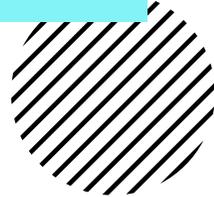
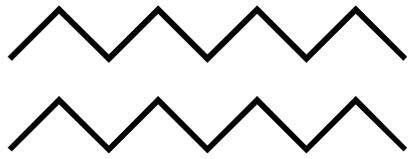


BÀI 26

HÀM TRONG

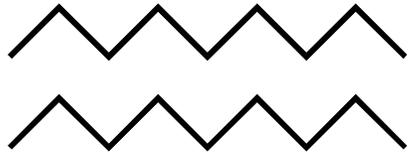
PYTHON





Các chương trình giải những bài toán thực tế phức tạp thường có rất nhiều dòng lệnh, trong đó có không ít những khối lệnh tương ứng với một số thao tác lặp đi lặp lại nhiều lần ở những vị trí khác nhau. Để đỡ công viết đi viết lại các khối lệnh đó, trong tổ chức chương trình viết bằng ngôn ngữ lập trình bậc cao, người ta thường gom các khối lệnh như vậy thành những chương trình con. Khi đó, trong chương trình người ta chỉ cần thay cả khối lệnh bằng một lệnh gọi chương trình con tương ứng. Trong Python, các hàm chính là các chương trình con.

Em có thể kể tên một số hàm trong số các lệnh đã học hay không? Các hàm đó có những đặc điểm chung gì?



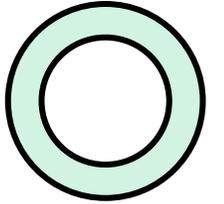
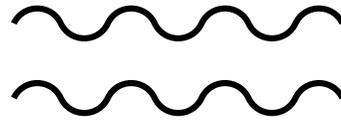
Quan sát một số câu lệnh trong bảng 26.1 và cho biết những câu lệnh này có điểm chung gì?

Bảng 26.1. Một số lệnh trong Python



abs()	len()	range()	bool()
list()	round()	chr()	input()
str()	divmod()	int()	print()
float()	ord()	type()	

1. Một số hàm thiết kế sẵn của Python



Ví dụ: Ta có các lệnh như sau

lệnh in chuỗi kí tự "Thời khóa biểu" trong dấu ngoặc ra màn hình

```
print("Thời khóa biểu")
```

```
x = int("52") # chuyển chuỗi "52" thành số nguyên 52
```

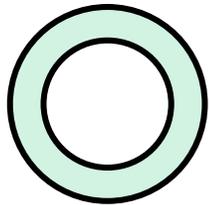
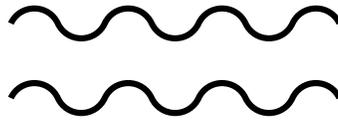
```
type(y) # trả lại kiểu dữ liệu của biến y
```

```
x = input( ) # nhập một chuỗi bất kì từ bàn phím
```

=> Các lệnh trong Bảng 26.1 chính là các chương trình con được thiết kế sẵn của Python, cho phép người dùng tùy ý sử dụng trong các chương trình của riêng mình.



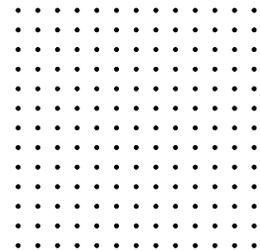
1. Một số hàm thiết kế sẵn của Python

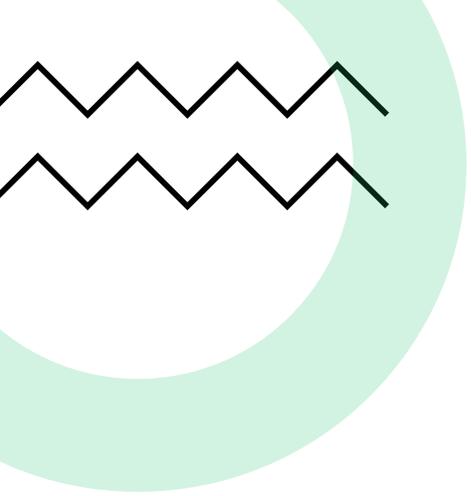


⇒ chuỗi kí tự bên trong ngoặc của các hàm `int ()` và `print()` là tham số của hàm.

