

Algoritmos y Estructuras de Datos

Práctica 1

8 de abril de 2021

Índice

1. Estructura secuencial	2
2. Estructura de Decisión Simple (IF)	14
3. Estructura de Iteración o repetición con cantidad conocida de veces (FOR)	29
4. Estructura de Iteración o repetición con cantidad desconocida de veces (WHILE)	34
5. Estructura de Iteración o repetición con cantidad desconocida de veces (REPEAT UNTIL)	37
6. Estructura de Selección Múltiple (CASE)	39

1. Estructura secuencial

1. Modificando el siguiente ejemplo, agregar el cálculo del producto e informar los resultados:

Algoritmo Secuencial1
Leer N1
Leer N2
SUMA \leftarrow N1+N2
Escribir SUMA
FinAlgoritmo

Solución

- NSD:

Algoritmo Secuencial1
Definir N1,N2,SUMA,PRODUCTO Como Entero
Leer N1
Leer N2
SUMA \leftarrow N1+N2
PRODUCTO \leftarrow N1*N2
Escribir SUMA
Escribir PRODUCTO
FinAlgoritmo

La declarativa de variables como se ve en los diagramas aquí presentes es tanto correcta como frecuente. Desafortunadamente la cátedra lo considera incorrecto.

- Python:

```
N1      = int(input())
N2      = int(input())
SUMA    = N1 + N2
PRODUCTO = N2 * N2
print(SUMA)
print(PRODUCTO)
```

2. Describa lo que realiza el diagrama de Chapin. Indicar el valor que se muestra si las variables tendrían los siguientes valores: $A = 10$, $B = 20$ y $C = 2$.

Algoritmo Secuencial2
Leer A
Leer B
Leer C
$R \leftarrow A+B$
$R \leftarrow R/C$
Escribir R
FinAlgoritmo

Solución Muestra en la pantalla el resultado de sumar los primeros dos números ingresados y dividirlos por el tercero.

3. Dados como datos cinco números obtener el promedio de los mismos e informar el resultado.

Solución

- NSD:

Algoritmo Secuencial3
Definir NUMERO1,NUMERO2,NUMERO3,NUMERO4,NUMERO5 Como Real
Definir SUMA,PROMEDIO Como Real
Escribir 'Ingrese el primer numero'
Leer NUMERO1
Escribir 'Ingrese el segundo numero'
Leer NUMERO2
Escribir 'Ingrese el tercer numero'
Leer NUMERO3
Escribir 'Ingrese el cuarto numero'
Leer NUMERO4
Escribir 'Ingrese el quinto numero'
Leer NUMERO5
$SUMA \leftarrow NUMERO1+NUMERO2+NUMERO3+NUMERO4+NUMERO5$
$PROMEDIO \leftarrow SUMA/5$
Escribir 'El promedio es:',PROMEDIO
FinAlgoritmo

- Python:

```

NUMERO1 = int(input("Ingrese el primer numero: "))
NUMERO2 = int(input("Ingrese el segundo numero: "))
NUMERO3 = int(input("Ingrese el tercer numero: "))
NUMERO4 = int(input("Ingrese el cuarto numero: "))
NUMERO5 = int(input("Ingrese el quinto numero: "))
SUMA     = NUMERO1 + NUMERO2 + NUMERO3 + NUMERO4 + NUMERO5
PROMEDIO = SUMA / 5
print("El promedio es:", PROMEDIO)

```

Existen varias formas de resolver cada algoritmo. La siguiente por ejemplo, es equivalente a la anterior pero utiliza una sola variable:

```

PROMEDIO = int(input("Ingrese el primer numero: ")) / 5
PROMEDIO += int(input("Ingrese el segundo numero: ")) / 5
PROMEDIO += int(input("Ingrese el tercer numero: ")) / 5
PROMEDIO += int(input("Ingrese el cuarto numero: ")) / 5
PROMEDIO += int(input("Ingrese el quinto numero: ")) / 5
print("El promedio es:", PROMEDIO)

```

4. Dados como dos números obtener su suma, resta, multiplicación y división.

Solución

- NSD:

Algoritmo Secuencial4
Definir NUMERO1, NUMERO2 Como Real
Definir SUMA, RESTA, MULTIPLICACION, DIVISION Como Real
Escribir 'Ingrese el primer numero'
Leer NUMERO1
Escribir 'Ingrese el segundo numero'
Leer NUMERO2
SUMA \leftarrow NUMERO1+NUMERO2
RESTA \leftarrow NUMERO1-NUMERO2
MULTIPLICACION \leftarrow NUMERO1*NUMERO2
DIVISION \leftarrow NUMERO1/NUMERO2
Escribir 'La suma es:', SUMA
Escribir 'La resta es:', RESTA
Escribir 'La multiplicacion es:', MULTIPLICACION
Escribir 'La division es:', DIVISION
FinAlgoritmo

- Python:

```

NUMERO1 = int(input("Ingrese el primer numero: "))
NUMERO2 = int(input("Ingrese el segundo numero: "))
SUMA     = NUMERO1 + NUMERO2
RESTA    = NUMERO1 - NUMERO2
PRODUCTO = NUMERO1 * NUMERO2
DIVISION = NUMERO1 / NUMERO2
print("La suma es:", SUMA)
print("La resta es:", RESTA)
print("La mutiplicacion es:", PRODUCTO)
print("La division es:", DIVISION)

```

5. Dado un número mostrar el producto de ese número por 6, suponiendo que se cuenta solamente con el operador suma.

Solución

- NSD:

Algoritmo Secuencial5
Definir NUMERO,SEXTUPLE Como Real
Escribir 'Ingrese un numero:'
Leer NUMERO
SEXTUPLE \leftarrow NUMERO+NUMERO+NUMERO+NUMERO+NUMERO+NUMERO
Escribir 'El producto por 6 es: ',SEXTUPLE
FinAlgoritmo

- Python:

```

NUMERO    = int(input("Ingrese un numero: "))
SEXTUPLE  = NUMERO + NUMERO + NUMERO + NUMERO + NUMERO + NUMERO
print("El producto por 6 es:", SEXTUPLE)

```

6. Dadas las medidas de dos ángulos de un triángulo, determinar la medida del tercero e informar el resultado.

Solución

- NSD:

Algoritmo Secuencial6
Definir ANGULO1,ANGULO2,ANGULO3 Como Real
Escribir 'Ingrese el primer angulo:'
Leer ANGULO1
Escribir 'Ingrese el segundo angulo:'
Leer ANGULO2
$ANGULO3 \leftarrow 180 - ANGULO1 - ANGULO2$
Escribir 'El angulo restante es: ',ANGULO3
FinAlgoritmo

- Python:

```
ANGULO1 = int(input("Ingrese el primer angulo: "))
ANGULO2 = int(input("Ingrese el segundo angulo: "))
ANGULO3 = 180 - ANGULO1 - ANGULO2
print("El ángulo restante es:", ANGULO3)
```

7. Dado el valor del lado de un cuadrado, calcular su perímetro y su superficie, e informar los mismos con carteles aclaratorios.

