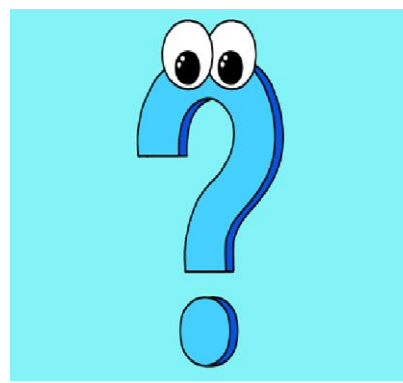


BÀI 27

THAM SỐ CỦA HÀM





Quan sát các lệnh sau và cho biết sự khác nhau giữa tham số (parameter) và đối số (argument).

Tham số

```
>>> def f_sum(a,b,c):  
    return a+b+c
```

Đối số

```
>>> x,y =5,3  
>>> f_sum(10,x,y):  
18
```

1. THAM SỐ VÀ ĐỔI SỐ CỦA HÀM

Ví dụ. Cách truyền dữ liệu qua tham số

```
1 >>> def f(a,b,c):          # Hàm f() có 3 tham số a, b, c
```

```
2         return a+b+c
```

```
3 >>> f(1,2,3)              # Hàm f() được gọi với ba giá trị cụ thể
```

```
4 6
```

```
5 >>> x,y,z = 10,20,5
```

```
6 >>> f(x,y,z)    # Hàm f() được gọi với ba biến đã có giá trị
```

```
7 35
```

```
8 >>> f(a,b,c)    # bị lỗi vì các tham số được truyền vào chưa có giá trị
```

```
9 Traceback (most recent call last):
```

```
10   File "<pyshell#6>", line 1, in <module>
```

```
11     f(a,b,c)
```

```
12 NameError: name 'a' is not defined
```

```
13 >>>
```




Ghi nhớ:

Tham số của hàm được định nghĩa khi khai báo hàm và được dùng như biến trong định nghĩa hàm.

Đối số là giá trị được truyền vào khi gọi hàm. Khi gọi hàm, các **tham số (parameter)** sẽ được truyền bằng giá trị thông qua **đối số (argument)** của hàm, số lượng giá trị được truyền vào hàm bằng với số tham số trong khai báo của hàm.