⇒ Cú pháp câu lệnh gọi hàm trong Python có dạng chung như sau:

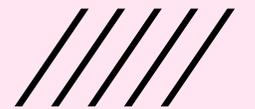
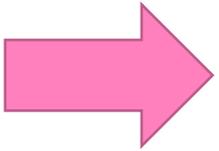
<tên hàm>(**<danh sách tham số hàm>**)

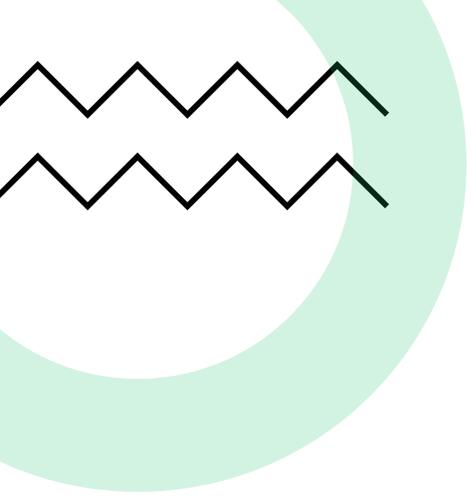




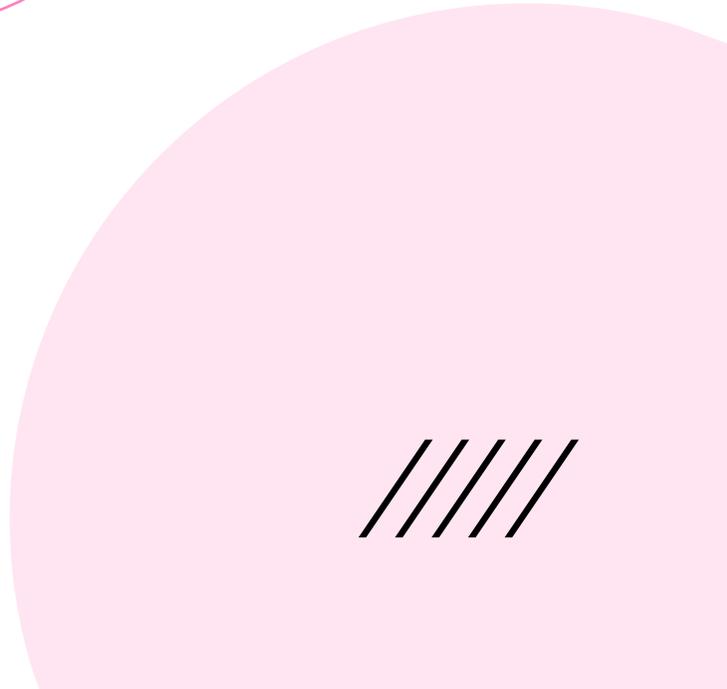
Ghi nhớ

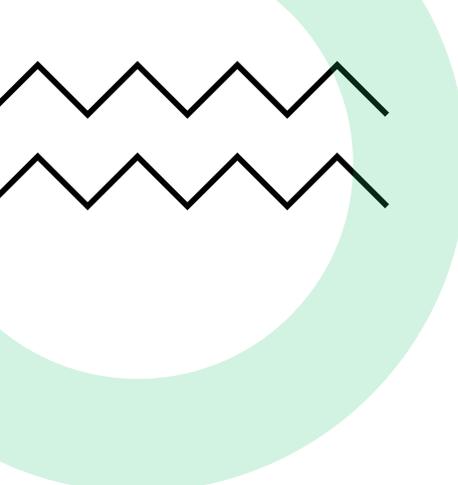
- Python cung cấp sẵn nhiều hàm thực hiện những công việc khác nhau cho phép người dùng được tùy ý sử dụng khi viết chương trình bằng các câu lệnh gọi hàm tương ứng.





