**ESCUELA POLITÉCNICA NACIONAL**

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

**ALGORITMOS Y ESTRUCTURAS DE DATOS (TSDS)**

|  |  |
| --- | --- |
|  | |
| ASIGNATURA: | ALGORITMOS Y ESTRUCTURAS DE DATOS |
| PROFESOR: | Ing. Lorena Chulde MSc. |
| PERÍODO ACADÉMICO: | 2023-B |
|  | |

**TAREA**

**Grupal**

|  |
| --- |
|  |
| **TÍTULO:**  **ARREGLOS**  **Nombres de los estudiantes:**  **Guerra Lovato Josué**  **Pérez Orosco Carlos**  **Soria Ansa Richard** |
|  |
|  |
| Arreglos en Python: Introducción al uso de vectores y matrices (Numpy) -  YouTube |
|  |

**2023-B**

**PROPÓSITO DE LA TAREA**

Reutilizar el código mediante funciones para una programación óptima.

**OBJETIVO GENERAL**

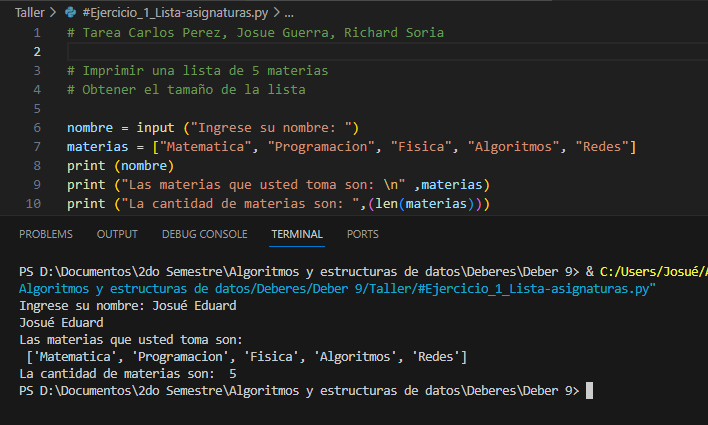
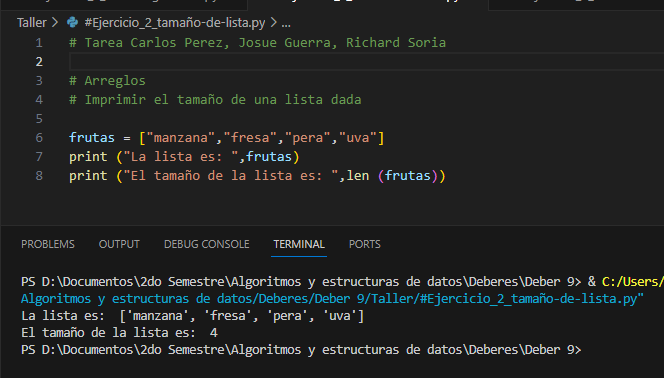
