Fundamentos de Programación

ESTRUCTURA DE CONTROL 2
Semana 06

OBJETIVO DEL LABORATORIO

Aplica las condicionales dobles y anidadas de un algoritmo.

MARCO TEÓRICO

Entendiendo el antecedente y consecuente, se podrán diseñar algoritmos que respondan ante determinada situación que se nos plantee. No solamente de 2 opciones, sino de más escenarios.

El uso de las condiciones implica, que podamos contemplar las diversas opciones que tengamos en un solo algoritmo.

Como fuente de información, se tiene la misma plataforma del Netacad. Tanto para realizar los ejemplos propuestos en la interfaz integrada, así como para repasar lo visto en clase.

RECURSOS

a. **Hardware**
- Pc Pentium IV a superior
- Conexión de red

b. **Software**
- Sistema Operativo Windows XP a superior
- Navegador Chrome o Firefox
- Edube Sandbox de Python desde Netacad

PROCEDIMIENTO

1. SENTENCIAS CONDICIONALES ANIDADAS

   Una sentencia condicional puede contener a su vez otra sentencia anidada.

   Por ejemplo, el programa siguiente "adivina" el número pensado por el usuario:

```
print("Piense un número de 1 a 4.")
print("Conteste S (sí) o N (no) a mis preguntas.")
primera = input("¿El número pensado es mayor que 2? ")
if primera == "S":
    segunda = input("¿El número pensado es mayor que 3? ")
    if segunda == "S":
        print("El número pensado es 4.")
    else:
        print("El número pensado es 3")
else:
    segunda = input("¿El número pensado es mayor que 1? ")
    if segunda == "S":
        print("El número pensado es 2.")
    else:
        print("El número pensado es 1.")
print("¡Hasta la próxima!")
```

   RESULTADO:

```
Piense un número de 1 a 4.
Conteste S (sí) o N (no) a mis preguntas.
¿El número pensado es mayor que 2? S
¿El número pensado es mayor que 3? N
El número pensado es 3
¡Hasta la próxima!
```

2. MÁS DE DOS ALTERNATIVAS: if ... elif ... else ...

   La estructura de control if ... elif ... else ... permite encadenar varias condiciones. elif es una contracción de else if. La orden en Python se escribe así:

```
if condición_1:
    bloque 1
elif condición_2:
    bloque 2
else:
    bloque 3
```

   - Si se cumple la condición 1, se ejecuta el bloque 1
   - Si no se cumple la condición 1 pero sí que se cumple la condición 2, se ejecuta el bloque 2

- Si no se cumplen ni la condición 1 ni la condición 2, se ejecuta el bloque 3.

Esta estructura es equivalente a la siguiente estructura de if ... else ... anidados:

```
if condición_1:
    bloque 1
else:
    if condición_2:
        bloque 2
    else:
        bloque 3
```

Se pueden escribir tantos bloques elif como sean necesarios. El bloque else (que es opcional) se ejecuta si no se cumple ninguna de las condiciones anteriores.

En las estructuras if ... elif ... else ... el orden en que se escriben los casos es importante y, a menudo, se pueden simplificar las condiciones ordenando adecuadamente los casos.

Podemos distinguir dos tipos de situaciones:

A. Cuando los casos son mutuamente excluyentes.

Consideremos un programa que pide la edad y en función del valor recibido da un mensaje diferente. Podemos distinguir, por ejemplo, tres situaciones:

- si el valor es negativo, se trata de un error
- si el valor está entre 0 y 17, se trata de un menor de edad
- si el valor es superior o igual a 18, se trata de un menor de edad

Los casos son mutuamente excluyentes, ya que un valor sólo puede estar en uno de los casos.

- Un posible programa es el siguiente:

```
edad = int(input("¿Cuántos años tiene? "))
if edad < 0:
    print("No se puede tener una edad negativa")
elif edad < 18:
    print("Es usted menor de edad")
else:
    print("Es usted mayor de edad")
```

B. Cuando unos casos incluyen a otros.
Consideremos un programa que pide un valor y nos dice:

- si es múltiplo de dos,
- si es múltiplo de cuatro (y de dos)
- si no es múltiplo de dos

Nota: El valor 0 se considerará múltiplo de 4 y de 2.

Los casos no son mutuamente excluyentes, puesto que los múltiplos de cuatro son también múltiplos de dos.

Opción 1

```
numero = int(input("Escriba un número: "))
if numero % 2 == 0 and numero % 4 != 0:
    print(numero, " es múltiplo de dos")
elif numero % 2 == 0:
    print(numero, " es múltiplo de cuatro y de dos")
else:
    print(numero, " no es múltiplo de dos")
```

Opción 2

```
numero = int(input("Escriba un número: "))
if numero % 4 == 0:
    print(numero, " es múltiplo de cuatro y de dos")
elif numero % 2 == 0:
    print(numero, " es múltiplo de dos")
else:
    print(numero, " no es múltiplo de dos")
```

3. LAUNCH IN EDUBE
   - usando la sentencia if-elif-else;
   - encontrar la correcta implementación de reglas verbalmente definidas;
   - Código de prueba usando entrada y salida de muestra.

Como seguramente sabrá, debido a algunas razones astronómicas, los años pueden ser saltados o comunes. Los primeros tienen una duración de 366 días, mientras que los últimos tienen una duración de 365 días.

Desde la introducción del calendario gregoriano (en 1582), se utiliza la siguiente regla para determinar el tipo de año:

- si el número del año no es divisible por cuatro, es un año común;
- de lo contrario, si el número del año no es divisible por 100, es un año bisiesto;
- de lo contrario, si el número del año no es divisible por 400, es un año común;
- De lo contrario, es un año bisiesto.

Mire el código a continuación: solo lee un número de año y debe completarse con las instrucciones que implementan la prueba que acabamos de describir. El código debe mostrar uno de los dos mensajes posibles, que son el año bisiesto o el año común, según el valor ingresado.

Sería bueno verificar si el año ingresado cae en la era gregoriana y emitir una advertencia de lo contrario.

Pruebe su código utilizando los datos que hemos proporcionado.

```
# Sample Solution

year = int(input("Introduzca un año: "))
if year < 1582:
      print("No dentro del periodo del calendario gregoriano")
else:
      if year % 4 != 0:
            print("año común")
```

```
elif year % 100 != 0:
      print("año bisiesto")
elif year % 400 != 0:
      print("año común")
else:
      print("año bisiesto")
```

CONCLUSIONES Y RECOMENDACIONES DE LA EXPERIENCIA

- Al momento de construir la condición, hay que verificar los intervalos para saber hasta qué rango considera; para evitar un error de lectura.
- Se recomienda usar el Edube SandBox para realizar nuestros ejercicios sin necesidad de instalar algún software adicional.
- Así como en algún momento se solicita cálculos con números enteros, sería factible buscar realizar lo mismo; pero con números reales.

ACTIVIDAD VIRTUAL

1. Observa y analiza el siguiente video: https://www.youtube.com/watch?v=CAnQUgif4q8, y responde las siguientes preguntas:

- ¿Cuáles son las entradas y las salidas del caso observado?
- ¿Cuál es la condición que aparece en el video?
- Realiza ese algoritmo según los datos proporcionados. (Desarróllalo en Python)