? Mô tả tham số và giá trị
trả lại của mỗi hàm sau:
float(), str(), len(), list()





2. THIẾT LẬP CÁC HÀM TỰ ĐỊNH NGHĨA

Ví dụ 1. Cách viết hàm có trả lại giá trị

```
>>> def inc(n):
```

```
    return n+1
```

```
>>> inc(3)
```

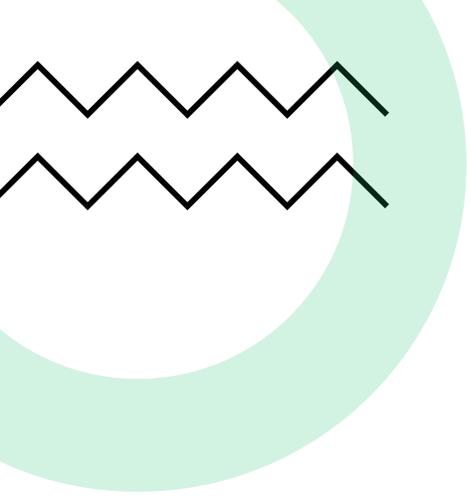
4

Tên hàm: inc

Tham số hàm: số n

Giá trị trả lại: số n + 1





Ví dụ 2. Cách viết hàm không trả lại giá trị.

```
>>> def thong_bao(msg):  
    print("Xin chào bạn", msg)  
    return
```

```
>>> thong_bao("Trần Quang Minh")
```

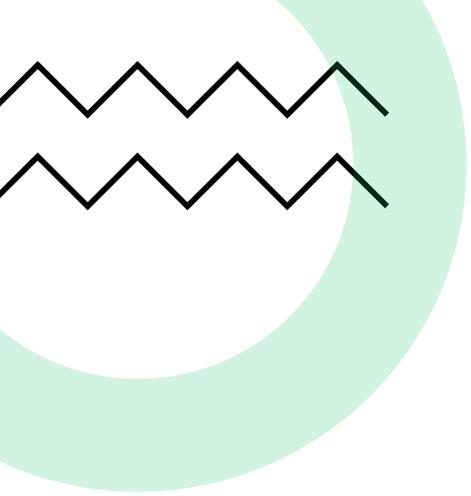
Xin chào bạn Trần Quang Minh

Tên hàm: thong_bao

Tham số hàm: xâu kí tự msg

Giá trị trả lại: không có

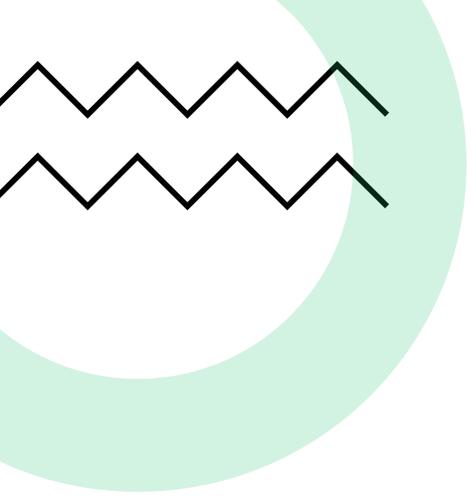




Ghi nhớ

Hàm trong Python được định nghĩa bằng từ khóa **def**, theo sau là tên hàm (tên hàm sẽ theo quy tắc đặt tên định danh). Hàm có thể có hoặc không có tham số. Khối lệnh mô tả hàm được viết sau dấu “:” và viết lùi vào, thẳng hàng. Hàm có thể có hoặc không có giá trị trả lại sau từ khóa **return**.