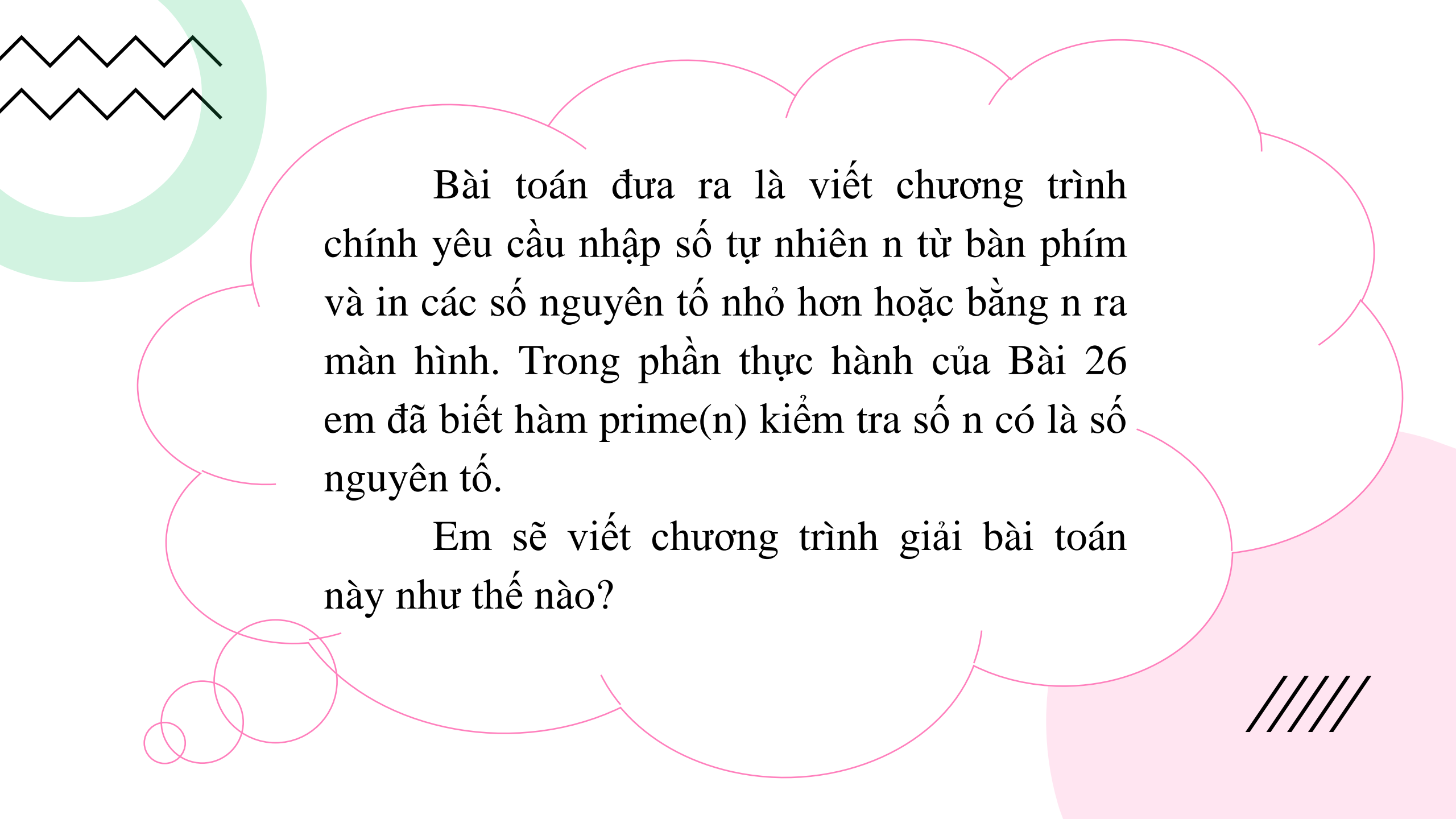




1. Một hàm khi khai báo có một tham số, nhưng khi gọi hàm có thể có hai đối số được không ?

2. Giả sử hàm f có hai tham số x, y khi khai báo, hàm sẽ trả lại giá trị $x + 2y$.
Lời gọi hàm $f(10, a)$ có lỗi hay không?





Bài toán đưa ra là viết chương trình chính yêu cầu nhập số tự nhiên n từ bàn phím và in các số nguyên tố nhỏ hơn hoặc bằng n ra màn hình. Trong phần thực hành của Bài 26 em đã biết hàm $\text{prime}(n)$ kiểm tra số n có là số nguyên tố.

Em sẽ viết chương trình giải bài toán này như thế nào?



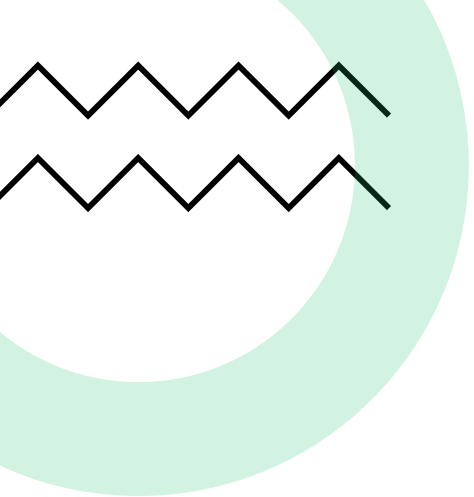


2. CÁCH SỬ DỤNG CHƯƠNG TRÌNH CON

Ví dụ 1. Việc kiểm tra một số có là số nguyên tố được lặp đi lặp lại từ 1 đến n và do đó nên sử dụng hàm `prime(n)` để kiểm tra sẽ giúp chương trình cấu trúc rõ ràng và dễ hiểu hơn.