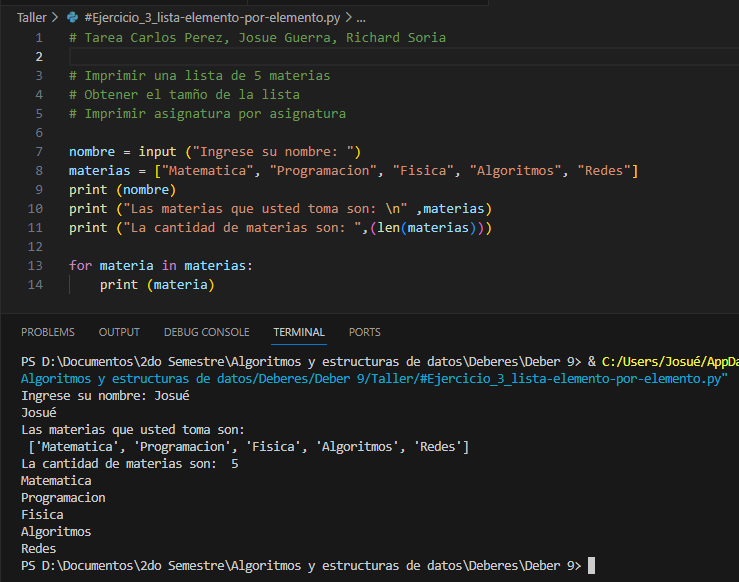
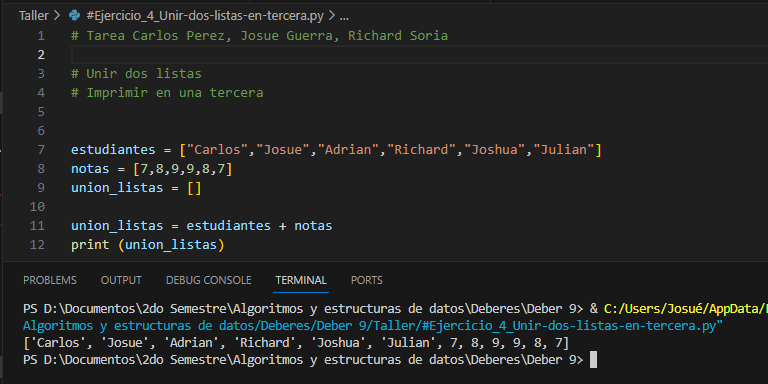
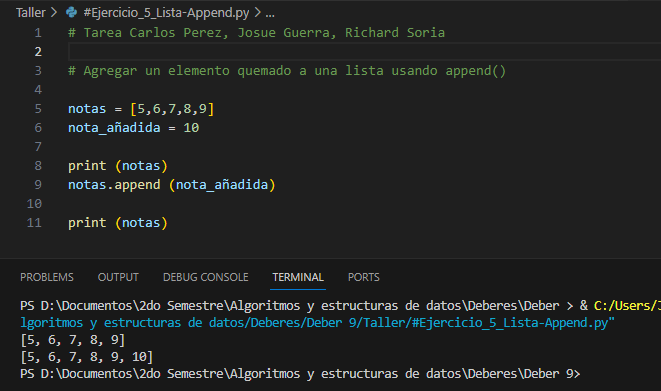
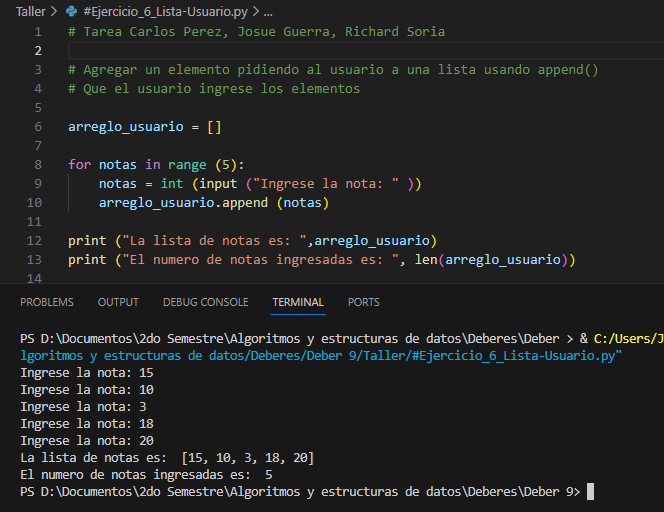
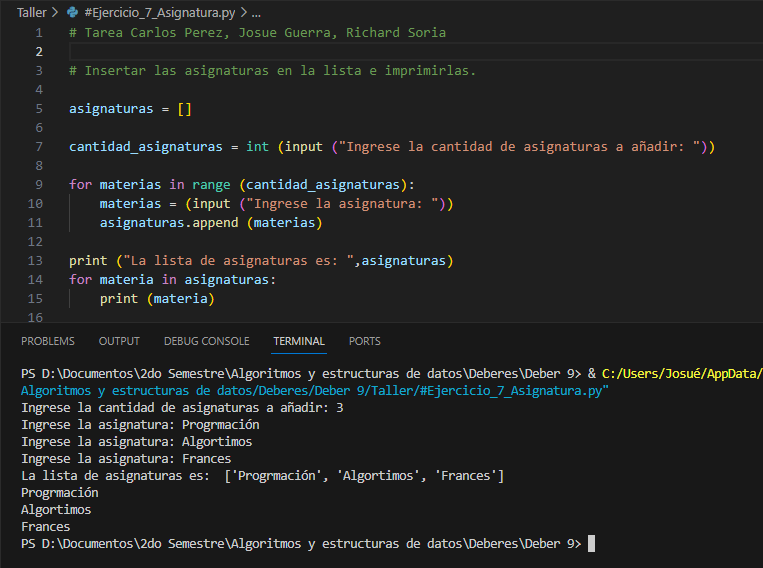
Se tienen como objetivo profundizar en la capacidad de los arreglos para almacenar y manipular datos eficientemente, concentrándose en contextos específicos como la visualización de datos, el procesamiento de imágenes y la implementación de algoritmos avanzados de análisis numérico. Para evitar cualquier riesgo de similitud en sus trabajos, planea abordar este objetivo mediante el estudio autónomo, consultando fuentes académicas, participando activamente en foros especializados y, sobre todo, aplicando los conocimientos adquiridos en proyectos prácticos originales.

