**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 |
|  | |

**LABORATORIO 1**

**Grupal**

|  |
| --- |
|  |
| **TÍTULO:**  **FUNCIONES**  **Nombres de los estudiantes:**   |  | | --- | | **Guerra Lovato Josué**  **Pérez Orosco Carlos**  **Soria Ansa Richard** | |
|  |
|  |
|  |
|  |

**2023-B**

**PROPÓSITO DE LA TAREA**

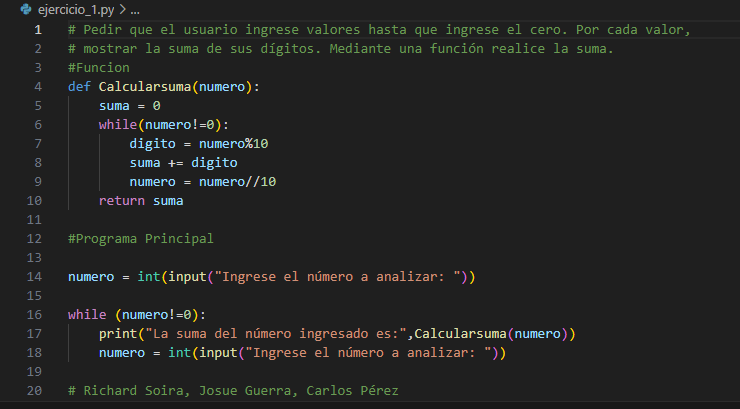
Reutilizar el código mediante funciones para una programación óptima.

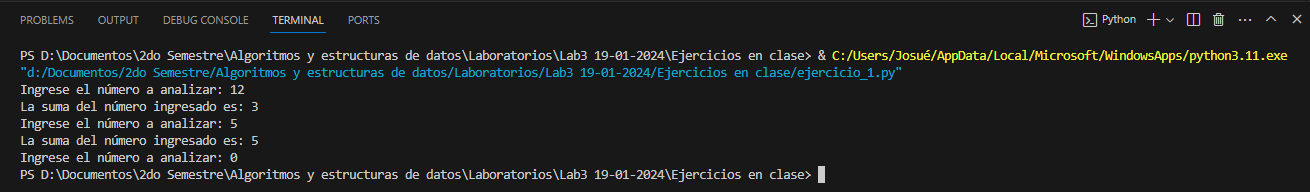
**OBJETIVO GENERAL**

El propósito principal de las funciones y la recursividad en Python es ofrecer soluciones efectivas y sofisticadas para resolver problemas que requieren repeticiones o cálculos complejos. La recursividad permite que una función se llame a sí misma, lo que puede ser muy útil para abordar problemas mediante la subdivisión en subproblemas similares al problema original. Este enfoque busca reemplazar estructuras repetitivas con una solución más elegante y eficiente. Sin embargo, es crucial aplicar la recursividad con cautela, definiendo casos base, gestionando la asignación de memoria y evitando bucles infinitos.

**Parte I**

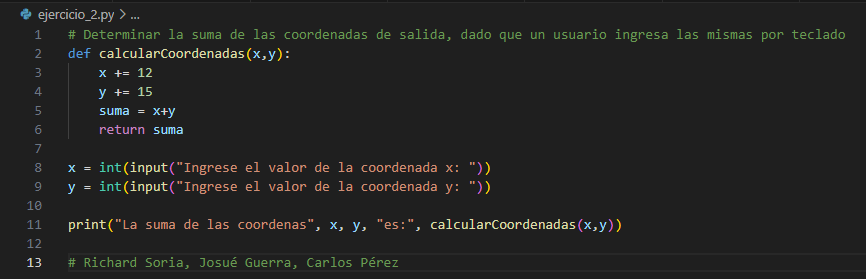
**TALLER:**

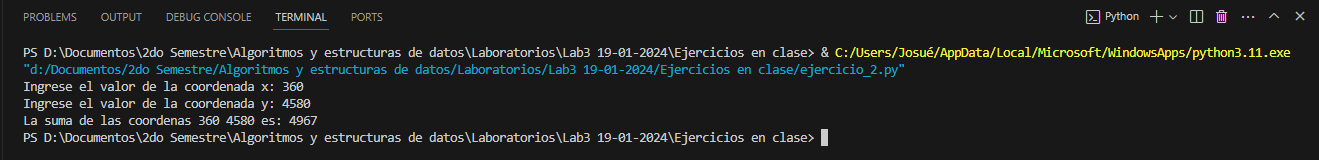
1. ****Pedir que el usuario ingrese valores hasta que ingrese el cero. Por cada valor, mostrar la suma de sus dígitos. Mediante una función realice la suma.

