

SAU BÀI NÀY EM SẼ:

- Biết cách thiết lập các tham số của hàm. Hiểu được cách truyền giá trị thông qua đối số hàm.
- Biết viết chương trình có sử dụng chương trình con.



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 ĐỐI SỐ CỦA HÀM

Hoạt động 1 Phân biệt tham số và đối số

Quan sát ví dụ sau, tìm hiểu cách dữ liệu được truyền qua tham số vào hàm. Thảo luận để giải thích kết quả.



Ví dụ. Cách truyền dữ liệu qua tham số.

```
1 >>> def f(a,b,c):
2     return a+b+c
3 >>> f(1,2,3)
4 6
5 >>> x,y,z = 10,20,5
6 >>> f(x,y,z)
7 35
8 >>> f(a,b,c)
Traceback (most recent call last):
  File "<pyshell#6>", line 1, in <module>
    f(a,b,c)
NameError: name 'a' is not defined
13 >>>
```

Hàm f() có ba tham số a, b, c.

Hàm f() được gọi với ba giá trị cụ thể.

Hàm f() được gọi với ba biến đã có giá trị.

Lời gọi hàm bị lỗi nếu các tham số được truyền vào chưa có giá trị.

Giải thích:

- Dòng 1: Hàm $f()$ đã được định nghĩa với ba tham số a, b, c . Hàm có trả lại giá trị là $a + b + c$. Chú ý trong định nghĩa hàm, các tham số được coi như biến.
- Dòng 3: Hàm $f()$ được gọi với ba giá trị cụ thể là 1, 2, 3. Các giá trị được truyền qua tham số được gọi là đối số. Đối số tại dòng 3 là các số cụ thể.
- Dòng 6: Hàm $f()$ được gọi với ba biến x, y, z , đã được gán giá trị (dòng 5). Các biến được truyền qua tham số được gọi là đối số, kết quả trả lại là $x + y + z$.
- Dòng 10: Hàm $f()$ được gọi với ba biến a, b, c không xác định giá trị nên lời gọi hàm $f(a, b, c)$ báo lỗi do không xác định được giá trị của a, b, c .

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?

2. CÁCH SỬ DỤNG CHƯƠNG TRÌNH CON

Hoạt động 2 Khi nào nên sử dụng chương trình con?

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?



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 $\text{prime}(n)$ 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:

```
def prime(n):  
    C = 0  
    k = 1  
    while k < n:  
        if n%k == 0:  
            C = C + 1  
        k = k + 1
```

```

    if C == 1:
        return True
    else:
        return False

# Chương trình chính
n = int(input("Nhập số tự nhiên n: "))
for k in range(1,n+1):
    if prime(k):
        print(k, end = " ")

```

Sử dụng hàm prime đã định nghĩa ở trên để kiểm tra số k: Hàm prime(k) sẽ trả lại True nếu k là số nguyên tố.

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 0 của một dãy A. Chương trình chính sẽ gọi hàm tongduong(A).

Chương trình có thể như sau:

```

>>> def tongduong(A):
    S = 0
    for k in A:
        if k > 0:
            S = S + k
    return S

# Chương trình chính
>>> A = [0,2,-1,5,10,-3]
>>> B = [1,-10,-11,8,2,0,-5]
>>> tongduong(A)
17
>>> tongduong(B)
11
>>> print(B)

```

Sử dụng hàm tongduong tính tổng các số dương của dãy A

Sử dụng hàm tongduong tính tổng các số dương của dãy B

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.



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:

```
def f_sum(A,b):
    S = 0
    for x in A:
        if b == 0:
            S = S + x
        else:
            if x > 0:
                S = S + x
    return S
```

Nhiệm vụ 2. Thiết lập hàm `f_dem(msg, sep)` có chức năng đếm số các từ của một chuỗi `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 chuỗi `msg` thành các từ, chúng ta dùng lệnh `split()`. Tham số `sep` chính là tham số của lệnh `split()`.

Chương trình có thể như sau:

```
def f_dem(msg,sep):
    xlist = msg.split(sep)
    return len(xlist)
```

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

Hàm này sẽ gộp hai chuỗi `s1, s2` theo cách, lấy lần lượt từng kí tự của `s1, s2` đưa vào chuỗi kết quả. Nếu có một chuỗi hết kí tự thì đưa phần còn lại của chuỗi dài hơn vào chuỗi kết quả. Ví dụ nếu `s1 = "1111", s2 = "0000", c = 1` thì chuỗi kết quả là `"10101010"`.

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

```
def merge_str(s1, s2):  
    S = ""  
    l1 = len(s1)  
    l2 = len(s2)  
    l = min(l1, l2)  
    for i in range(l):  
        S = S + s1[i] + s2[i]  
    if l1 < l2:  
        for i in range(l, l2):  
            S = S + s2[i]  
    if l2 < l1:  
        for i in range(l, l1):  
            S = S + s1[i]  
    return S
```



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. 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.



VẬN DỤNG

1. 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.
2. Thiết lập hàm change() có hai tham số là chuỗi ho_ten và số c. Hàm sẽ trả lại chuỗi kí tự ho_ten là chữ in hoa nếu c = 0. Nếu tham số c khác 0 thì hàm trả lại chuỗi ho_ten là chữ in thường.

Gợi ý: Sử dụng các phương thức s.upper() và s.lower() để chuyển đổi các kí tự của chuỗi s sang chữ in hoa và in thường.