3. Trabajando con condicionales y bucles (Tema 3)

Enunciado de la práctica

Objetivos:

- Dominar el almacenamiento de datos en variables.
- Dominar las estructuras de control if y while.

Se programarán las soluciones en este mismo Notebook en la celda que se suministra al efecto.

IMPORTANTE

Cuando trabajamos con bucles es muy fácil crear bucles infinitos, es decir, bucles que no terminan nunca porque no se llega a cumplir la condición de parada. La forma de parar la ejecución infinita de una celda es pulsando en el botón de parar ejecución (en VSC a la izquierda de la celda y en Jupyter en la parte superior).

Programa #1. Bucles nivel básico

Calcular el factorial y el sumatorio de un número dado.

```
numero = 5

El sumatorio de 5 es 15
El factorial de 5 es 120
```

Solución en la celda siguiente

```
numero=5
i=numero
sumatorio=0
fact=1
while i>=1:   ## sustituir con variable ?
    sumatorio+=i
    fact *=i
    i-=1


print ("numero=", numero)
print("El sumatorio de", numero, "es", sumatorio)
print("El factorial de", numero ,"es", fact)
```

```
numero= 5
El sumatorio de 5 es 15
El factorial de 5 es 120
```

Programa #2a. Bucles nivel medio

Mostrar una lista de números sucesivos del 1 al 10 y del 10 al 1 con dos bucles while.

```
1, 2,...., 9, 10, 9, ..., 2, 1
```

Solución en la celda siguiente

```
i=1
while i<10 : ##sustituir con variable
    print (i,",")
    i+=1

while i>=1:
    print(i,",")
    i-=1
```

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

Programa #2b. Bucles nivel avanzado

Mostrar una cuenta de números del 1 al 10 y del 10 al 1 **con un solo bucle while**.

```
1, 2,...., 9, 10, 9, ..., 2, 1
```

Solución en la celda siguiente

```
i =1
count =1
while i>=1 and i<=10: ## variable ?
    if 1<=count<=9:
        print (i, ",")
        count+=1
        i+=1
    elif count>9:
        print(i,",")
        count +=1
        i-=1


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

Programa #3. Condiciones anidadas

Crear un sistema experto (principios de la IA) basado en reglas para que en función de los datos de entrada del usuario, diagnóstiquemos la enfermedad de un paciente. Se implementará este problema utilizando condicionales anidados.

Primero se le preguntará por teclado si tiene fiebre el paciente. Si tiene fiebre, habrá que saber si le duele la garganta. Si la respuesta es cierta, se le preguntará otra vez si posee alguna dificultad respiratoria. Si fuese así, tendrá COVID, si no, una infección de garganta. Si no le duele la garganta habrá que preguntarle por la tos. Si tose, tendrá bronquitis y si no, una infección desonocida. Si no tiene fiebre, estará sano como un roble. :)

Entrada del programa

Una sugerencia puede ser esta:

```
"¿Tienes dolor de garganta? (S/N)"
```

Salida del programa

```
El Dr. IA te diagnostica:  COVID
```

Realizar el diagrama de flujo antes de implementar este programa

Solución en la celda siguiente

```
resp_1=input("Tiene fiebre ?")       ## camiar resp ?
if resp_1=="Si":
    resp_2=input("Le duele la garganta ?")
    if resp_2=="Si":
        resp_3=input("Posee alguna dificultad respiratoria")
        if resp_3 =="Si":
            print("Tiene COVID")
        else :
            print("Tiene una infección de garganta")
    else :
        resp_4=input("tose?")
        if resp_4=="Si":
            print("Tiene bronquitis")
        else : print("Tiene infección desonocida")
else : print ("Esta sano como un roble")
```

Programa #4. Cálculo de una función

Dada la siguiente función, calcular los valores de ésta en todos los números enteros del intervalo [-5,5] y mostrarlos por pantalla, como se sugiere a continuación.

$f(x)=x^3+2x^2-3x+7$

Salida del programa

```
( -5 ,  -53 )
( -4 ,  -13 )
( -3 ,  7 )
( -2 ,  13 )
( -1 ,  11 )
( 0 ,  7 )
( 1 ,  7 )
( 2 ,  17 )
( 3 ,  43 )
( 4 ,  91 )
( 5 ,  167 )
```

Solución en la celda siguiente

```
i=-5
def f(x):       ## podemos utilizar funciones
```

```
    return x**3+2*x**2-3*x+7
while i<=5 :
    print("(", i, ",", f(i), ")")
    i+=1

( -5 , -53 )
( -4 , -13 )
( -3 , 7 )
( -2 , 13 )
( -1 , 11 )
( 0 , 7 )
( 1 , 7 )
( 2 , 17 )
( 3 , 43 )
( 4 , 91 )
( 5 , 167 )
```

Programa #5. Conversión de decimal a binario

Realizar un programa que convierta del sistema numérico decimal a binario. Programarlo de tal manera para que funcione para las bases desde 2 hasta 10.

Entrada de datos

Se inicializará las variables del numero a calcular y la base, no se pedirá por teclado.

```
numero_decimal = 8
base = 2
```

Salida

Se mostrará por pantalla el número obtenido de forma inversa, tal y como lo hacemos en papel.

```
0
0
0
1


numero_decimal=752
base=2
quotient=numero_decimal
rest = 0

while quotient!=0:
    rest=quotient%base
    quotient= quotient//base
    print (rest)
```

```
0
0
0
0
1
1
1
1
0
1
```

Programa #5bis. Conversión a hexadecimal (Opcional)

Modifica el programa anterior para que funcione para el Sistema Numérico Hexadecimal

```
numero_decimal=654
base=16
quotient=numero_decimal
rest = 0
hexa=[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, "A", "B", "C", "D", "E", "F"]

while quotient!=0:
    rest=quotient%base
    quotient= quotient//base
    print (hexa[rest])
```

```
E
8
2
```