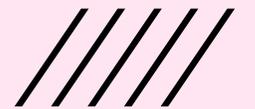
- Cú pháp thiết lập hàm có trả lại giá trị

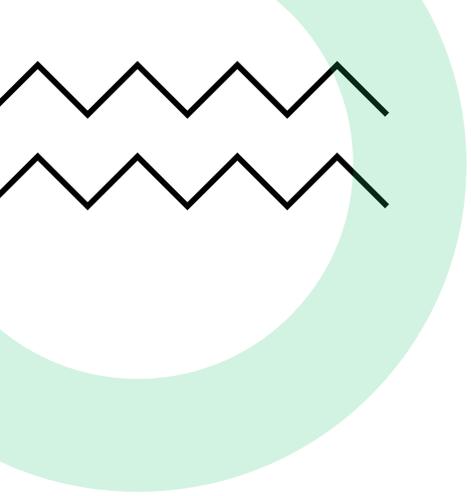
`def` <tên hàm> (<tham số>):

 <khối lệnh>

`return` <giá trị>

Cần có lệnh `return` <giá trị>. Hàm sẽ kết thúc khi gặp lệnh `return` và trả lại <giá trị>





- Cú pháp thiết lập hàm không trả lại giá trị

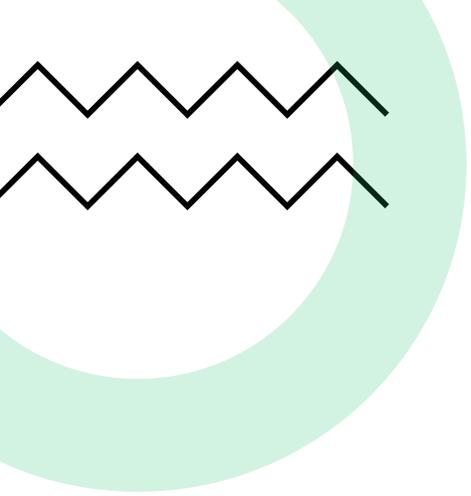
`def` <tên hàm> (<tham số>):

<khối lệnh>

`return`

Lệnh `return` không có giá trị trả lại. Hàm số kết thúc khi gặp lệnh `return`. Nếu hàm không trả lại giá trị thì có thể không cần lệnh `return`

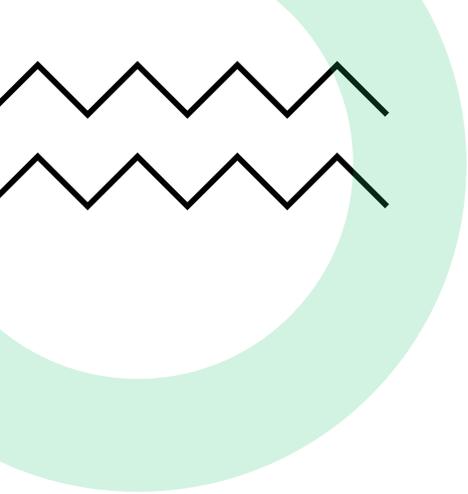




Ghi nhớ:

Để thiết lập hàm trả lại giá trị, câu lệnh **return** trong khai báo hàm cần có <giá trị> đi kèm. Để thiết lập hàm không trả lại giá trị có thể dùng **return** không có <giá trị> hoặc không cần có **return**





Quan sát các hàm sau, giải thích cách thiết lập và chức năng của mỗi hàm



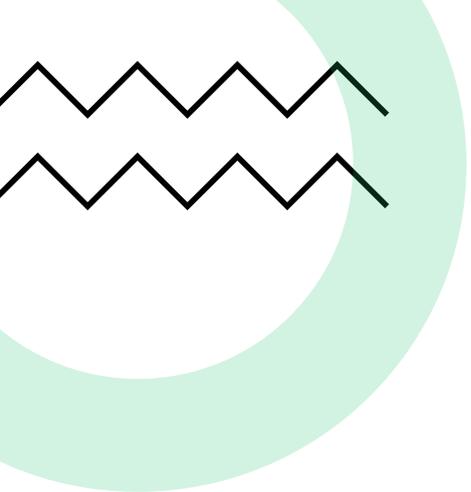
a)

```
def Nhap_xau( ) :  
    msg = input(“Nhập một xâu: “)  
    return msg
```

b)

```
def Inday(n) :  
    for k in range(n) :  
        print(k, end = “ “)
```





