường có 3 cấu trúc chính

- ❖ Cấu trúc tuần tự
- ❖ Cấu trúc quyết định chọn lựa rẽ nhánh
- ❖ Cấu trúc lặp



📖 Lệnh lặp được sử dụng để điều khiển các công việc được lặp đi lặp lại nhiều lần. Mỗi lệnh lặp luôn có điều kiện dừng. Ngôn ngữ C có 3 lệnh lặp:

- ❖ While
- ❖ Do While
- ❖ For

📖 Lệnh ngắt được sử dụng để ngắt vòng lặp bất kỳ khi nào chúng ta muốn. Ngôn ngữ C có 3 lệnh ngắt:

- ❖ break
- ❖ Continue
- ❖ goto





PHẦN 1: VÒNG LẶP WHILE – DO...WHILE

❑ Ví dụ

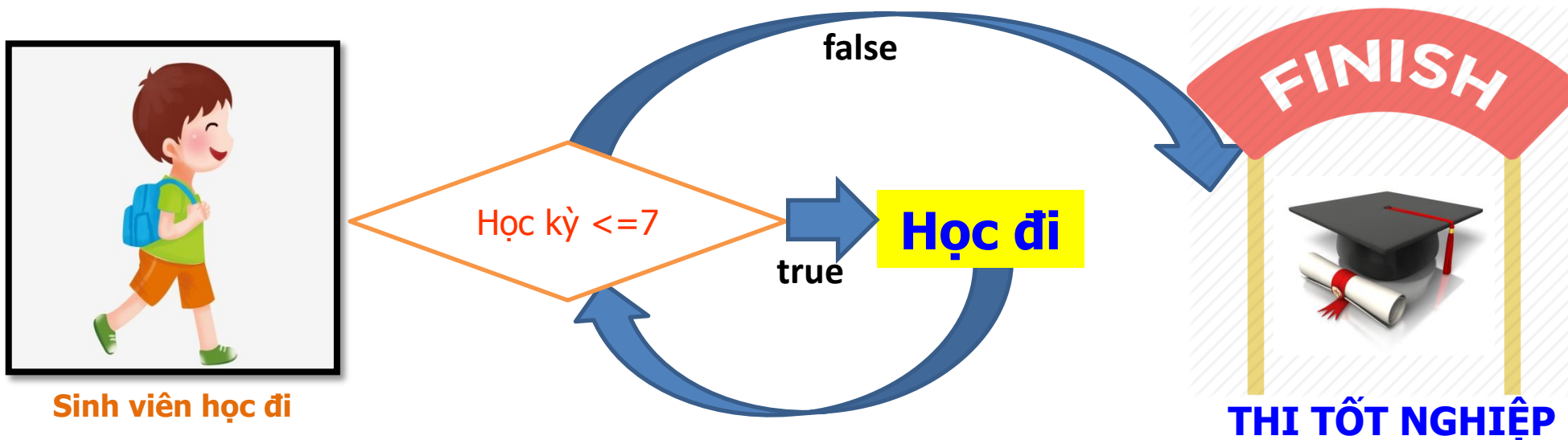
- ❖ Viết chương trình xuất các số từ 1 đến 10
=> Sử dụng 10 câu lệnh printf
- ❖ Viết chương trình xuất các số từ 1 đến 1000
=> Sử dụng 1000 câu lệnh printf !

❑ Giải pháp

- ❖ Sử dụng cấu trúc lặp lại một hành động trong khi còn thỏa một điều kiện nào đó.
- ❖ 3 lệnh lặp: for, while, do... while

❑ Sinh viên A đăng kí thi tốt nghiệp tại trường Fpoly.

- Nếu sinh viên A học không đủ số học kỳ (≤ 7) thì sinh viên A sẽ phải học tiếp.
- Trong trường hợp sinh viên A học tiếp lần thứ 2, chúng ta lại nói rằng nếu sinh viên A học không đủ số học kỳ để được thi tốt nghiệp thì sinh viên A phải học tiếp...
- Vậy việc sinh viên A “học đi” cho từng học kỳ là công việc sẽ được lặp đi lặp lại nhiều lần khi điều kiện sinh viên A đủ số học kỳ để thi tốt nghiệp vẫn chưa đạt.

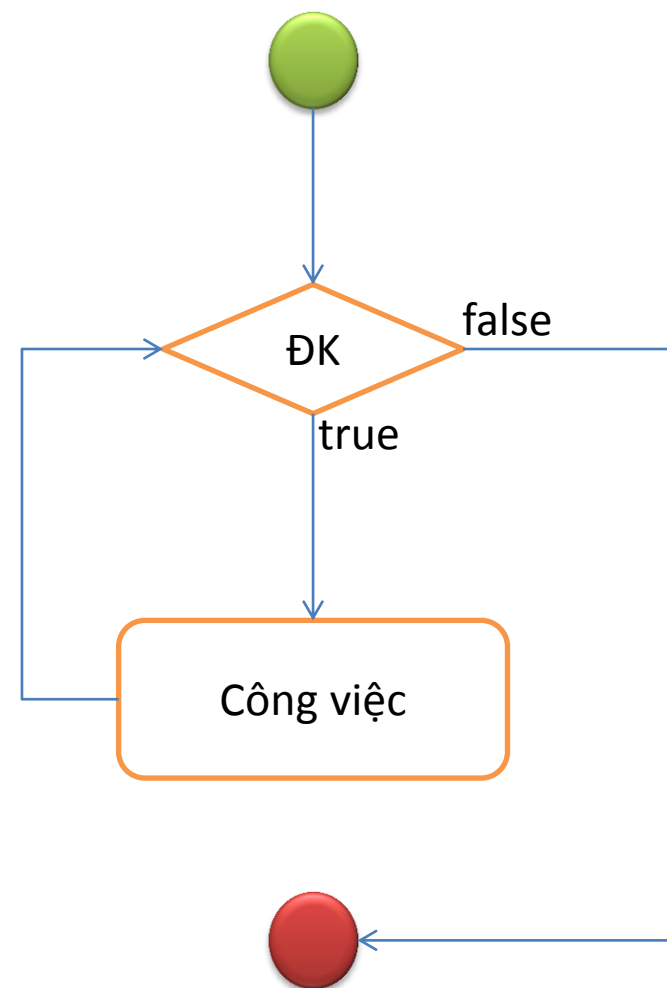


□ Cú pháp

```
while (<<điều kiện>>) {  
    // công việc;  
}
```

□ Diễn giải:

