



CẤU TRÚC LẬP TRONG PYTHON

ĐẶT VẤN ĐỀ

Bài toán: Nhập vào tuổi của bạn. Nếu tuổi > 40 thì thông báo "Bạn không còn trẻ", ngược lại thông báo "Bạn vẫn trẻ"

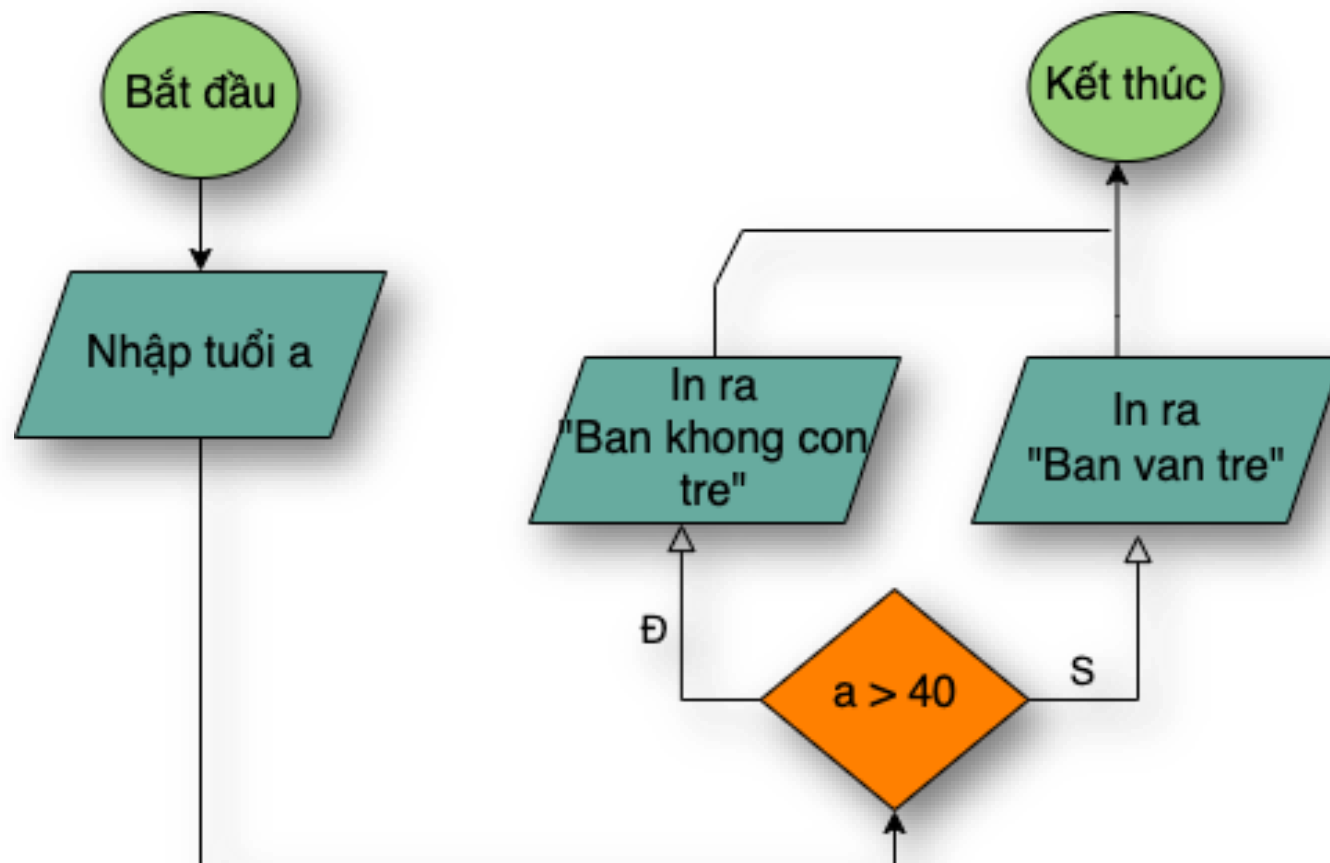
Ký hiệu (tham số/biến):