Solución

- NSD:

Algoritmo Secuencial7
Definir LADO,PERIMETRO,SUPERFICIE Como Real
Escribir 'Ingrese el lado:'
Leer LADO
$PERIMETRO \leftarrow LADO * 4$
$SUPERFICIE \leftarrow LADO * LADO$
Escribir 'El perimetro es: ',PERIMETRO
Escribir 'La superficie es: ',SUPERFICIE
FinAlgoritmo

- Python:

```
LADO = int(input("Ingrese el lado: "))
PERIMETRO = LADO * 4
SUPERFICIE = LADO * LADO
print("El perimetro es:", PERIMETRO)
print("La superficie es:", SUPERFICIE)
```

8. Dado un número, mostrar el resultado de dicho número elevado a la octava.

Solución

- NSD:

Algoritmo Secuencial8
Definir N,RESULTADO Como Real
Escribir 'Ingrese el numero:'
Leer N
RESULTADO $\leftarrow N * N * N * N * N * N * N * N$
Escribir 'El resultado es: ',RESULTADO
FinAlgoritmo

- Python:

```
N = int(input("Ingrese el numero: "))
RESULTADO = N * N * N * N * N * N * N * N
print("El resultado es:", RESULTADO)
```

9. Tener en cuenta, que solo se reconocen 4 operaciones básicas. ¿Cual sería el algoritmo si la máquina cuenta además con la operación potencia?

Solución

- NSD:

Algoritmo Secuencial9
Definir N,RESULTADO Como Real
Escribir 'Ingrese el numero:'
Leer N
RESULTADO $\leftarrow N^8$
Escribir 'El resultado es: ',RESULTADO
FinAlgoritmo

- Python:

```
N = int(input("Ingrese el numero: "))
RESULTADO = N ** 8
print("El resultado es:", RESULTADO)
```

10. Dado como dato el importe neto de una factura, calcular el valor correspondiente al IVA (21 %).

Solución

- NSD:

Algoritmo Secuencial10
Definir NETO,BRUTO,IVA Como Real
Escribir 'Ingrese el importe neto:'
Leer NETO
$BRUTO \leftarrow NETO / 1.21$
$IVA \leftarrow NETO - BRUTO$
Escribir 'El IVA es: ',IVA
FinAlgoritmo

- Python:

```
NETO = int(input("Ingrese el importe neto: "))
BRUTO = NETO / 1.21
IVA = NETO - BRUTO
print("El IVA es: ", IVA)
```

11. Calcular el sueldo de un operario conociendo la cantidad de horas que trabajó en el mes y el jornal horario.

Solución

- NSD:

Algoritmo Secuencial11
Definir HORAS,JORNAL,SUELDO Como Real
Escribir 'Ingrese las horas:'
Leer HORAS
Escribir 'Ingrese el jornal horario:'
Leer JORNAL
$SUELDO \leftarrow HORAS * JORNAL$
Escribir 'El sueldo es: ',SUELDO
FinAlgoritmo

- Python:

```
HORAS = int(input("Ingrese las horas: "))
JORNAL = int(input("Ingrese el jornal horario: "))
SUELDO = HORAS * JORNAL
print("El sueldo es: ", SUELDO)
```

- Determinar el número de horas, minutos y segundos que hay en 6250 segundos.

Solución

- NSD:

Algoritmo Secuencial12
Definir SEGUNDOS,MINUTOS,HORAS Como Real
SEGUNDOS \leftarrow 6250
MINUTOS \leftarrow SEGUNDOS/60
HORAS \leftarrow MINUTOS/60
Escribir 'Horas: ',HORAS
Escribir 'Minutos: ',MINUTOS
Escribir 'Segundos: ',SEGUNDOS
FinAlgoritmo

- Python:

```
SEGUNDOS = 6250
MINUTOS = SEGUNDOS / 60
HORAS = MINUTOS / 60
print("Horas: ", HORAS)
print("Minutos: ", MINUTOS)
print("Segundos: ", SEGUNDOS)
```

- Dado el importe bruto de una factura, calcular el resultado de bonificarlo (descuento) con un 4%. Al monto obtenido, calcularle el IVA (21%). Finalmente informar: el importe bruto, el valor de la bonificación, el importe bruto bonificado, el monto correspondiente al IVA y el importe neto resultante.

Solución

- NSD:

Algoritmo Secuencial13
Definir BRUTO,BONIFICACION,SUBTOTAL,IVA,NETO Como Real
Escribir 'Ingrese el importe bruto: '
Leer BRUTO
$BONIFICACION \leftarrow BRUTO * 4 / 100$
$SUBTOTAL \leftarrow BRUTO - BONIFICACION$
$IVA \leftarrow SUBTOTAL * 21 / 100$
$NETO \leftarrow SUBTOTAL + IVA$
Escribir 'Bruto: ',BRUTO
Escribir 'Bonificacion: ',BONIFICACION
Escribir 'Subtotal: ',SUBTOTAL
Escribir 'IVA: ',IVA
Escribir 'Neto: ',NETO
FinAlgoritmo

- Python:

```
BRUTO          = int(input("Ingrese el importe bruto: "))
BONIFICACION   = BRUTO * 4 / 100
SUBTOTAL       = BRUTO - BONIFICACION
IVA            = SUBTOTAL * 21 / 100
NETO           = SUBTOTAL + IVA
print("Bruto:", BRUTO)
print("Bonificacion:", BONIFICACION)
print("Subtotal:", SUBTOTAL)
print("IVA:", IVA)
print("neto:", NETO)
```

14. Calcular cuántos pesos tiene un banco en monedas si dispone de $N1$ monedas de 1 peso, $N2$ de medio peso, $N3$ de un cuarto de peso, $N4$ de 10 centavos y $N5$ de 5 centavos de peso.