- ❖ Thực hiện công việc trong khi biểu thức điều kiện có giá trị là true.

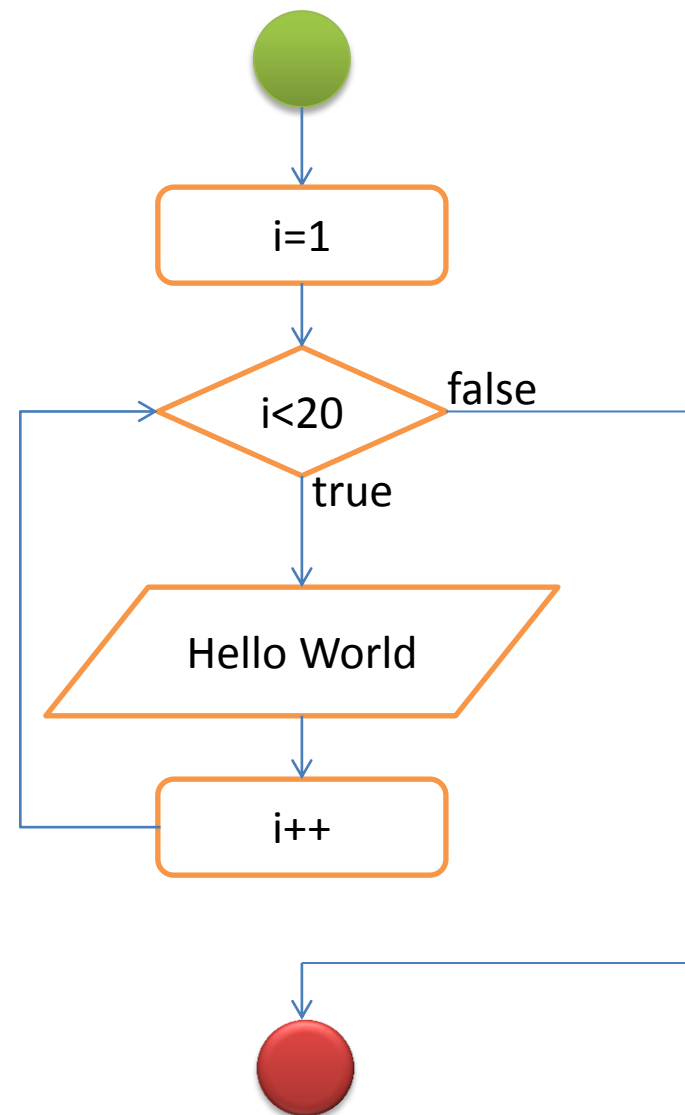


□ Ví dụ

```
int i = 1;
while (i < 20) {
    printf("Hello World !");
    i++;
}
```

□ Diễn giải:

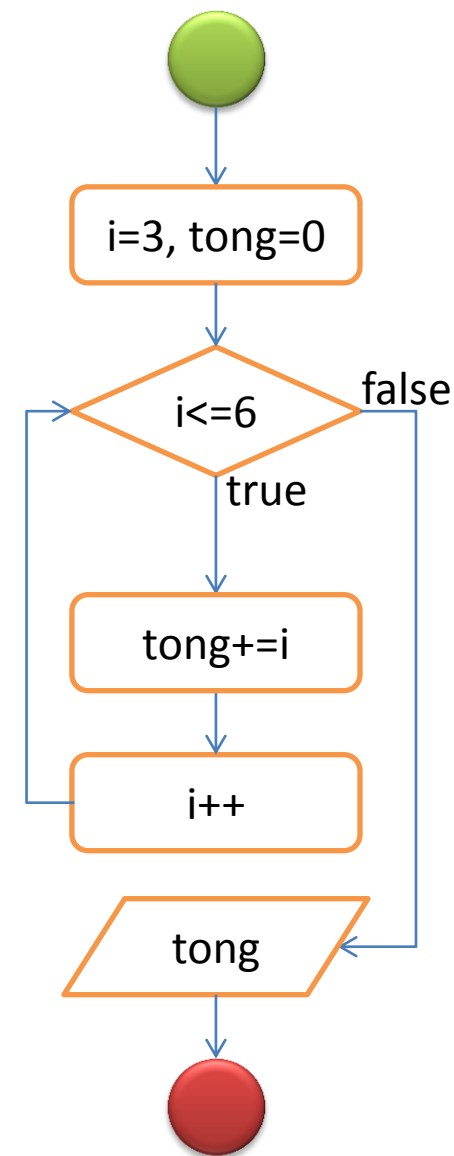
- ❖ Đoạn mã trên xuất 19 chữ "Hello World" ra màn hình



□ Tính tổng các số từ 3 đến 6

```
int i=3;
int tong=0;
while(i<=6){
    tong+=i;
    i++;
}
printf("%d", tong);
```