a: số tuổi

Đầu vào: a

Ràng buộc: a nguyên dương

Đầu ra (mục tiêu): Thông báo người có số tuổi a là trẻ hay không còn trẻ

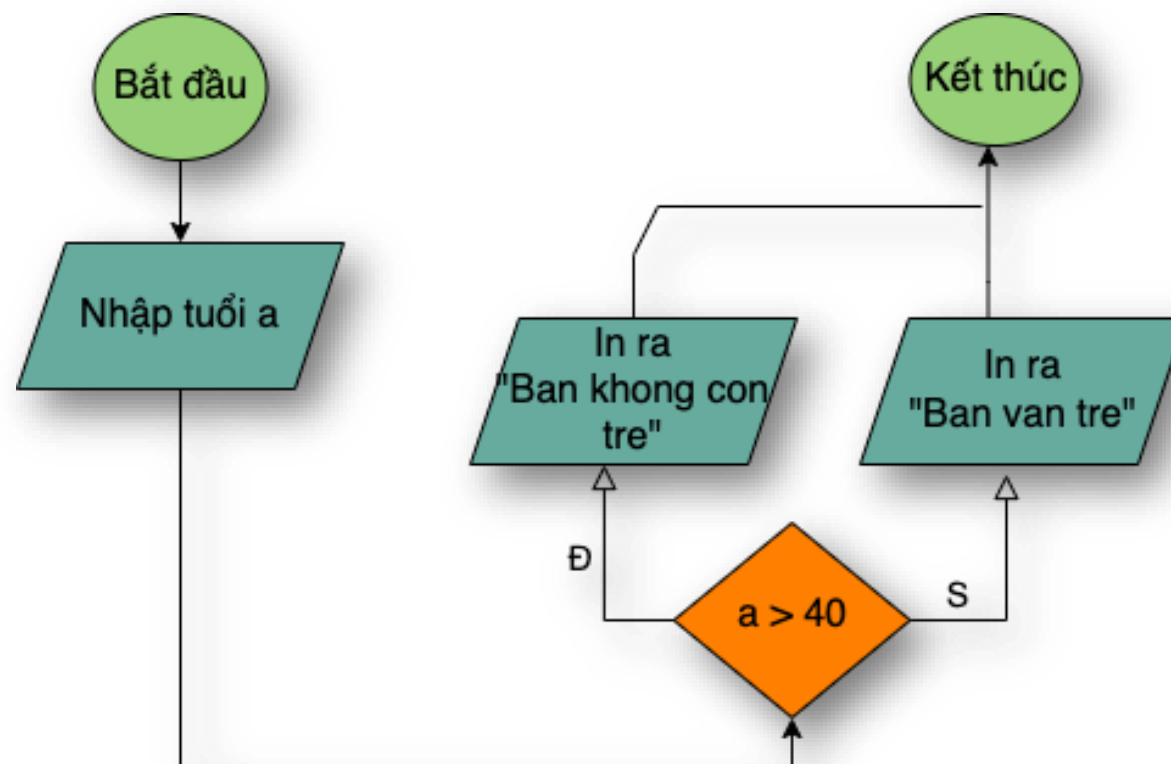


ĐẶT VẤN ĐỀ

Bài toán: Nhập vào tuổi của bạn. Nếu tuổi > 40 thì thông báo "Bạn không còn trẻ", ngược lại thông báo "Bạn vẫn trẻ"

Thuật toán

1. Nhập a
2. Kiểm tra $a > 40$ hay không?
 - a. Nếu có, in ra thông báo "Bạn không còn trẻ"
 - b. Ngược lại, in ra thông báo "Bạn vẫn trẻ"
3. Kết thúc



ĐẶT VẤN ĐỀ

- Kiểm soát đầu vào:**

- Nếu nhập $a \leq 0$ thì yêu cầu nhập lại.
- Chỉ thực hiện bước tiếp theo khi giá trị a nhập vào lớn hơn 0

1. Nhập a

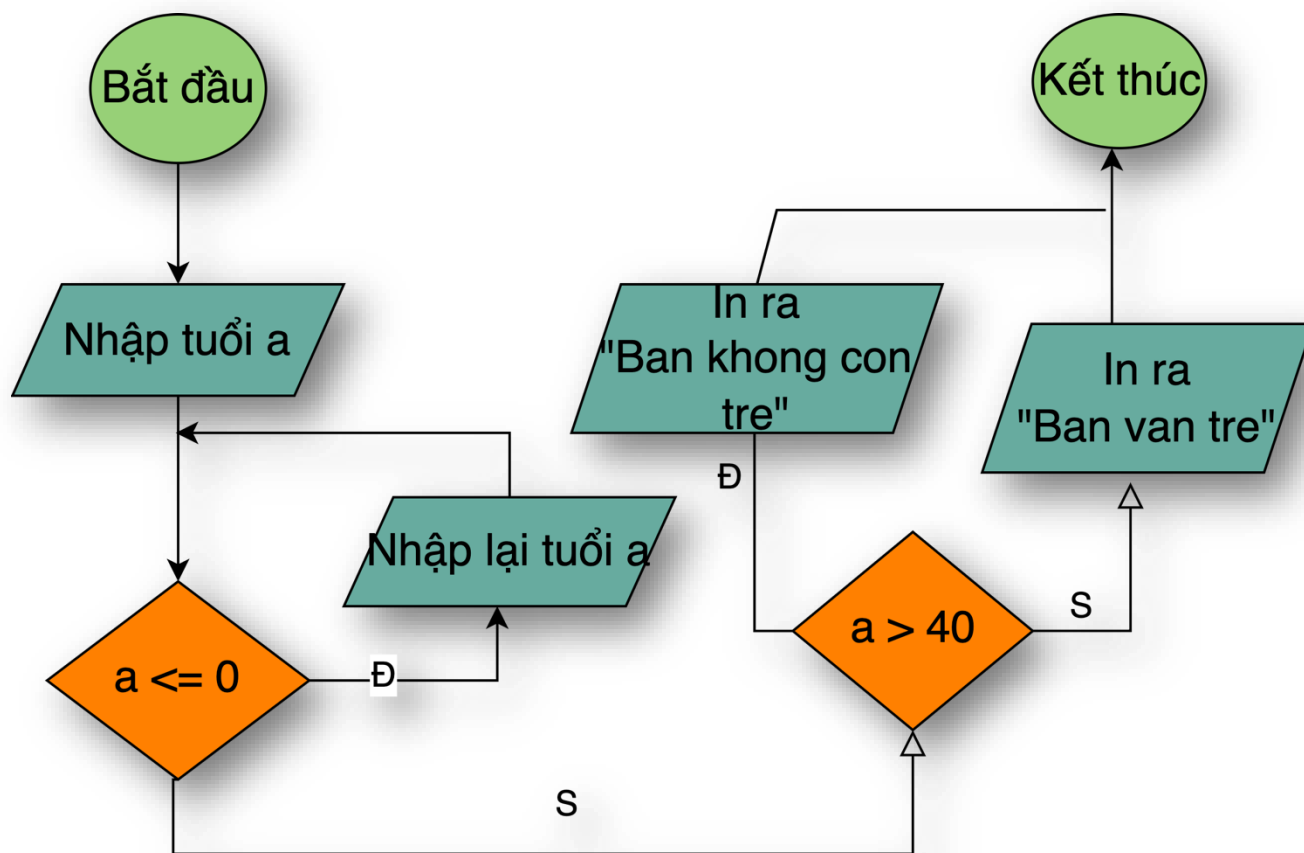
2. Kiểm tra $a \leq 0$ hay không?