Chương trình hoàn chỉnh giải bài toán trên có thể được viết 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  # Chương trình chính
10 n = int(input("Nhập số tự nhiên n: "))
11 for k in range(1, n+1) :
12     if prime(k):
13         print(k, end = " ")
```

Shell ×

```
>>> %Run HoangThithanhTam_TL.py
```

```
Nhập số tự nhiên n: 15
```

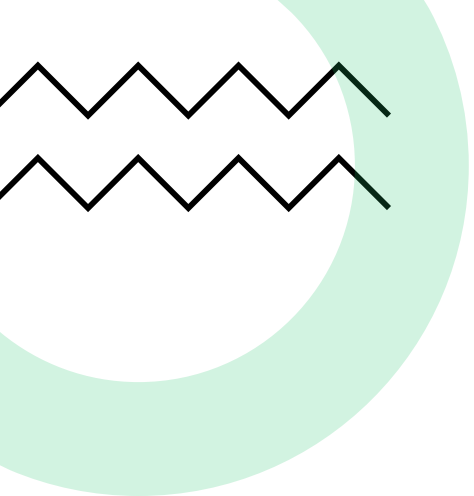
```
2 3 5 7 11 13
```

```
>>> %Run HoangThithanhTam_TL.py
```

```
Nhập số tự nhiên n: 41
```

```
2 3 5 7 11 13 17 19 23 29 31 37 41
```



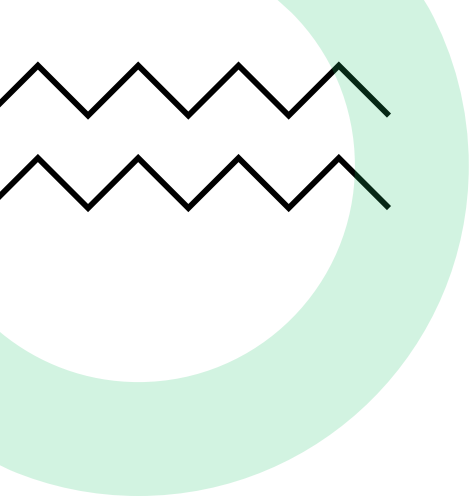


Ví dụ 2. Chương trình sử dụng chương trình con.

Cho trước hai dãy số B, C, chương trình chính cần tính tổng các số hạng dương của mỗi dãy này. Chúng ta sẽ thiết lập hàm **tongduong**(A) để tính tổng các số hạng lớn hơn của một dãy A. Chương trình chính sẽ gọi hàm **tongduong**(A)

Chương trình có thể như sau:





HoangThithanhTam_TL.py * x

```
1 def tongduong(A) :
2     S=0
3     for k in A:
4         if k > 0:      S = S + k
5     return S
6 # Chương trình chính
7 A = [0, 2, -1, 5, 10, -3]
8 B = [1, -10, -11, 8, 2, 0, -5]
9 # Sử dụng hàm tongduong tính tổng các số dương của dãy A
10 print("Tổng các số dương trong dãy A = ",tongduong(A))
11 # Sử dụng hàm tongduong tính tổng các số dương của dãy B
12 print("Tổng các số dương trong dãy B = ",tongduong(B))
13
```

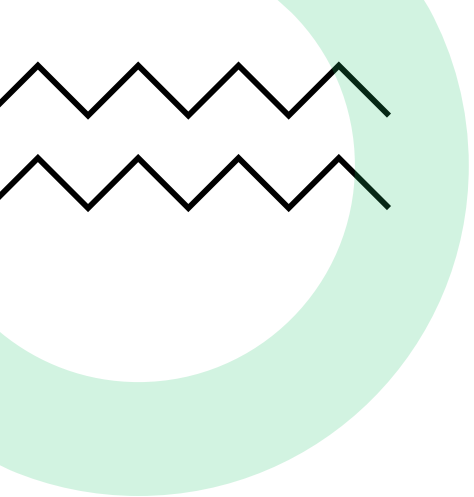
Shell x

