

## Problemas de estructuras repetitivas

1) Programa que permite calcular el factorial de un número entero.

Matemáticamente el factorial de un número se representa así:  $3!$ , que se lee factorial de 3, lo cual equivale a lo siguiente:

$$3! = 1 * 2 * 3 = 6, \text{ por lo tanto el factorial de 3 es 6, otros ejemplos}$$

$$4! = 1 * 2 * 3 * 4 = 24$$

$$5! = 1 * 2 * 3 * 4 * 5 = 120$$

Aquí conocemos la cantidad de veces que ciclo se va ejecutar en el caso de  $3!$  el ciclo se ejecutara 3 veces, usamos un **for**. El código en Python es:

```
1 """
2     Programa que permite calcular el factorial
3     de un numero entero.
4 """
5 # Inicializacion de variables
6 num = 0
7 fact = 1
8 # Entrada datos
9 num = int(input("De numero entero: "))
10 # Proceso
11 for i in range(1, num + 1):
12     fact *= i
13 print("\nEl factorial de ", num, " es: ", fact)
```

Nota: El factorial de **8** es: **40320**, mientras más grande es el numero cuyo factorial queremos calcular su factorial crece en forma exponencial por lo que es conveniente que la variable en algunos casos sea **float** y no **int**.

2) Programa que lee una cantidad de valores enteros y un valor b. Determine e imprima el porcentaje de valores leídos menores que b.

De acuerdo al enunciado se va procesar una lista de valores enteros cuyo tamaño es conocido, por lo tanto el ciclo a usar será un ciclo **for**, el programa tendrá como entrada la cantidad de valores a leer y valor de **b**, cual nos servirá para determinar el porcentaje de los valores leídos que son menores que el.

**Código:**

```
1 """
2     Programa que lee una lista de valores enteros y un valor b
3     Determine e imprima el porcentaje de valores leídos
4     menores que b.
5 """
6 # Inicializar variables
7 cant = 0 # Cantidad de enteros a leer.
8
9 valor = 0 # Guarda el valor leído.
10 porc=0.0 # Para guardar el porcentaje.
11 b=0 # Valor a comparar
12 cont=0 # Cuenta valores menores que comp.
```

```

13  # Entrada de datos
14  cant=int(input("Indique cuantos valores seran leidos: "))
15  b=int(input("Indique de b: "))
16  # Ciclo de lectura y Proceso
17  for i in range(1,cant+1):
18      valor=int(input("De un valor entero: "))
19      if (valor<b):
20          cont+=1
21  # Calculos finales salida resultados
22  porc=cont/cant*100
23  print("\nPorcentaje = ",porc)

```

Ejecución:

```

Indique cuantos valores seran leidos: 10
Indique de b: 8
De un valor entero: 5
De un valor entero: 12
De un valor entero: 7
De un valor entero: 9
De un valor entero: 8
De un valor entero: 25
De un valor entero: 4
De un valor entero: 16
De un valor entero: 12
De un valor entero: 67

Porcentaje = 30.0

```

- 3) Si desconociéramos el tamaño de la lista no podríamos usar un ciclo for, sino un ciclo while con centinela.

Código:

```

1  """
2      Programa que lee ua lista de valores enteros y un valor b
3      Determine e imprima el porcentaje de valores leidos
4      menores que b.
5  """
6  # Inicializar variables
7  resp = "s" # variable centinela
8  valor = 0 # Guarda el valor leído.
9  cant = 0 # Catidad de valores leidos.
10 porc = 0.0 # Para guardar el porcentaje.
11 b = 0 # Valor a comparar
12 cont = 0 # Cuenta valores menores que comp.
13 # Entrada de datos y Ciclo leyendo y procesando la lista
14 b = int(input("Indique de b: "))

```

```

15 # Ciclo de lectura y Proceso
16 while (resp.upper() == "S"):
17     valor = int(input("De un valor entero: "))
18     cant += 1
19     if (valor < b):
20         cont += 1
21     resp = input("Lee mas valore Si-->S, No-->N ")
22 # Calculos finales salida resultados
23 porc = cont / cant * 100
24 print("\nPorcentaje = ", porc)

```

Línea 16: aquí usamos la función de cadena **upper** aplicada a la variable centinela, la razón, esto nos da la posibilidad de ingresar el carácter “s” en mayúscula o minúscula, ya que **upper** convierte en mayúscula el carácter leído, otra forma seria escribir el código así:

```
while(resp=="s" or resp=="S"):
```

Aquí evaluamos si el carácter leído esta en minúscula o mayúscula, esto logra el mismo efecto que al usar **upper**. En el programa nos vemos en la necesidad de contar la cantidad de caracteres leídos, ya que no conocemos el tamaño de la lista, línea 20, en la 21 controlamos el valor del centinela introduciendo una s si queremos continuar o una n si queremos detener el proceso de lectura de los valores de la lista.

