**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 – PARÁMETROS POR REFERENCIA Y VALOR - MATRICES**  **Nombres de los estudiantes:**  Incluir el nombre de todos los estudiantes del grupo |
|  |
|  |
| 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**

Facilitar la modularidad y reutilización de código, permitiendo a los programadores dividir programas en bloques más pequeños y manejables, tomando en cuenta que, al utilizar funciones, podemos reutilizar código existente en diferentes partes de un programa, evitando así la repetición innecesaria de líneas de código. Además, las funciones nos permiten encapsular lógica específica en un solo lugar, lo que facilita la depuración y el mantenimiento del código a largo plazo.

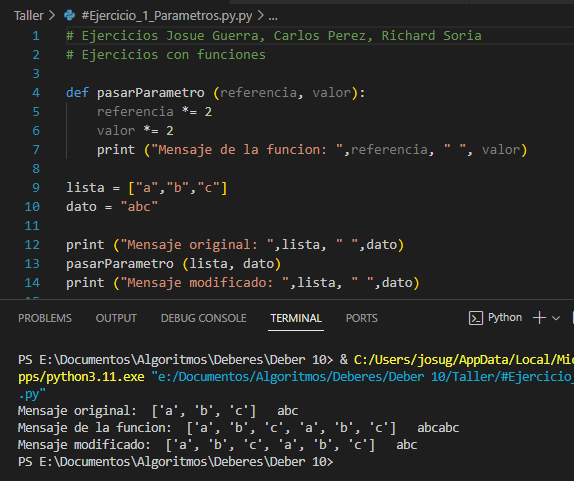
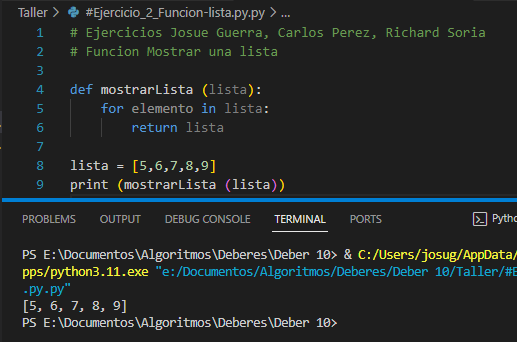
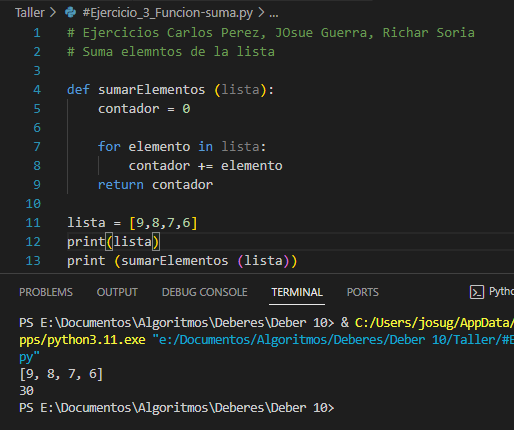
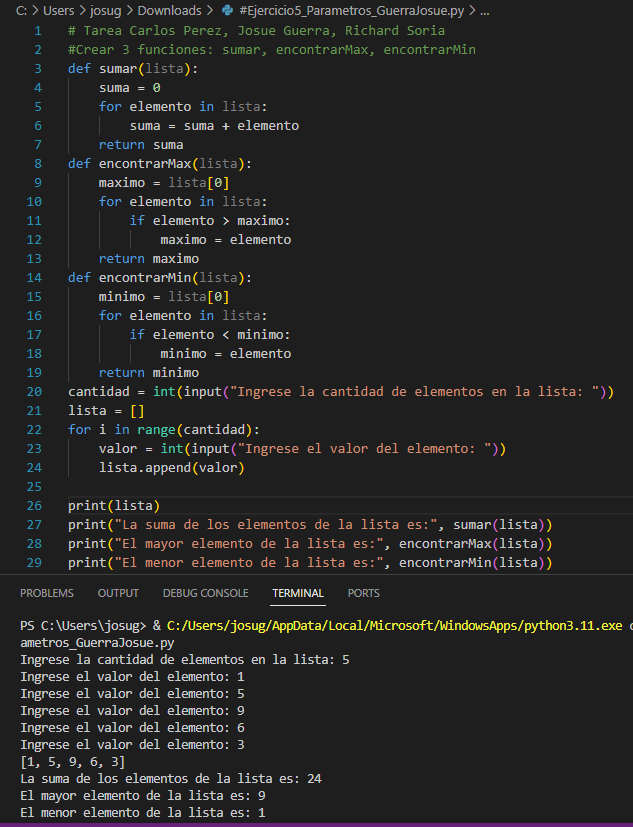
**Parte I**

**TALLER:**

**Listas – Funciones**

Leer la guía sobre Paso de parámetros por valor y referencia del siguiente link:

[Paso de parámetros por lista](file:///C:\Users\josug\Downloads\Funciones-paso-parametros-referencia.pdf)

1. Crear una función con dos parámetros paso por referencia y valor, que doble el valor de la lista y el dato enviado
2. Crear una función que muestre una lista dada
3. Crear una función que sume los valores de una lista dada
4. ****Elaborar tres funciones, la primera recibe la lista y retorna la suma de todos sus elementos, la segunda recibe la lista y retorna el mayor valor y la última recibe la lista y retorna el menor.

**Algoritmos de búsqueda lineal**

1. Crear una función que imprima la posición de un valor enviado

def buscarLinealmente(lista,numero):

    i = 0 //posicion de la lista

    for elemento in lista: *# estamos en la posicion i, z contiene el valor de lista[i]*

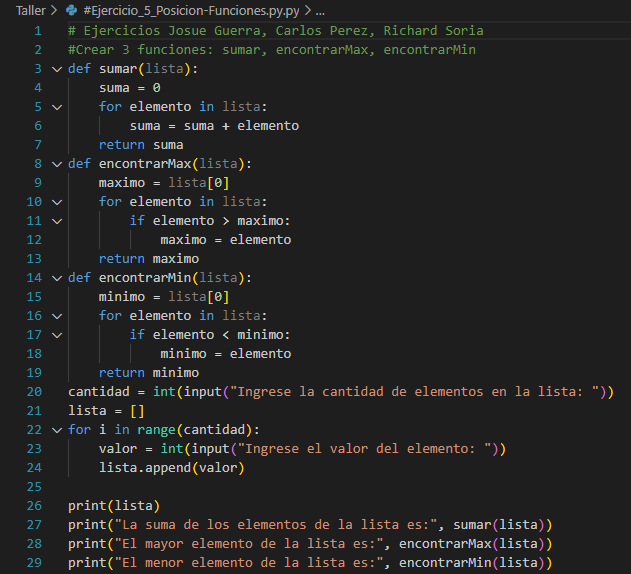
      if elemento == numero:

            return i

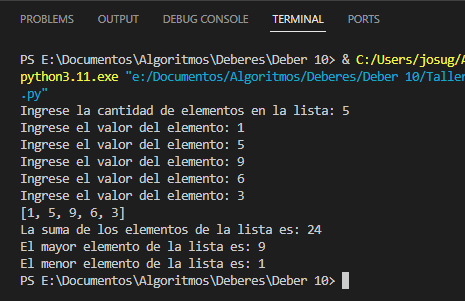
        i = i+1

    return -1

lista = [2,5,3,6,7,8,4,9,1,0]

print(buscarLinealmente(lista, 7))