```
>>> %Run HoangThithanhTam_TL.py
```

```
Tổng các số dương trong dãy A = 17
Tổng các số dương trong dãy B = 11
```

```
>>> |
```

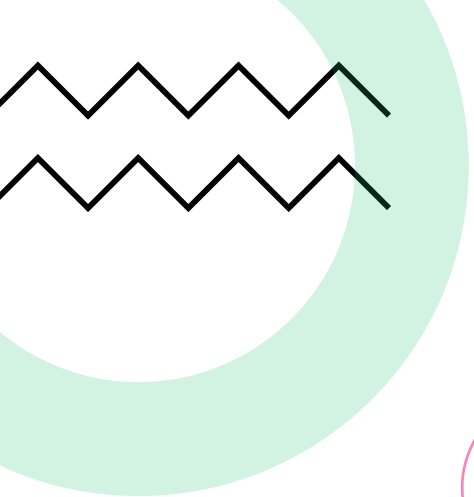




Tóm lại:

Sử dụng chương trình con có thể giúp phân chia việc giải một bài toán lớn thành giải quyết các bài toán nhỏ và phát huy được tinh thần làm việc nhóm. Chương trình chính có cấu trúc rõ ràng, dễ hiểu hơn; Nếu cần hiệu chỉnh, phát triển và nâng cấp cũng thuận tiện hơn.

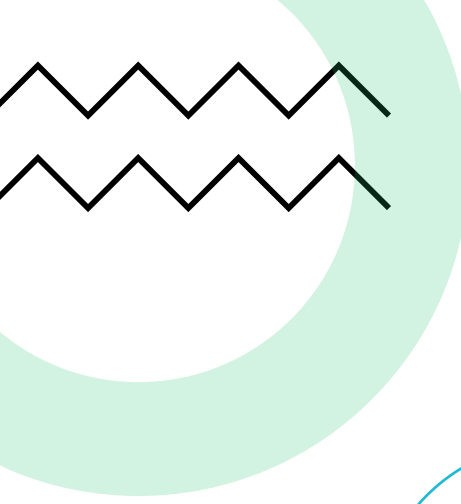




1. Sử dụng hàm prime, em hãy viết chương trình in ra các số nguyên tố trong khoảng từ m đến n, với m, n là hai số tự nhiên và $1 < m < n$

2. Em hãy nêu một công việc/bài toán nào đó mà có thể sử dụng hàm để giải





Viết chương trình chính yêu cầu nhập số tự nhiên n từ bàn phím và in các số nguyên tố nhỏ hơn hoặc bằng n ra màn hình. Trong phần thực hành của Bài 26 em đã biết hàm $\text{prime}(n)$ kiểm tra số n có là số nguyên tố.

Em sẽ viết chương trình giải bài toán này như thế nào?





3. THỰC HÀNH

Truyền giá trị cho đối số của hàm

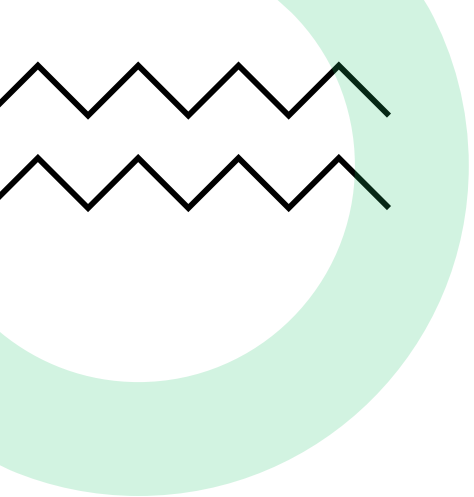
Nhiệm vụ 1. Thiết lập hàm $f_sum(A, b)$ có chức năng tính tổng các số của danh sách A theo quy định sau:

- Nếu $b = 0$ thì tính tổng các số của danh sách A
- Nếu b khác 0 thì chỉ tính tổng các số dương của A

Hướng dẫn. Chương trình luôn kiểm tra giá trị của đối số b khi tính tổng các số của danh sách A

Chương trình có thể như sau:





HoangThithanhTam_TL.py * x

```
1  def f_sum(A, b):
2      S = 0
3      for x in A:
4          if b == 0:
5              S = S + x
6          else:
7              if x > 0:
8                  S = S + x
9      return S
```



Nhiệm vụ 2. Thiết lập hàm `f_dem(msg, sep)` có chức năng đếm số từ của một xâu `msg` với kí tự tách từ là `sep`

Ví dụ:

```
f_dem("Mùa thu lịch sử", " ")
```

```
# trả lại giá trị 4
```

```
f_dem("Mùa thu lịch sử", ". ")
```

```
# trả lại giá trị 1
```

Hướng dẫn. Để tách xâu `msg` thành các từ, ta dùng lệnh `split()`. Tham số `sep` chính là tham số của lệnh `split()`.

Chương trình có thể như sau:

HoangThithanhTam_TL.py ×

