# NỘI DUNG ÔN TẬP GIỮA HỌC KỲ II
# MÔN: TIN HỌC 10

## BÀI 4. CÁC KIỂU DỮ LIỆU SỐ VÀ CÂU LỆNH VÀO – RA ĐƠN GIẢN

### 1. Kiểu dữ liệu số nguyên và số thực

- Trong ngôn ngữ lập trình bậc cao có kiểu dữ liệu số nguyên và kiểu dữ liệu số thực
- Câu lệnh **type** (biến) cho kiểu dữ liệu hiện thời của biến.

```
>>> type(20/5)
<class 'float'>
>>> type(20//3)
<class 'int'>
>>> type(20%3)
<class 'int'>
>>> type(5.0%3)
<class 'float'>
```

### 2. Các câu lệnh vào ra đơn giản

***a. Nhập dữ liệu từ bàn phím***

- Câu lệnh nhập dữ liệu của biến là:

+ Biến = **input** (dòng thông báo)
+ Biến = **int** (**input** (dòng thông báo)) → với biến kiểu nguyên
+ Biến = **float** (**input** (dòng thông báo)) → với biến kiểu thực

***b. Xuất dữ liệu ra màn hình***

- Câu lệnh đưa các giá trị biểu thức ra màn hình là:

**print** (danh sách biểu thức)

### 3. Hằng trong Python

- Hằng là những biến có giá trị chỉ định trước và không thể thay đổi trong quá trình thực hiện chương trình.
- Python không cung cấp công cụ khai báo hằng.
- Khi lập trình bằng Python, người ta thường sử dụng hằng số như một loại biến với cách đặt tên đặc biệt.

## BÀI 5. THỰC HÀNH VIẾT CHƯƠNG TRÌNH ĐƠN GIẢN

### 1. Giải phương trình bậc nhất

- Hoàn thiện chương trình:

```
a = float(input("a = "))
b = float (input("b = "))
```

print("Nghiệm của phương trình là ", -b/a)
- Chạy thử với a = 1, b = 2:

```
1   a = float(input("a = "))
2   b = float (input("b = "))
3   print("Nghiệm của phương trình là ", -b/a)
4
```

=> Chương trình vừa hoàn thiện có cho kết quả giống như Hình 1b.
- Chương trình sẽ đưa ra màn hình lỗi nếu giá trị a nhập vào là 0:
*ZeroDivisionError: float division by zero*

**2. An ninh lương thực**

Chương trình:

```
a = float(input("Nhập số kg gạo cần thiết "))
b = int (input("Nhập số người dân của một nước "))
print("Số gạo cần dự trữ là ", b*a)
```

```
>>> a = float(input("Nhập số kg gạo cần thiết "))
... b = int (input("Nhập số người dân của một nước "))
... print("Số gạo cần dự trữ là ", b*a)
```

```
Nhập số kg gạo cần thiết 365
Nhập số người dân của một nước 91086294
Số gạo cần dự trữ là  33246497310.0
```

**3. Tìm ước chung lớn nhất**

```
import math
a = int(input("Nhập a "))
b = int(input("Nhập b "))
print("Ước chung lớn nhất là ", math.gcd(a, b))
```

```
2   import math
3   a = int(input("Nhập a "))
4   b = int(input("Nhập b "))
5   print("Ước chung lớn nhất là ", math.gcd(a, b))
```

Kết quả:

```
Nhập a 6
Nhập b 9
Ước chung lớn nhất là  3
```

**4. Làm quen với ghi chú trong chương trình**

- Chương trình có chú thích:

```
#Giải phương trình bậc hai
import math
```

```
a = 1
b = -5
c = 6
x1 = (-b - math.sqrt(b * b - 4 * a * c) / (2 * a))
x2 = -b / a - x1 #Định lí Viet
print(x1)
print(x2)
```

- Chương trình không có chú thích:

```
import math
a = 1
b = -5
c = 6
x1 = (-b - math.sqrt(b * b - 4 * a * c) / (2 * a))
x2 = -b / a - x1
print(x1)
print(x2)
```

=> Nhận xét: Kết quả của chúng giống nhau.

