

TC1028

Pensamiento computacional para ingeniería

Darío Cuauhtémoc Peña Mariano

Programa 1

Funcionamiento:

En este programa desarrollé un algoritmo para que si un número es par se dividiera entre 2 y si el número es impar se multiplicara por tres y sumara uno.

Para iniciar declaré una variable que sería recibida por la terminal, un número entero, después inicié con un while, el cual evaluaría que el número sea mayor a uno para que el ciclo se ejecute, después anidé un If dentro del while, con este while evaluamos si el número es par o impar, para esto evaluamos que el residuo de dividir el número entre de dos sea cero, si esto se cumple significa que el número es par, de lo contrario, sería impar y se ejecutaría un else.

Para el caso par lo que hicimos fue que el "numero" tomara el valor de el mismo "numero dividido entre dos, y lo mostraría en consola. En el caso de ser el impar, "numero" sería multiplicado por tres y sumado un 1 y lo mostraría en consola. Con esto logramos que nuestro código funcionara correctamente y en consola muestra la serie de números al aplicar éstas reglas.

Pseudocódigo:

Variables:

Enteros: numero <-- Recibe de terminal

Inicio

mostrar ("conjetura de Collatz para el número" + numero)

while (numero > 1) entonces:

 if(numero % 2 == 0) entonces:

 numero = numero / 2

 mostrar(numero)

 else(numero % 2 != 0) entonces:

 numero = (numero x3) +1

 mostrar(numero)

regresar 0

Fin

Funcionamiento del programa:

```

ejer_1.py x  ejer_2.py
porgra 1 > tareaOpcional >  ejer_1.py > ...
1  numero = int(input("Ingrese número de serie"))
2
3  print("conjetura de Collatz para el número",numero, "es:")
4
5  while (numero>1):
6
7      if (numero % 2 == 0):
8          numero = int(numero / 2)
9          print(numero)
10     else:
11         numero= int(numero * 3 +1)
12         print(numero)
13
14

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SQL CONSOLE  COMMENTS
Python  + v  [ ]  [ ]  [ ]  [ ]  [ ]  [ ]

PS F:\Tec\programacion\python> & C:/Users/User/AppData/Local/Microsoft/WindowsApps/python3.11.exe "f:/Tec/prog
ramacion/python/porgra 1/tareaOpcional/ejer_1.py"
Ingrese número de serie3
conjetura de Collatz para el número 3 es:
10
5
16
8
4
2
1
PS F:\Tec\programacion\python> 
```

```

ejer_1.py x  ejer_2.py
porgra 1 > tareaOpcional >  ejer_1.py > ...
1  numero = int(input("Ingrese número de serie"))
2
3  print("conjetura de Collatz para el número",numero, "es:")
4
5  while (numero>1):
6
7      if (numero % 2 == 0):
8          numero = int(numero / 2)
9          print(numero)
10     else:
11         numero= int(numero * 3 +1)
12         print(numero)
13
14

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SQL CONSOLE  COMMENTS
Python  + v  [ ]  [ ]  [ ]  [ ]  [ ]  [ ]

PS F:\Tec\programacion\python> & C:/Users/User/AppData/Local/Microsoft/WindowsApps/python3.11.exe "f:/Tec/prog
ramacion/python/porgra 1/tareaOpcional/ejer_1.py"
Ingrese número de serie7
conjetura de Collatz para el número 7 es:
22
11
34
17
52
26
13
40
20
10
5
16
8
4
2
1
PS F:\Tec\programacion\python> 
```

Programa 2

Funcionamiento:

Para este programa tenía que replicar la figura de abajo, al analizarla llegué a la conclusión que la figura tiene 3 tipos de patrones diferentes, el primer patrón que va aparecer en la primera línea, el segundo patrón que va aparecer de la segunda línea a la penúltima línea de la figura, y el tercer patrón que aparecerá en el último renglón.

Para iniciar el código declararé 4 variables; "lineas" que es ingresado por el usuario, "asteriscos" que le asignamos el valor de "lineas", "n" que es un número entero donde iniciara nuestro while y le asignamos un 1, y "renglonString" que es una variable vacía la cuál le vamos a dar su valor más adelante en el código.

Como tenemos que hacer una iteración para que se imprima cada renglón usaremos un while, el cuál se ejecutará siempre y cuando "n" sea menor o igual a "asteriscos", además a esto en cada ciclo sumamos en 1 a "n", el cual nos daría el valor del renglón en el que se encuentra el ciclo, también restaremos en 1 a "lineas", ya que esta variable nos brinda información de los espacios que tendrá al principio cada renglón, como la relación es que al principio hay todos los espacios y al final hay no hay por eso por cada renglón se resta en 1.