```
1 def f_dem(msg, sep):  
2     xlist = msg.split(sep)  
3     return len(xlist)  
4 print(f_dem("Hoàng Thị Thanh Tâm THPT Thăng Long", " "))
```

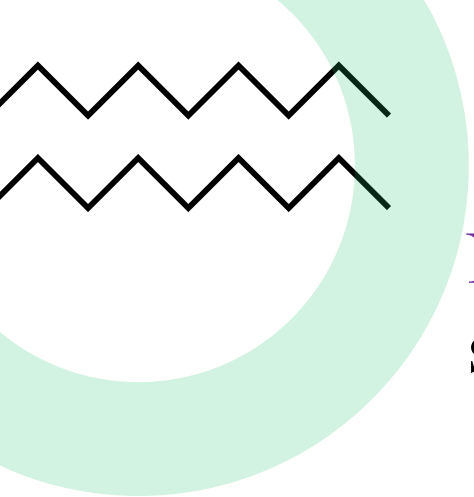
Shell ×

```
>>> %Run HoangThithanhTam_TL.py
```

```
7
```

```
>>> |
```





Nhiệm vụ 3. Thiết lập hàm `merge_str(s1, s2)` với `s1, s2` là hai xâu cần gộp .

Hàm này sẽ gộp hai xâu `s1, s2` theo cách, lấy lần lượt kí tự `s1, s2` đưa vào xâu kết quả. Nếu có một xâu hết kí tự thì đưa phần còn lại của xâu dài hơn vào xâu kết quả. Ví dụ nếu `s1 = “1111”`, `s2 = “0000”`, thì xâu kết quả là “10101010”

Hướng dẫn. Gọi `S` là xâu kết quả trước và sau khi gộp hai xâu `s1` và `s2`, chương trình có thể như sau:



```
1 def merge_str(s1, s2) :
2     S = ""
3     l1 = len(s1)
4     l2 = len(s2)
5     l = min(l1,l2)
6     for i in range (l):
7         S = S + s1[i] + s2[i]
8         if l1 < l2 :
9             for i in range(1, l2):
10                 S = S + s2[i]
11         if l1 > l2:
12             for i in range (1, l1):
13                 S = S + s1[i]
14     return S
15 a=input("Nhập xâu a = ")
16 b=input("Nhập xâu b = ")
17 print(merge_str(a,b))
```

Python 3.7.9 (bundled)

>>> %Run HoangThithanhTam

Nhập xâu a = 1111

Nhập xâu b = 0000

10101010

>>> %Run HoangThithanhTam

Nhập xâu a = 111111

Nhập xâu b = 0000

1011101110111011

/////



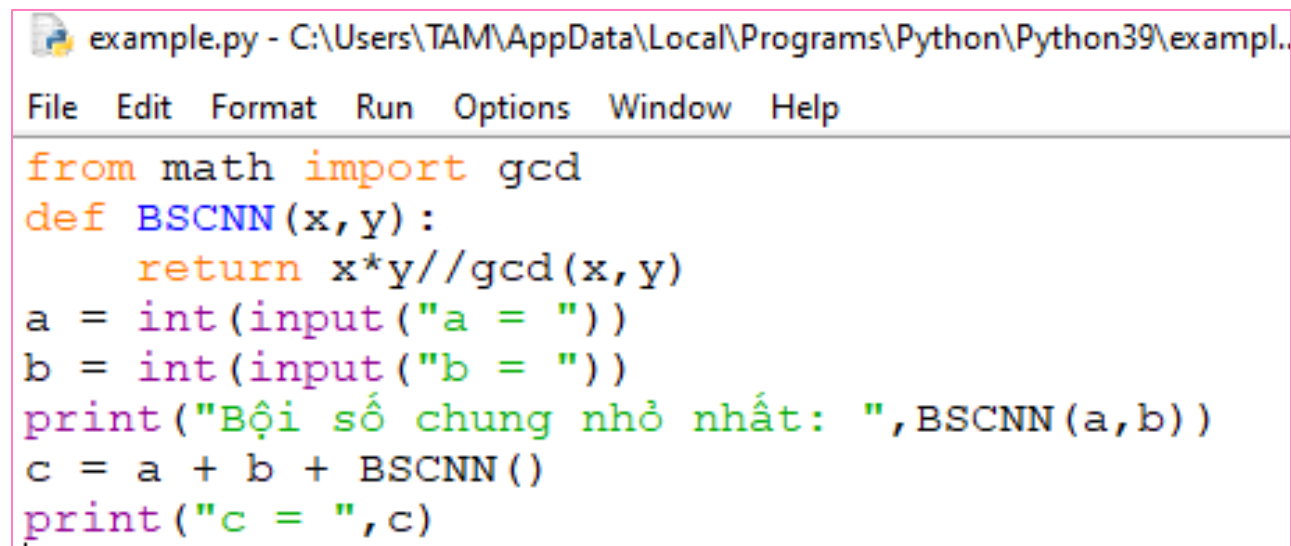
LUYỆN TẬP

1. Thiết lập hàm `power(a, b, c)` với `a, b, c` là số nguyên. Hàm trả lại giá trị $(a+b)^c$
2. Thiết lập hàm `change()` có hai tham số là xâu `ho_ten` và số `c`. Hàm sẽ trả lại xâu kí tự `ho_ten` là chữ in hoa nếu `c = 0`. Nếu tham số `c` khác 0 thì hàm trả lại xâu `ho_ten` là chữ in thường.
3. Viết chương trình thực hiện. Nhập hai số tự nhiên từ bàn phím, hai số cách nhau bởi dấu phẩy, in ra ước chung lớn nhất (UCLN) của hai số.
4. Viết chương trình thực hiện. Nhập `n` số tự nhiên từ bàn phím, hai số cách nhau bởi dấu cách. Tính và in ra tổng của các số này.



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)
```