Solución

- NSD:

Algoritmo Secuencial14
Definir N1,N2,N3,N4 Como Entero
Definir PESOS Como Real
Escribir 'Ingrese N1:'
Leer N1
Escribir 'Ingrese N2:'
Leer N2
Escribir 'Ingrese N3:'
Leer N3
Escribir 'Ingrese N4:'
Leer N4
$PESOS \leftarrow N1 + N2 * 0.5 + N3 * 0.25 + N4 * 0.1$
Escribir '\$', PESOS
FinAlgoritmo

- Python:

```
N1 = int(input("Ingrese N1: "))
N2 = int(input("Ingrese N1: "))
N3 = int(input("Ingrese N1: "))
N4 = int(input("Ingrese N1: "))
PESOS = N1 + N2 * 0.5 + N3 * 0.25 + N4 * 0.1
print("$", PESOS)
```

15. Ingresar 3 valores en 3 variables X , Y y Z . Se desea obtener una rotación de sus valores, es decir que el contenido de Z pase a X , el contenido de X pase a Y , y el contenido de Y pase a Z . Se debe mostrar las variables X , Y y Z con sus valores originales y mostrar X , Y y Z con los valres luego de la rotación.

Solución

- NSD:

Algoritmo Secuencial15
Definir X,Y,Z,T Como Entero
Escribir 'Ingrese X: '
Leer X
Escribir 'Ingrese Y: '
Leer Y
Escribir 'Ingrese Z: '
Leer Z
Escribir 'X, Y, Z: ',X,' ',Y,' ',Z
$T \leftarrow X$
$X \leftarrow Z$
$Z \leftarrow Y$
$Y \leftarrow T$
Escribir 'X, Y, Z: ',X,' ',Y,' ',Z
FinAlgoritmo

- Python:

```
X = int(input("Ingrese X: "))
Y = int(input("Ingrese Y: "))
Z = int(input("Ingrese Z: "))
print("X, Y, Z:", X, Y, Z)
T = X
X = Z
Z = Y
Y = T
print("X, Y, Z:", X, Y, Z)
```

Python ofrece una sintaxis mas cómoda para el intercambio de variables. El mismo programa puede realizarse de la siguiente manera:

```
X = int(input("Ingrese X: "))
Y = int(input("Ingrese Y: "))
Z = int(input("Ingrese Z: "))
print("X, Y, Z:", X, Y, Z)
X, Y, Z = Z, X, Y
print("X, Y, Z:", X, Y, Z)
```

16. Indicar cómo será la salida luego de realizar lo pedido en el ejercicio anterior, si se ingresa 10 en la variable X , 15 en la variable Y y 20 en la variable Z .

Solución

X: 10

Y: 15

Z: 20

X: 20

Y: 10

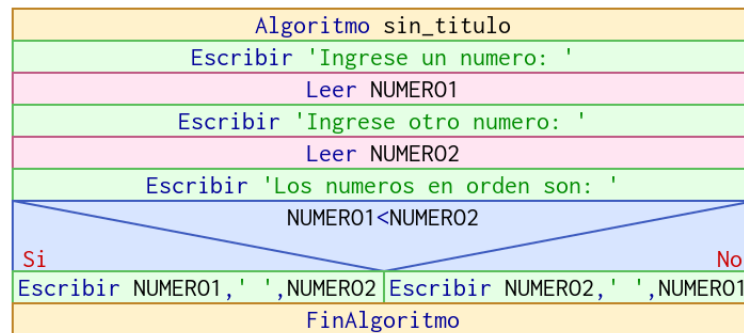
Z: 15

2. Estructura de Decisión Simple (IF)

1. Dados dos números distintos, mostrarlos ordenados en forma creciente.

Solución

- NSD:



- Python:

```

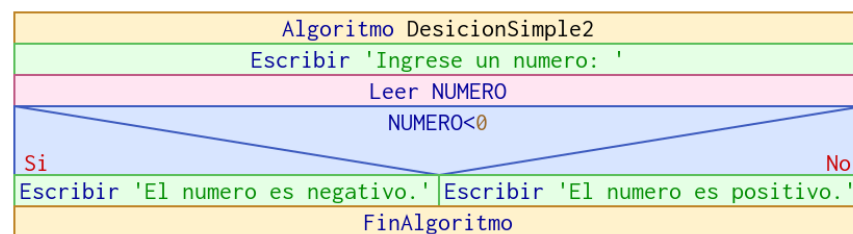
NUMERO1 = int(input("Ingrese un numero: "))
NUMERO2 = int(input("Ingrese un numero: "))
print("Los numeros en orden son: ")
if NUMERO1 < NUMERO2:
    print(NUMERO1, NUMERO2)
else:
    print(NUMERO2, NUMERO1)

```

2. Dado un número determinar si es positivo o negativo.

Solución

- NSD:



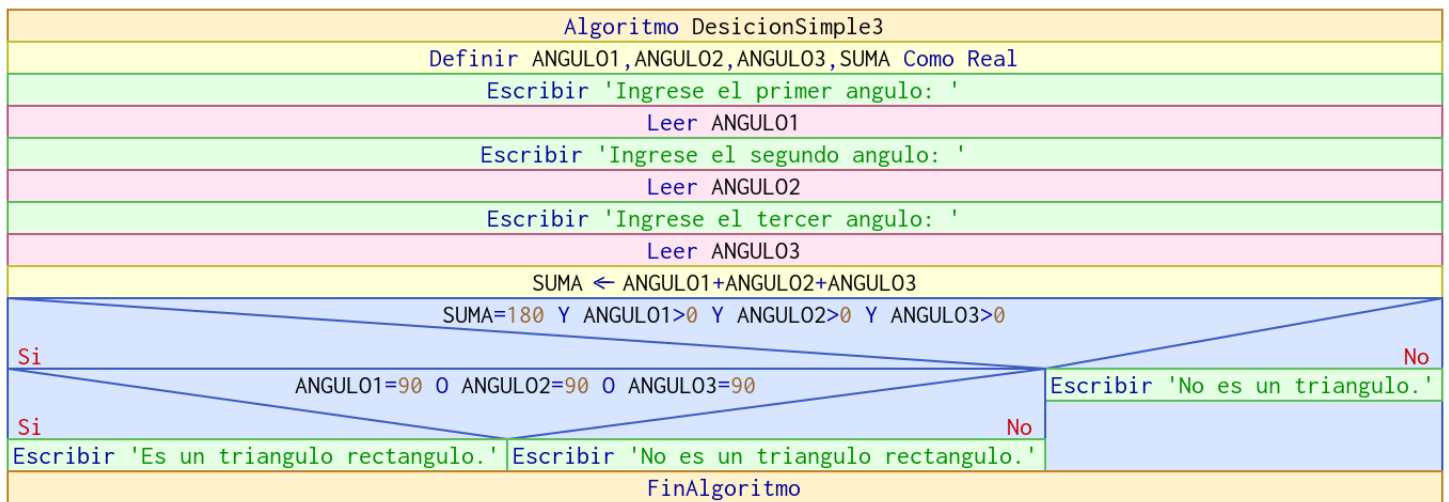
- Python:

```
NUMERO = int(input("Ingrese un numero: "))
if NUMERO < 0:
    print("El numero es negativo.")
else:
    print("El numero es positivo.")
```