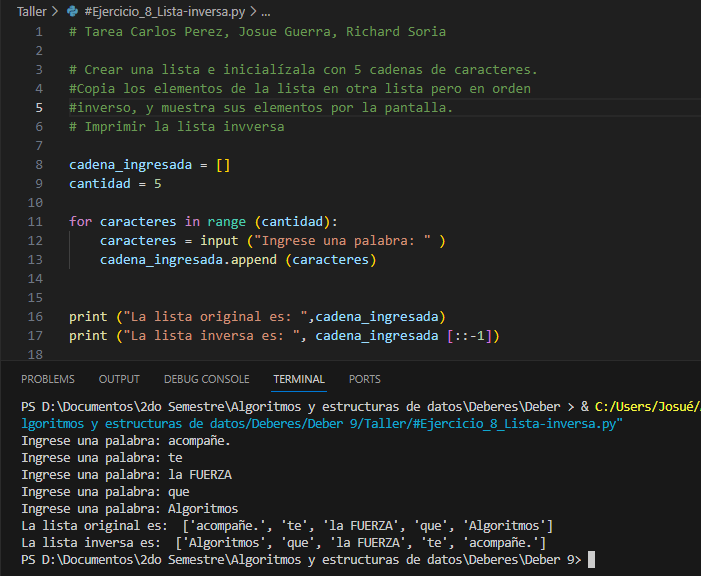
**OBJETIVOS ESPECÍFICOS**

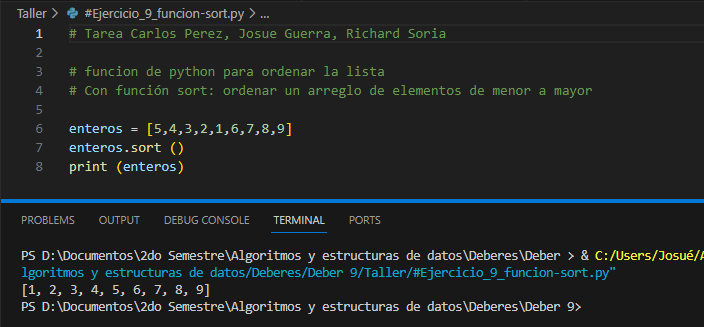
* Desarrollar habilidades avanzadas en la manipulación de arreglos.
* Explorar la aplicación de arreglos en la visualización de datos
* Profundizar en el uso de arreglos para el procesamiento de imágenes

**Parte I**

**TALLER:**

1. Programa que imprime las asignaturas definidas en una lista.
2. Imprimir el tamaño de una lista dada.
3. Programa que imprime las asignaturas con for (elemento por elemento
4. Unir dos listas y mostrar en una tercera lista
5. Agregar un elemento quemado a una lista usando append()
6. Agregar un elemento pidiendo al usuario a una lista usando append()
7. Insertar las asignaturas en la lista e imprimirlas.
8. Crear una lista e inicialízala con 5 cadenas de caracteres. Copia los elementos de la lista en otra lista, pero en orden inverso, y muestra sus elementos por la pantalla.



1. **Con función sort:** ordenar un arreglo de elementos de menor a mayor
2. Crear una fn para recorrer una lista.

**Método burbuja:** Crear una función para ordenar elementos de un array de menor a mayor.

Cree una función llamada ordenar() (estudiar para la lección)