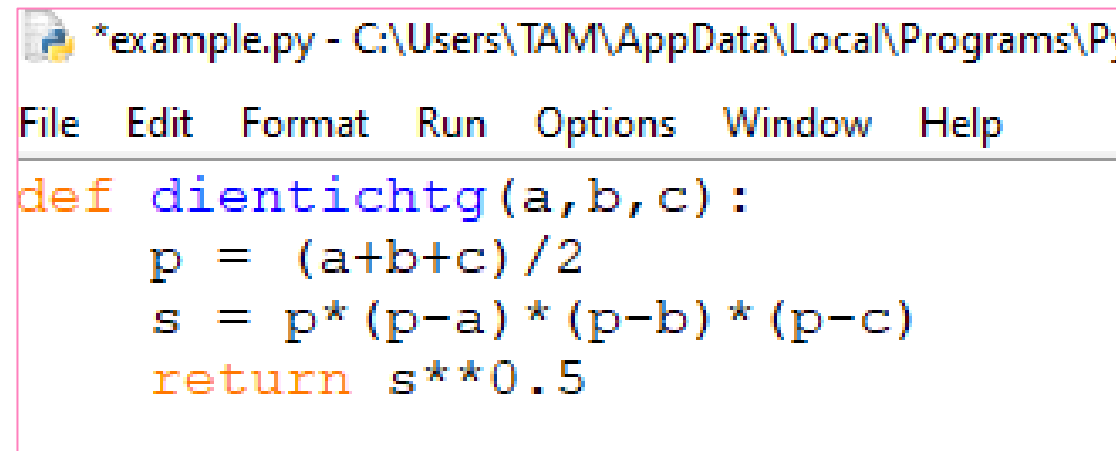
Hình 8. Chương trình về hàm bội số chung nhỏ nhất



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

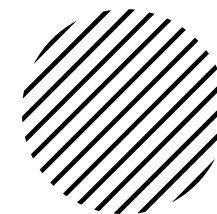
Bài 3. Sử dụng kết quả của Bài 2 phần *Luyện tập*, em hãy viết chương trình giải bài toán ở *Hoạt động 1*

```
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))
```

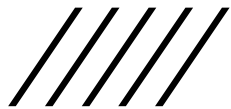
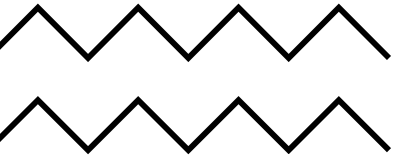

BÀI TẬP



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

