

</ Clase 03 fundamentos de la programación

/>

} /> [

Nombre: Alex Mellado
Gamboa

</ ¿Que se vio anteriormente?

Estructura (if, else, elif)

Ejemplos de aplicación de esta estructura

Ciclo while

Ejercicios enfocados en ciclos y listas

Ejercicio de tipo prueba

Aplicando conceptos de estructura de condicionales, y ciclo while



</Consejo de un desarrollador/>

“Ser constante, practicar y
revisar bien la
documentación”

Ejercicios con python

1 0 1 1 0 1 1 0 1 1 0 1 1 0 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 1 0 1 1 0 1 1 0 1 1 1 1 1 0 1

Ejercicio 1

Construir un programa que permita ordenar una lista de números en listas de pares e impares

1 0 1 1 0 1 1 0 1 1 0 1 1 0 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 1 0 1 1 0 1 1 0 1 1 1 1 1 0 1

Ejercicio nº2

Construir un programa que permita obtener la distancia entre dos puntos, además de saber el punto más cercano entre los puntos ingresados

Ejercicio nº3

Construya un programa que permita calcular la raíz digital de un número ingresado por pantalla

1 0 1 1 0 1 1 0 1 1 0 1 1 0 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 1 0 1 1 0 1 1 0 1 1 1 1 1 0 1

Ejercicio nº4 tipo pep

Uno de los algoritmos de ordenamiento más extraños en el mundo de la programación es el algoritmo conocido como random sort o shotgun sort, el cual funciona de la siguiente forma:

- Si la **lista está ordenada**, se **entrega la lista**.
 - Si la **lista no está ordenada**, se **revuelven aleatoriamente sus elementos y se revisa nuevamente**.
- Este algoritmo es altamente ineficiente, pero se desea conocer en la práctica cuán ineficiente es y cómo varía en la medida que la lista a ordenar crece, por lo que se le solicita a usted que construya un programa que ingrese una lista, de números enteros positivos, e indique **cuántas veces debieron reordenarse los elementos** para ordenarse de menor a mayor.