

PRELIMINARES

En su versión más simple, un ciclo anidado es un ciclo (ciclo interno) dentro de otro ciclo (ciclo externo). El registro del tiempo es un ejemplo de ciclos anidados fácil de entender:

- el segundero del reloj es un ciclo anidado dentro del ciclo del minuterero: por cada vuelta completa que da el segundero (ciclo interno), el minuterero ejecuta una iteración (avanza un minuto, ciclo externo).
- Cada 30 días (ciclo "interno" de 30 ciclos), se avanza 1 mes (ciclo "externo"). Tres ciclos "externos" (3 meses) corresponden a 90 días: 30 días en la primera iteración del ciclo externo (mes 1), 30 días en la segunda iteración del ciclo externo (mes 2) y 30 días en la tercera iteración del ciclo externo (mes 3).

Cada vez que hay una repetición dentro de otra, se usan ciclos. Por ejemplo, para dibujar el siguiente patrón por pantalla:

```
#####
#####
#####
```

Basta con reconocer que hay 3 repeticiones de la misma línea: #####

Pero esta línea es, a su vez, otra repetición: se repite 5 veces el símbolo #.

Por lo tanto, aquí se puede reconocer un ciclo anidado: el ciclo externo (línea) se repite 3 veces (3 líneas), mientras que el ciclo interno (símbolo) se repite 5 veces.

Un ejemplo de código que implementa este patrón por pantalla es el siguiente:

```
for i in range(1,4,1):
    for j in range(1,6,1):
        print("#",end="")
    print("")
```

Atención con la 3^{ra} línea de código (`print("#",end="")`). Esta línea, permite mostrar por pantalla el carácter # y no saltar a la siguiente línea (`end=""`). Por este motivo, la 6^{ta} línea de código es: `print("")`, lo que permite saltar una línea.

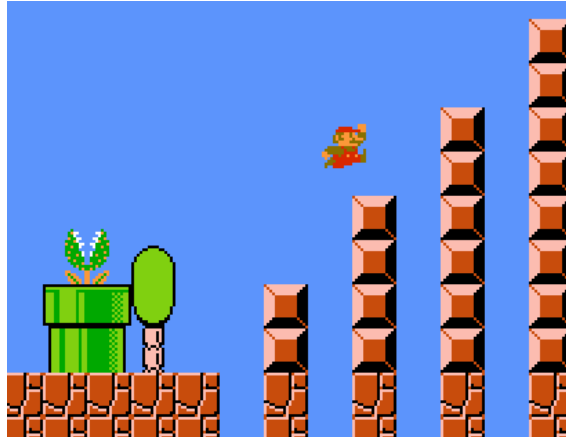
PROBLEMAS RESUELTOS

1. Escribe un programa que dibuje una matriz de símbolos #. El número de filas y el número de columnas de la matriz debe ser ingresado por el usuario.

```
numFilas=int(input("ingrese numero de filas "))
numColumnas=int(input("ingrese numero de columnas "))

for i in range(1,numFilas+1,1):
    for j in range(1,numColumnas+1,1):
        print("#",end="")
    print("")
```

2. En el mundo 8-1, del juego Super Mario Bros de Nintendo, el personaje debe superar unos obstáculos de escaleras y precipicios. A continuación se muestra una imagen con el obstáculo (una escalera):



Si uno voltea la escalera en 180 grados y representa cada escalón con el símbolo #, el programa en Python que imprime la escalera desplegaría los siguientes caracteres por pantalla:

```
#####
#####
###
###
##
##
#
#
```

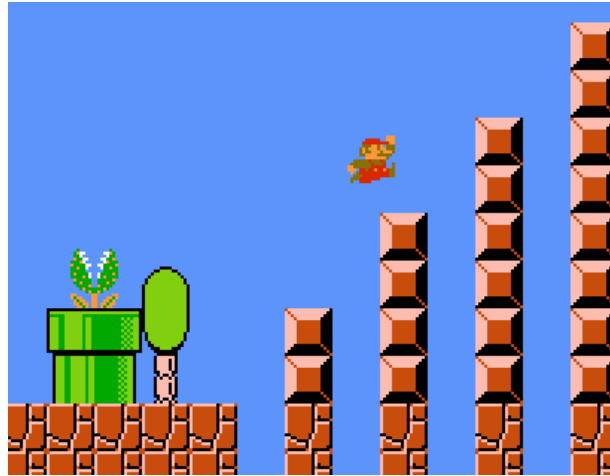
Escribe un programa en Python que solicite al usuario ingresar el número de escalones y despliegue en pantalla la escalera correspondiente con asteriscos. En el caso del ejemplo anterior, el número de escalones es 4.