EJECUCIÓN:



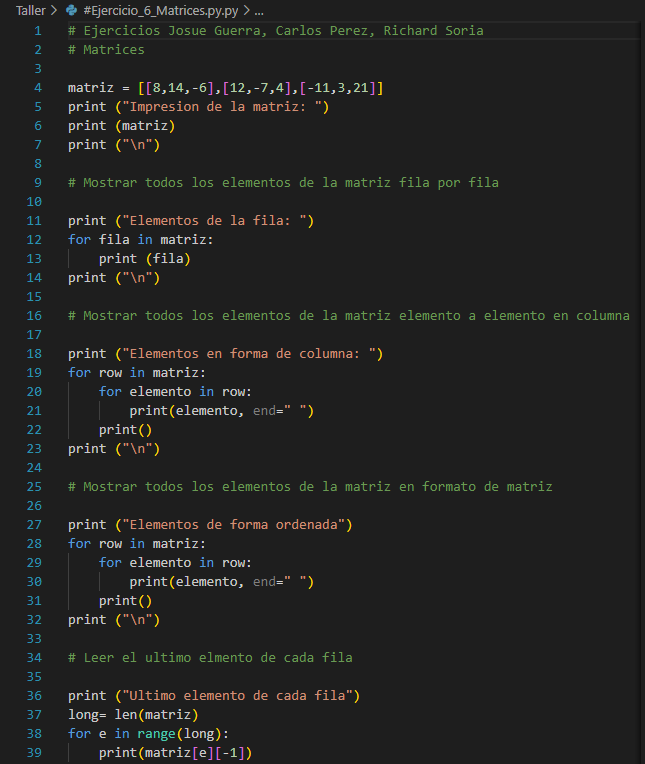
**Matrices**

**valores = [[8,14,-6],**

**[12,-7, 4],**

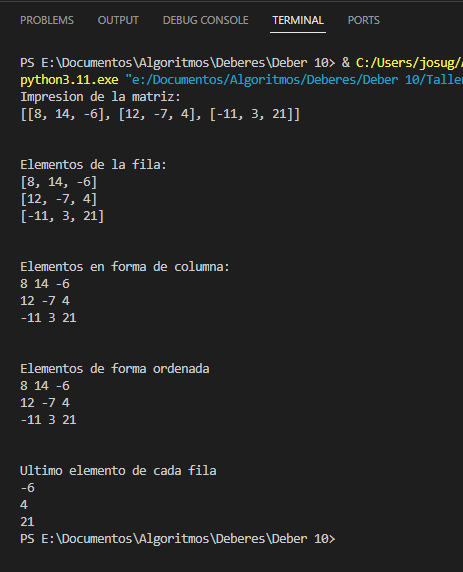
**[-11,3, 21]]**

1. Dada la siguiente matriz:

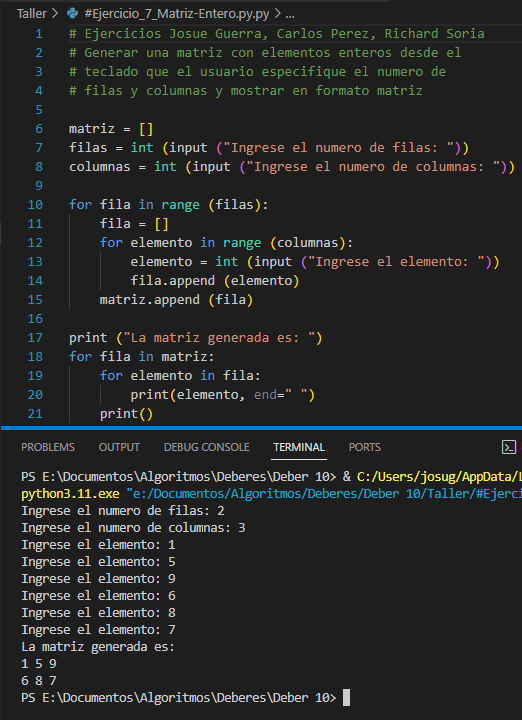
* Mostrar todos los elementos de la matriz fila por fila
* Mostrar todos los elementos de la matriz elemento a elemento en columna
* Mostrar todos los elementos de la matriz en formato de matriz
* Leer el último elemento de cada fila

LA EJECUCIÓN DEL CÓDIGO SE ENCUENTRA EN LA SIGUIENTE PÁGINA

Ejecución:



EL EJERCICIO 7 DEL TALLER SE ENCUENTRA EN LA SIGUIENTE PÁGINA

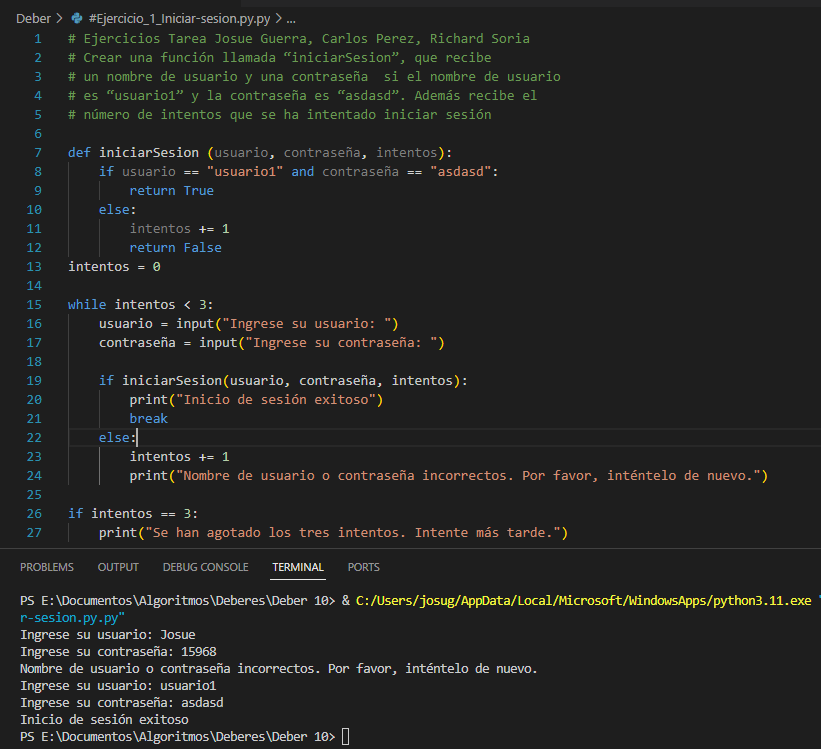
1. ****Generar una matriz con elementos de tipo entero desde el teclado. El usuario debe especificar las filas y las columnas de la matriz e ingresar los elementos, mostrar en formato de matriz

