

Mục tiêu:

Sinh viên thực hành những bài tập đơn giản bằng ngôn ngữ Python, làm việc với biến, thực hiện các lệnh nhập xuất. Sinh viên thực hành thành thạo các thao tác cơ bản trên Google Colab để lập trình các chương trình bằng ngôn ngữ Python. Sinh viên tham khảo các bài tập mẫu để làm các bài tập được yêu cầu.

Bài tập mẫu

Bài tập mẫu, giải và biện luận phương trình bậc nhất $ax+b=0$, với a, b được nhập từ bàn phím

```
a = float(input('nhập a = '))
b = float(input('nhập b = '))

if a == 0:
    if b == 0:
        print('Phương trình vô số nghiệm')
    else:
        print('Phương trình vô nghiệm')
else:
    x = -b/a
    print('Phương trình có nghiệm x = ',x)
```

```
nhập a = 0
nhập b = 1
Phương trình vô nghiệm
```

Trong chương trình này, sử dụng lệnh if/else để thực hiện kiểm tra các tham số của phương trình. Về cơ bản lệnh if / else trong Python hoạt động giống như các ngôn ngữ khác, tuy nhiên cần chú ý đến dấu hai chấm ':', dấu này là bắt buộc phải có. Các lệnh trong if sẽ thực thi dòng, Python không giống hàng theo dấu ngoặc nhọn {}, do vậy việc thực thi dòng để giống hàng là bắt buộc, tương tự với lệnh else.

Module math

Khi làm bài tập cần sử dụng đến các phép toán, các hàm cơ bản trong toán học, sinh viên import module math. Một số hàm trong module math được liệt kê dưới đây.

- `sqrt`
Computes the square root of a number: $\text{sqrt}(x) = \sqrt{x}$
- `exp`
Computes e raised a power: $\text{exp}(x) = e^x$
- `log`
Computes the natural logarithm of a number: $\text{log}(x) = \log_e x = \ln x$
- `log10`

Computes the common logarithm of a number: $\log(x) = \log_{10} x$

- `cos`

Computes the cosine of a value specified in radians: $\cos(x) = \cos x$; other trigonometric functions include sine, tangent, arc cosine, arc sine, arc tangent, hyperbolic cosine, hyperbolic sine, and hyperbolic tangent

- `pow`

Raises one number to a power of another: $\text{pow}(x;y) = x^y$

- `degrees`

Converts a value in radians to degrees: $\text{degrees}(x) = 180 \pi x$

- `radians`

Converts a value in degrees to radians: $\text{radians}(x) = 180 \pi x$

- `fabs`

Computes the absolute value of a number: $\text{fabs}(x) = |x|$

Bài tập mẫu: Nhập vào số nguyên n, in ra các số nguyên tố nhỏ hơn n

```
import math

def isPrime(n):
    if n < 2:
        return False
    for i in range(2, int(math.sqrt(n))+1):
        if n % i == 0:
            return False

    return True

n = int(input('Nhập vào số nguyên dương n = '))
count = 0
print('Các số nguyên tố nhỏ hơn', n, 'là:')
for i in range(2, n):
    if isPrime(i):
        print(i, ', ', end='')
        count += 1 # Không viết được count++
if count == 0:
    print('Không có số nguyên tố nào nhỏ hơn', n)
print()
```

Nhập vào số nguyên dương n = 2
Các số nguyên tố nhỏ hơn 2 là:
Không có số nguyên tố nào nhỏ hơn 2

Bài này sử dụng phép tính căn bậc hai sqrt trong module math Trong bài có sử dụng vòng lặp for, ví dụ for i in range(2,n): Khi đó i sẽ nhận lần lượt giá trị từ 2 cho đến n-1.

Trong Python, để in ra các thông tin về một biến, hàm, module,... ta dùng hàm dir(). In ra tài liệu của các hàm, đối tượng ta gọi đến thuộc tính **`__doc__`**

```
from math import sqrt

def isPrime(n):
    if(n < 2):
        return False
    for i in range(2,int(sqrt(n))+1):
        if n% i == 0:
            return False

    return True

p = isPrime

ip = isPrime(5)

array = [1,2,3,4,5]

print(dir(isPrime))
print(isPrime.__name__)
print(dir(array))
print(array.count.__doc__)
['__annotations__', '__call__', '__class__', '__closure__', '__code__', '__defaults__', '__delattr__', '__dict__',
 '__dir__', '__doc__', '__eq__', '__format__', '__ge__', '__get__', '__getattr__', '__globals__', '__gt__',
 '__hash__', '__init__', '__init_subclass__', '__kwdefaults__', '__le__', '__lt__', '__module__', '__name__',
 '__ne__', '__new__', '__qualname__', '__reduce__', '__reduce_ex__', '__repr__', '__setattr__', '__sizeof__',
 '__str__', '__subclasshook__']
isPrime
['__add__', '__class__', '__contains__', '__delattr__', '__delitem__', '__dir__', '__doc__', '__eq__', '__format__',
 '__ge__', '__getattr__', '__getitem__', '__gt__', '__hash__', '__iadd__', '__imul__', '__init__',
 '__init_subclass__', '__iter__', '__le__', '__len__', '__lt__', '__mul__', '__ne__', '__new__', '__reduce__',
 '__reduce_ex__', '__repr__', '__reversed__', '__rmul__', '__setattr__', '__setitem__', '__sizeof__', '__str__',
 '__subclasshook__', 'append', 'clear', 'copy', 'count', 'extend', 'index', 'insert', 'pop', 'remove', 'reverse', 'sort']
L.count(value) -> integer -- return number of occurrences of value
```