**def ordenar(array):**

**tamanio = len(array)**

**#for para recorrer las posiciones de 0 a 5**

**for i in range(0,tamanio-1):**

**#con estbucle comparamos**

**for j in range(0,tamanio-1):**

**if array[j] > array[j+1]:**

**aux = array[j]**

**array[j] = array[j+1]**

**array[j+1] = aux**

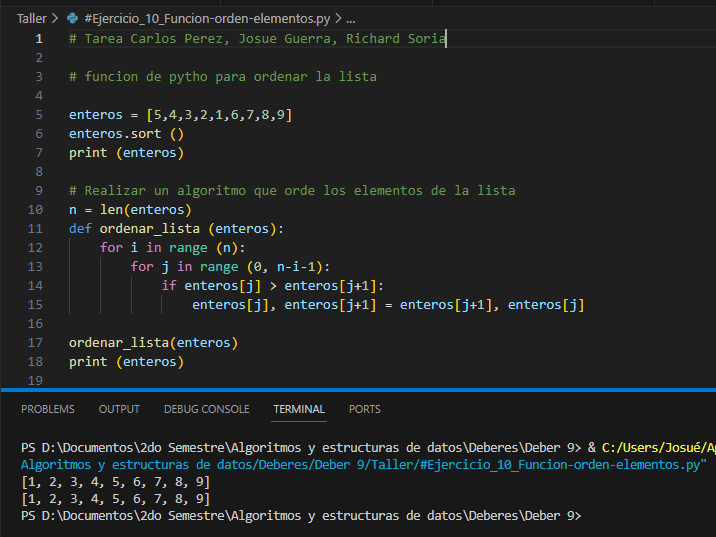
**return array**

**numeros = [6,3,8,2,7]**

**print("la lista original es: ", numeros)**

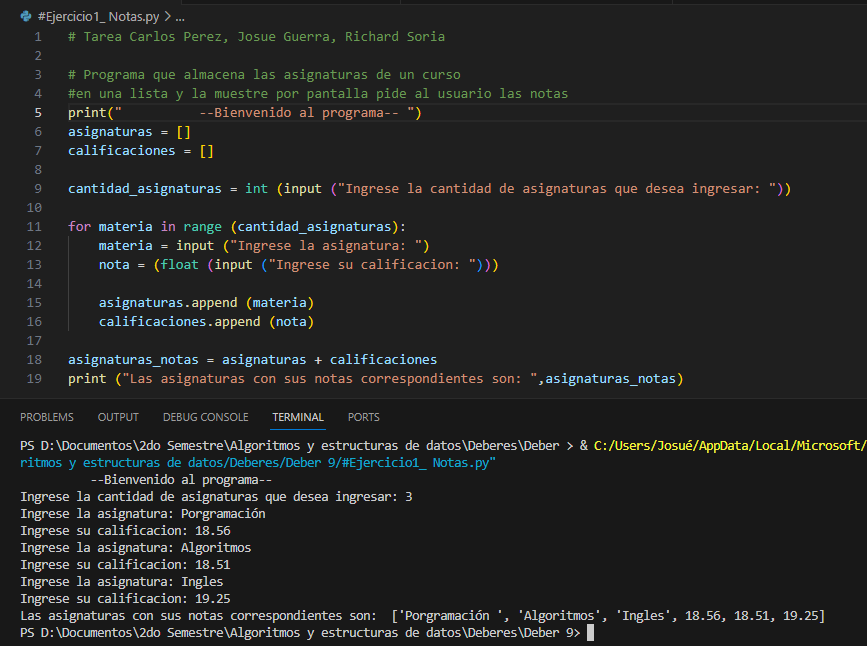
**#ordenar(numeros)**

**print("la lista ordenada es: ", ordenar(numeros))**

****

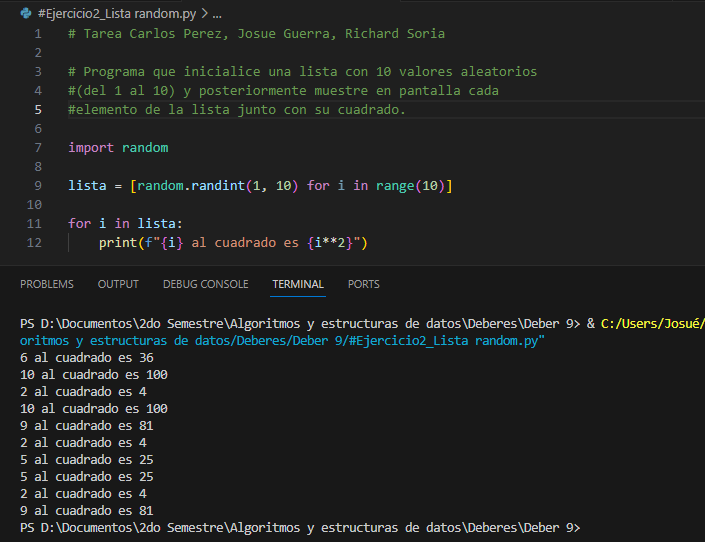
**Parte II**

**TAREA:**

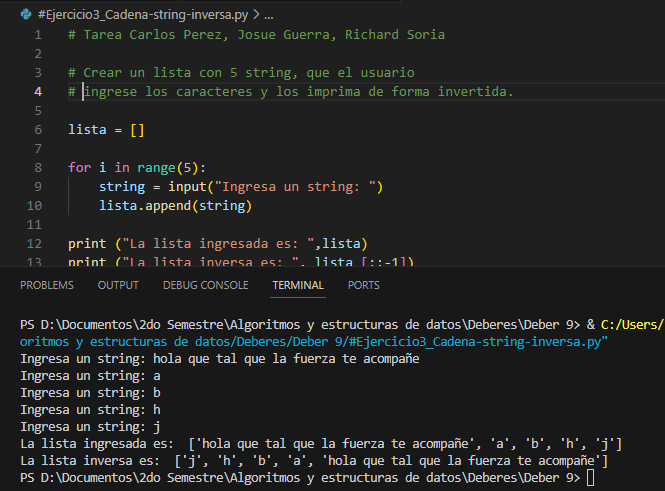
1. Programa que almacena las asignaturas de un curso en una lista y la muestre por pantalla pide al usuario las notas

* Dentro del programa se crea dos arreglos vacíos de inicio para asignar tanto lo que son las asignaturas como las calificaciones, de ahí se solicita para que ingrese la cantidad que el usuario desea de asignaturas, dependiendo de eso se guardara dentro del arreglo pero solo la cantidad escrita, en la ejecución se puede ver como el arreglo mismo guarda lo que son los nombres de las asignaturas, y también las notas
* Tomando en cuenta que nosotros no quemamos el número si no es lo que el usuario decide poner dentro del programa, por ende el arreglo actuara de otra manera si se quemara el número inicialmente.

