	<b>Trabajo Práctico 02 2024</b>	<b>Clean Code - Resolución de Ejercicios</b>
---	---	-------------------------------------	--

## Caso de Estudio

Realizar un programa que permita cargar los resultados de un examen mediante un menú de opciones, para cada alumno se consideran los siguientes atributos dni, nombre y nota:

- Cargar las notas hasta que se ingrese un dni = 0, validar que las notas estén en el intervalo [0, 10]. Si el valor ingresado NO es válido mostrar un mensaje de error y volver a solicitarlo hasta que el valor sea correcto.
- Mostrar la lista de alumnos con sus respectivas notas, al final del listado mostrar el promedio de las notas registradas.
- Mostrar los alumnos cuyas notas que superan el promedio e indicar cuántas fueron.
- Solicitar una nota y mostrar los alumnos que obtuvieron una nota mayor o igual a dicha nota.
- Calcular la nota máxima y mostrar los alumnos que la obtuvieron.
- Salir

## Ejercicio para la Práctica



### Ejercicio 1

Utilizando listas, diseñar un programa modular que permita gestionar la información de mascotas de una veterinaria. Las funciones solicitadas son:

- Agregar una nueva mascota con los siguientes atributos:
  - Nombre: String, entre 3 y 20 caracteres
  - Especie: (P= Perro, G= Gato, H= Hamster)
  - Edad: Entero, en años, rango [0, 20]
  - Peso: Float, en kilogramos
  - Estado de Salud: (S=Saludable, E=Enfermo, C=Control)
  - Dueño: Lista que contenga la información del dueño (DNI, Nombre, Teléfono, Dirección)
- Buscar una mascota por su nombre y mostrar el detalle de sus datos.
- Actualizar el estado de salud de una mascota en base a la evolución de su tratamiento o chequeo.
- Mostrar un listado de todas las mascotas que pertenecen a un dueño específico.
- Mostrar un listado de todas las mascotas que se encuentran en estado "Enfermo", mostrando su nombre, especie y dueño.

### Consideraciones:

- El programa debe cargar la lista en con al menos 10 mascotas en el momento en que se inicia.
- Realice todas las validaciones necesarias para asegurar que los datos ingresados sean consistentes y correctos.

		<b>Trabajo Práctico 02</b> <b>2024</b>	<b>Clean Code - Resolución de Ejercicios</b>
---	---	---	--

## Ejercicio 2

Dados cinco números enteros positivos, encuentre los valores mínimo y máximo que se pueden calcular sumando exactamente cuatro de los cinco números enteros. Luego imprima los valores mínimo y máximo respectivos como una sola línea de dos números enteros largos separados por espacios.

### Ejemplo

`arr = [1, 3, 5, 7, 9]`

La suma mínima es `1 + 3 + 5 + 7 = 16` y la suma máxima es `3 + 5 + 7 + 9 = 24`

La función imprime `16 24`

### Función descriptiva

Complete la función `miniMaxSum`.

`miniMaxSum` tiene los siguientes parámetros:

- `arr` : una lista de **5** números enteros

### Imprimir

Imprima dos números enteros separados por espacios en una línea: la suma mínima y la suma máxima de **4** de **5** elementos.

### Formato de entrada

Una sola línea de cinco números enteros separados por espacios.

### Restricciones

$1 \leq arr[i] \leq 10^9$

### Formato de salida

Imprima dos números enteros largos separados por espacios que indiquen los valores mínimo y máximo respectivos que se pueden calcular sumando exactamente cuatro de los cinco números enteros. (El resultado puede ser mayor que un número entero de 32 bits).

### Entrada de muestra

`1 2 3 4 5`

### Ejemplo de salida

`10 14`

### Explicación

Los números son **1, 2, 3, 4, y 5** Calcule las siguientes sumas utilizando cuatro de los cinco números enteros:

Suma todo excepto 1, la suma es `2 + 3 + 4 + 5 = 14`



Suma todo excepto 2, la suma es `1 + 3 + 4 + 5 = 13`

Suma todo excepto 3, la suma es `1 + 2 + 4 + 5 = 12`

Suma todo excepto 4, la suma es `1 + 2 + 3 + 5 = 11`

Suma todo excepto 5, la suma es `1 + 2 + 3 + 4 = 10`

Sugerencias: ¡Cuidado con el desbordamiento de números enteros! Utilice un entero de 64 bits.

		<b>Trabajo Práctico 02 2024</b>	<b>Clean Code - Resolución de Ejercicios</b>
---	---	-------------------------------------	--

### Código para implementar la solución

```
import math
import os
import random
import re
import sys



#
# Complete la función 'miniMaxSum' siguiente.
#
#
def miniMaxSum(arr):
    # Escriba aquí su código
    return ''

# Pedir al usuario que ingrese los números separados por espacios
arr = list(map(int, input("Ingrese 5 números separados por espacios: ").split()))

print (miniMaxSum(arr))
```

### Casos de Prueba

Input	Output esperado
1 2 3 4 5	10 14
256741038 623958417 467905213 714532089 938071625	2063136757 2744467344
396285104 573261094 759641832 819230764 364801279	2093989309 2548418794
140638725 436257910 953274816 734065819 362748590	1673711044 2486347135
769082435 210437958 673982045 375809214 380564127	1640793344 2199437821
156873294 719583602 581240736 605827319 895647130	2063524951 2802298787
426980153 354802167 142980735 968217435	1659655705 2484892405

		<b>Trabajo Práctico 02 2024</b>	<b>Clean Code - Resolución de Ejercicios</b>
---	---	-------------------------------------	--

734892650	
942381765 627450398 954173620 583762094 236817490	2390411747 3107767877
539674108 549382170 270968351 746219035 140597628	1500622257 2106243664
254961783 604179258 462517083 967304281 860273491	2181931615 2894274113
501893267 649027153 379408215 452968170 487530619	1821800271 2091419209
140537896 243908675 670291834 923018467 520718469	1575456874 2357937445
793810624 895642170 685903712 623789054 468592370	2572095760 2999145560
5 5 5 5 5	20 20
7 69 2 221 8974	299 9271