THỰC HÀNH

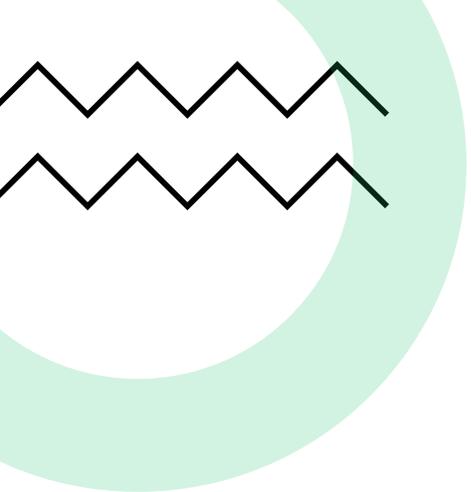
Thiết lập hàm trong Python

Nhiệm vụ 1. Viết hàm yêu cầu người dùng nhập họ tên rồi đưa lời chào ra màn hình

Hướng dẫn: Chương trình có thể như sau:

```
def meeting():  
    ten = input("Nhập họ tên của em:")  
    print("Xin chào", ten)  
meeting()
```





Nhiệm vụ 2. Viết hàm **prime** (n) với tham số tự nhiên n và trả lại True nếu n là số nguyên tố, trả lại False nếu n không phải số nguyên tố

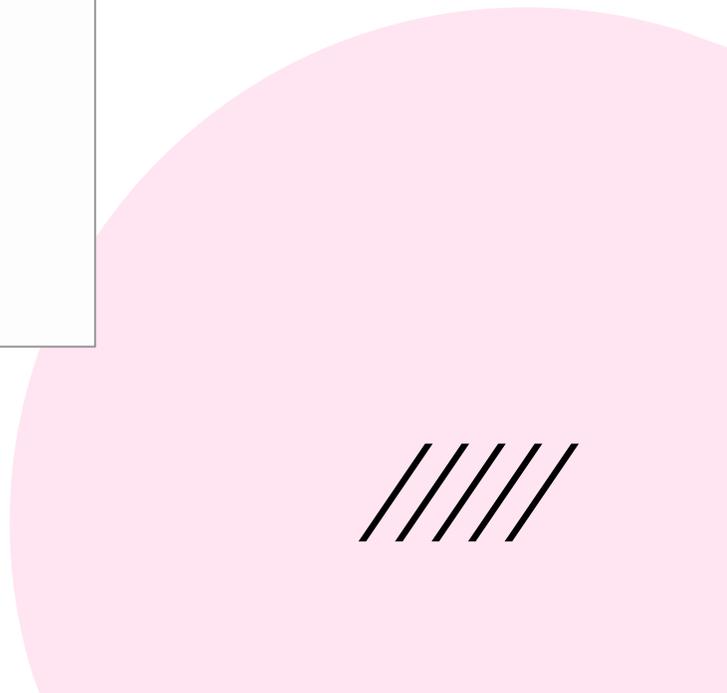
Hướng dẫn: Số nguyên tố là số tự nhiên lớn hơn 1, không có ước nào ngoài 1 và chính nó. Để thiết lập hàm **prime** (n) chúng ta cần tính số ước thật sự của n (từ 1 đến n-1). Biến C dùng để đếm số các ước thật sự của n. Khi đó, n sẽ là số nguyên tố khi và chỉ khi $C = 1$

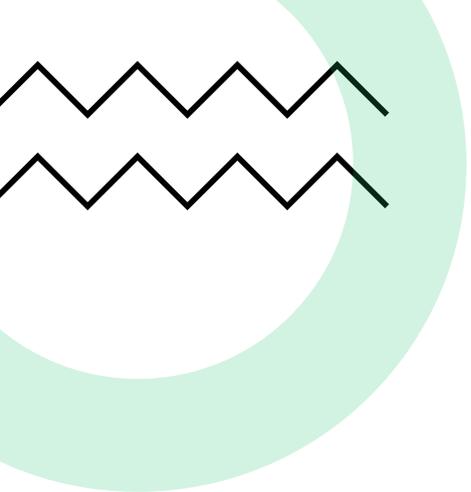
Hàm **prime** (n) và chương trình có thể được thiết lập của như sau:





```
HoangThithanhTam_TL.py * ×  
1  def prime (n):  
2      C = 0  
3      k = 1  
4      while k < n:  
5          if n%k == 0: C = C + 1  
6          k = k + 1  
7      if C == 1: return True  
8      else: return False  
9
```