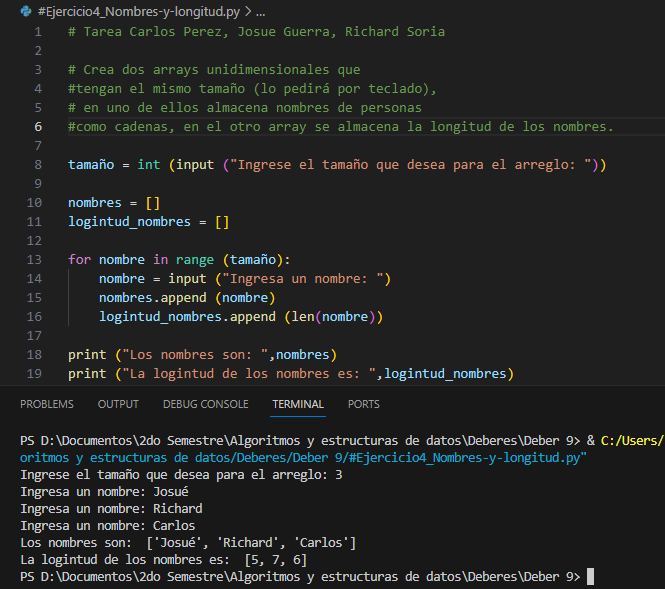
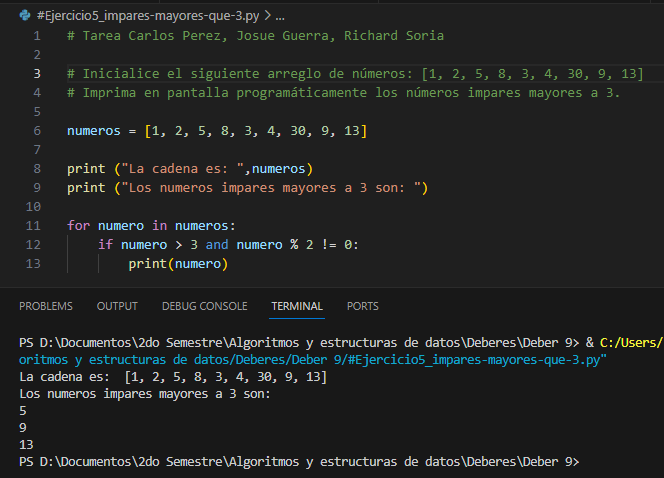
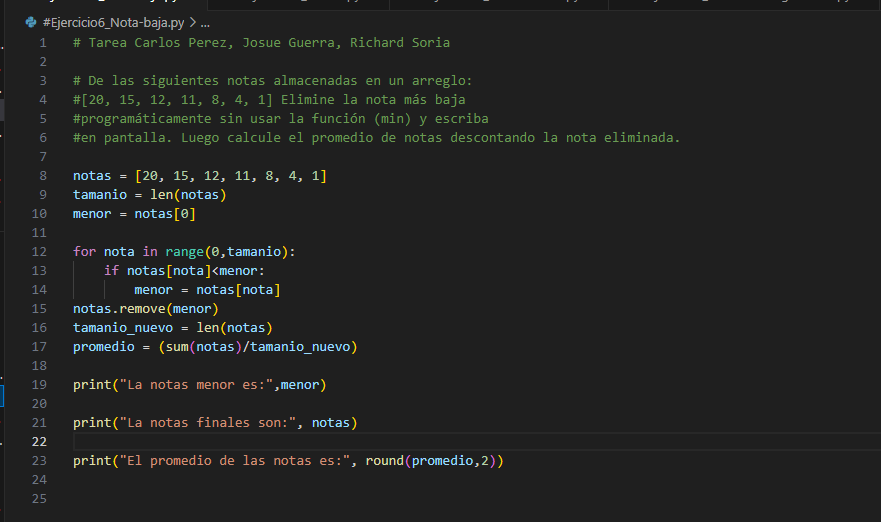
1. Programa que inicialice una lista con 10 valores aleatorios (del 1 al 10) y posteriormente muestre en pantalla cada elemento de la lista junto con su cuadrado.

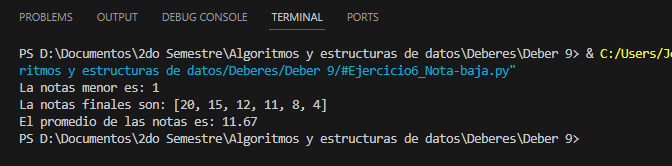


Al tener una lista quemada y en este caso con números random generados al azar en ocasiones el arreglo podrá ser más fácil programado como se muestra a continuación, por lo tanto solo se hace el cuadrado de un n número que se genera automáticamente y se presenta en pantalla.