## BÀI 6. CÂU LỆNH RẼ NHÁNH

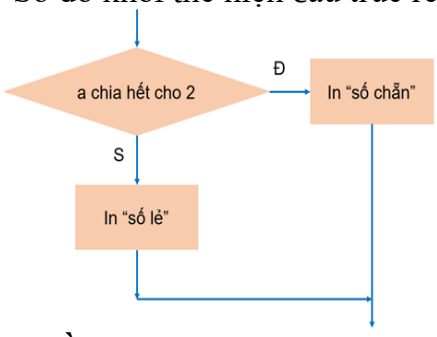
### 1. Cấu trúc rẽ nhánh trong mô tả thuật toán

- Cấu trúc rẽ nhánh:

**Nếu** <*điều kiện*>**:**
*Nhánh đúng*
**Trái lại:**
*Nhánh sai*
**Hết nhánh**

*Hình 1a. Mẫu cấu trúc rẽ nhánh*

- Sơ đồ khối thể hiện cấu trúc rẽ nhánh của Hình 1b:




### 2. Điều kiện rẽ nhánh

- Trong mô tả thuật toán, <*điều kiện*>
rẽ nhánh là một biểu thức nhận giá trị logic True hoặc False.
- Phép so sánh hai giá trị hay so sánh hai biểu thức sẽ cho ta một biểu thức logic.

*Bảng 1. Kí hiệu phép so sánh trong Python*

| So sánh | Kí hiệu trong Python |
|---|---|
| Lớn hơn | > |
| Lớn hơn hoặc bằng | >= |
| Nhỏ hơn | < |
| Nhỏ hơn hoặc bằng | <= |
| Bằng | == |
| Khác | != |

| | | |
|---|---|---|
| and | x and y | Cho kết quả True khi và chỉ khi x và y đều nhận giá trị True |
| or | x or y | Cho kết quả False khi và chỉ khi x và y đều nhận giá trị False |
| not | not x | Đảo giá trị logic của x |

**3. Câu lệnh rẽ nhánh trong chương trình python**

- Python cung cấp hai câu lệnh rẽ nhánh cơ bản:

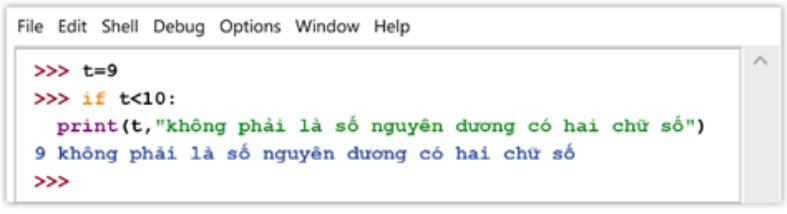
**+ Câu lệnh rẽ nhánh if:**

Cách viết:

*if <điều kiện>:*

*câu lệnh hay nhóm câu lệnh*

Ví dụ:




*Hình 4. Chương trình kiểm tra số nguyên dương có hai chữ số*

**+ Câu lệnh rẽ nhánh if - else:**

Cách viết:

*if <điều kiện>:*

*câu lệnh hay nhóm câu lệnh 1*

*else :*

*câu lệnh hay nhóm câu lệnh 2*

Ví dụ:

*Chương trình*

File Edit Format Run Options Window Help

```
A = int(input("Nhập vào một số nguyên: "))
if A%2 == 0:
    print(A," là số chẵn.")
else:
    print(A," là số lẻ.")
```

*Kết quả thực hiện*

File Edit Shell Debug Options Window Help

```
Nhập vào một số nguyên: 15
15 là số lẻ.
>>>
```

Chú ý:

- Các câu lệnh ở khối trong viết lùi vào các đầu dòng nhiều hơn các câu lệnh khối ngoài.
- Các câu lệnh ở cùng một khối có khoảng cách ở đầu dòng như nhau.

## BÀI 7. THỰC HÀNH CÂU LỆNH RẼ NHÁNH

**1. Lấy ví dụ về câu lệnh if**

| Mô tả | Câu lệnh if |
|---|---|
| Ví dụ 1: Kiểm tra số đã cho có phải số chẵn không? Nếu là số chẵn hiện thông điệp "Đây là số chẵn" | |
| Ví dụ 2: Kiểm tra lớp học có học sinh đi học đủ không, số học sinh của lớp là 40, nếu lớp đi học đủ hiện thông điệp "Lớp đã đi học đủ". | |