Datos de prueba:

```

Indique de b: 12
De un valor entero: 3
Lee mas valore Si-->S, No-->N s
De un valor entero: 34
Lee mas valore Si-->S, No-->N s
De un valor entero: 3
Lee mas valore Si-->S, No-->N s
De un valor entero: 41
Lee mas valore Si-->S, No-->N n

Porcentaje = 50.0

```

- 4) Programa que lee una lista de N enteros y determina cual de los enteros leídos es el mayor y cual el menor. Aquí usaremos un **for** es conocida el tamaño de la lista.

Para procesar la lista debemos usar una variable bandera con la finalidad de determinar si el primer valor de la lista ya fue procesado, el cual usaremos como valor de referencia de partida para comparar con valores siguientes de la lista y determinar el mayor y el menor valor de la lista. Les recuerdo que una bandera es una variable que procesa dos valores, cierto o falso, 0 o 1 etc., al pasar de un valor a otro nos está indicando que el proceso o evento que queremos controlar ya tuvo lugar.

## Código:

```
1  """
2      Programa que lee una lista de N enteros y determina
3      cual de los enteros leídos es el mayor y cual
4      el menor.
5  """
6  # Inicializar variables
7  cant = 0 # Cantidad de enteros a leer.
8  band = 0 # Variable que verifica si el 1er valor fue leído. Bandera
9  mayor = menor = 0 # Guardan el mayor o el menor valor procesado en la list.
10 valor = 0 # Guarda el valor leído.
11 # Entrada de datos
12 cant = int(input("Cantidad de numeros a leer: "))
13 # Ciclo de lectura y Proceso datos.
14 for i in range(1, cant + 1):
15     valor = int(input("De un numero entero: "))
16     if (band == 0):
17         # Fijamos el 1er valor leído como el mayor y el menor
18         mayor = valor
19         menor = valor
20         band = 1 # Cambia la bandera indicando que el 1er valor fue procesado
21     elif (mayor < valor):
22         mayor = valor
23     elif (menor > valor):
24         menor = valor
25 # Salida de resultados
26 print("\nEl mayor valor leído fue: ", mayor)
27 print("El menor valor leído fue: ", menor)
```

Una vez procesado el 1er valor de la lista, líneas desde la 16 a la 20, considerando este valor el mayor y el menor de la lista, los siguientes valores se comparan contra las variables **mayor** y **menor**, línea 21, aquí verificamos si el valor leído es mayor que el valor que considerábamos el mayor de la lista, si esto se cumple, debemos cambiar el contenido de **mayor** al nuevo valor encontrado, de no cumplirse, se procede con la línea 23, aquí verificamos si el valor leído es menor que el valor que considerábamos el menor de la lista, si esto se cumple, debemos cambiar el contenido de **menor** al nuevo valor encontrado, de no cumplirse, se procesa el siguiente valor de la lista. Al terminar el ciclo en mayor tendremos el mayor de la lista y en menor el menor de la lista. **Ejecución:**

```
Cantidad de numeros a leer: 7
De un numero entero: 3
De un numero entero: 15
De un numero entero: 10
De un numero entero: 1
De un numero entero: 25
De un numero entero: -3
De un numero entero: 12

El mayor valor leído fue: 25
El menor valor leído fue: -3
```

5) Programa que lea un entero positivo y determina si el entero leído es primo no.

Consideraciones:

- a) Un entero positivo es primo si es divisible por 1 y el mismo.
- b) Un número primo es impar.
- c) El procedimiento a seguir es hallar un entero entre 2 y  $\sqrt{N}$  que lo divida exactamente, si lo hay no es primo, si no lo hay es primo. Pasos a seguir
- d) Hallar la  $\sqrt{N}$ .
- e) Obtener la parte entera P del paso anterior.
- f) Hallar un entero entre 2 y P que divida exactamente a N.
- g) Si encontramos un entero no es primo, de lo contrario es primo.

Código:

```
1  # Importamos la funcion sqrt
2  from math import sqrt
3  # Inicializar variables
4  pnum = 0 # Valor a evaluar si es primo o no
5  raiz=0.0 # Guarda la raiz dl numero
6  eraiz=0 # Guarda la parte entera de la raiz
7  i=0 # Variable controladora del ciclo
8  # Entrada dato
9  pnum=int(input("De un entero positivo: "))

10 if (pnum%2==0):
11     print("No es primo")
12 else:
13     raiz = sqrt(pnum) #Obtenemos la raiz cuadrada
14     eraiz = int(raiz) # Obtemos la parte entera
15     i = 2
16     # Ciclo para encontrar un numero i que divida exactamente
17     # el numero leído.
18     while (pnum % i != 0 and i < eraiz):
19         i += 1
20     # Se verifica si i divide exactamente al numero leído
21     # si lo no hay es primo, de lo cotrari es primo.
22     if (pnum % i != 0):
23         print("Es primo")
24     else:
25         print(" No Es primo")
```

Ejecución:

De un entero positivo: 17  
Es primo

De un entero positivo: 721  
No Es primo