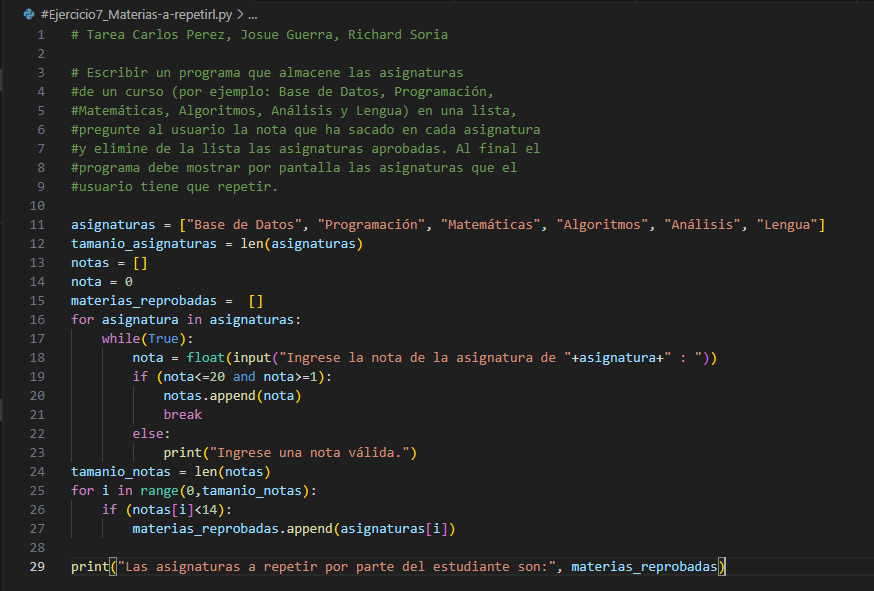
1. Crear una lista con 5 string, que el usuario ingrese los caracteres y los imprima de forma invertida.

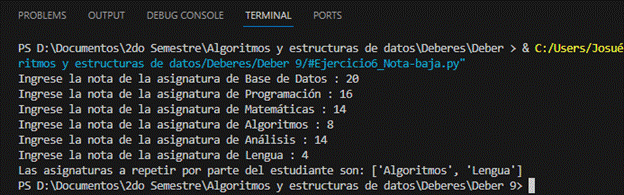
El string se lo puede considerar tanto como una frase como una sola palabra o letra, entonces lo que se hace en este programa es pedir al usuario 5 string y los mismos ya nos los muestra en orden si no desde abajo hacia arriba o inversamente.

1. Crea dos arrays unidimensionales que tengan el mismo tamaño (lo pedirá por teclado), en uno de ellos almacena nombres de personas como cadenas, en el otro array se almacena la longitud de los nombres.
2. Inicialice el siguiente arreglo de números: [1, 2, 5, 8, 3, 4, 30, 9, 13]  
   Imprima en pantalla programáticamente los números impares mayores a 3.
3. De las siguientes notas almacenadas en un arreglo: [20, 15, 12, 11, 8, 4, 1]  
   Elimine la nota más baja programáticamente sin usar la función (min) y escriba en pantalla. Luego calcule el promedio de notas descontando la nota eliminada.

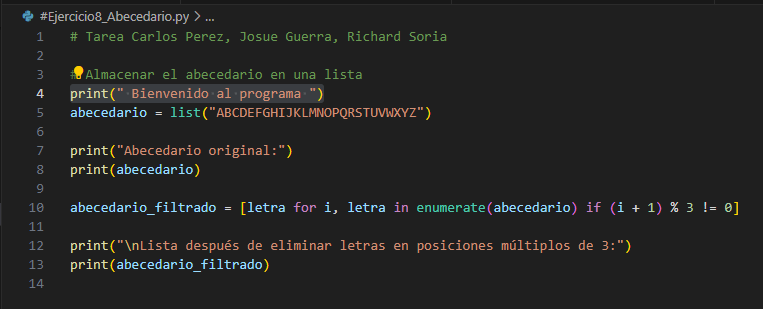
Ejecución del programa:

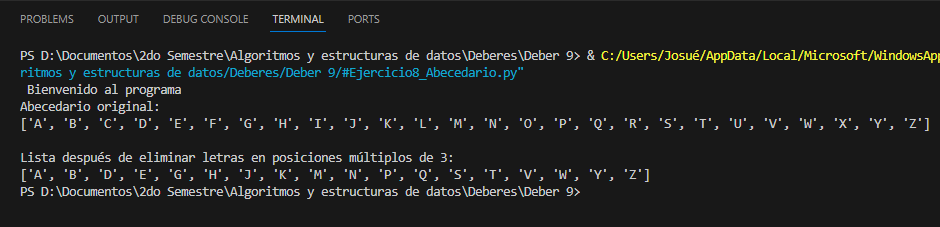
La lista que se ubica dentro del código ya viene predefinida por el problema en si entonces lo que se hace es comprara cual es menor quitar la nota como tal y eso sacar el promedio total.

1. Escribir un programa que almacene las asignaturas de un curso (por ejemplo: Base de Datos, Programación, Matemáticas, Algoritmos, Análisis y Lengua) en una lista, pregunte al usuario la nota que ha sacado en cada asignatura y elimine de la lista las asignaturas aprobadas. Al final el programa debe mostrar por pantalla las asignaturas que el usuario tiene que repetir.

Ejecución del programa:

La nota para pasar se establece en 20 por ende el estudiante al ingresar una nota tendría que tomar en cuenta esta característica, además cabe recalcar que la nota mínima para pasar es de 14 así que si se inserta una menor a esta el arreglo lo detectara y por consiguiente se añadirá a la lista de materias a repetir que se muestra al final de toda la ejecución del programa.