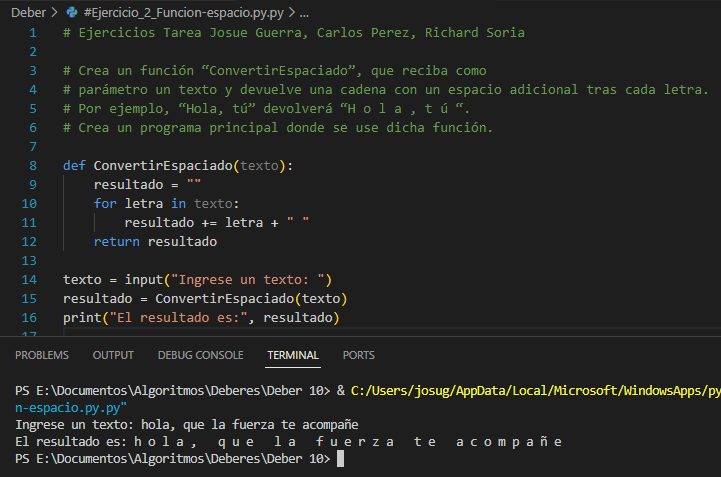
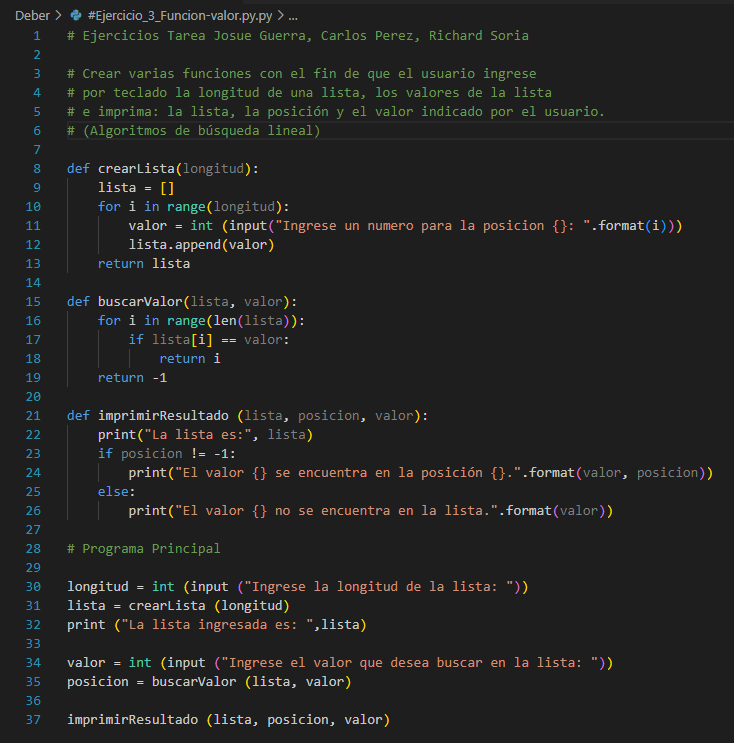
**Parte II**

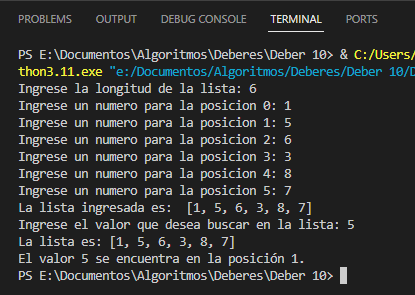
**TAREA:**

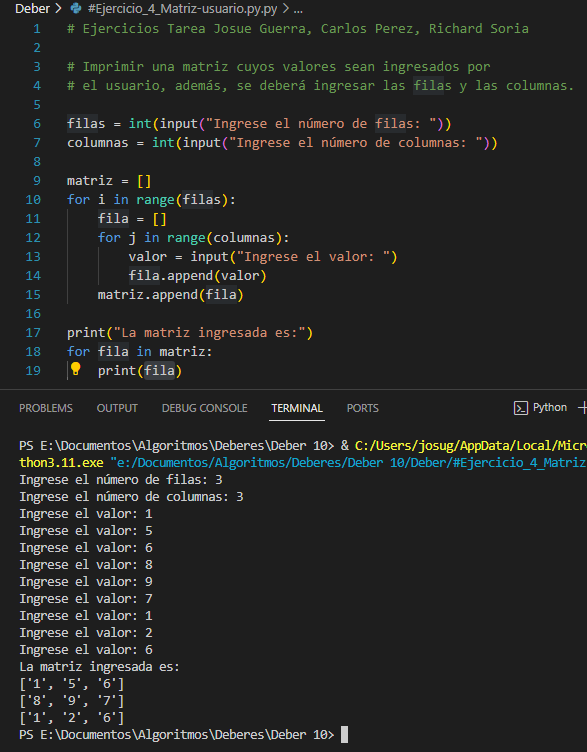
Funciones

1. Crear una función llamada “iniciarSesion”, que recibe un nombre de usuario y una contraseña y te devuelve Verdadero si el nombre de usuario es “usuario1” y la contraseña es “asdasd”. Además, recibe el número de intentos que se ha intentado iniciar sesión y si no se ha podido iniciar la sesión incremente este valor.

Crear un programa principal donde se pida un nombre de usuario y una contraseña y se intente hacer login, solamente tenemos tres oportunidades para intentarlo.

1. Crea una función “ConvertirEspaciado”, que reciba como parámetro un texto y devuelve una cadena con un espacio adicional tras cada letra. Por ejemplo, “Hola, tú” devolverá “H o la, t ú “. Crea un programa principal donde se use dicha función.
2. Crear varias funciones con el fin de que el usuario ingrese por teclado la longitud de una lista, los valores de la lista e imprima: la lista, la posición y el valor indicado por el usuario. (Algoritmos de búsqueda lineal)

Ejecución:

1. Imprimir una matriz cuyos valores sean ingresados por el usuario, además, se deberá ingresar las filas y las columnas.
2. Crear un programa para trabajar con una pila. Una pila es una estructura de datos que nos permite guardar un conjunto de variables. La característica fundamental es que el último elemento que se añade al conjunto es el primero que se puede sacar.

Para representar una pila vamos a utilizar una lista de cadenas de caracteres.

Vamos a crear varias funciones para trabajar con la pila:

LongitudPila: Función que recibe una pila y devuelve el número de elementos que tiene.

EstaVaciaPila: Función que recibe una pila y que devuelve si la pila está vacía, no tiene elementos.

EstaLlenaPila: Función que recibe una pila y que devuelve si la pila está llena.

AddPila: función que recibe una cadena de caracteres y una pila, y añade la cadena a la pila, si no está llena. sí está llena muestra un mensaje de error.

SacarDeLaPila: Función que recibe una pila y devuelve el último elemento añadido y lo borra de la pila. Si la pila está vacía muestra un mensaje de error.

EscribirPila: Función que recibe una pila y muestra en pantalla los elementos de la pila.

1. Realiza un programa principal que nos permita usar las funciones anteriores, que nos muestre un menú, con las siguientes opciones:

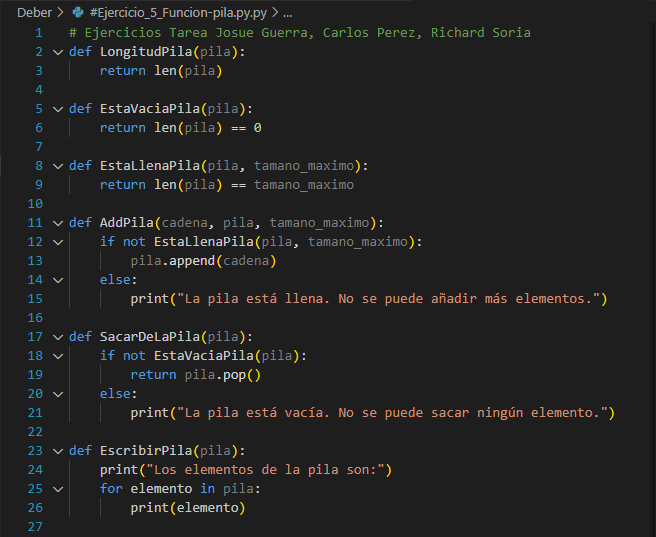
Añadir elemento a la pila

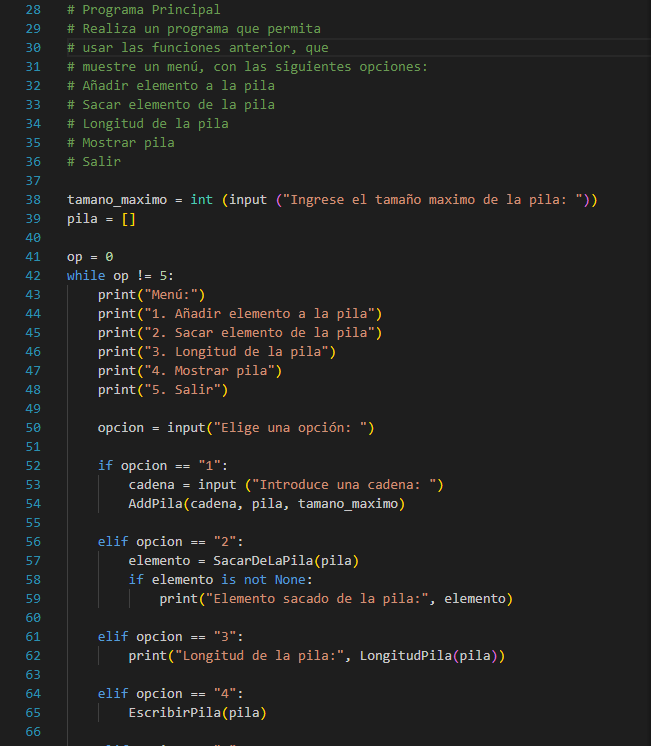
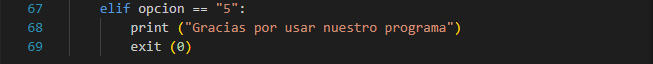
Sacar elemento de la pila

Longitud de la pila

Mostrar pila

Salir

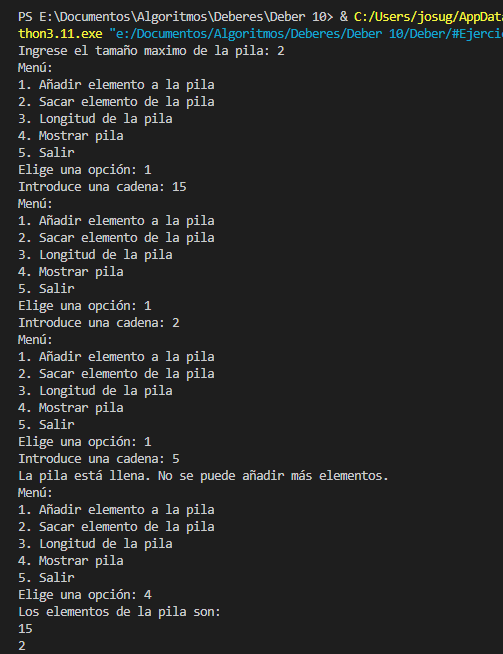


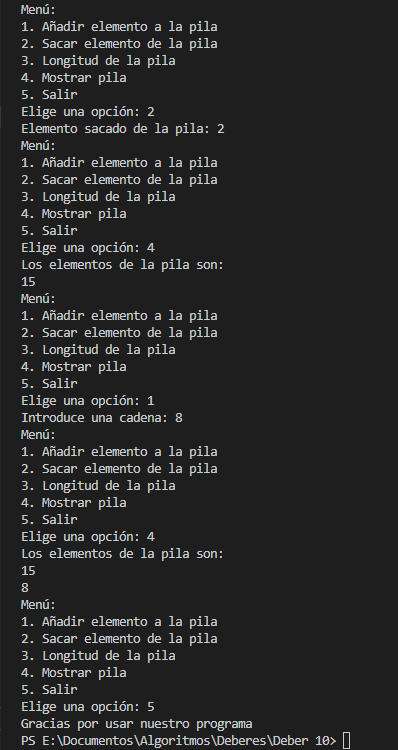


El programa toma en consideración la orden del ejercicio 5 y la del ejercicio, juntando todo en un programa el cual se presenta en el siguiente código, por ende, se muestran en las capturas presentadas, su ejecución se encuentra en la siguiente página.

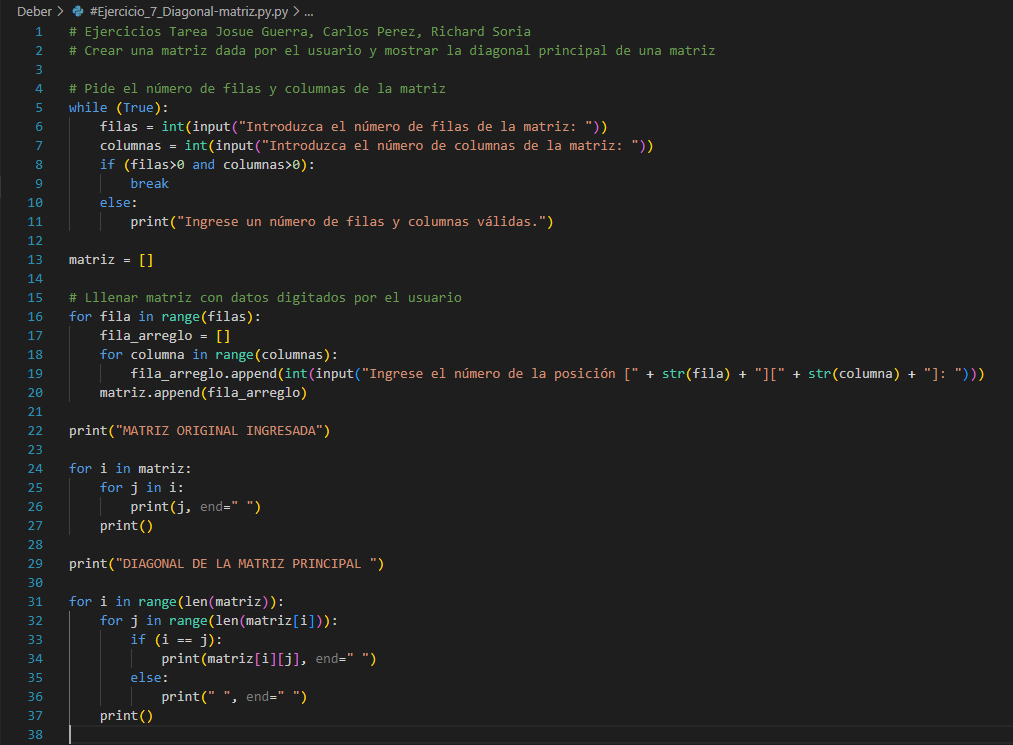
Ejecución:

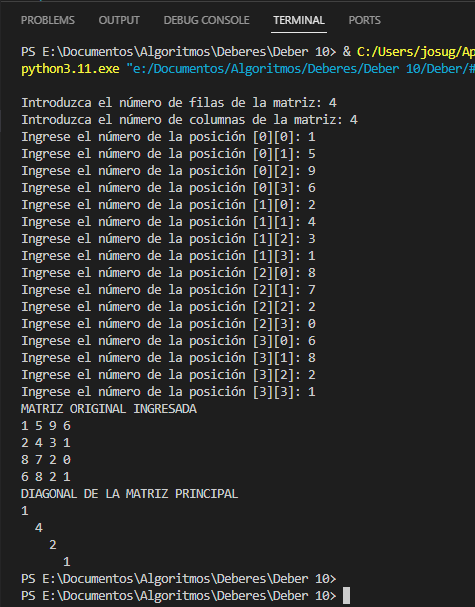
Al ejecutar el código se comprueba que todas las opciones cumplan con lo solicitado, entonces se procede a escoger cada una de las opciones para visualizar en pantalla

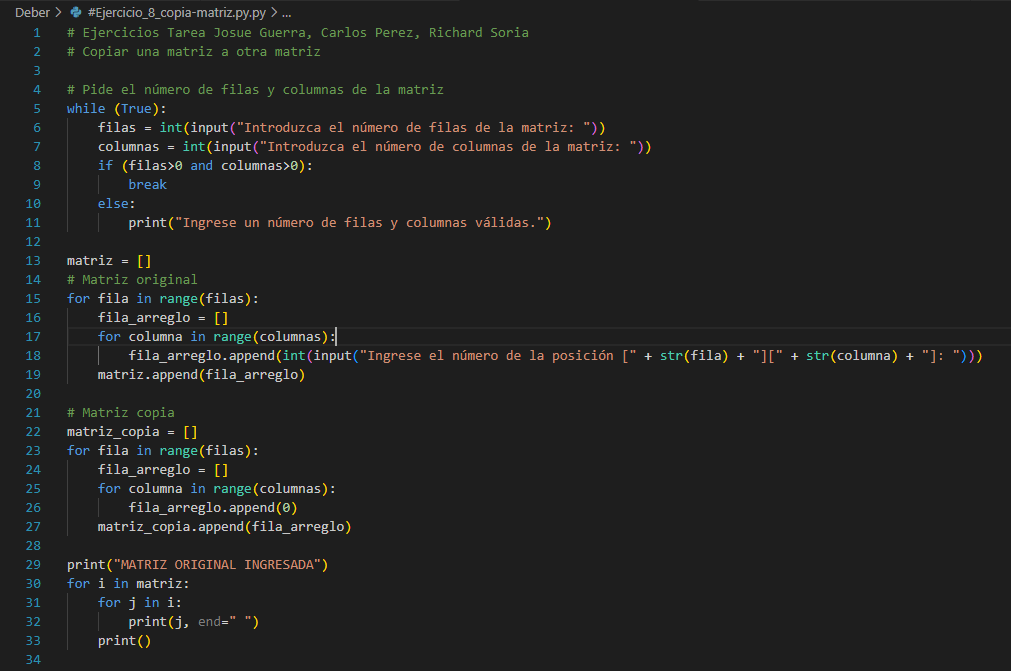
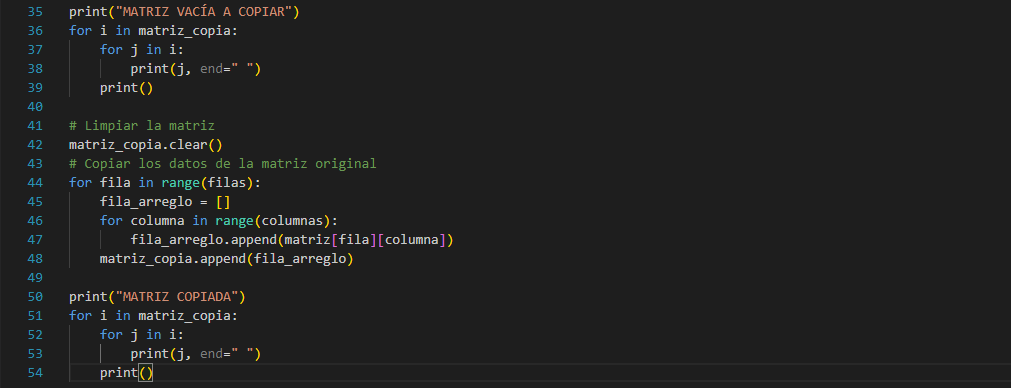




1. Crear una matriz dada por el usuario y mostrar la diagonal principal de una matriz

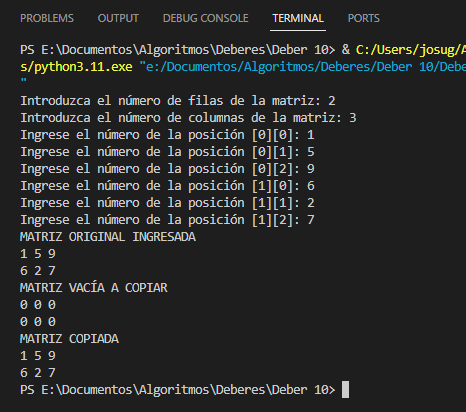


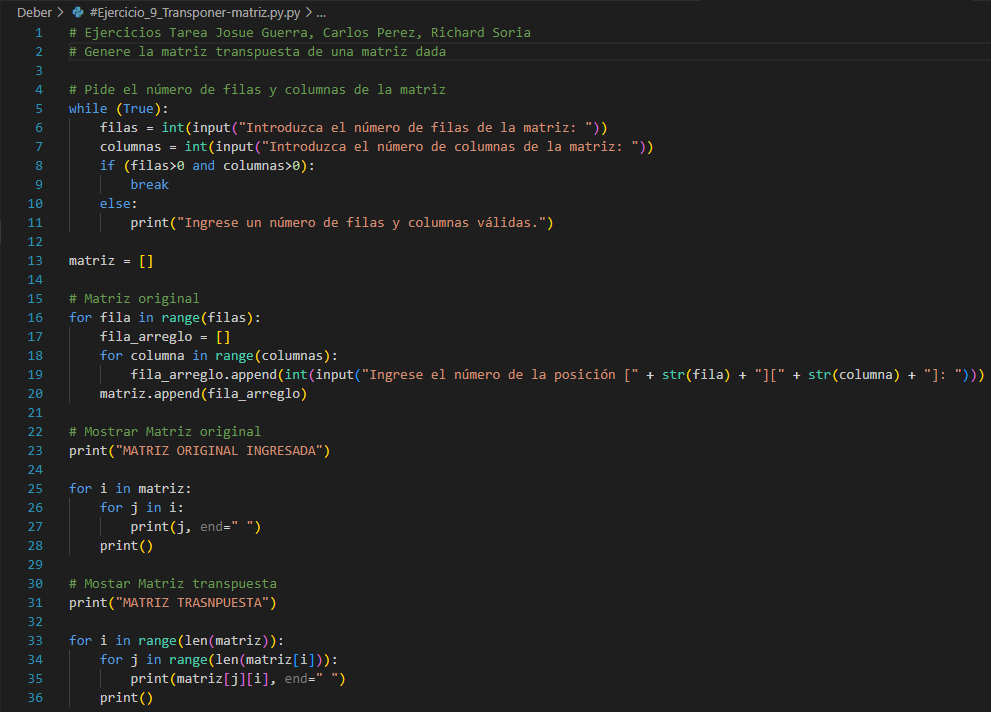
Ejecución:

1. Copiar una matriz a otra matriz

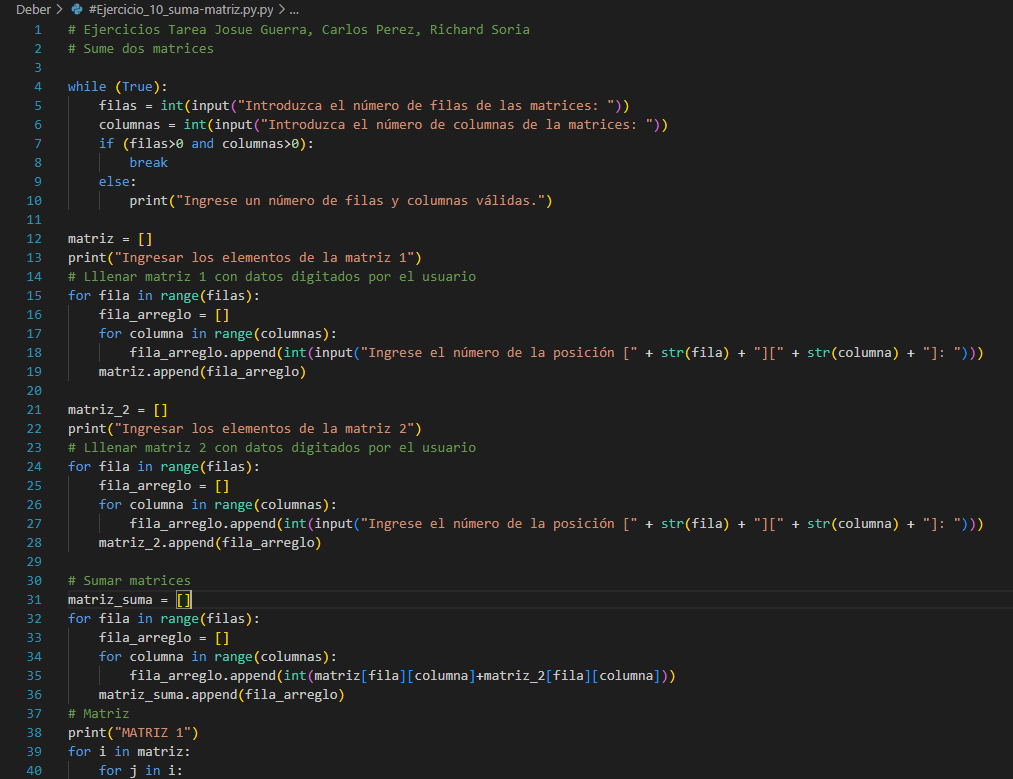
La ejecución del código se encuentra en la siguiente hoja.

Ejecución:

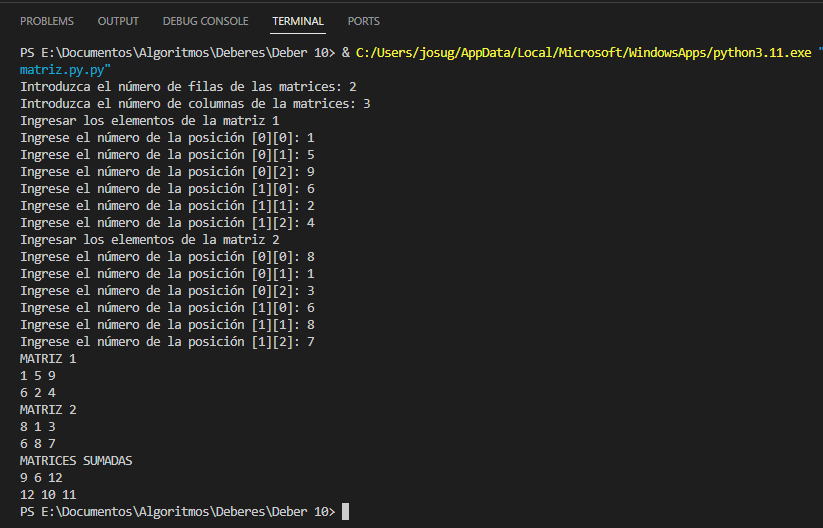


1. Genere la matriz transpuesta de una matriz dada

EJECUCIÓN

1. Sume dos matrices



EJECUCIÓN:

Se solicita al usuario que inicialmente inserte tanto el número de filas como columnas, para ambas matrices ya que en esa relación para poder sumarlas deberían ser iguales, al insertar esos valores, el usuario insertara también los dígitos de cada matriz, mostrando en pantalla la matriz uno la dos y la suma de ambas.

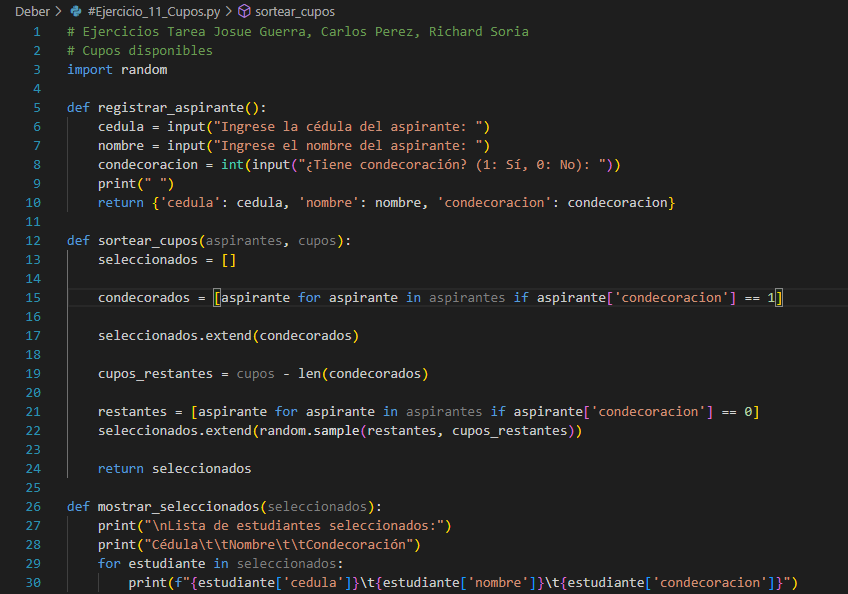
1. [](http://blog.espol.edu.ec/ccpg1001/files/2015/03/enes_exam.png)Debido a la alta demanda y limitada oferta de cupos para el curso pre-politécnico de la EPN, se realiza el sorteo de cupos entre los aspirantes. Los estudiantes que han sido abanderados o escoltas en sus colegios, ingresan sin sorteo.

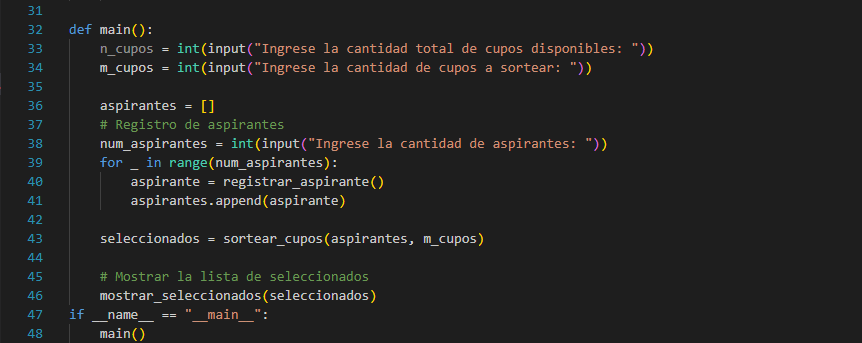
Se requiere un programa para realizar el registro de los aspirantes que participaran en el sorteo, ingresando su cédula, nombre, condecoración (1:abanderado/escolta, 0: ninguna).