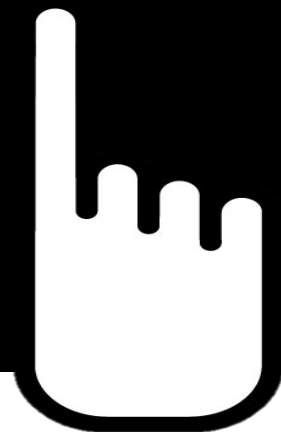
Lần	i	Tổng
1	3	3
2	4	7
3	5	12
4	6	18





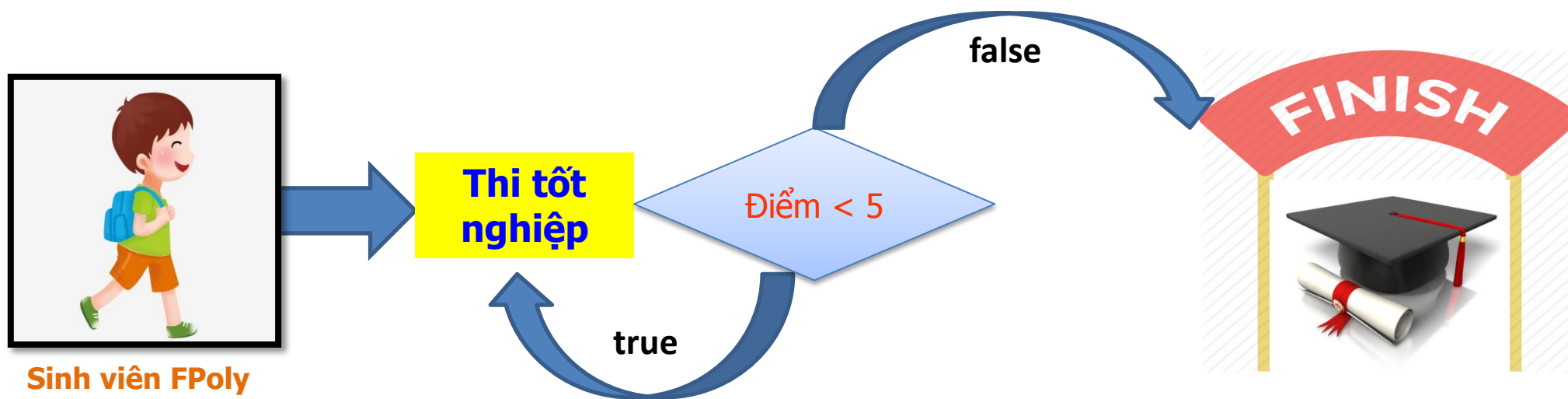
DEMO

1. Xuất bảng cửu chương 7
2. Tính trung bình cộng các số chia hết cho 3 từ 1 đến 20.



❑ Sinh viên A bắt đầu thi tốt nghiệp học tại trường Fpoly.

- Nếu sinh viên A thi tốt nghiệp dưới 5 điểm thì sinh viên A sẽ phải thi lại.
- Trong trường hợp sinh viên A thi điểm tốt nghiệp từ 5 điểm trở lên thì sinh viên A được xem là hoàn thành yêu cầu tốt nghiệp.
- Vậy việc sinh viên A “Thi tốt nghiệp” là công việc sẽ được lặp đi lặp lại nhiều lần khi điều kiện điểm thi tốt nghiệp của sinh viên A vẫn còn chưa đạt.



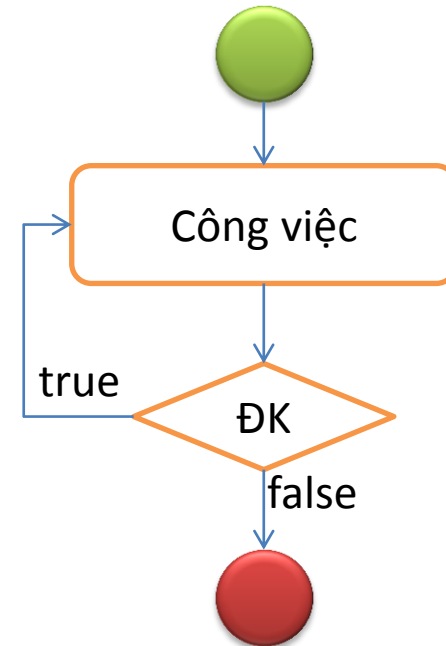
□ Cú pháp:

```
do {  
    // công việc  
}
```

```
while (<<điều kiện>>);
```

□ Diễn giải:

- ❖ Tương tự lệnh lặp while chỉ khác ở chỗ điều kiện được kiểm tra sau, nghĩa là công việc được thực hiện ít nhất 1 lần.