3. Dada la medida de cada uno de los tres ángulos de un triángulo determinar e informar mediante un mensaje si pertenecen o no a un triángulo rectángulo

Solución

- NSD:



- Python:

```

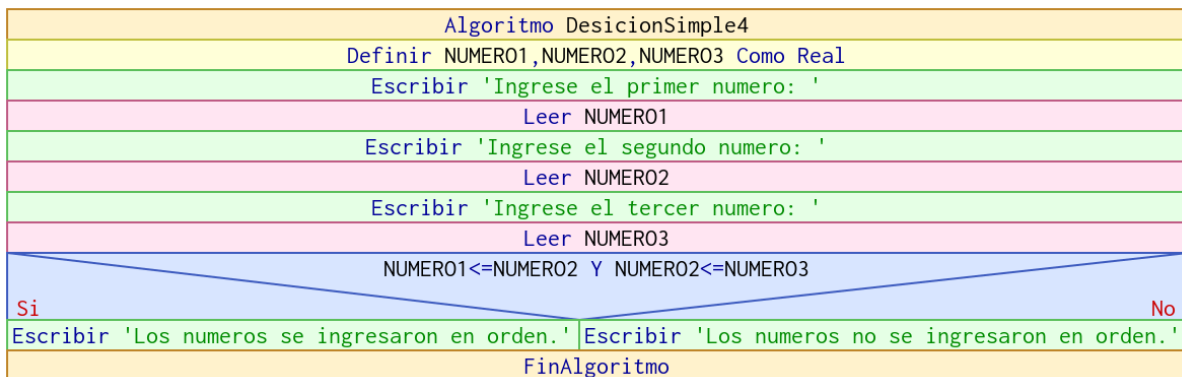
ANGULO1 = int(input("Ingrese el primer angulo: "))
ANGULO2 = int(input("Ingrese el segundo angulo: "))
ANGULO3 = int(input("Ingrese el tercer angulo: "))
SUMA    = ANGULO1 + ANGULO2 + ANGULO3
if SUMA == 180 and ANGULO1 > 0 and ANGULO2 > 0 and ANGULO3 > 0:
    if ANGULO1 == 90 or ANGULO2 == 90 or ANGULO3 == 90:
        print("Es un triangulo rectangulo.")
    else:
        print("No es un triangulo rectangulo.")
else:
    print("No es un triangulo")

```

4. Ingresar tres números enteros distintos. Determinar y mostrar si ingresaron en orden creciente.

Solución

- NSD:



- Python:

```

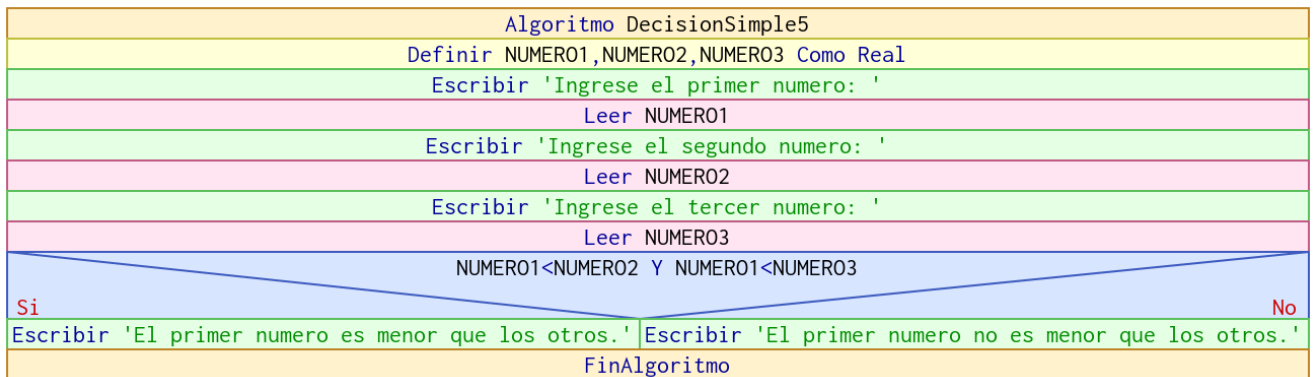
NUMERO1 = int(input("Ingrese el primer numero: "))
NUMERO2 = int(input("Ingrese el segundo numero: "))
NUMERO3 = int(input("Ingrese el tercer numero: "))
if NUMERO1 <= NUMERO2 and NUMERO2 <= NUMERO3:
    print("Los numeros se ingresaron en orden.")
else:
    print("Los numeros no se ingresaron en orden.")

```


5. Determinar si el primero de un conjunto de tres números dados, es menor que los otros dos.

Solución

- NSD:



- Python:

```

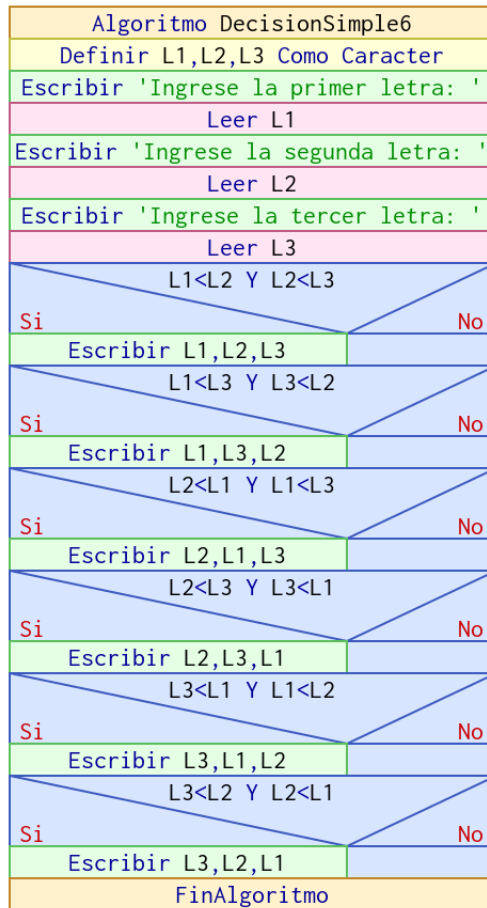
NUMERO1 = int(input("Ingrese el primer numero: "))
NUMERO2 = int(input("Ingrese el segundo numero: "))
NUMERO3 = int(input("Ingrese el tercer numero: "))
if NUMERO1 < NUMERO2 and NUMERO1 < NUMERO3:
    print("El primer numero es menor que los otros.")
else:
    print("El primer numero no es menor que los otros.")