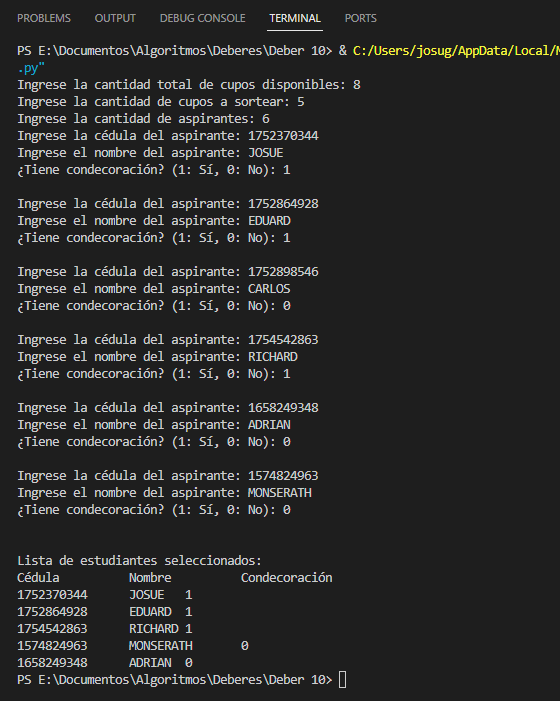
De los n cupos disponibles, se les descuenta la cantidad de estudiantes que tienen condecoración, y se sortean los m cupos restantes entre los estudiantes.

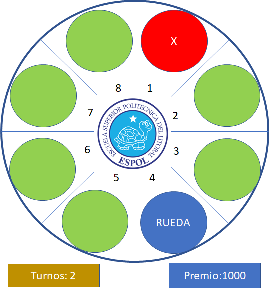
Muestre la lista de todos estudiantes seleccionados.

|  |  |  |
| --- | --- | --- |
| Cédula | Nombre | Condecoración |
| 0912345678 | Juan Pedro Moreno | 0 |
| 0987654321 | Maria Ana Campos | 1 |
| … | … | … |

NOTA: n se ingresa al inicio y m se cuenta luego del registro de aspirantes  
“[El sorteo de cupos para el ‘pre’ generó alegría y decepción](http://www.eluniverso.com/2010/01/21/1/1445/sorteocupospregeneroalegriadecepcion.html)”. www.eluniverso.com. Enero 21 del 2010.

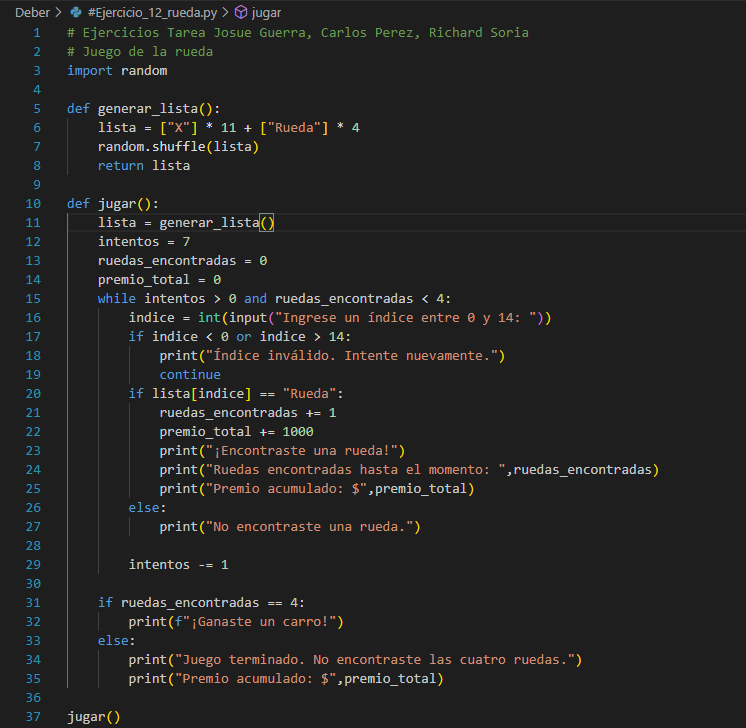


EJECUCIÓN:

1. [](http://blog.espol.edu.ec/ccpg1001/files/2019/08/JuegoRueda01.png)Escriba un programa que implemente el “Juego de las Ruedas”.

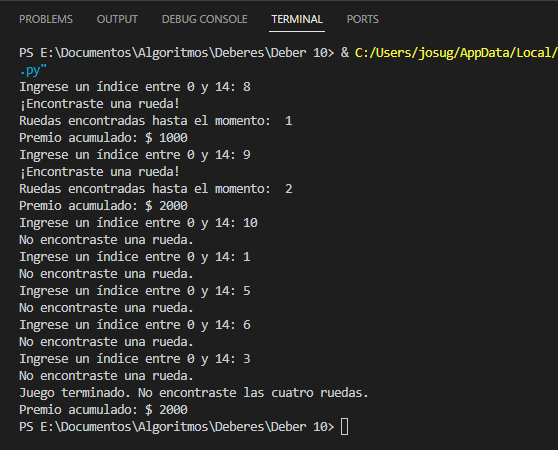
Para esto genere aleatoriamente una lista de 15 elementos, donde cuatro elementos deben decir “Rueda” y los otros once “X”.

Luego el programa deberá pedirle al jugador que ingrese por teclado índices entre 0 y 14 (​validar) correspondientes a la posición de la rueda.

* Asuma que el jugador siempre ingresa índices distintos.
* Si el índice ingresado por el usuario corresponde al de una “Rueda”, gana $1000.
* Si el jugador encuentra las cuatro “Ruedas” se gana un carro. ​
* El jugador tiene siete intentos para hallar las cuatro “Ruedas”.
* El juego termina cuando encuentra las cuatro “Ruedas” o ha usado todos los intentos​.

En cada intento muestre por pantalla el número total de “Ruedas” encontradas hasta el momento. Al final muestre el premio que recibe el jugador (cantidad de dólares o la palabra “carro” si encontró las cuatro ruedas).

EJECUCIÓN:



Se intento varias veces para comprobar que el programa establezca correctamente cada función, además cuando el usuario gana también se va a cumulando el premio que es en dinero si el participante acierta con el índice del número, tomando en cuenta que en cada ejecución los índices ganadores irán cambiando ya que el premio se almacena en diferentes números generados con la biblioteca random, y esto permite que no solo este en ciertos números predeterminados, más bien rotan entre el rango de 14.

Parte III

Consulta:

Algoritmos de búsqueda binaria

La búsqueda binaria es un algoritmo eficiente para encontrar un elemento en una lista ordenada de elementos. Funciona al dividir repetidamente a la mitad la porción de la lista que podría contener al elemento, hasta reducir las ubicaciones posibles a solo una.

Un ejemplo de ello es el siguiente: un usuario que tiene una lista ordenada de elementos, como una lista de números. Si desea encontrar un número específico en esta lista, la búsqueda binaria ofrece una estrategia inteligente y eficiente. En lugar de comenzar desde el principio de la lista, el algoritmo comienza en el medio. Si el número buscado es mayor que el número en el medio de la lista, el usuario sabe que no puede estar en la mitad inferior de la lista, por lo que descarta esa mitad y se queda solo con la mitad superior.