**Ejecución:**

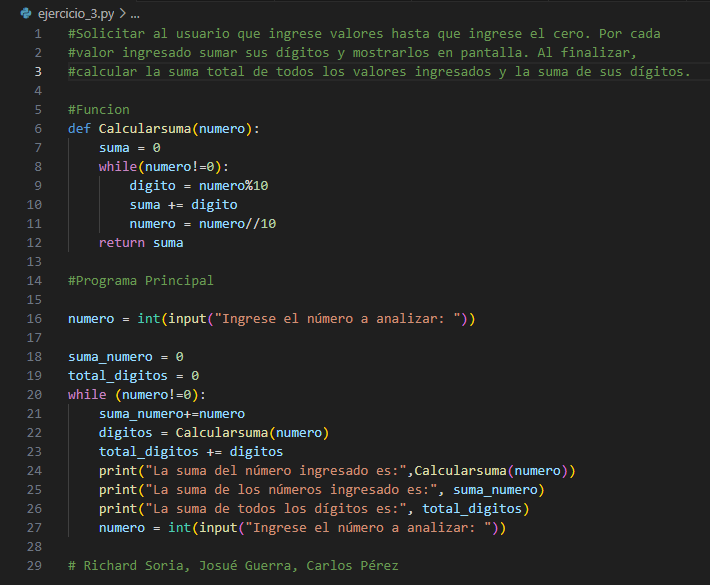
1. Determinar la suma de las coordenadas de salida, dado que un usuario ingresa las mismas por teclado.

X = x + 12

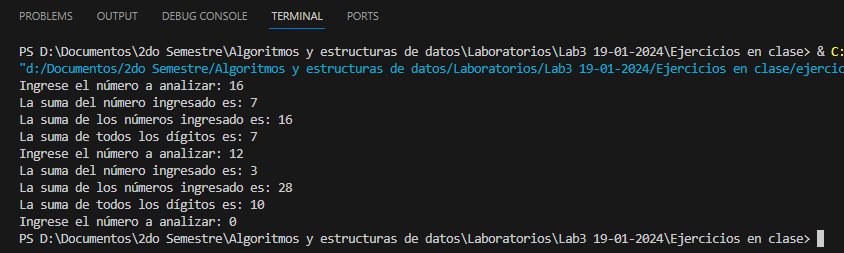
Y = y +15

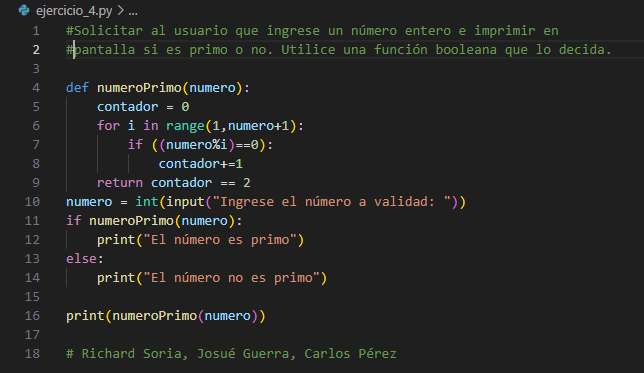
Ejecución:

1. Solicitar al usuario que ingrese valores hasta que ingrese el cero. Por cada valor ingresado sumar sus dígitos y mostrarlos en pantalla. Al finalizar, calcular la suma total de todos los valores ingresados y la suma de sus dígitos.

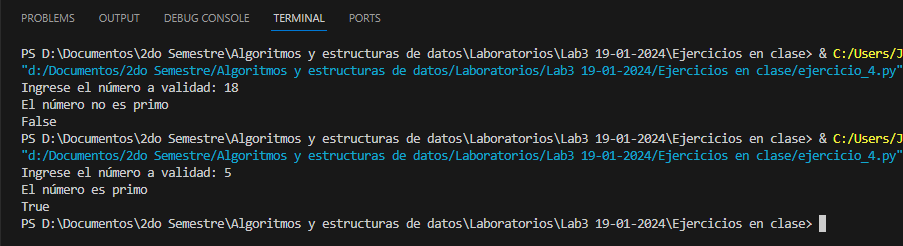


Ejecución:

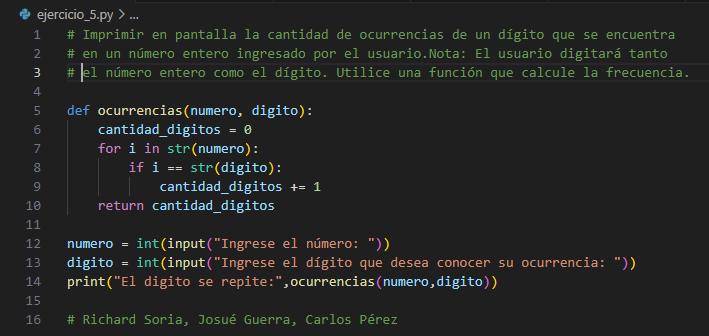


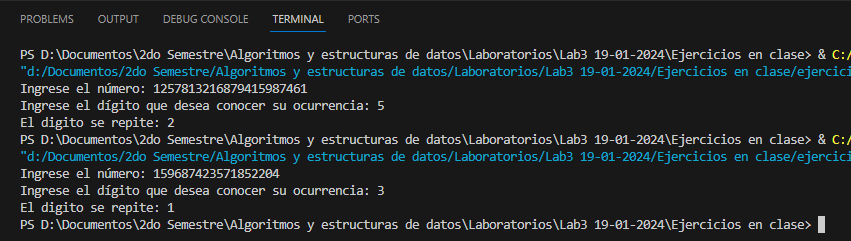
1. Solicitar al usuario que ingrese un número entero e imprimir en pantalla si es primo o no. Utilice una función booleana que lo decida.

Ejecución:



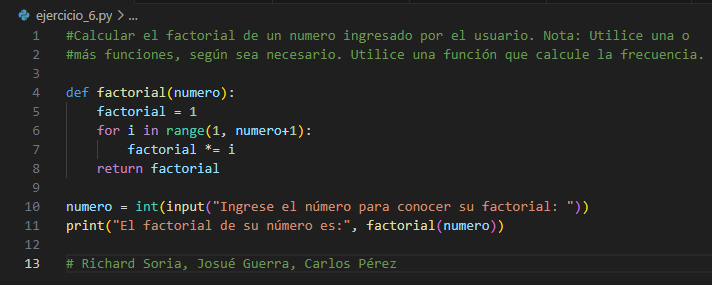
1. Imprimir en pantalla la cantidad de ocurrencias de un dígito que se encuentra en un número entero ingresado por el usuario.

Nota: El usuario digitará tanto el número entero como el dígito.

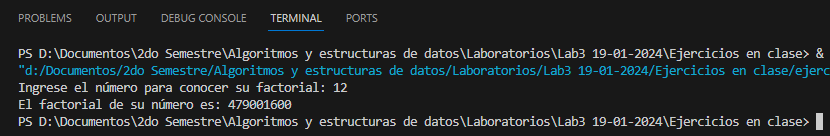
Ejecución:

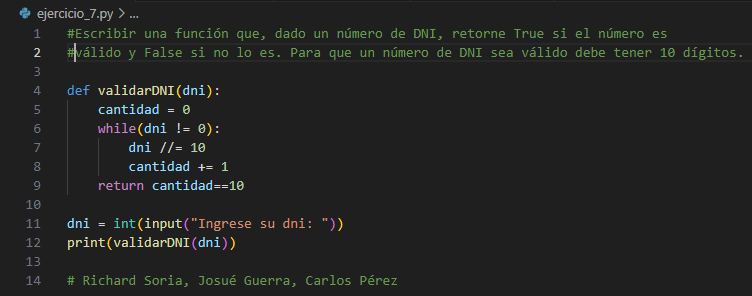
1. Calcular el factorial de un numero ingresado por el usuario.

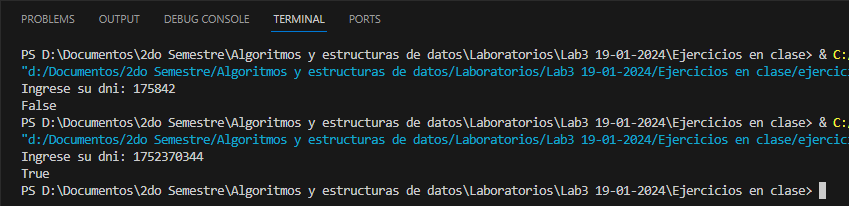
Nota: Utilice una o más funciones, según sea necesario.

Utilice una función que calcule la frecuencia.