**2. Chia kẹo**

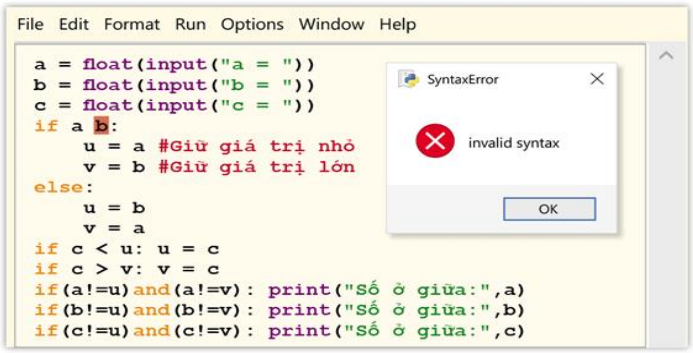
Chương trình:

```
n = int (input("Nhập số kẹo: "))
m = int (input("Nhập số em bé: "))
if n % m == 0:
    print("Có")
else:
    print ("Không")
```

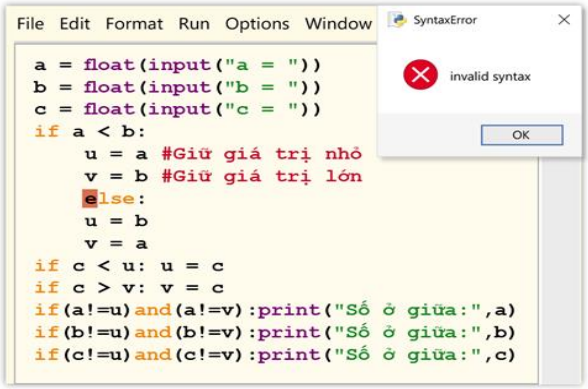
Kết quả thực hiện (chạy chương trình 3 lần, mỗi lần với bộ dữ liệu n, m khác nhau), ví dụ:

```
Nhập số kẹo: 10
Nhập số em bé: 2
Có
>>>
====== RESTART:
Nhập số kẹo: 15
Nhập số em bé: 6
Không
>>>
====== RESTART:
Nhập số kẹo: 14
Nhập số em bé: 7
Có
```

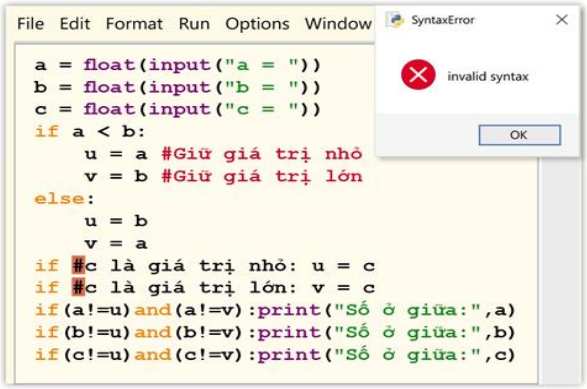
## 3. Tìm lỗi sai




*Hình 1a. Báo lỗi ở chương trình của bạn Bình*




*Hình 1b. Báo lỗi ở chương trình của bạn An*




*Hình 1c. Báo lỗi ở chương trình của bạn Phúc*

- Lỗi sai của Bình xuất hiện ở lệnh ***if a b*** vì sau if là một biểu thức logic.
Sửa lại: ***if a < b***
- Lỗi sai của An xuất hiện ở dòng else, else phải được viết thẳng hàng với if.
Sửa lại: Lùi else ra đầu dòng thẳng hàng với if.
- Lỗi sai của Phúc xuất hiện ở hai dấu # sau if vì sau if là một biểu thức logic.
Sửa lại: Đổi *if # c là giá trị nhỏ: u = c* thành *if c < u: u = c*
*if # c là giá trị lớn: v = c* *if c > v: v = c*

* Chương trình chuẩn:

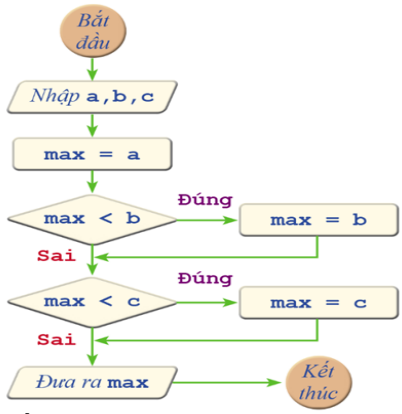
```
a = float (input("a = "))
b = float (input("b = "))
c = float (input("c = "))
if a < b:
    u = a #Giữ giá trị nhỏ
    v = b #Giữ giá trị lớn
else:
    u = b
    v = a
if c < u: u = c
if c > u: v = c
if (a!= u) and (a!= v): print ("số ở giữa: ", a)
if (b!= u) and (b!= v): print ("số ở giữa: ", b)
if (c!= u) and (c!= v): print ("số ở giữa: ", c)
```

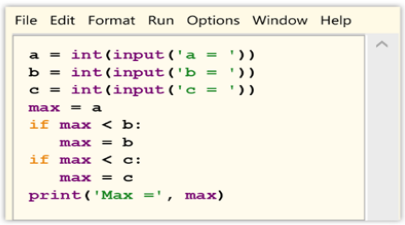
Kết quả:

```
====== RESTART: C:
a = 6
b = 15
c = 105
số ở giữa:  15.0
>>>
```

**4. Tìm số lớn nhất**

Sơ đồ khối và chương trình giải bài 4.