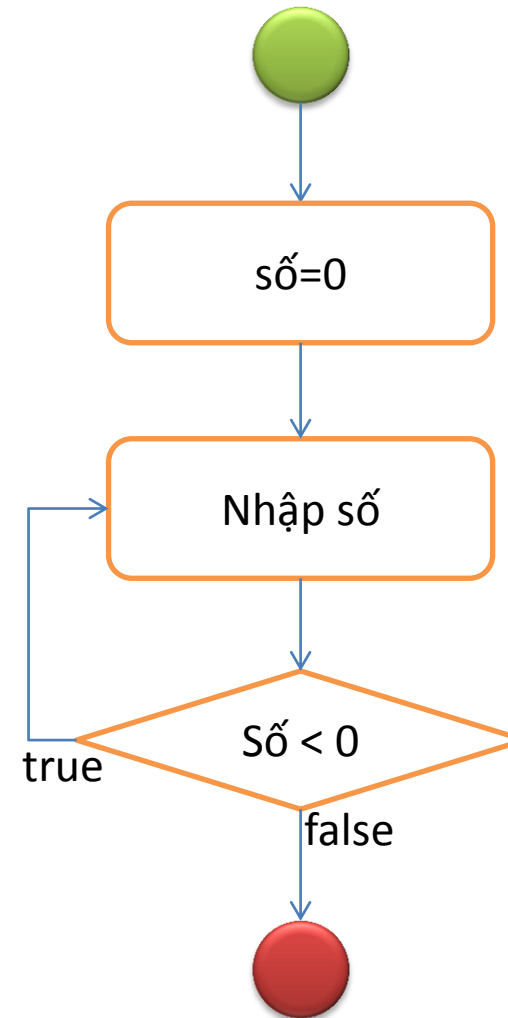
VÍ DỤ 1 LỆNH LẶP DO....WHILE

□ Ví dụ

```
int so = 0;  
do {  
    scanf("%d", &so);  
}  
while (so < 0);
```

□ Diễn giải:

- ❖ Đoạn mã trên chỉ cho phép nhập số nguyên dương từ bàn phím.



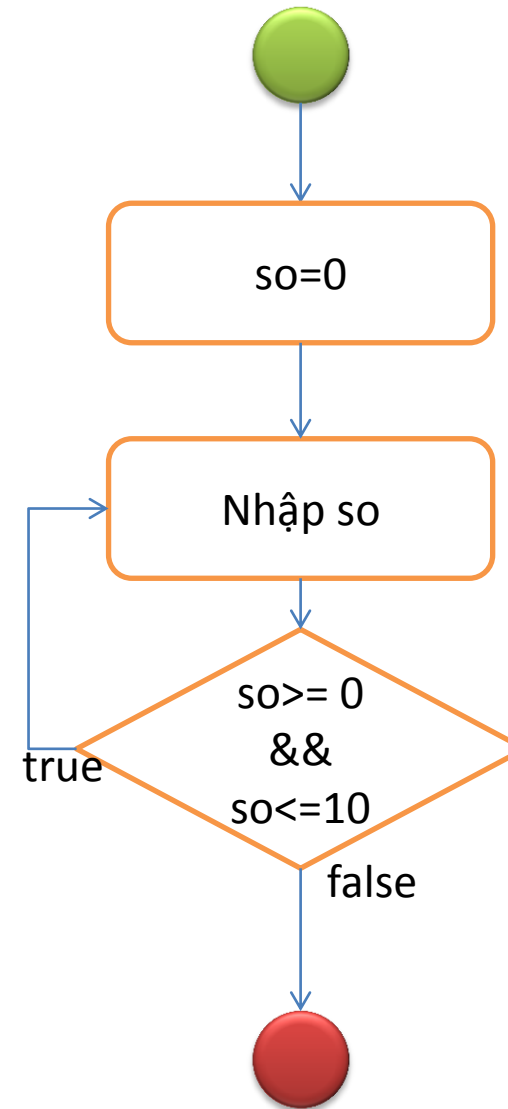
VÍ DỤ 2 LỆNH LẶP DO...WHILE

□ Ví dụ

```
int so=0;
do{
    printf("Nhập số: ");
    scanf("%d", &so);
}
while(so >= 0 && so <= 10);
```

□ Diễn giải:

- ❖ Đoạn mã trên chỉ cho phép nhập số nguyên dương từ 0-10 từ bàn phím.







CHÈN QUIZ

- ☑ Cấu trúc lập trình
 - ❖ Cấu trúc tuần tự
 - ❖ Cấu trúc rẽ nhánh
 - ❖ Cấu trúc lặp
- ☑ Lệnh lặp
 - ❖ while
 - ❖ do...While

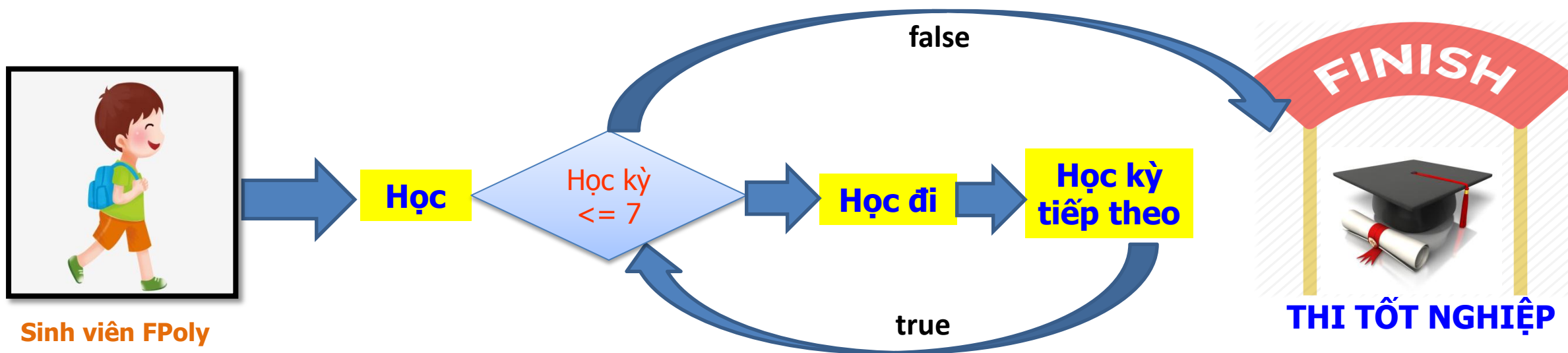




PHẦN 2: VÒNG LẶP FOR, BREAK, CONTINUE, GOTO

□ Sinh viên A bắt đầu học tại trường Fpoly.