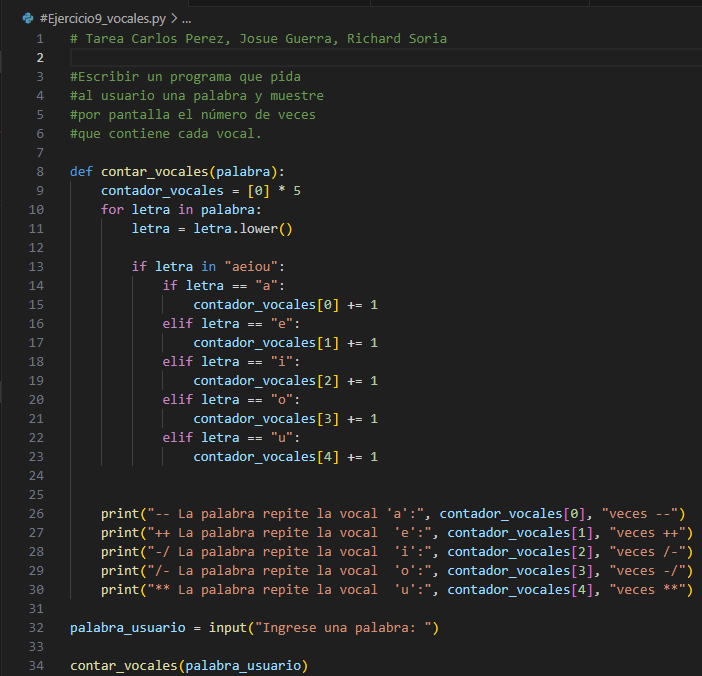
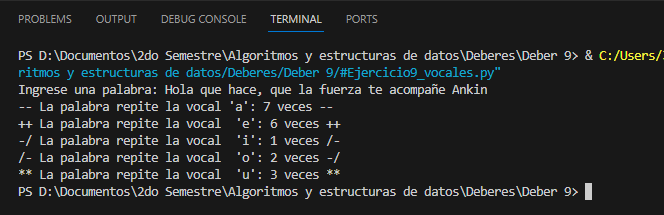
1. Escribir un programa que almacene el abecedario en una lista, elimine de la lista las letras que ocupen posiciones múltiplos de 3, y muestre por pantalla la lista resultante.

Ejecución del programa:

Dentro del código muestra el abecedario inicial y como el problema pide que se elimine cada letra que sea múltiplo de 3 entonces den el arreglo, se lo manda a que detecte la posición de cada letra y según eso vaya eliminando, con eso imprime en pantalla el nuevo abecedario el cual contienen las letras cuyas posiciones no son múltiplos de 3

1. Escribir un programa que pida al usuario una palabra y muestre por pantalla el número de veces que contiene cada vocal.

El código define las 5 vocales del abecedario que ya se las conoce dentro de un elif para que vaya contando según la palabra ingresada, añadiéndolas al contador como tal y eso se va acumulando hasta el final lo que muestra al usuario la cantidad de vocales dentro de la palabra ingresada.

Ejecución

1. Una vez conocidas las 32 selecciones que participarán del próximo **mundial de fútbol** se necesita realizar el sorteo entre las 8 series o grupos de competencia. Las selecciones se encuentran numeradas del 1 al 32, las mejores han sido pre asignadas como “cabeza de serie”; una por cada grupo y no se sorteará su ubicación en la serie. Las selecciones restantes se sortearán la ubicación en cada serie (grupo) para completar los cuatro participantes por serie.