Kết quả:

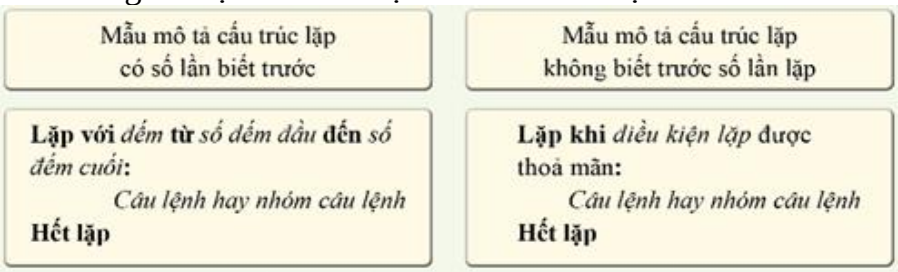
```
====== RESTART:
a = 10
b = 152
c = 124
Max =  152
>>>
```

Nhận xét: Chương trình đúng với sơ đồ khối.

## BÀI 8. CÂU LỆNH LẶP

**1. Cấu trúc lặp trong mô tả thuật toán**

- Khi có một hay nhiều thao tác cần được thực hiện lặp lại một số lần liên tiếp trong quá trình thực hiện thuật toán thì cần dùng cấu trúc lặp.
- Có những thuật toán biết trước được số lần lặp của những thao tác cần lặp lại.
- Có những thuật toán không biết trước được số lần lặp mà chỉ đến khi thực hiện thuật toán với những dữ liệu đầu vào cụ thể mới biết được.




*Hình 1. Mẫu mô tả cấu trúc lặp trong mô tả thuật toán*

- Một số tình huống thực tế:

| **Có số lần lặp biết trước** | **Có số lần lặp không biết trước** |
|---|---|
| - Vận động viên chạy 20 vòng sân.<br>- Em làm 5 bài tập về nhà thầy cô giáo giao. | - Vận động viên chạy nhiều vòng xung quanh sân trong thời gian 20 phút.<br>- Em làm bài tập về nhà đến giờ ăn cơm thì dừng lại. |

**- *Hoạt động 1***:

+ Mô tả thuật toán ứng với Ví dụ 1:

**Lặp với** *đếm* từ 1 đến 10:

*In ra màn hình* "*Xin chào Python*"

**Hết lặp**

+ Mô tả thuật toán ứng với Ví dụ 2:

**Lặp khi** *số nhập vào ≠ mật khẩu*: *Yêu cầu nhập lại mật khẩu*

**Hết lặp**

**2. Câu lệnh lặp với số lần biết trước trong python**

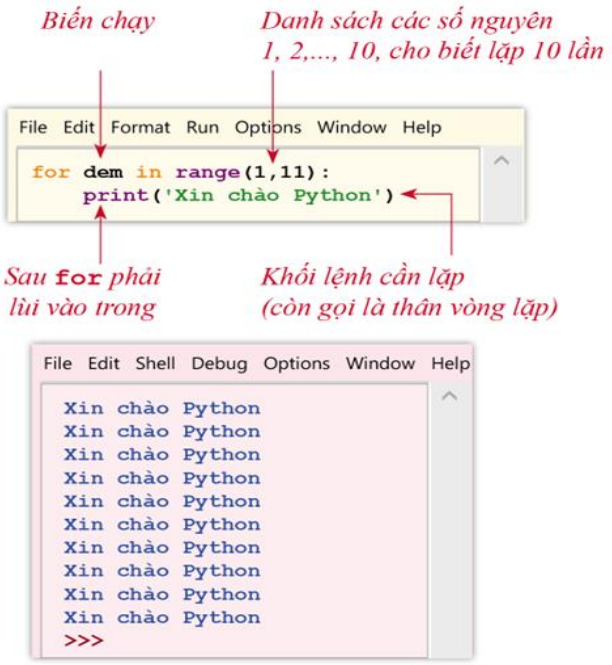
- Dạng câu lệnh:

for <*biến chạy*> in range(m, n):