- Nếu sinh viên A học chưa đủ 7 học kỳ tại Fpoly thì sinh viên A sẽ phải học tiếp.
- Trong trường hợp sinh viên A học đủ 7 học kỳ thì sinh viên A được xem là hoàn thành đủ số học kỳ yêu cầu.
- Vậy việc sinh viên A “học đi” là công việc sẽ được lặp đi lặp lại nhiều lần trong từng học kỳ khi điều kiện sinh viên A hoàn thành tổng số học kỳ vẫn chưa đạt.



□ Cú pháp

```

1 2 5 4 7
for (khởi đầu ; điều kiện ; bước nhảy) {
    // công việc 3 6
}
    
```

□ Diễn giải

❖ B1: Thực hiện <khởi đầu>

❖ B2: Kiểm tra <điều kiện>

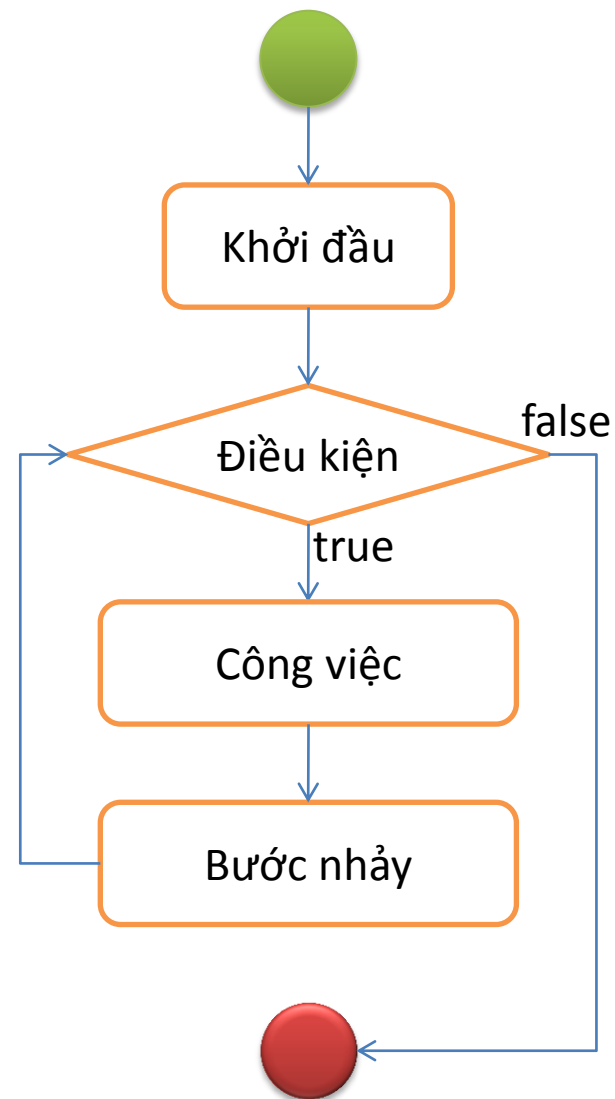
➤ True: thực hiện

✓ <công việc>

✓ <bước nhảy>

✓ Trở lại B2

➤ False: dừng lệnh lặp



```
int i;  
for(i=1;i<=10;i++){  
    printf("%d x %d = %d\n",7,i,7*i);  
}
```

7 x 1 = 7
7 x 2 = 14
7 x 3 = 21
7 x 4 = 28
7 x 5 = 35
7 x 6 = 42
7 x 7 = 49
7 x 8 = 56
7 x 9 = 63
7 x 10 = 70

Debug: Lặp lần 11



```
int i;  
for(i=11;i<=10;i++){ //11<=10 → false  
    printf("%d x %d = %d\n",7,i,7*i);  
}
```


❑ Ví dụ: Cần lấy tất cả hồ sơ sinh viên năm cuối để trong kho để xét tốt nghiệp

- ❖ Trong cuộc sống thực: Đi lấy từng hồ sơ trong kho
- ❖ Trong lập trình: Sử dụng vòng lặp biến đếm for để lấy ra từng sinh viên có trong kho dữ liệu