LUYỆN TẬP

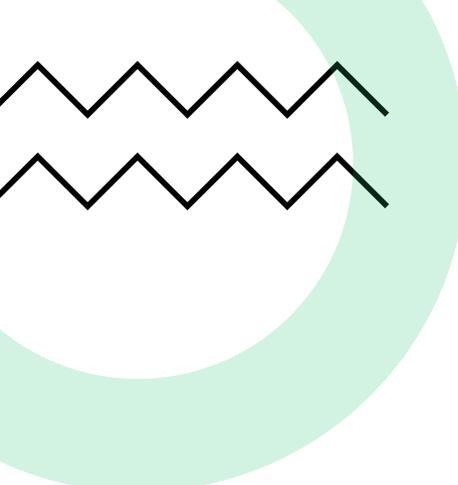
1. Viết hàm với tham số là số tự nhiên n in ra các số là ước nguyên tố của n

Gợi ý: sử dụng hàm `prime()` trong phần thực hành.

2. Viết hàm `numbers(s)` đếm số các chữ số có trong chuỗi s

Ví dụ `numbers("0101abc") = 4`.

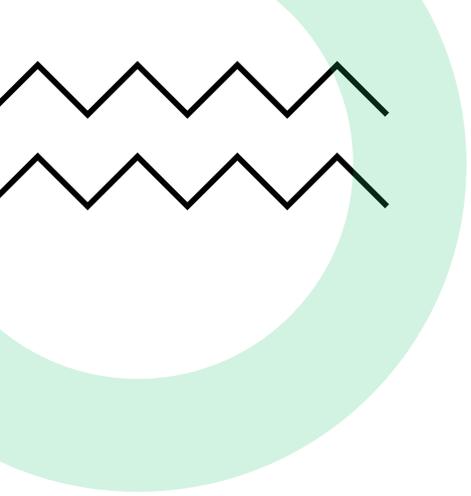




3. Trong khi viết hàm có thể có nhiều lệnh return. Quan sát hàm sau và giải thích ý nghĩa của những lệnh return. Hàm này có điểm gì khác so với hàm `prime ()` đã được mô tả trong phần thực hành.

```
HoangThithanhTam_TL.py * x
1  def prime(n):
2      if n < 2: return False
3      C = 0
4      k = 2
5      while k < n:
6          if n%k == 0: return False
7          k = k + 1
8      return True
```





4. Viết chương trình yêu cầu nhập từ bàn phím một xâu kí tự, sau đó thông báo:

- Tổng số các kí tự là chữ số của xâu
- Tổng số các kí tự là chữ cái tiếng Anh trong xâu

Viết hàm cho mỗi yêu cầu trên.

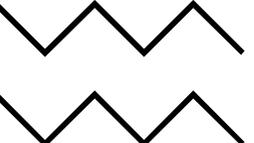


BÀI TẬP

Bài 1. Với hàm BSCNN được xây dựng ở chương trình sau đây (Hình 8), trong những dòng lệnh có sử dụng hàm BSCNN, dòng lệnh nào đúng, dòng lệnh nào sai và tại sao?

```
example.py - C:\Users\TAM\AppData\Local\Programs\Python\Python39\exampl.
File Edit Format Run Options Window Help
from math import gcd
def BSCNN(x, y):
    return x*y//gcd(x, y)
a = int(input("a = "))
b = int(input("b = "))
print("Bội số chung nhỏ nhất: ", BSCNN(a, b))
c = a + b + BSCNN()
print("c = ", c)
```