Khối lệnh cần lặp

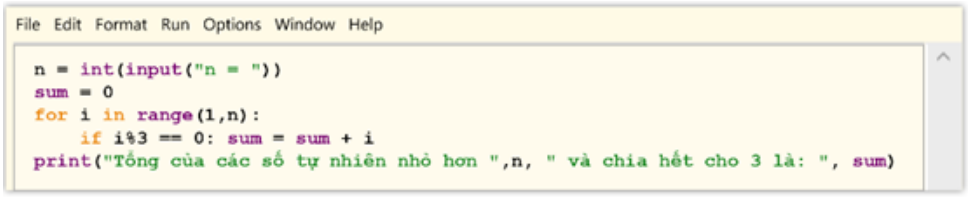
Trong đó:

• Hàm range(m, n) dùng để khởi tạo dãy số nguyên từ m đến n – 1 (với m < n)• Trường hợp m = 0, hàm range(m, n) có thể viết gọn là range(n) Ví dụ 3: minh họa một câu lệnh for trong Python và kết quả thực hiện.




*Hình 3. Ví dụ một câu lệnh* **for**

- Ví dụ 4: Viết chương trình nhập từ bàn phím và tính tổng các số tự nhiên chia hết cho 3 nhỏ hơn n.




*Hình 4. Ví dụ một chương trình sử dụng câu lệnh* **for**

### 3. Câu lệnh lặp với số lần không biết trước trong python

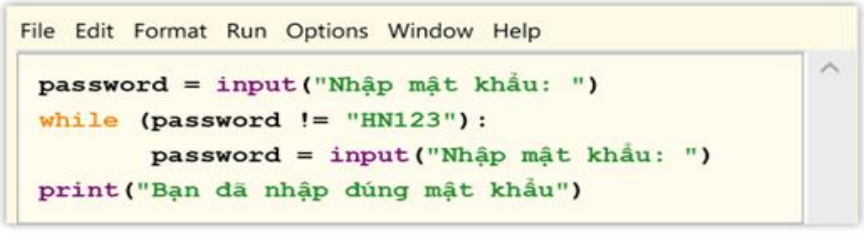
- Dạng câu lệnh:

while <điều kiện>:

Câu lệnh hay nhóm câu lệnh

Trong đó: điều kiện là biểu thức nhận giá trị logic là True hoặc False

- Ví dụ 5: Chương trình nhập mật khẩu




- Ví dụ 6: Chương trình sử dụng câu lệnh while

```
File Edit Format Run Options Window Help
sodem = 1
while (sodem <= 6):
    print (sodem)
    sodem = sodem + 1
```

```
1
2
3
4
5
6
>>>
```

- Hoạt động 2:

```
1 for sodem in range(1, 7):
2     print(sodem)
3
```

* Câu lệnh while cũng có thể thực hiện được cấu trúc lặp với số lần lặp biết trước.

## BÀI 9. THỰC HÀNH CÂU LỆNH LẶP (2 TIẾT)

### 1. Làm quen với câu lệnh lặp trong python

```
File Edit Format Run Options Window Help
total = 0
for i in range (1, 101):
    total = total + i
    print(i, total)
```

Dự đoán: Chương trình sẽ chạy ra 100 dòng, với mỗi dòng sẽ bằng tổng trước đó cộng với số vòng hiện tại. Ví dụ:

1 1
2 3
3 6
4 10
...

Kết quả thực hiện:

### 2. Đếm các ước thực sự của một số nguyên

```
File Edit Format Run Options Window Help
n = int(input("n = "))
i = 2
so_uoc = 0
while i <= n/2
    if (n%i) == 0: so_uoc = so_uoc + 1
    i = i+1
    print(n,"có số ước thực sự là: ", so_uoc)
```

*Hình 2. Chương trình của bạn Hà*

Sửa lại:

```
n = int (input("n = "))
i = 2
so_uoc = 0
while i < n:
    if n % i == 0:
        so_uoc = so_uoc + 1
    i = i + 1
print (n, "có số ước thực sự là", so_uoc)
```

Kết quả:

```
====== RESTART: C:/Users/Us
n = 10
10 có số ước thực sự là 2
>>>
```

**3. Nhập dữ liệu có kiểm tra**

Chương trình như sau:

```
n = 0
while n < 100000000:
      n =  int(input("Nhập số nguyên lớn hơn 100 000 000:"))
print ("Cảm ơn, bạn đã nhập dữ liệu đúng yêu cầu")
```