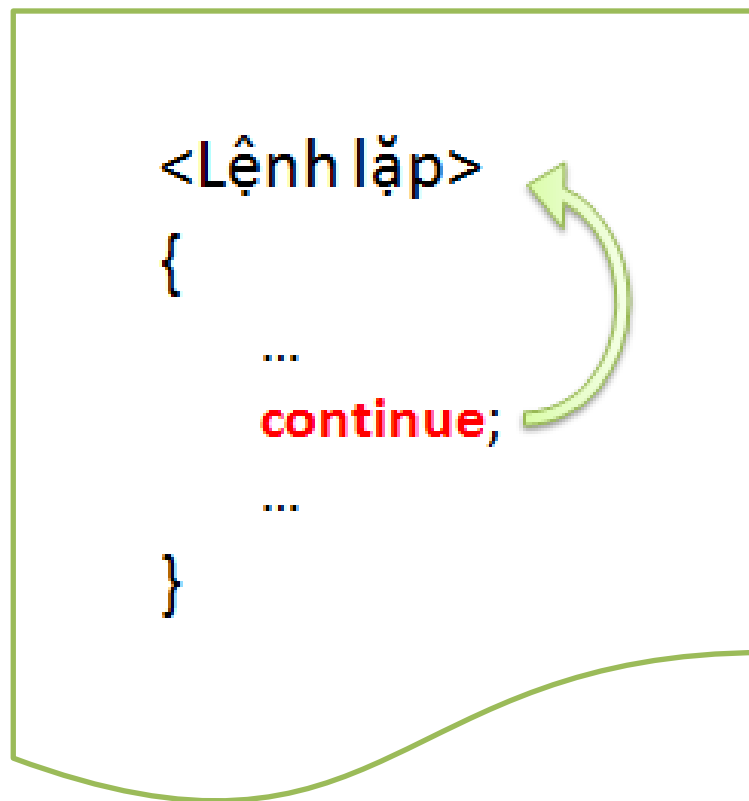
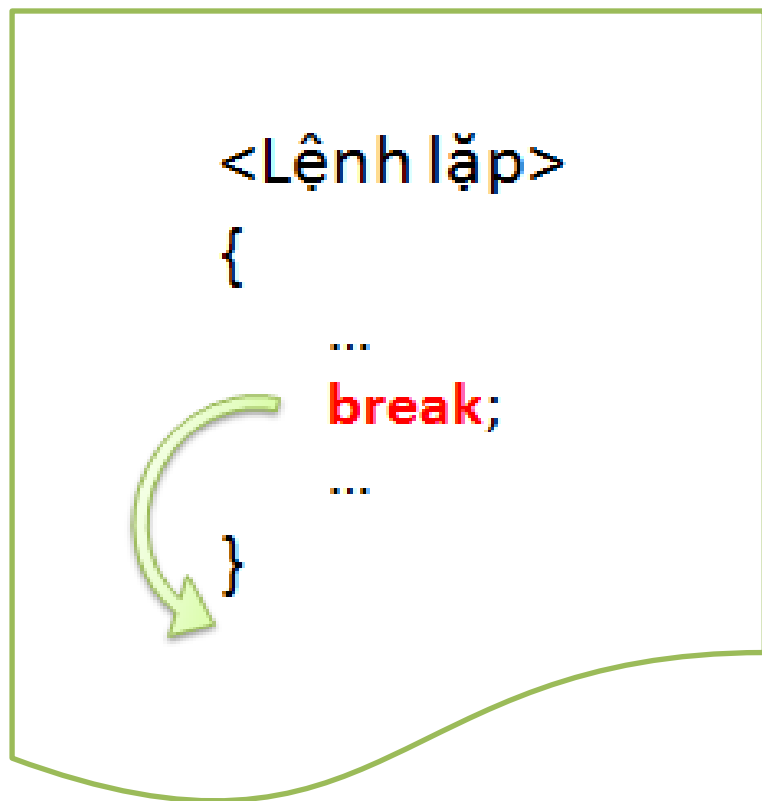




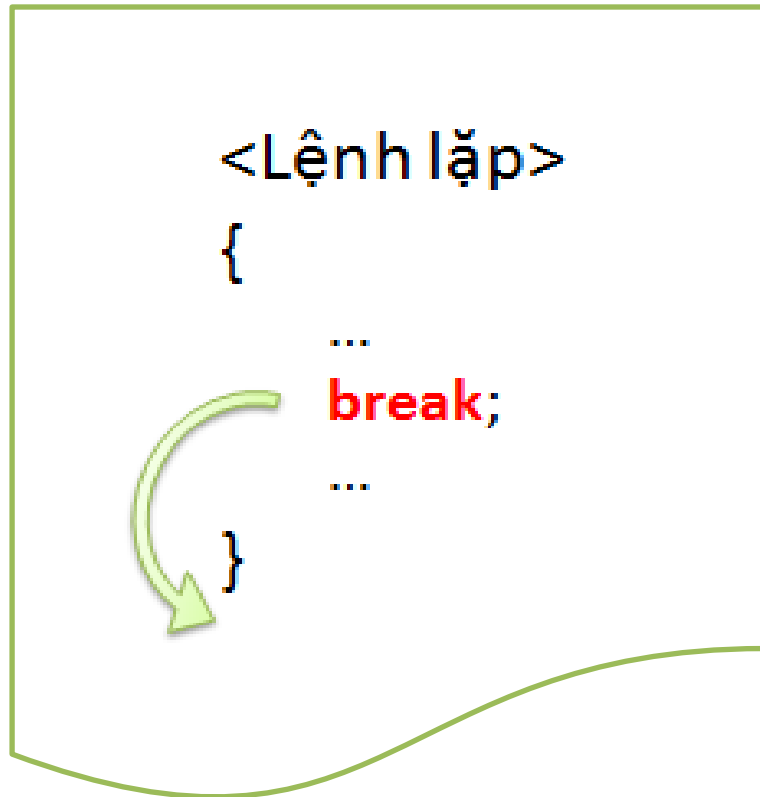
-  Lệnh ngắt được sử dụng để ngắt vòng lặp bất kỳ khi nào chúng ta muốn.
-  Ngôn ngữ C có 3 lệnh ngắt:
 - ❖ break
 - ❖ continue
 - ❖ goto