```

6. Ingresar tres letras mayúsculas y mostrarlas ordenadas alfabéticamente.

Solución

- NSD: Existen múltiples formas de resolver este algoritmo, la mas evidente utiliza múltiples estructuras de decisión simple anidadas. La que muestro a continuación utiliza estructuras de decisión simple secuenciales, lo cual considero mas simple.



■ Python:

```

L1 = input("Ingrese la primer letra: ")
L2 = input("Ingrese la segunda letra: ")
L3 = input("Ingrese la tercer letra: ")

if L1 < L2 and L2 < L3:
    print(L1, L2, L3)
if L1 < L3 and L3 < L2:
    print(L1, L3, L2)
if L2 < L1 and L1 < L3:
    print(L2, L1, L3)
if L2 < L3 and L3 < L1:
    print(L2, L3, L1)

```

```

if L3 < L1 and L1 < L2:
    print(L3, L1, L2)
if L3 < L2 and L2 < L1:
    print(L3, L2, L1)

```

Como alternativa al anterior algoritmo, se presenta el siguiente programa que solo utiliza 3 estructuras de decisión simple. Dicho algoritmo se llama «ordenamiento burbuja».

```

L1 = input("Ingrese la primer letra: ")
L2 = input("Ingrese la segunda letra: ")
L3 = input("Ingrese la tercer letra: ")

if L1 > L2:
    L1, L2 = L2, L1
if L2 > L3:
    L2, L3 = L3, L2
if L1 > L2:
    L1, L2 = L2, L1

print(L1, L2, L3)

```

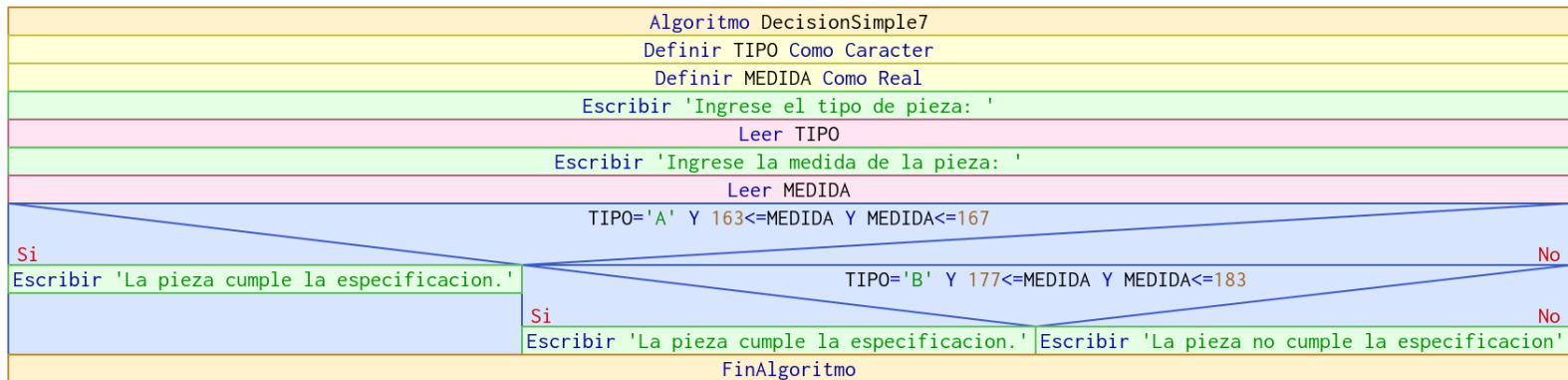
7. Se desea controlar en una fábrica la calidad de dos tipos de piezas que denominaremos A y B.

Se dan como datos el tipo de pieza y su medida en milímetros y se debe indicar si cumple con las especificaciones sabiendo que:

- Las piezas de tipo A deben medir 165 mm y se admite un error de ± 2 mm.
- Las piezas de tipo B deben medir 180 mm y se admite un error de ± 3 mm.

Solución

- NSD:



- Python:

```

TIPO  = input("Ingrese el tipo de pieza: ")
MEDIDA = int(input("Ingrese la medida de la pieza: "))

if TIPO == "A" and 163 <= MEDIDA and MEDIDA <= 167:
    print("La pieza cumple la especificacion.")
elif TIPO == "B" and 177 <= MEDIDA and MEDIDA <= 183:
    print("La pieza cumple la especificacion.")
else:
    print("La pieza no cumple la especificacion.")

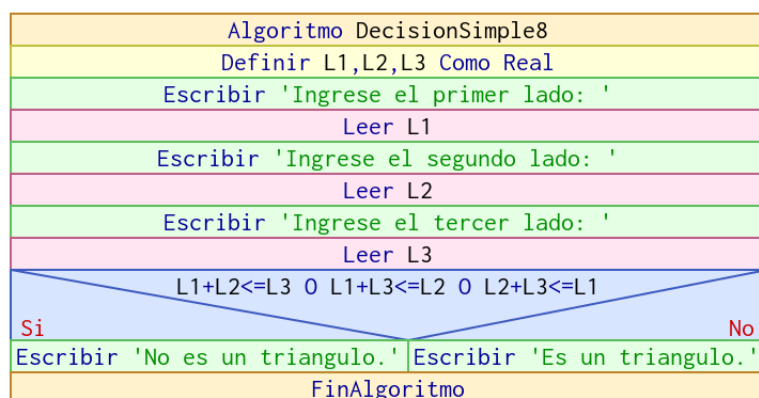
```

8. Se leen tres números positivos. Determinar si son las longitudes de los lados de un triángulo: Recordar que en todo triangulo cada lado es menor o igual que la suma de los otros dos y menor que su diferencia (basta mostrarlo para un lado). En caso afirmativo, informar si el mismo es equilátero (3 lados iguales), isósceles (2 lados iguales) o escaleno (3 lados distintos).

Solución

El enunciado que da origen al problema, y la supuesta solución que brindó la cátedra no determinan en forma correcta algunos triángulos. La solución que se presenta a continuación es correcta.

■ NSD:



■ Python:

```

L1 = int(input("Ingrese el primer lado: "))
L2 = int(input("Ingrese el segundo lado: "))
L3 = int(input("Ingrese el tercer lado: "))

if L1 + L2 <= L3 or L1 + L3 <= L2 or L2 + L3 <= L1:
    print("No es un triangulo.")
else:
    print("Es un triangulo.")