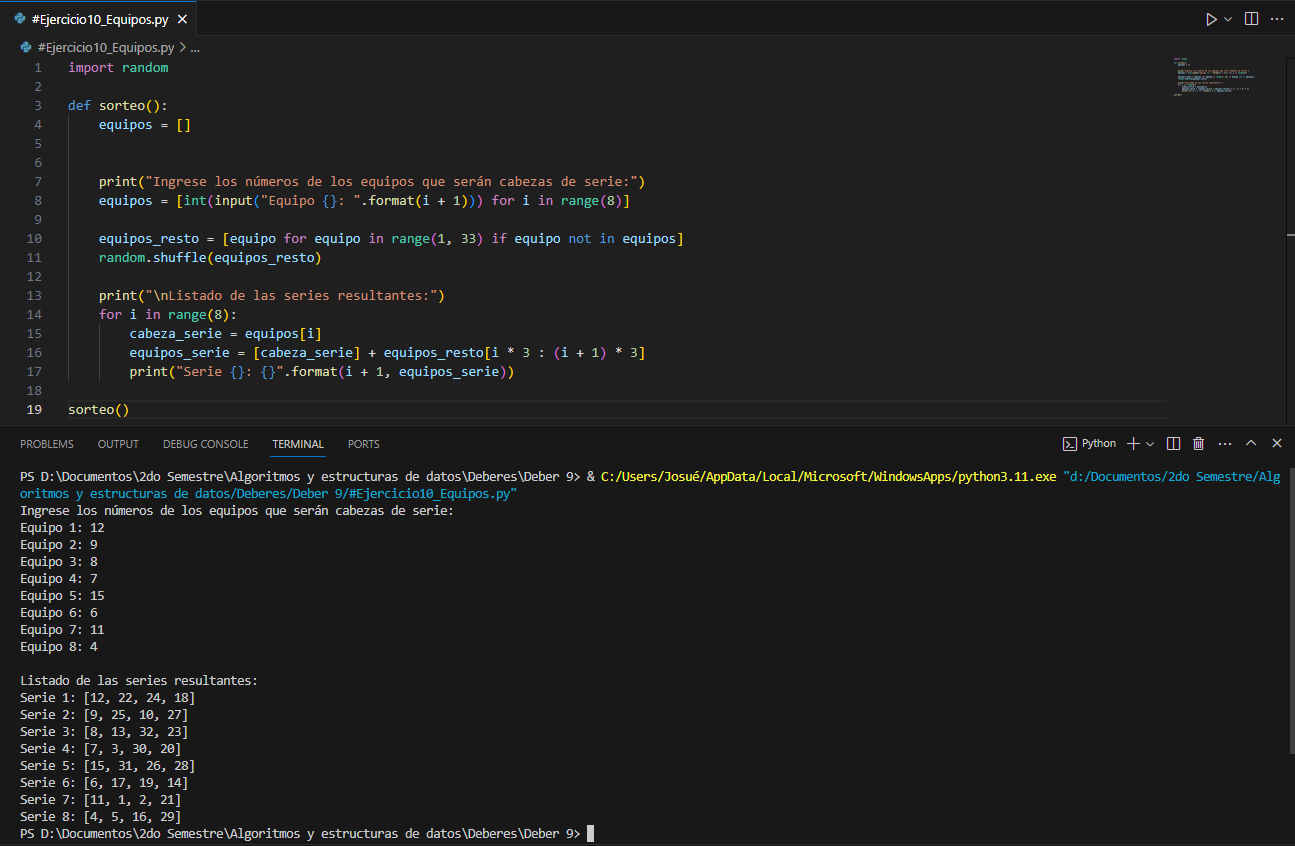
Imagen que contiene Gráfico

Descripción generada automáticamente

El sorteo de serie (luego de copiar los cabezas de grupo) ser realizará en un vector como el mostrado:

Interfaz de usuario gráfica, Aplicación, Tabla

Descripción generada automáticamente

Elabore un algoritmo que solicite cuáles son los 8 equipos que serán cabezas de serie, asigne aleatoriamente (y sin repeticiones) los 24 equipos restantes, al final muestre el listado de las series resultantes.

El algoritmo incluye una generación de números hasta el 32 una vez el usuario inserte los 8 primeros grupos cabezas automáticamente dará el resultado de la serie junto con los números random no repetidos dentro de is con los que se enfrentara.

**ENTREGABLES:**

* Una vez culminada tu tarea, capturar las pantallas de la ejecución del problema con tus datos y súbela en el apartado del aula virtual “S10-Tarea2
* Subir los ejercicios al git o al drive y entrega la url de los archivos .py o, a su vez, entregue el archivo.
* Recordar que el nombre del archivo deberá ser: S11\_Deber2\_Algoritmos\_2023B\_NApellido(de todos los integrantes)

**RECURSOS NECESARIOS**

* Acceso a Internet.
* Imaginación.
* VSC

**RECOMENDACIONES**

Se aconseja abordar el estudio de arreglos en Python con un enfoque integral y auténtico. Para evitar cualquier indicio de similitud, es fundamental explorar diversas fuentes de aprendizaje, como documentación oficial, tutoriales especializados y libros académicos. La participación activa en plataformas de aprendizaje en línea, donde se puedan resolver dudas específicas y discutir conceptos, también puede ser beneficioso.

Se recomienda aplicar los conocimientos adquiridos en proyectos prácticos, asegurándose de comprender cada línea de código y adaptándola a contextos propios. Este enfoque no solo fortalecerá las habilidades, sino que también garantizará la originalidad en cualquier trabajo relacionado con arreglos en Python.

**CONCLUSIONES**

En resumen, la comprensión profunda y la aplicación práctica de arreglos en Python son esenciales para cualquier estudiante. Al adoptar un enfoque genuino en el aprendizaje, evitando simplemente copiar y pegar códigos, se construye un conocimiento sólido y se minimiza el riesgo de similitud en trabajos académicos. La combinación de diversidad en las fuentes de información, participación activa en la comunidad educativa y la creación de proyectos originales no solo cumple con estándares éticos, sino que también contribuye a un aprendizaje más efectivo y duradero.

**ENLACE GITHUB(Dentro del enlace se encuentran los códigos escritos):**

<https://github.com/JosueGuerra2023B/Estructuras-Datos2023B/tree/master/Deberes/Deber%209>

**BIBLIOGRAFÍA**

Los recursos usados dentro del deber son los que nos dio la ingeniera en clase, solo se aplico lo aprendido en los diferentes ejercicios.