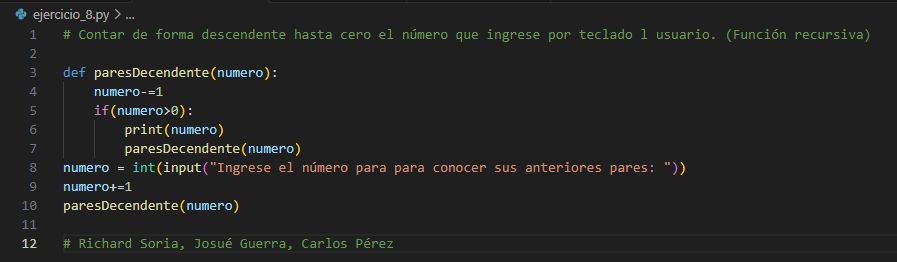
Ejecución:



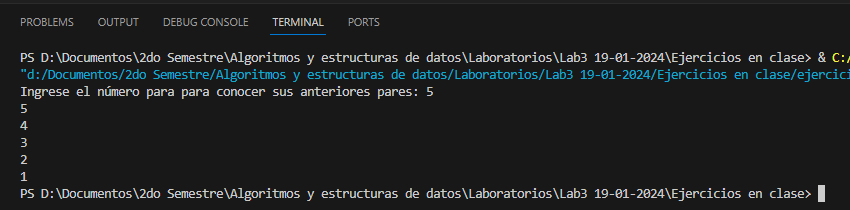
1. Escribir una función que, dado un número de DNI, retorne True si el número es válido y False si no lo es. Para que un número de DNI sea válido debe tener 10 dígitos.

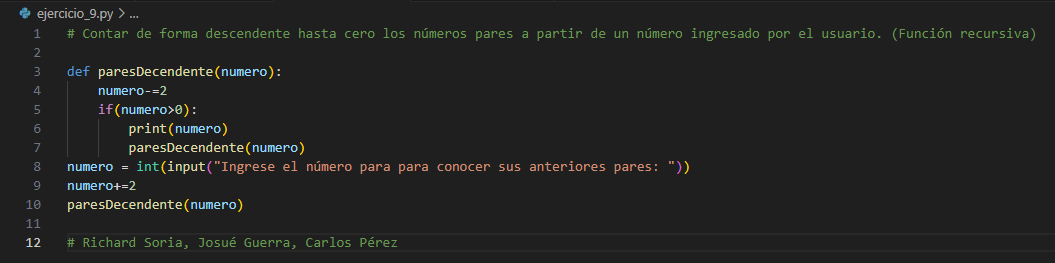
Ejecución:

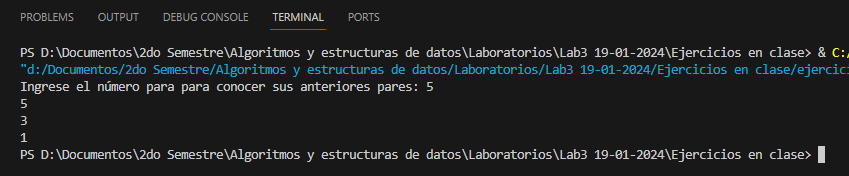
El siguiente ejercicio se encuentra en la siguiente página.

1. Contar de forma descendente hasta cero el número que ingrese por teclado l usuario. (Función recursiva).

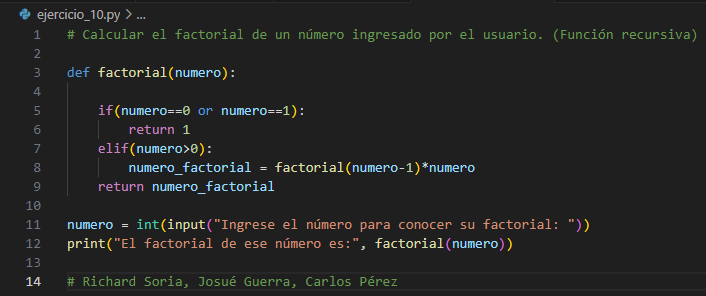
Ejecución:

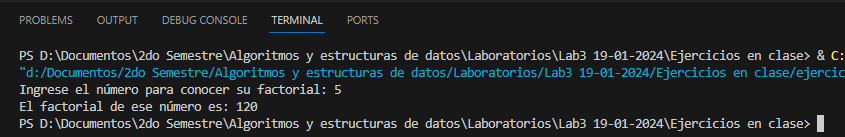


1. Contar de forma descendente hasta cero los números pares a partir de un número ingresado por el usuario. (Función recursiva)