```

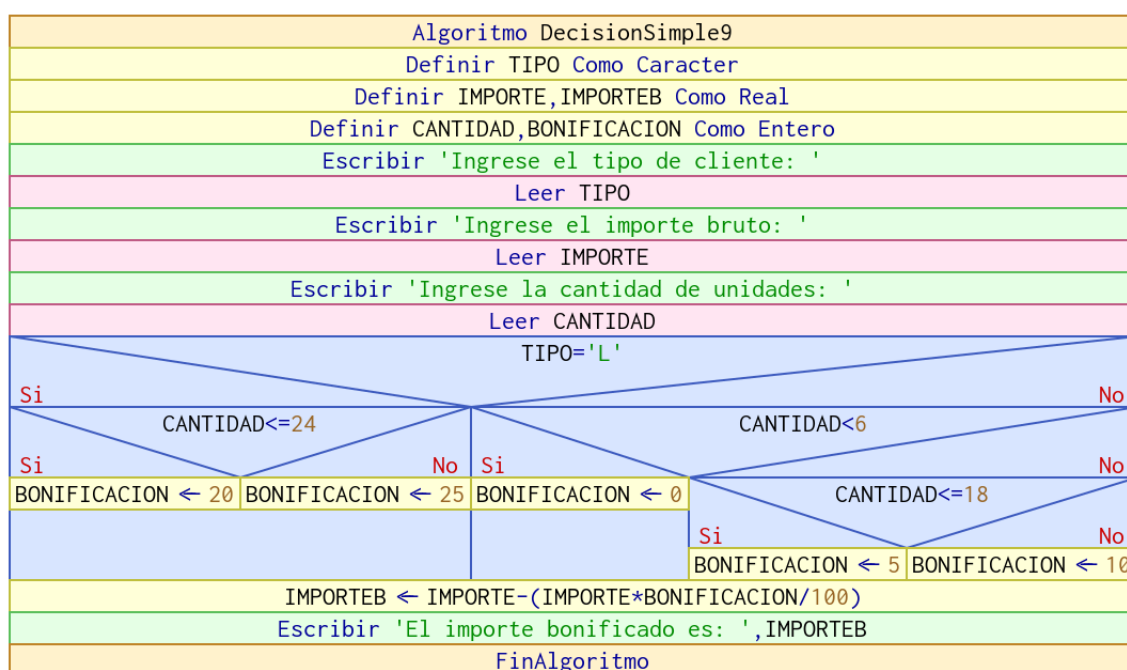
9. Una distribuidora de libros vende a librerías y a particulares. Aplica bonificaciones por cantidad según el siguiente criterio:

- Librerías: hasta 24 unidades, el 20 %; más de 24 unidades, el 25 %.
- Particulares: menos de 6 unidades, nada; desde 6 hasta 18 unidades, el 5 %; y más de 18 unidades, el 10 %.

El tipo de cliente está codificado: «L» para librerías y «P» para particular. Dado el importe bruto total de una compra de libros, el tipo de cliente y la cantidad total pedida por el mismo, determinar el importe bruto bonificado.

Solución

- NSD:



- Python:

```

TIPO      = input("Ingrese el tipo de cliente: ")
IMPORTE   = int(input("Ingrese el importe bruto: "))
CANTIDAD  = int(input("Ingrese la cantidad de unidades: "))

```

```

if TIPO == "L":
    if CANTIDAD <= 24:
        BONIFICACION = 20
    else:
        BONIFICACION = 25
else:
    if CANTIDAD < 6:
        BONIFICACION = 0
    elif CANTIDAD <= 18:
        BONIFICACION = 5
    else:
        BONIFICACION = 10

IMPORTEB = IMPORTE - (IMPORTE * BONIFICACION / 100)
print("El importe bonificado es:", IMPORTEB)

```

11) Dado como dato la cantidad de kilowatios consumidos por un usuario en un mes, calcular el importe a pagar por el mismo teniendo en cuenta que:

- Si la cantidad de kilowatios consumidos es menor ó igual a 200, el precio del kilowatio es de 0.05 pesos.
- Si la cantidad de kilowatios consumidos es mayor que 200 y menor que 1000, el precio del kilowatio es de 0.1 pesos.
- Si la cantidad de kilowatios consumidos es mayor ó igual que 1000, el precio del kilowatio es de 0.15 pesos.

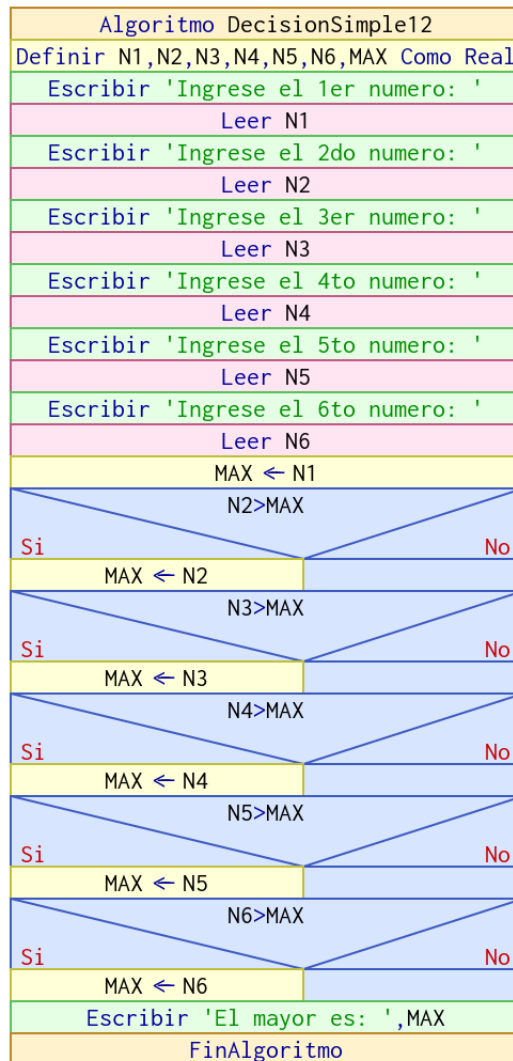
Solución

- NSD: COMPLETAR.
- Python: COMPLETAR.

