

Patrones comunes

Programación

<http://progra.usm.cl>

UTFSM



Departamento de Informática
Universidad Técnica Federico Santa María

Sumar cosas

Escriba una función que reciba como parámetro un número entero. La función debe retornar la suma de los números al cuadrado desde el 1 hasta el valor ingresado como parámetro:

$$1^2 + 2^2 + 3^2 + \dots + (n - 1)^2 + n^2.$$

Solución:

```
def sumar(n):  
    suma = 0  
    c = 1  
    while c <= n:  
        d = c ** 2  
        suma = suma + d  
        c = c + 1  
    return suma
```

Sumar cosas

Escriba una función que retorne la suma de los cuadrados de los números ingresados por el usuario. La función no recibe parámetro y termina cuando el usuario ingrese un cero:

```
>>> sumatoria()  
5  
1  
4  
0  
42
```

Solución:

```
def sumar():  
    suma = 0  
    flag = True  
    while flag:  
        n = int(raw_input()) ** 2  
        if n == 0:  
            flag = False  
        else:  
            suma = suma + n  
    return suma
```

Patrón *sumar cosas*:

```
suma = 0
```

```
ciclo:
```

```
    n = calcular()
```

```
    suma = suma + n
```

Patrón *multiplicar cosas*:

```
prod = 1
```

```
ciclo:
```

```
    n = calcular()
```

```
    prod = prod * n
```

Patrón *contar cosas*:

```
cuenta = 0
```

```
ciclo:
```

```
    ...
```

```
    cuenta = cuenta + 1
```


Encontrar el máximo de enteros positivos

Escriba una función que reciba como parámetro la cantidad de números a ingresar, luego solicite los números enteros positivos y retorne el número mayor ingresado:

```
>>> mayor(10)
```

```
4
```

```
2
```

```
7
```

```
5
```

```
5
```

```
3
```

```
7
```

```
2
```

```
4
```

```
7
```

Solución 1:

```
def mayor(n):  
    my = -1  
    c=1  
    while c <= n:  
        a = int(raw_input())  
        if a > my:  
            my = a  
        c = c + 1  
    return my  
x = int(raw_input('Cuantos datos: '))  
print "El mayor es", mayor(x)
```

Solución 2:

```
def mayor(n):  
    my = -1  
    c = 1  
    while c <= n:  
        a = int(raw_input())  
        my = max(my, a)  
        c = c + 1  
    return my  
x = int(raw_input('Cuantos datos: '))  
print 'El mayor es', mayor(x)
```

Encontrar el máximo de números reales

Escriba una función que pida al usuario ingresar varios números reales (positivos o negativos) hasta que se ingrese un cero, la función no recibe parámetro y retorna cuál fue el valor mayor ingresado:

```
>>> mayor ()  
-2.5  
7.1  
-5.0  
6.1  
0  
7.1
```

Solución:

```
def mayor():  
    my = -float('inf')  
    flag = True  
    while flag:  
        n = float(raw_input())  
        if n == 0:  
            flag = False  
        else:  
            my = max(my, n)  
    return my  
  
print 'El mayor es', mayor()
```

Patrón *encontrar el mayor*:

```
# Ejemplo 1
mayor = numero_muy_chico
ciclo:
    n = calcular()
    mayor = max(mayor, n)
```

```
# Ejemplo 2
mayor = numero_muy_chico
ciclo:
    n = calcular()
    if n > mayor:
        mayor = n
```

Patrón *encontrar el menor*:

```
# Ejemplo 1
menor = numero_muy_grande
ciclo:
    n = calcular()
    menor = min(menor, n)
```

```
# Ejemplo 2
menor = numero_muy_grande
ciclo:
    n = calcular()
    if n < menor:
        menor = n
```

Combinaciones de dados

Escriba un programa que muestre todas las combinaciones de dos dados que entreguen un puntaje mayor que siete.

Solución:

```
i=1
while i<7:
    j=1
    while j<7:
        if i + j > 7:
            print i, j
        j+=1
    i+=1
```

Patrón *pares de cosas*:

```
while i < n:  
    while j < m:  
        hacer algo con i y j  
        j+=1  
    i+=1
```