Una solución posible:

```
numEscalones=int(input("ingrese numero de escalones "))

for i in range(numEscalones,0,-1):
    for j in range(1,3,1): #imprimiendo dos veces la linea de #
        for k in range(1,i+1,1):
            print("#",end="")
        print("")
```

3. Para el mismo escenario de la pregunta anterior:



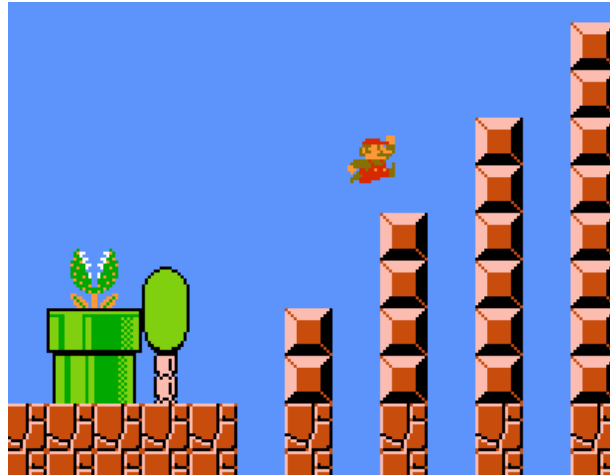
Si uno voltea la escalera en 90 grados a la derecha, y representa cada escalón con el símbolo * (asterisco), el programa en Python que imprime la escalera desplegaría los siguientes caracteres por pantalla:

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

Escribe un programa en Python que pida al usuario ingresar el número de escalones y despliegue en pantalla la escalera correspondiente con asteriscos. En el caso del ejemplo anterior, el número de escalones es 4.

```
numEscalones=int(input("ingrese numero de escalones "))
for i in range(1,numEscalones+1,1):
    for j in range(1,2*i+1,1):
        print("#",end="")
    print("")
```

4. Para el mismo escenario de la pregunta anterior:



Si uno no voltea la escalera, y representa cada escalón con el símbolo * (asterisco), el programa en Python que imprime la escalera desplegaría los siguientes caracteres por pantalla:

```

**
****
*****
*****

```

Escribe un programa en Python que pida al usuario ingresar el número de escalones y despliegue en pantalla la escalera correspondiente con asteriscos. En el caso del ejemplo anterior, el número de escalones es 4.

```

numEscalones=int(input("ingrese numero de escalones "))
for i in range(1,numEscalones+1,1):
    for j in range(1,2*(numEscalones-i)+1,1):
        print(" ",end="")
    for j in range(1,2*i+1,1):
        print("*",end="")
    print("")

```

No se preocupa si no le queda una escalera perfecta, esto es porque el ancho del carácter " " es mas pequeño, que el ancho del carácter "*"

5. Escriba un programa que despliegue en pantalla los valores de la función $f(x)$ para el rango de x que indique el usuario:

$$f(x) = \sum_{k=0}^x k^2$$

El programa debe solicitar al usuario el valor menor y mayor del rango de x . Un ejemplo de la ejecución del programa es la siguiente:

```
Ingrese valor inferior de x: 2
Ingrese valor superior de x: 5
x      f(x)
=====
2      5
3      14
4      30
5      55
```

```
limInf = int(input("ingrese límite inferior "))
limSup = int(input("ingrese límite superior "))
print("x      f(x)")
print("=====")
for i in range(limInf, limSup+1,1):
    suma=0
    for j in range(1,i+1,1):
        suma=suma+j*j
    print(i, "      ", suma)
```

6. Escribe el código que permite resolver la siguiente serie: $1! + 2! + 3! + \dots + N!$. El valor de N se solicita al usuario.

```
num=int(input("ingrese el valor N de la serie "))
total=0
for i in range(1,num+1):
    fact=1
    for j in range(1,i+1,1):
        fact=fact*j
    total=fact+total
print("el valor de la serie es ",total)
```

7. Crea un programa que despliegue el siguiente patrón de números:

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

El número de filas lo ingresa el usuario

Una solución posible:

```
num=int(input("ingrese el valor N del patron "))
for i in range(1,num+1,1):
    for j in range(1,i+1,1):
        print(j," ",end="")
    print (" ")
```

8. Crea un programa que despliegue el siguiente patrón de números:

```
1
2 3
4 5 6
7 8 9 10
```

El número de filas lo ingresa el usuario.

```
num=int(input("ingrese el valor N del patron "))
cont=1
for i in range(1,num+1,1):
    for j in range(1,i+1,1):
        print(cont," ",end="")
        cont=cont+1
    print (" ")
```

9. Escribe el código que escriba los N primeros números primos. El valor de N se solicita al usuario.

```

numN=int(input("Ingrese un valor N: "))
contador=0
numVerif=0

while contador<numN:
    numVerif=numVerif+1
    esPrimo = 1

    for posibleDivisor in range (2,numVerif,1):
        if (numVerif%posibleDivisor==0):
            esPrimo = 0

    if (esPrimo == 1):
        print(numVerif)
        contador=contador+1

```