Luego, el usuario vuelve a calcular el punto medio de la mitad superior y repite el proceso. Si el número buscado es menor que el número en el medio de esta nueva mitad descarta la mitad superior y se queda solo con la mitad inferior. Este proceso de división por la mitad se repite hasta que se encuentra el número deseado o hasta que solo queda un elemento en la lista.

Este enfoque es eficiente porque en cada paso se elimina la mitad de los elementos posibles en los que podría estar el número buscado. Esto contrasta con la búsqueda lineal, donde podría ser necesario revisar cada elemento de la lista uno por uno. En resumen, la búsqueda binaria ofrece una manera inteligente y rápida de encontrar un elemento en una lista ordenada, ya que reduce rápidamente el espacio de búsqueda a la mitad en cada paso. [1]

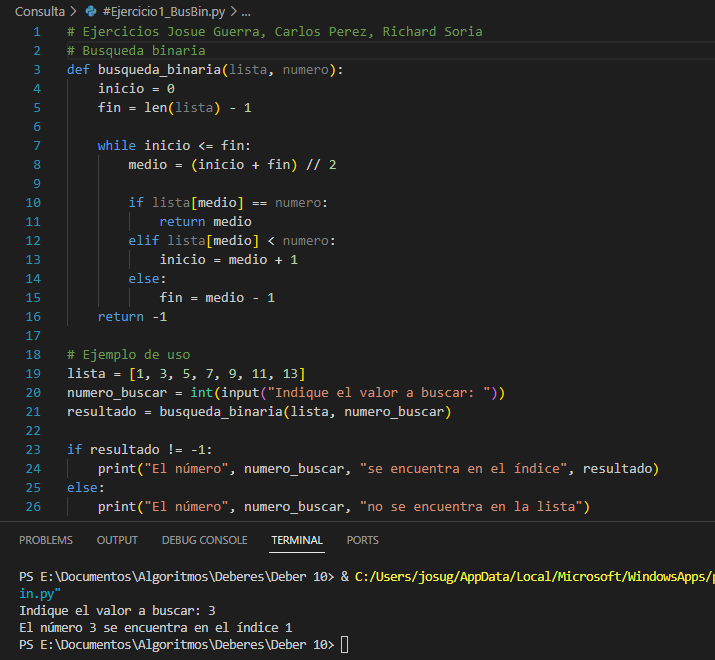
RESUMEN

La búsqueda binaria es un algoritmo eficiente para encontrar un elemento en una lista ordenada. Funciona dividiendo repetidamente la lista en mitades y descartando la mitad donde no puede estar el elemento buscado, hasta reducir las ubicaciones posibles a solo una. Por ejemplo, si se busca un número en una lista ordenada, la búsqueda binaria comienza en el medio y ajusta la búsqueda según el valor del elemento en ese punto. Este proceso se repite hasta encontrar el elemento deseado o hasta que solo quede un elemento en la lista. Este enfoque es más rápido que la búsqueda lineal, ya que elimina la mitad de los elementos posibles en cada paso, lo que lo hace eficiente para listas grandes.

Para demostrar mejor se realizan dos ejemplos, los cuales se presentan en la siguiente página

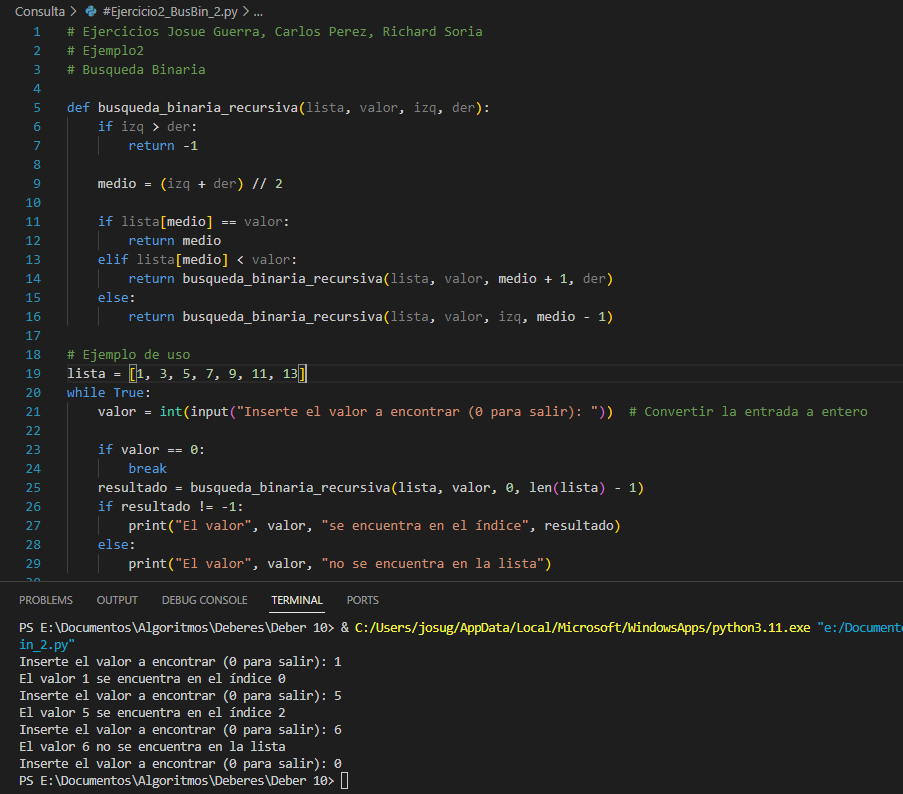
EJEMPLOS

EJEMPLO 1:



Se declara una lista predefinida donde el usuario inserta el valor a buscar, tomando en cuenta que el programa lee la posición desde el índice 0, así que el primer número de la lista tendría esa posición y según eso los siguientes tomaran la posición de 1, 2, entre los demás

EJEMPLO 2:



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 “S11-Tarea
* 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: Tarea8\_Algoritmos\_2023B\_NApellido(de todos los integrantes)

RECURSOS NECESARIOS

* Acceso a Internet.
* Imaginación.
* VSC

**RECOMENDACIONES**

1. Organiza tu código en funciones claras y bien definidas.
2. Reutiliza código existente para evitar duplicación y mejorar la legibilidad.
3. Documenta adecuadamente tus funciones para facilitar la colaboración y el mantenimiento del código.
4. Simplifica las recomendaciones para un código más eficiente.

**CONCLUSIÓN**

En resumen, el uso de funciones en Python es fundamental para mejorar el modularidad y la reutilización de código en nuestros programas. Al dividir el código en bloques más pequeños y manejables, podemos evitar la repetición innecesaria de líneas de código y facilitar el mantenimiento a largo plazo. Además, las funciones nos permiten encapsular lógica específica en un solo lugar, lo que facilita la depuración y el desarrollo de programas más eficientes.

# Bibliografía

|  |  |
| --- | --- |
| [1] | Khan Academy, «Khan Academy,» 2 Octubre 2015. [En línea]. Available: https://es.khanacademy.org/computing/computer-science/algorithms/binary-search/a/binary-search#:~:text=La%20b%C3%BAsqueda%20binaria%20es%20un,ubicaciones%20posibles%20a%20solo%20una.. [Último acceso: 1 Febrero 2024]. |
| [2] | M. ATEHORT, marzo 2002. [En línea]. Available: http://goo.gl/HbbfD. |