- Nếu đúng, thông báo a cần lớn hơn không và quay lại bước 1
- Nếu sai, chuyển đến bước tiếp theo

3. Kiểm tra $a > 40$ hay không?

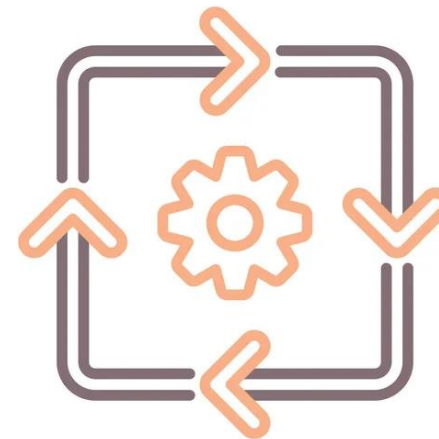
- Nếu có, in ra thông báo “Ban khong con tre”
- Ngược lại, in ra thông báo “Ban van tre”

4. Kết thúc



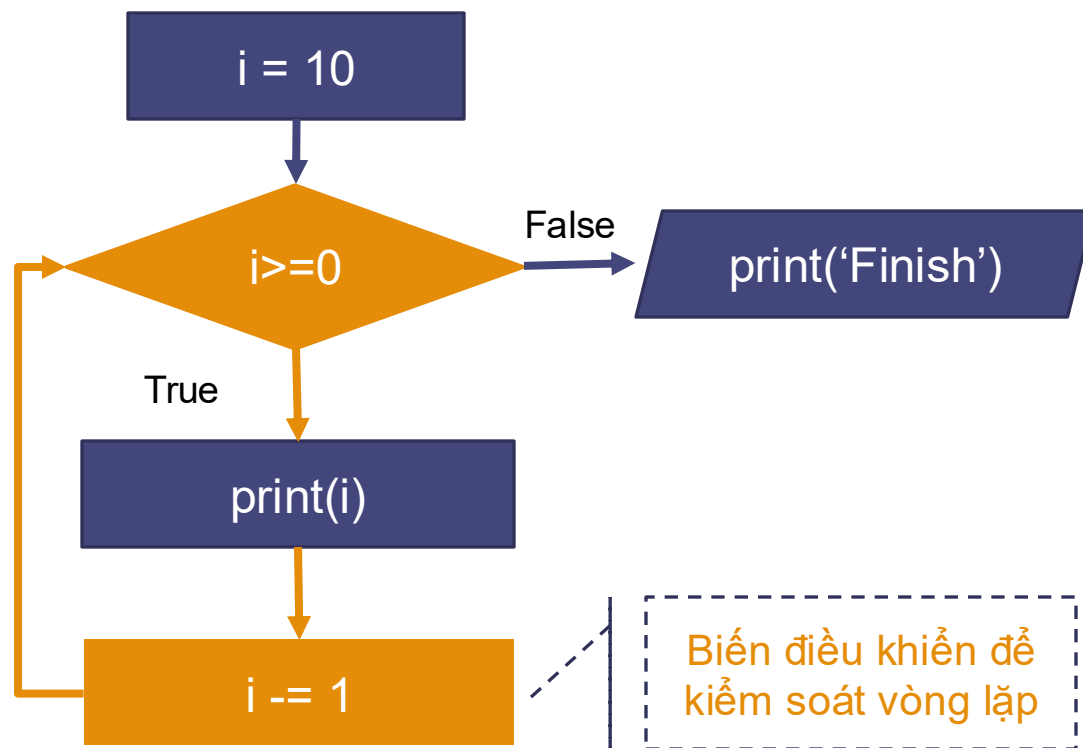
NỘI DUNG

- Cấu trúc lặp **while**
- Cấu trúc lặp **for**
- Cấu trúc lặp lồng nhau
- Lệnh nhảy **break** và **continue**
- Cấu trúc lặp với **else**



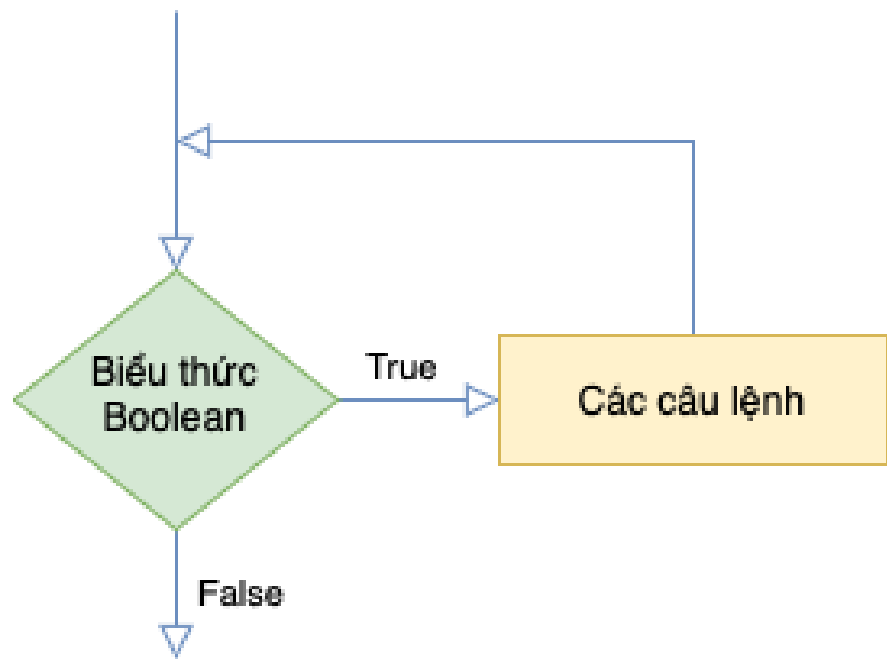
CẤU TRÚC LẶP WHILE

CẤU TRÚC LẶP WHILE



- Cho phép chương trình chỉ thực hiện một đoạn mã lệnh nào đó dưới một điều kiện cụ thể.
- Số lần lặp là không xác định, tùy thuộc vào biểu thức logic.

