

PSEUDOCÓDIGO	DFD	PYTHON
PARA (Variable control = valor inicial; variable control ? valor final, incremento) Instrucciones si condición V Instrucciones si condición V Instrucciones si condición V FIN-PARA	<pre> graph TD Start(()) --> Init[Para contr ← 0 , 10 , 1] Init --> Process[acumul ← acumul] Process --> Decision{Salidas parcia} Decision --> End([Fin (Para)]) </pre>	<pre> acumulador = 0 for control in range(10): print('SALIDA PARCIAL', control) acumulador = acumulador + control print('SALIDAS TIEMPO TOTAL', acumulador) </pre>

PSEUDOCÓDIGO	DFD	PYTHON
Variable control = valor inicial EJM (Variable control ? valor final) Instrucciones si condición V Instrucciones si condición V Instrucciones si condición V FIN-EJM	Representación gráfica de la estructura while: <pre> graph TD Start(()) --> Decision{Condición} Decision -- Verdadero --> Process[Operacion(es)] Process --> Decision Decision -- Falso --> Exit(()) </pre>	<pre> control = 1 final = 10 acumulador = 0 while control <= final: print('SALIDA PARCIAL', control) acumulador = acumulador + control control+=1 print('SALIDAS TIEMPO TOTAL', acumulador) </pre>

Para los siguientes ejercicios, resolver en Python, aplicando listas, bucles, funciones y condicionales donde se requieran:

1.	FOR Programa que permita mostrar un listado, con todos los caracteres imprimibles del teclado con su correspondiente ASCII al frente. Ejm: A->65 B->66.....Z->90; Tenga presente que el código ASCII va desde 1 hasta 255
2.	Realizar un programa que lea los lados de N triángulos. Informar después de cada triángulo si es equilátero (tres lados iguales), isósceles (dos lados iguales) o escaleno (ningún lado igual). Informar al final el total de triángulos de cada tipo. <div style="display: flex; justify-content: space-between;"> <div>TIPOS TRIANGULOS</div> <div>CANTIDAD</div> </div> <div style="display: flex; justify-content: space-between;"> <div>EQUILATERO</div> <div>1</div> </div> <div style="display: flex; justify-content: space-between;"> <div>ESCALENO</div> <div>2</div> </div> <div style="display: flex; justify-content: space-between;"> <div>ISOSCELES</div> <div>0</div> </div>

3.	Leer aleatoriamente un número N de edades entre 18 y 100. Mostrarlas y al final indicar el promedio de las edades. Utilizar la función de biblioteca de aleatorios respectiva.
4.	WHILE , Programa que permita escribir el alfabeto en minúscula o mayúsculas según el usuario lo desee: ejm: a – b- c- dz. o A – B – C – D.....Z
5.	Crear la función myFactorial(n) Hallar el Factorial de un número dado, mostrar la serie y el resultado Ejm: 4! Factorial de 4 es 24 porque $1 * 2 * 3 * 4 = 24$
6.	Programa que permita mostrar la conocida serie de FIBONACCI, crear un función que la invoque y la muestre, por ejemplo: myFibonacci (n). La siguiente es la serie FIBONACCI “0 - 1 – 1 – 2 – 3 – 5 – 8 – 13. – 21” <ul style="list-style-type: none">• Hallar la sumatoria de los todos los números de la serie• La serie finaliza cuando la sumatoria sobrepase el valor de 1000• NO utilizar el método de la biblioteca Python fibbo(), crear su propia función
7.	Programa que permita escribir, la tabla de multiplicar de un número dado (N). Ejm: la tabla del 9 mostrar: $9*1=9$ $9*2=18$ $9*10=90$

8.	<p>Leer N números e indicar si es un número primo o NO. Un número primo es aquel que solamente es divisible por sí mismo y por la unidad, además el único número par primo es el dos.</p> <p>Crear su propia función que le permita enviar un número y la misma devuelva si es primo o NO</p>
9.	<p>De su autoria plantee y resuelva 1 problema que contengan un ciclo PARA</p>
10.	<p>De su autoria plantee y resuelva 1 problema que contengan un ciclo MIENTRAS QUÉ, puede ser el mismo anterior.</p>