Bài tập mẫu: Tính e^x theo khai triển Taylor với x và n được nhập vào từ bàn phím.

$$e^x = \frac{1}{0!} + \frac{x}{1!} + \frac{x^2}{2!} + \dots + \frac{x^n}{n!}$$

```
import math

def exp(x, n):
    e = 0
    p = 1
    for i in range(1,n+2):
```

```
e = e+p
p = p*x/i
return e
```

```
x = float(input('Nhập vào giá trị x = '))
n = int(input('Nhập vào giá trị n = '))
print('e^',x,' = ',exp(x, n),sep = '')
```

```
print('Hàm tính e mũ x trong module math e^',x,' = ',math.exp(x), sep="");
```

Nhập vào giá trị x = 5
Nhập vào giá trị n = 100
e^5.0 = 148.41315910257654
Hàm tính e mũ x trong module math e^5.0 = 148.4131591025766

Nhấp đúp (hoặc nhấn Enter) để chỉnh sửa

Bài tập mẫu: Tính tổng các số chẵn nhỏ hơn n, với n nhập từ bàn phím

```
sum = 0
n = int(input('Nhập số nguyên dương n = '))

count = 0

while count < n:
    sum += count
    count +=2

print('Tổng các số chẵn nhỏ hơn',n,'là: ',sum)
```

Nhập số nguyên dương n = 10
Tổng các số chẵn nhỏ hơn 10 là: 20

Trong bài sử dụng vòng lặp while để thực hiện vòng lặp.

Bài tập thực hành

1. Giải và biện luận phương trình bậc hai $ax^2+bx+c = 0$, với a, b, c được nhập từ bàn phím
2. Nhập ba số a, b, c tương ứng là độ dài 3 cạnh của một tam giác. Kiểm tra xem 3 cạnh có lập thành tam giác hay không, nếu có tính chu vi và diện tích tam giác.
3. Nhập vào tọa độ (x, y) của một điểm trên mặt phẳng, in ra màn hình vị trí của điểm đó thuộc góc phần tư thứ mấy, hay thuộc trục tọa độ nào, hay là gốc tọa độ.
4. Nhập vào số nguyên n, in ra các số chính phương nhỏ hơn hoặc bằng n.

Số chính phương là các số có căn bậc 2 là 1 số nguyên, ví dụ số 9, số 25 là các số chính phương

5. Nhập vào số nguyên dương n , tính và in ra màn hình giai thừa kép $n!!$ của n .

$n!! = 2 \times 4 \times 6 \times \dots \times n$ với n là số chẵn

$n!! = 1 \times 3 \times 5 \times \dots \times n$ với n là số lẻ

6. Viết chương trình tính $\sin(x)$ theo khai triển Taylor, với x và n được nhập từ bàn phím.

$$\sin(x) = x/1! - x^3/3! + x^5/5! - \dots + (-1)^n x^{(2n+1)}/((2n+1)!)$$

7. Viết chương trình tính $\cos(x)$ theo khai triển Taylor, với x và sai số e được nhập từ bàn phím.

$$\cos(x) = 1/1! - x^2/2! + x^4/4! - \dots + (-1)^n x^{2n}/((2n)!)$$

Sai số e được sử dụng để xác định số hạng cuối cùng được tính trong dãy, kết quả sẽ tính đến số hạng thứ n thỏa mãn

$$p_n = |(-1)^n x^{2n}/((2n)!)| \leq e$$

Trên đây là một cách tính sai số, có một cách tính sai số khác là tính độ sai khác giữa hai số hạng liên tiếp $\text{diff} = |p_n - p_{n-1}| \leq e$

8. Nhập số nguyên dương n , in ra màn hình số hệ nhị phân của n .
9. Nhập số nguyên n , liệt kê các số nguyên a, b, c trong đoạn $[1, n]$ để bộ 3 (a, b, c) lập thành bộ Pitago.
10. Nhập các số nguyên dương a, b, c, d, e . Tìm các nghiệm nguyên dương của hệ phương trình

○ $x + y + z = e$

○ $ax + by + cz = d$

Áp dụng phương pháp giải hệ trên để giải bài toán vừa gà vừa chó 36 con, 100 chân; và bài 100 trâu, 100 cò.