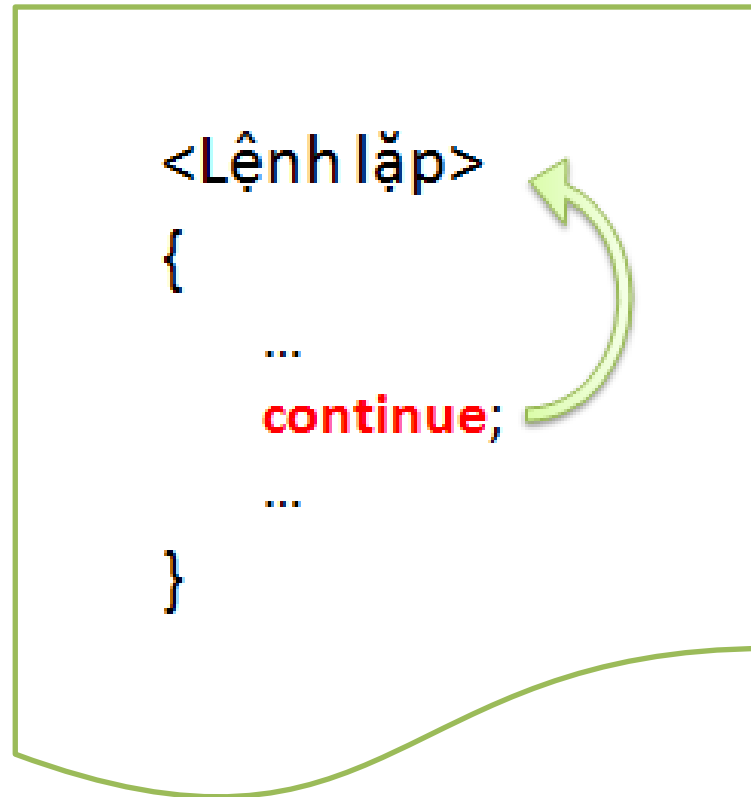
- ❑ **break** dùng để ngắt lệnh lặp
- ❑ **Continue** dùng để thực hiện lần lặp tiếp theo ngay lập tức



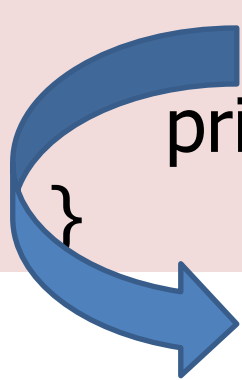
- ❑ Ví dụ: Đi vào kho hồ sơ và lấy ra sinh viên tên Trần Văn A
 - ❖ Khi lấy được sinh viên Trần Văn A thì đi ra, không cần thực hiện lấy các hồ sơ khác



- ❑ Ví dụ: Đóng mộc cho các sinh viên đã tốt nghiệp
 - ❖ Gặp những sinh viên nào chưa tốt nghiệp thì không cần thực hiện, tiếp tục xem các sinh viên tiếp theo

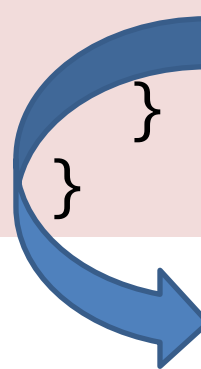


```
int i;
for(i=1;i<10;i++){
    if(i%2==0)
        break;
    printf("%d\n",i);
}
```



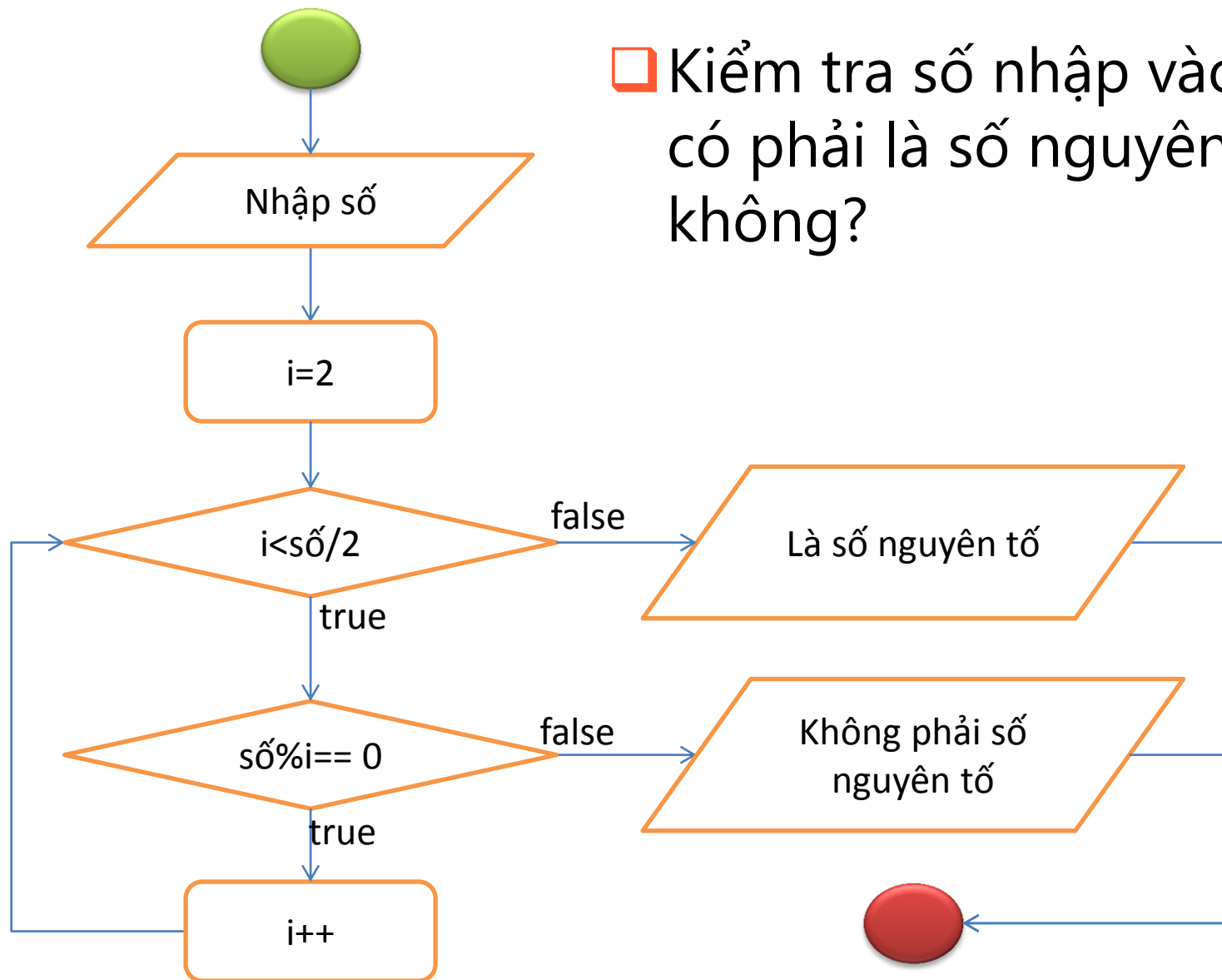
```
int a = 10;
while( a < 20 ) {
    printf("Gia tri cua a: %d\n", a);
    a++;

    if( a > 15) {
        /* ket thuc vong lap khi a lon hon 15 */
        break;
    }
}
```