CẤU TRÚC LẶP WHILE

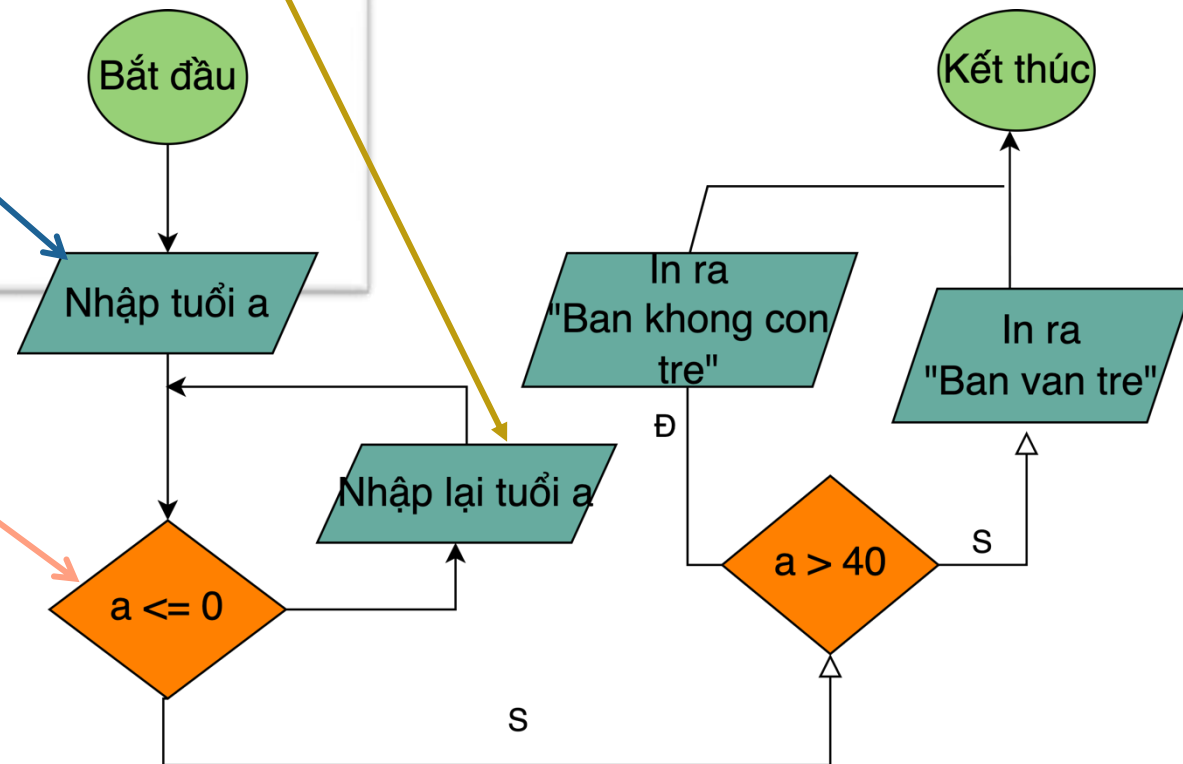


```
while <biểu thức logic>:  
    <các câu lệnh>
```

- Bắt đầu bằng từ khóa **while**, tiếp theo là **<biểu thức logic>** và kết thúc bởi dấu hai chấm ":"
- Các câu lệnh được thực thi và được thực hiện lặp đi lặp lại **trong khi** **<biểu thức logic>** có giá trị **True**.
- **Chú ý:** kiểm soát vòng lặp để tránh bị lặp vô hạn.

VÍ DỤ 4.1. KIỂM TRA TUỔI

```
1 a = int(input("Nhap vao so tuoi cua ban: "))
2 while (a <= 0):
3     a = int(input("Tuoi>0. Nhap lai tuoi: "))
4 if a < 40:
5     print("Ban van tre")
6 else:
7     print("Ban khong con tre")
8
```



VÍ DỤ 4.2. HIỂN THỊ SỐ

Ví dụ 5.1. Nhập vào một số nguyên n và in ra tất cả các số nguyên nhỏ hơn n.

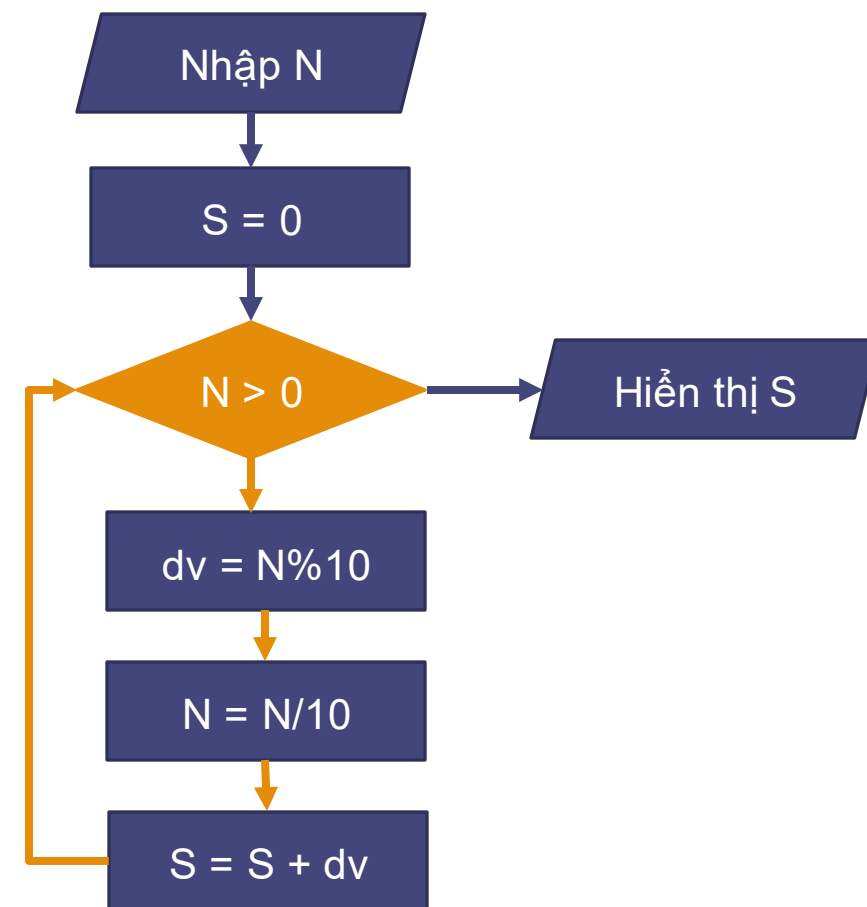
```
n = int(input("Nhập n = "))
i = 0
while i < n:
    print(i)
    i = i + 1
```