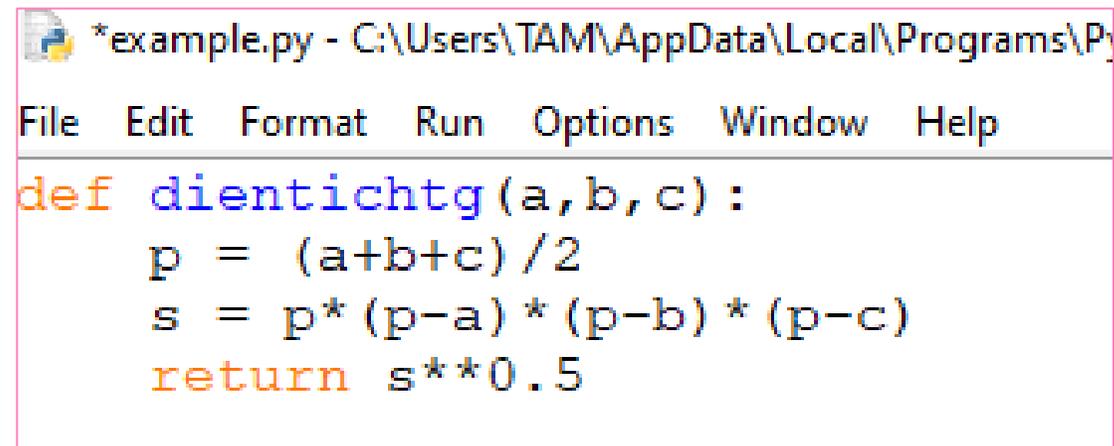
Hình 8. Chương trình về hàm bội số chung nhỏ nhất



BÀI TẬP



Bài 2. Chương trình ở (Hình 9), xây dựng một hàm tính diện tích một tam giác bằng công thức Heron theo ba cạnh của tam giác. Em hãy hoàn thiện chương trình bằng lời gọi hàm thích hợp để đưa ra màn hình kết quả diện tích của tam giác có ba cạnh là 3, 4, 5



```
*example.py - C:\Users\TAM\AppData\Local\Programs\Py
File Edit Format Run Options Window Help
def dientichtg(a, b, c):
    p = (a+b+c)/2
    s = p*(p-a)*(p-b)*(p-c)
    return s**0.5
```

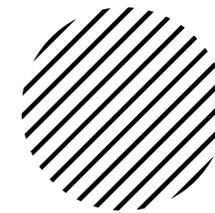
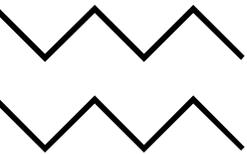
Hình 9. Chương trình tính diện tích tam giác

BÀI TẬP

Bài 3. Sử dụng kết quả của Bài 2 phần *Luyện tập*, em hãy viết chương trình giải bài toán ở *Hoạt động 1*

```
example.py - C:\Users\TAM\AppData\Local\Programs\Python\Python39\example.py (3.9.6)
File Edit Format Run Options Window Help
def dientich(a,b,c):
    p = (a+b+c)/2
    s = p*(p-a)*(p-b)*(p-c)
    return s**0.5
a = float(input("a = "))
b = float(input("b = "))
c = float(input("c = "))
u = float(input("u = "))
v = float(input("v = "))
w = float(input("w = "))
p = float(input("p = "))
q = float(input("q = "))
r = float(input("r = "))
s1 = dientich(a,b,c)
s2 = dientich(u,v,w)
s3 = dientich(p,q,r)
print("Diện tích của 3 tam giác lần lượt là:",s1,s2,s3)
print("Diện tích lớn nhất là:",max(s1,s2,s3))
```

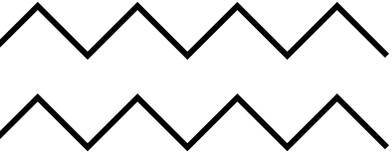
BÀI TẬP



Bài 4. Trong các câu sau đây, những câu nào đúng?

- 1) Sử dụng chương trình con sẽ làm chương trình dễ hiểu, dễ tìm lỗi hơn
- 2) Hàm chỉ được gọi một lần duy nhất ở chương trình chính
- 3) Hàm luôn trả một giá trị qua tên của hàm
- 4) Python chỉ cho phép chương trình gọi một hàm xây dựng sẵn trong các thư viện của Python
- 5) Khai báo hàm trong Python luôn có danh sách tham số





**THANK
YOU**

PRESENTER NAME
EMAIL
WEBSITE

TOPIC ONE

SUBTITLE