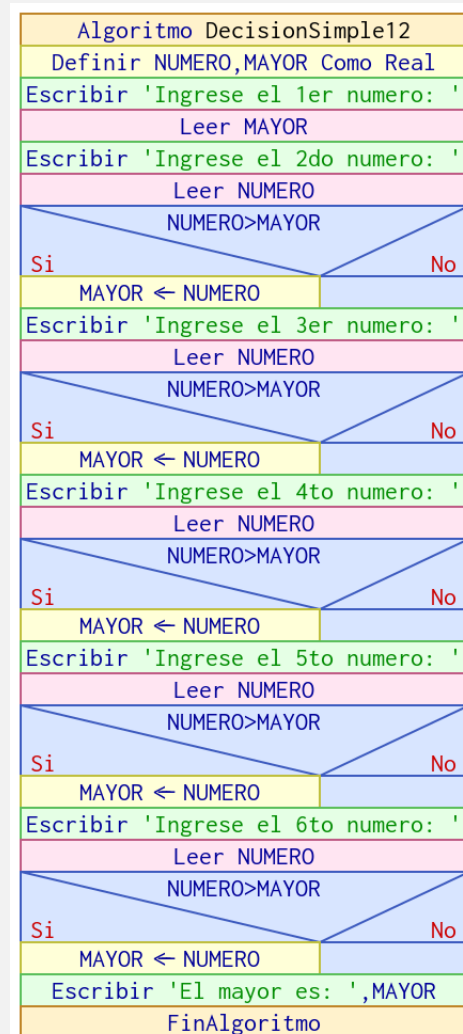
12) Se ingresan seis números positivos diferentes. Al final mostrar un cartel que diga: «El mayor número ingresado fue el xxx».

Solución

■ NSD:



El mismo programa puede realizarse con solo dos variables, si en vez de solicitar toda la entrada y luego computar, computamos la salida a medida que obtenemos la entrada.



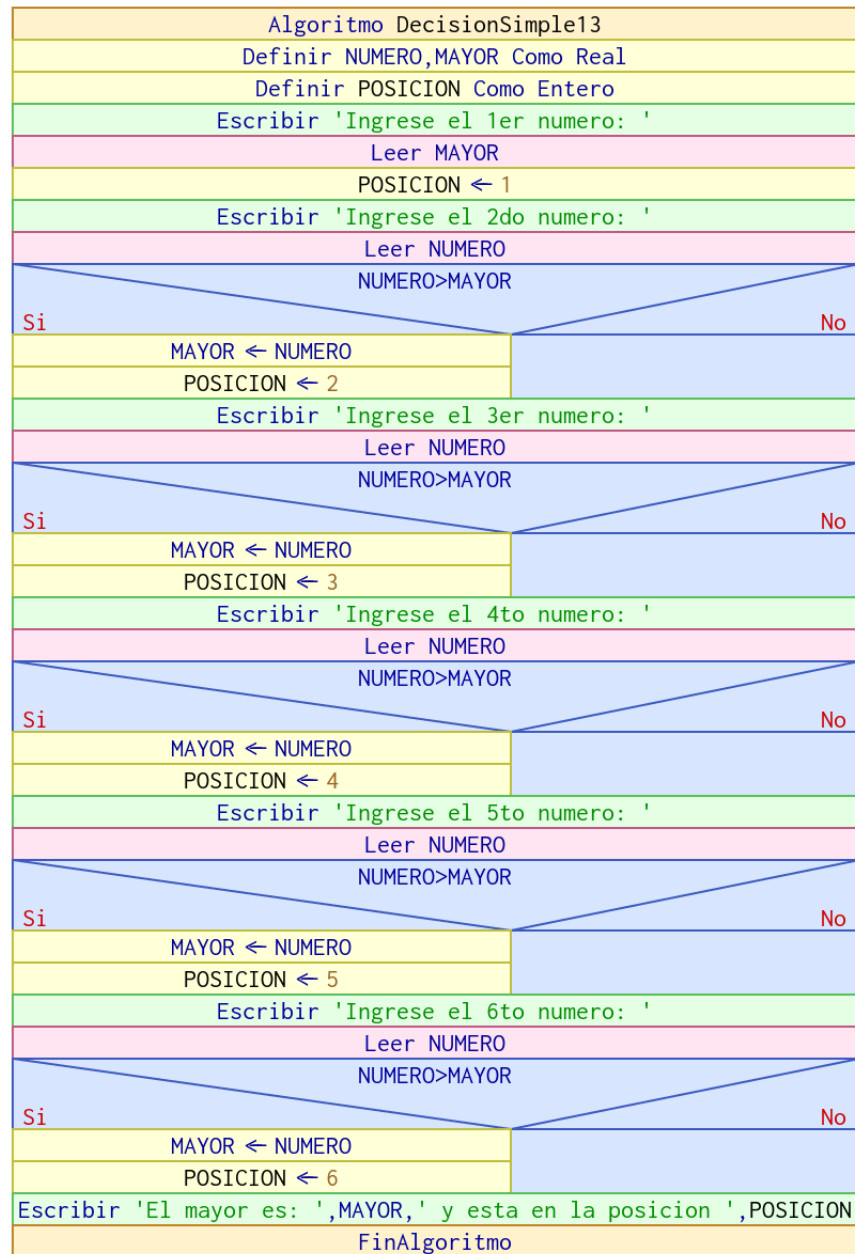
■ Python:

```
MAYOR = int(input("Ingrese el 1er numero: "))
NUMERO = int(input("Ingrese el 2do numero: "))
if NUMERO > MAYOR:
    MAYOR = NUMERO
NUMERO = int(input("Ingrese el 3er numero: "))
if NUMERO > MAYOR:
    MAYOR = NUMERO
NUMERO = int(input("Ingrese el 4to numero: "))
if NUMERO > MAYOR:
    MAYOR = NUMERO
NUMERO = int(input("Ingrese el 5to numero: "))
if NUMERO > MAYOR:
    MAYOR = NUMERO
NUMERO = int(input("Ingrese el 6to numero: "))
if NUMERO > MAYOR:
    MAYOR = NUMERO
print("El mayor es:", MAYOR)
```

- 13) El mismo que el anterior, pero indicar la posición en que entró el mayor.
(Ej. «El mayor fue xx y fue ingresado quinto»).