BÀI TẬP THỰC HÀNH

- Bài 4.1:** Cho một số nguyên N , tính tổng các chữ số trong biểu diễn thập phân của N ?

Ý tưởng:

- Tách dần** từng chữ số của N , từ chữ số hàng đơn vị trở đi bằng cách:
 - Lấy chữ số hàng đơn vị của N , $dv = ???$
 - Xóa chữ số hàng đơn vị của N , $N = ???$.
 - Cộng chữ số vừa tách được vào tổng tích lũy, $S = ???$
- Hiển thị kết quả S ra màn hình



CẤU TRÚC LẬP FOR

VÍ DỤ 4.3

- Viết chương trình in ra các bảng cửu chương bậc n.

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

n = 7

1 * 8 = 8
2 * 8 = 16
3 * 8 = 24
4 * 8 = 32
5 * 8 = 40
6 * 8 = 48
7 * 8 = 56
8 * 8 = 64
9 * 8 = 72
10 * 8 = 80

n = 8

1 * 9 = 9
2 * 9 = 18
3 * 9 = 27
4 * 9 = 36
5 * 9 = 45
6 * 9 = 54
7 * 9 = 63
8 * 9 = 72
9 * 9 = 81
10 * 9 = 90

n = 9

VÍ DỤ 4.3 – IN BANG CỬU CHƯƠNG

```
n = int(input("n = "))  
for i in [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]:  
    print(f"{i:2} * {n:2} = {i*n:2}")
```

Kết quả:

```
n = 4  
1 * 4 = 4  
2 * 4 = 8  
3 * 4 = 12  
4 * 4 = 16  
5 * 4 = 20  
6 * 4 = 24  
7 * 4 = 28  
8 * 4 = 32  
9 * 4 = 36  
10 * 4 = 40
```

```
n = int(input("n = "))  
i = 1  
while (i<=10):  
    print(f"{i:2} * {n:2} = {i*n:2}")  
    i = i + 1
```

CẤU TRÚC LẶP FOR

```
for <biến lặp> in <danh sách lặp>:  
    <các câu lệnh>
```

Biến lặp Danh sách lặp

```
for i in [5, 4, 3, 2, 1] :  
    print(i)  
print('Het!')
```

Câu lệnh lặp

- Thường được sử dụng để lặp với số lần hữu hạn xác định trước.
- Kiểm soát vòng lặp tốt hơn.
- Bắt đầu với từ khóa **for**
 - **<danh sách lặp>**: tập hợp hữu hạn các giá trị rời rạc có thể đếm được.
 - **<biến lặp>**: biến điều khiển số lần lặp, nhận từng giá trị trong **<danh sách>**
 - **<các câu lệnh>** là tập câu lệnh được thực hiện trong mỗi lần lặp

DANH SÁCH LẬP

```
name = input("Nhap vao ten ban: ")
print("Cac chu cai trong ten ban: ")
for c in name:
    print(f"{c}")
```

```
Nhap vao ten ban: Trung
Cac chu cai trong ten ban:
T
r
u
n
g
```

- Ngoài việc sử dụng hàm `range()` để tạo danh sách, một số cấu trúc dữ liệu đặc biệt của Python: string, list, tuple, ... cũng được sử dụng như là danh sách.

DANH SÁCH LẶP

```
range([<bắt đầu>], <kết thúc> [,<bước tăng>])
```

range() là một trong nhiều cách để tạo ra <danh sách lặp>

- *<bắt đầu>*: giá trị bắt đầu của danh sách. Mặc định, *<bắt đầu>* có giá trị là 0.
- *<kết thúc>*: tất cả các giá trị được sinh ra trong danh sách đều phải nhỏ hơn giá trị *<kết thúc>*.
- *<bước tăng>*: xác định khoảng cách giữa hai giá trị liên kề trong danh sách. Giá trị của *<bước tăng>* có thể là âm hoặc dương nhưng phải khác 0. Mặc định, *<bước tăng>* sẽ có giá trị là 1.

Ví dụ	Giá trị
range(3)	{0, 1, 2}
range(2, 5)	{2, 3, 4}
range(1, 6, 3)	{1, 4}
range(10)	???
range(1, 10, 2)	???