VÍ DỤ : VIẾT MÃ THEO LƯU ĐỒ THUẬT TOÁN

❑ Kiểm tra số nhập vào từ bàn phím có phải là số nguyên tố hay không?




```
int i;
for(i=1;i<10;i++){
    if(i%2==0)
        continue;
    printf("%d\n",i);
}
```

```
int a = 10;
do {
    if( a == 15) {
        // quay về do khi a = 15 (bỏ qua lệnh print)
        a = a + 1;
        continue;
    }

    printf("Gia tri cua a: %d\n", a);
    a++;
} while( a < 20 );
```

1
3
5
7
9

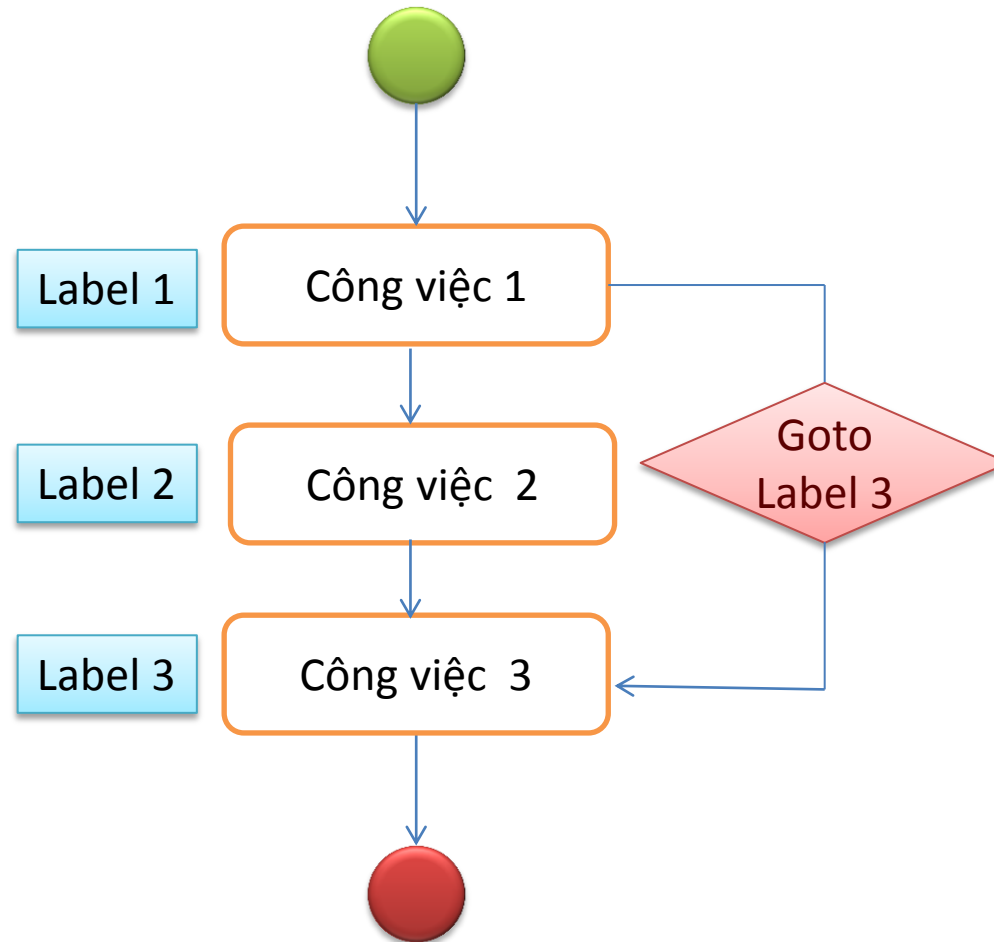
Gia tri cua a: 10
Gia tri cua a: 11
Gia tri cua a: 12
Gia tri cua a: 13
Gia tri cua a: 14
Gia tri cua a: 16
Gia tri cua a: 17
Gia tri cua a: 18
Gia tri cua a: 19

```
int x = 2;
while (x < 7) {
    if (x % 2 == 0) {
        x++;
        continue;
    } else {
        x++;
    }
    printf("%d \n", x);
}
```

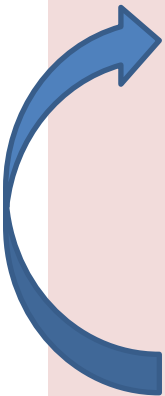
4
6



- ❑ **goto** cung cấp một bước nhảy vô điều kiện đến một câu lệnh có nhãn trong cùng một hàm.



```
int a = 10;
```



```
TEST:do {  
    if( a == 15) {  
        // quay ve do khi a = 15 (bỏ qua lệnh printf)  
        a = a + 1;  
        goto TEST;  
    }  
    printf("Gia tri cua a: %d\n", a);  
    a++;  
} while( a < 20 );
```



☑️ Lệnh lặp & lệnh ngắt

- ❖ for(;điều kiện;)
- ❖ break
- ❖ continue
- ❖ goto





thank
you!