Dentro del while quise anidar un switch, como en python no existen traté de emularlo con elifs, evalué tres casos para nuestros tres patrones, **para el primer patrón** usamos un if, en dónde evalúa que si está en su primera iteración, es decir "n" es igual a 1 entonces a "renglonString" le daremos el valor de concatenar un " " multiplicado por "lineas" menos 1, más un "*", como resultado el primer patrón, ya que a la izquierda se van a encontrar esos espacios multiplicados por "lineas" -1 más un asterisco, por último imprimimos "renglonString".

Para el patrón 3, usé un elif, el cual evalúa si estamos en la última línea de la figura, es decir si "n" es igual a "asteriscos", en caso de que se cumpla "renglonString" va tomar el valor de "*" multiplicado por "asteriscos" por 2 menos 1, ya que en el último renglón los asteriscos es igual al número de ciclos del while menos 1, por último imprimimos "renglonString" para que aparezca en consola.

Para el patrón 2, los asteriscos sólo aparecen en la parte interna de la figura, es decir en los renglones que no son ni el primero ni el último, usamos un elif para hacerlo, este evaluará que se ejecute el cuerpo de la función si no es la primera ni la última fila, para esto tuvimos que evaluar que las condiciones del patrón 1 y 3 no se cumpliera, entonces evalué que "n" **no** sea igual a 1 o que "n" **no** sea igual a "asteriscos", si alguna de estas dos condiciones no se cumplen entonces a "renglonString" le damos el valor de concatenar los espacios que aparecen a la izquierda de los asteriscos, osea " " multiplicado por lineas -1, más un "*", más los espacios que aparecen a la derecha del primer asterisco que lo obtenemos de multiplicar un " " por n -1 multiplicado por 2 y al resultado le restamos 1, por última concatenación agregamos un "*". Al tener "renglonString" lo imprimimos y mostramos en pantalla.

```
      *
     * *
    *   *
   *     *
  *       *
 *         *
*****
```

Pseudocódigo:

Variables:

Enteros: lineas <-- Recibe de terminal, asteriscos <-- lineas, "n" <-- 1

Strings: renglonString <-- ""

Inicio

while (n <= asteriscos) entonces:

if (n == 1) entonces:

renglonString <-- " " x lineas - 1 + "*"

mostrar(renglonString)

elif (n == asteriscos) entonces:

renglonString <-- "*" x ((asteriscos x 2) - 1)

mostrar(renglonString)

elif (not(n == lineas) or not(n == 1)) entonces:

renglonString <-- " " x lineas - 1 + "*" + " " x ((n - 1) x 2) - 1 + "*"

mostrar(renglonString)

n <-- n + 1

lineas <-- lineas - 1

regresar 0

Fin

Funcionamiento del programa:

```

+ ejer_1.py  + ejer_2.py  X
porgra 1 > tareaOpcional > + ejer_2.py > ...
1  lineas= int(input("ingresar renglones"))
2  asteriscos = lineas
3  n =1
4  renglonString =""
5
6  while (n <= asteriscos):
7
8      if (n ==1):
9          renglonString=(" " * int(lineas -1)+ "")
10         print(renglonString)
11
12     elif(n == asteriscos):
13         renglonString=("*" * (int(asteriscos *2 )-1))
14         print(renglonString)
15
16     elif(not(n == lineas) or not(n ==1)):
17         renglonString=(" " * int(lineas -1)+"*"+ " " * int(((n -1)*2 )-1 )+ "")
18         print(renglonString)
19
20
21
22     n = n+1
23     lineas = lineas - 1

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SQL CONSOLE COMMENTS Python + ▾ ▢ ✕

```

PS F:\Tec\programacion\python> & C:/Users/User/AppData/Local/Microsoft/WindowsApps/python3.11.exe "f:/Tec/prog
ramacion/python/porgra 1/tareaOpcional/ejer_2.py"
ingresar renglones6
  *
 * *
*   *
*   *
*   *
*****
PS F:\Tec\programacion\python> & C:/Users/User/AppData/Local/Microsoft/WindowsApps/python3.11.exe "f:/Tec/prog
ramacion/python/porgra 1/tareaOpcional/ejer_2.py"
ingresar renglones10
  *
  * *
   * *
  * *
 *   *
*     *
*     *
*     *
*     *
*     *
*     *
*****
PS F:\Tec\programacion\python>

```

Connect Server not selected tabnine starter Ln 3, Col 5 Spaces: 4 UTF-8 CRLF Python 3.11.5 64-bit (microsoft store)