VÍ DỤ 4.3

```
n = int(input("n = "))
i = 1
while (i<=10):
    print(f"{i} * {n} = {i*n}")
    i = i + 1
```

```
n = int(input("n = "))
for i in [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]:
    print(f"{i} * {n} = {i*n}")
```



```
n = int(input("n = "))
for i in range(???):
    print(f"{i} * {n} = {i*n}")
```

BÀI TẬP THỰC HÀNH

Bài 4.2: Hãy viết chương trình tính tổng S theo công thức sau, với n được nhập từ bàn phím:

$$S = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n-1} + \frac{1}{n}$$

Ý tưởng:

- Xác định công thức qui nạp cho S như sau:
 - Với $n = 1$, ta có: $S_1 = 1 = S_0 + 1/1 \Rightarrow S_0 = ???$
 - Với $n = 2$, ta có: $S_2 = 1/1 + 1/2 = S_1 + 1/2$
 - Với $n = i$, ta có: $S_i = 1/1 + 1/2 + \dots + 1/i = S_{i-1} + 1/i$
- Vòng lặp với biến i sẽ được chạy từ 1 đến $n \Rightarrow \text{range}(????)$

CẤU TRÚC LẬP KẾT HỢP RỄ NHÁNH

VÍ DỤ 4.4

Bài toán:

Trong chiến dịch phòng chống dịch nCoV, một bệnh viện muốn phân bổ số bác sĩ và y tá trong bệnh viện thành các nhóm sao cho số lượng bác sĩ và y tá trong các nhóm đều bằng nhau. Hãy xác định số lượng nhóm lớn nhất có thể phân bổ tại bệnh viện đó

Ký hiệu (tham số/biến):

a: số bác sĩ

b: số y tá

n: số nhóm

Đầu vào: a, b

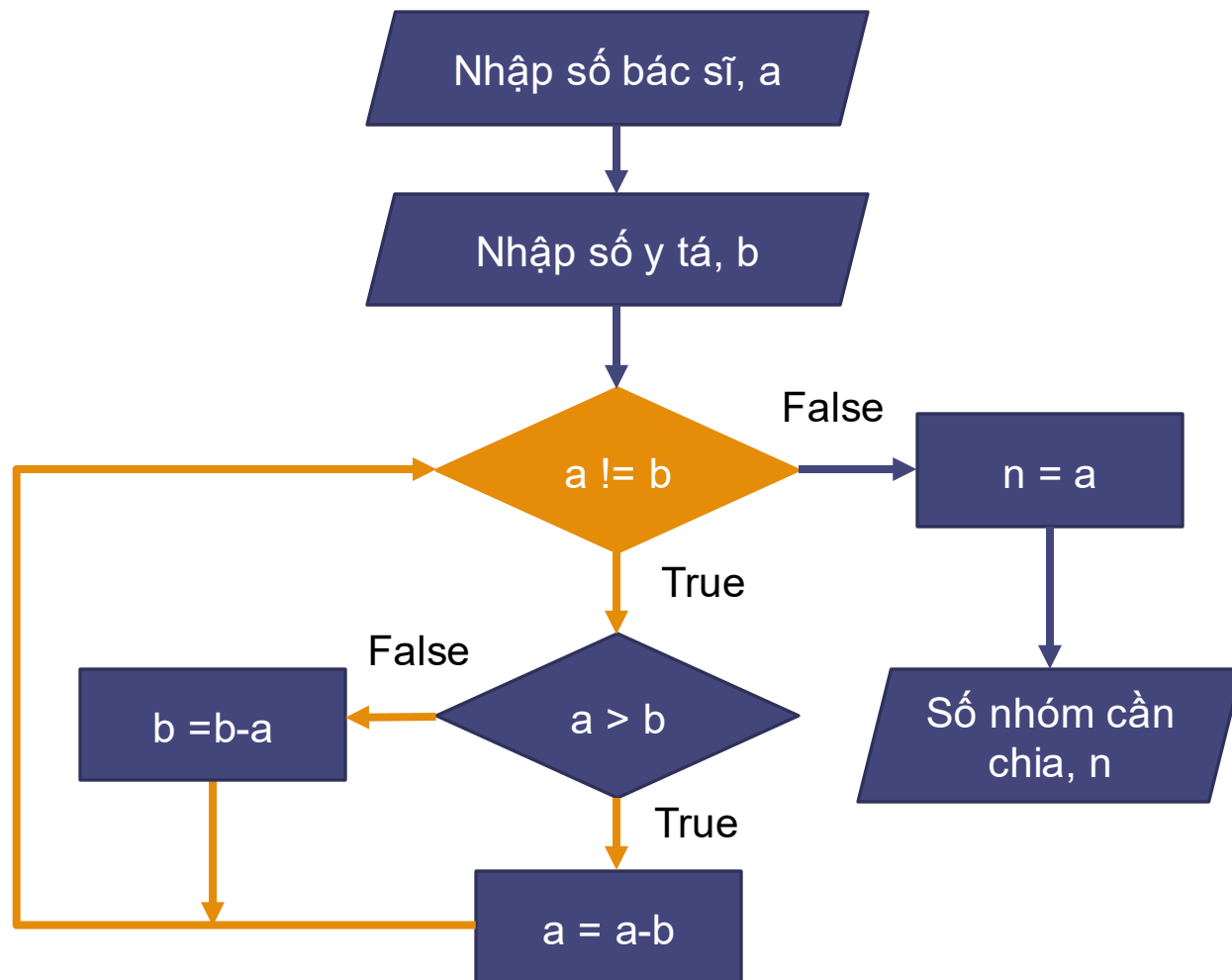
Ràng buộc:

a, b, n: nguyên dương

a, b chia hết cho n

Đầu ra (mục tiêu): giá trị lớn nhất của n

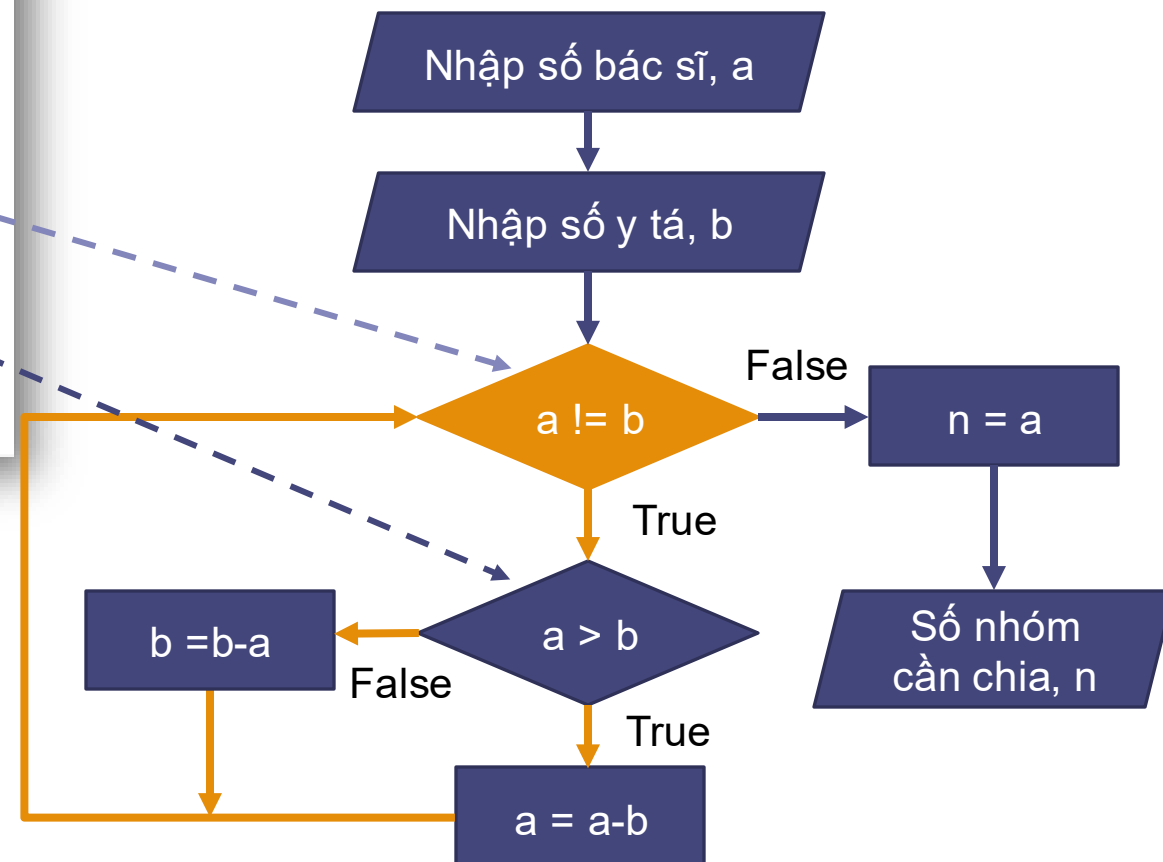
VÍ DỤ 4.4



a	b	Số nhóm
108	64	4
1800	3000	600
2890	2468	2
6835	4540	5

VÍ DỤ 4.4

```
1 a = int(input("Số bác sĩ a = "))
2 b = int(input("Số y tá b = "))
3 while (a != b):
4     if (a > b):
5         a = a-b
6     else:
7         b = b-a
8 n = a
9 print(f"Số nhóm: {n}")
```



CẤU TRÚC LẬP LỒNG NHAU

VÍ DỤ 4.5 – IN BẢNG CỬU CHƯƠNG

- Viết chương trình in ra bảng cửu chương như sau.

	1	2	3	4	5	6	7	8	9	10
1	1	2	3	4	5	6	7	8	9	10
2	2	4	6	8	10	12	14	16	18	20
3	3	6	9	12	15	18	21	24	27	30
4	4	8	12	16	20	24	28	32	36	40
5	5	10	15	20	25	30	35	40	45	50
6	6	12	18	24	30	36	42	48	54	60
7	7	14	21	28	35	42	49	56	63	70
8	8	16	24	32	40	48	56	64	72	80
9	9	18	27	36	45	54	63	72	81	90
10	10	20	30	40	50	60	70	80	90	100

VÍ DỤ 4.5 – IN BẢNG CỬU CHƯƠNG

```
n = int(input("n = "))  
for i in range(1, 11):  
    print(f"{i*n}", end=' ')
```



3	6	9	12	15	18	21	24	27	30
---	---	---	----	----	----	----	----	----	----

```
for n in range(1, 11):  
    for i in range(1, 11):  
        print(f"{i*n:5}", end=' ')  
    print()
```



1	2	3	4	5	6	7	8	9	10
2	4	6	8	10	12	14	16	18	20
3	6	9	12	15	18	21	24	27	30
4	8	12	16	20	24	28	32	36	40
5	10	15	20	25	30	35	40	45	50
6	12	18	24	30	36	42	48	54	60
7	14	21	28	35	42	49	56	63	70
8	16	24	32	40	48	56	64	72	80
9	18	27	36	45	54	63	72	81	90
10	20	30	40	50	60	70	80	90	100

BÀI TẬP THỰC HÀNH

Bài 4.3. Viết chương trình cho phép nhập vào từ bàn phím một số nguyên n và hiển thị ra màn hình hình dạng như hình bên với kích thước tùy theo giá trị nhập vào. Ví dụ ở hình bên với $n = 5$.

```
*  
***  
*****  
*****  
*****  
*****  
*****  
*****  
***  
*
```

LỆNH NHẢY

LỆNH NHẢY

- **Lệnh break;**
 - Khi break được thực hiện trong một vòng lặp, vòng lặp đó sẽ ngưng ngay
 - Bản chất: thoát khỏi vòng lặp, nhảy tới câu lệnh sau vòng lặp
- **Lệnh continue;**
 - Khi continue được thực hiện trong một vòng lặp, toàn bộ lệnh phía sau continue trong thân vòng sẽ bị bỏ qua, nhảy đến vị trí kiểm tra điều kiện lặp
 - Bản chất: nhảy tới vị trí kiểm tra điều kiện lặp

LỆNH NHẢY

```
for i in range(1, 11):  
    print(f"{i}", end=' ')
```



1 2 3 4 5 6 7 8 9 10

```
for i in range(1, 11):  
    if i==5:  
        break  
    print(i, end=' ')
```



???

```
for i in range(1, 11):  
    if i==5:  
        continue  
    print(i, end=' ')
```



???