Solución

■ NSD:



- Python:

```
MAYOR = int(input("Ingrese el 1er numero: "))
POSICION = 1
NUMERO = int(input("Ingrese el 2do numero: "))
if NUMERO > MAYOR:
    MAYOR = NUMERO
    POSICION = 2
NUMERO = int(input("Ingrese el 3er numero: "))
if NUMERO > MAYOR:
    MAYOR = NUMERO
    POSICION = 3
NUMERO = int(input("Ingrese el 4to numero: "))
if NUMERO > MAYOR:
    MAYOR = NUMERO
    POSICION = 4
NUMERO = int(input("Ingrese el 5to numero: "))
if NUMERO > MAYOR:
    MAYOR = NUMERO
    POSICION = 5
NUMERO = int(input("Ingrese el 6to numero: "))
if NUMERO > MAYOR:
    MAYOR = NUMERO
    POSICION = 6
print("El mayor es:", MAYOR, "y la posicion es", POSICION)
```

- 14) Se dan dos valores cualesquiera enteros. Si el primero es mayor que el segundo, restarle al primero un 20%, pero si el segundo es el mayor restarle al segundo un 15%. Con estos nuevos valores, si el primero quedo mayor a 100 y el segundo mayor a 150, se lo consideran valores correctos, de lo contrario es un intervalo de riesgo.

Solución

- NSD: COMPLETAR.
- Python: COMPLETAR.

3. Estructura de Iteración o repetición con cantidad conocida de veces (FOR)

1. COMPLETAR.

Solución

- NSD: COMPLETAR.
- Python: COMPLETAR.

2. COMPLETAR.

Solución

- NSD: COMPLETAR.
- Python: COMPLETAR.

3. COMPLETAR.

Solución

- NSD: COMPLETAR.
- Python: COMPLETAR.

4. COMPLETAR.

Solución

- NSD: COMPLETAR.
- Python: COMPLETAR.

5. COMPLETAR.

Solución

- NSD: COMPLETAR.
- Python: COMPLETAR.

6. COMPLETAR.

Solución

- NSD: COMPLETAR.
- Python: COMPLETAR.

7. COMPLETAR.

Solución

- NSD: COMPLETAR.
- Python: COMPLETAR.

8. COMPLETAR.

Solución

- NSD: COMPLETAR.
- Python: COMPLETAR.

9. COMPLETAR.

Solución

- NSD: COMPLETAR.
- Python: COMPLETAR.

10. COMPLETAR.

Solución

- NSD: COMPLETAR.
- Python: COMPLETAR.

11. COMPLETAR.

Solución

- NSD: COMPLETAR.
- Python: COMPLETAR.

12. COMPLETAR.

Solución

- NSD: COMPLETAR.
- Python: COMPLETAR.

13. COMPLETAR.

Solución

- NSD: COMPLETAR.
- Python: COMPLETAR.

14. COMPLETAR.

Solución

- NSD: COMPLETAR.
- Python: COMPLETAR.

15. COMPLETAR.

Solución

- NSD: COMPLETAR.
- Python: COMPLETAR.

16. COMPLETAR.

Solución

- NSD: COMPLETAR.
- Python: COMPLETAR.

17. COMPLETAR.

Solución

- NSD: COMPLETAR.
- Python: COMPLETAR.

18. COMPLETAR.

Solución

- NSD: COMPLETAR.
- Python: COMPLETAR.

19. COMPLETAR.

Solución

- NSD: COMPLETAR.
- Python: COMPLETAR.

20. COMPLETAR.

Solución

- NSD: COMPLETAR.
- Python: COMPLETAR.

21. COMPLETAR.

Solución

- NSD: COMPLETAR.
- Python: COMPLETAR.

22. COMPLETAR.

Solución

- NSD: COMPLETAR.
- Python: COMPLETAR.

23. COMPLETAR.

Solución

- NSD: COMPLETAR.
- Python: COMPLETAR.

4. Estructura de Iteración o repetición con cantidad desconocida de veces (WHILE)

1. COMPLETAR.

Solución

- NSD: COMPLETAR.
- Python: COMPLETAR.

2. COMPLETAR.

Solución

- NSD: COMPLETAR.
- Python: COMPLETAR.

3. COMPLETAR.

Solución

- NSD: COMPLETAR.
- Python: COMPLETAR.

4. COMPLETAR.

Solución

- NSD: COMPLETAR.
- Python: COMPLETAR.

5. COMPLETAR.

Solución

- NSD: COMPLETAR.
- Python: COMPLETAR.

6. COMPLETAR.

Solución

- NSD: COMPLETAR.
- Python: COMPLETAR.

7. COMPLETAR.

Solución

- NSD: COMPLETAR.
- Python: COMPLETAR.

8. COMPLETAR.

Solución

- NSD: COMPLETAR.
- Python: COMPLETAR.

9. COMPLETAR.

Solución

- NSD: COMPLETAR.
- Python: COMPLETAR.

10. COMPLETAR.

Solución

- NSD: COMPLETAR.
- Python: COMPLETAR.

11. COMPLETAR.

Solución

- NSD: COMPLETAR.
- Python: COMPLETAR.

12. COMPLETAR.

Solución

- NSD: COMPLETAR.
- Python: COMPLETAR.

5. Estructura de Iteración o repetición con cantidad desconocida de veces (REPEAT UNTIL)

1. COMPLETAR.

Solución

- NSD: COMPLETAR.
- Python: COMPLETAR.

2. COMPLETAR.

Solución

- NSD: COMPLETAR.
- Python: COMPLETAR.

3. COMPLETAR.

Solución

- NSD: COMPLETAR.
- Python: COMPLETAR.

4. COMPLETAR.

Solución

- NSD: COMPLETAR.
- Python: COMPLETAR.

5. COMPLETAR.

Solución

- NSD: COMPLETAR.
- Python: COMPLETAR.

6. COMPLETAR.

Solución

- NSD: COMPLETAR.
- Python: COMPLETAR.

7. COMPLETAR.

Solución

- NSD: COMPLETAR.
- Python: COMPLETAR.

8. COMPLETAR.

Solución

- NSD: COMPLETAR.
- Python: COMPLETAR.

9. COMPLETAR.

Solución

- NSD: COMPLETAR.
- Python: COMPLETAR.

10. COMPLETAR.

Solución

- NSD: COMPLETAR.
- Python: COMPLETAR.

6. Estructura de Selección Múltiple (CASE)

1. COMPLETAR.

Solución

- NSD: COMPLETAR.
- Python: COMPLETAR.

2. COMPLETAR.

Solución

- NSD: COMPLETAR.
- Python: COMPLETAR.

3. COMPLETAR.

Solución

- NSD: COMPLETAR.
- Python: COMPLETAR.

4. COMPLETAR.

Solución

- NSD: COMPLETAR.
- Python: COMPLETAR.

5. COMPLETAR.

Solución

- NSD: COMPLETAR.
- Python: COMPLETAR.

6. COMPLETAR.

Solución

- NSD: COMPLETAR.
- Python: COMPLETAR.

7. COMPLETAR.

Solución

- NSD: COMPLETAR.
- Python: COMPLETAR.