BÀI TẬP THỰC HÀNH

Bài 4.4. Hiển thị ra màn hình tất cả các số nguyên từ 1 đến n ($0 \leq n < 1000$) mà không chia hết cho 5.

- Chú ý kiểm soát dữ liệu đầu vào. Yêu cầu người dùng nhập lại n cho đến khi thỏa mãn điều kiện: $0 \leq n \leq 1000$
- Ví dụ: 1, 2, 3, 4, 6, 7, 8, 9, 11, 12, 13, 14, 16, 17, 18, ...

CẤU TRÚC LẶP VỚI ELSE

CẤU TRÚC LẶP VỚI ELSE

```
while <biểu thức logic>:  
    <các câu lệnh 1>  
else:  
    <các câu lệnh 2>
```

```
for <biến lặp> in <danh sách>:  
    <các câu lệnh 1>  
else:  
    <các câu lệnh 2>
```

- **while**: <Các câu lệnh 2> được thực hiện khi <biểu thức logic> có giá trị False
- **for**: <Các câu lệnh 2> được thực hiện khi <biến lặp> đã lấy hết giá trị trong <danh sách>
- <các câu lệnh 2> không được thực hiện nếu vòng lặp thoát bởi lệnh **break**

CẤU TRÚC LẶP FOR VỚI ELSE

```
i = 0
while True:
    print(f"Vong lap thu {i}")
    i = i + 2
    if i >= 10:
        break;
else:
    print("Ket thuc vong lap")
```

Kết quả ???

```
i = 0
for i in range(10):
    print(f"Vong lap thu {i}")
else:
    print("Ket thuc vong lap")
```

Kết quả ???

BÀI TẬP THỰC HÀNH

BÀI TẬP THỰC HÀNH

Bài 4.4: Cho một số nguyên dương N ($N \leq 10^8$), số N được gọi là đối xứng nếu viết từ trái sang phải hoặc từ phải sang trái thì giá trị không thay đổi

- Ví dụ:
 - Số 12321, 363, 5665 là các số đối xứng
 - Số 12333, 5661, 366 không phải là số đối xứng.

ST T	N	dv	M
1	5665	#	0
2	566	5	5
3	56	6	56
4	5	6	566
5	0	5	5665

Ý tưởng:

1. Tách dần từng chữ số của N , từ hàng đơn vị trở đi, $dv = N \% 10$
2. Tạo số M dựa trên các số được tách theo chiều ngược lại, $M = M * 10 + dv$
3. Kiểm tra $M == N$ hay không?

BÀI TẬP THỰC HÀNH

Bài 4.5. Viết chương trình cho phép nhập vào số nguyên dương n , hãy tính và đưa ra màn hình các tổng sau đây (với kết quả có 3 chữ số sau dấu phẩy).

$$s1 = \frac{1}{1*2} + \frac{1}{2*3} + \dots + \frac{1}{n*(n+1)}$$

$$s2 = \sqrt{3 + \sqrt{3 + \sqrt{3 + \dots \sqrt{3}}}} \quad (\text{có } n \text{ dấu căn})$$

$$s3 = 1 + \frac{1}{1+2} + \frac{1}{1+2+3} + \dots + \frac{1}{1+2+3+\dots+n}$$

BÀI TẬP THỰC HÀNH

Bài 4.6. Viết chương trình cho phép nhập vào từ bàn phím một số nguyên n và hiển thị ra màn hình hình dạng như hình bên với kích thước tùy theo giá trị nhập vào. Ví dụ ở hình bên với $n = 5$.

```
*  
  
***  
  
*****  
  
*****  
  
*****  
  
*****  
  
***  
  
*
```

BÀI TẬP THỰC HÀNH

Bài 4.7: Hãy tính số pi được tính gần đúng theo công thức Euler sau, với n được nhập từ bàn phím:

$$\frac{\pi^2}{6} = \frac{1}{1^2} + \frac{1}{2^2} + \frac{1}{3^2} + \dots + \frac{1}{(n-1)^2} + \frac{1}{n^2}$$



BÀI TẬP THỰC HÀNH

Bài 4.8. Viết chương trình tìm số sinh viên **tối thiểu** của một lớp nếu biết số học sinh này xếp thành a hàng, b hàng, c hàng hoặc d hàng cũng đều vừa đủ với a, b, c, d là các số nguyên được nhập từ bàn phím.

- Ví dụ: $a = 2$; $b = 3$; $c = 4$; $d = 8$ thì số sinh viên tối thiểu là 24.
- **Ý tưởng:** Tìm BCNN

BÀI THỰC HÀNH

Bài 4.9. Số *nguyên tố* là số chỉ có ước là 1 và chính nó. Viết chương trình cho phép nhập vào số nguyên dương n ($0 \leq n < 10^6$), kiểm tra xem n có phải là số nguyên tố hay không? Ví dụ:

- $n = 11 \Rightarrow n$ là số nguyên tố
- $n = 77 \Rightarrow n$ không phải là số nguyên tố

Bài 4.10. Viết chương trình cho phép nhập vào số nguyên dương n ($0 \leq n \leq 10^4$), liệt kê tất cả các số nguyên tố nhỏ hơn n . Ví dụ:

- $n = 20 \Rightarrow 2, 3, 5, 7, 11, 13, 17, 19$

CÂU HỎI