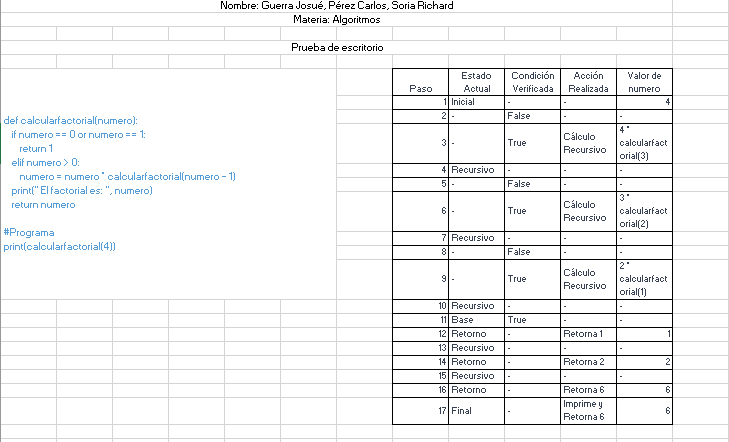
Ejecución:

1. Calcular el factorial de un número ingresado por el usuario. (Función recursiva)



Ejecución:

Prueba de escritorio:

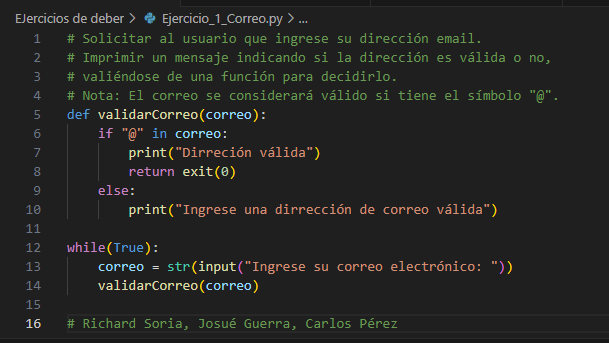


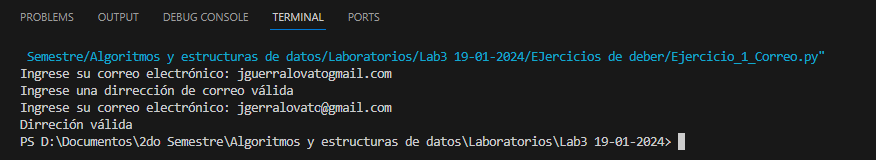
**Parte II**

**TAREA:**

1. Solicitar al usuario que ingrese su dirección email.

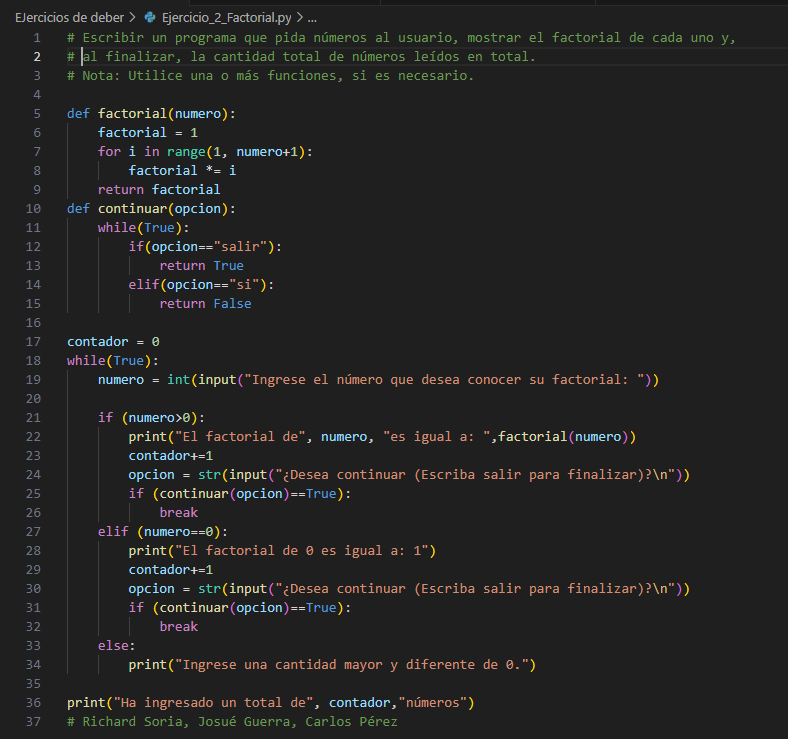
Imprimir un mensaje indicando si la dirección es válida o no, valiéndose de una función para decidirlo.

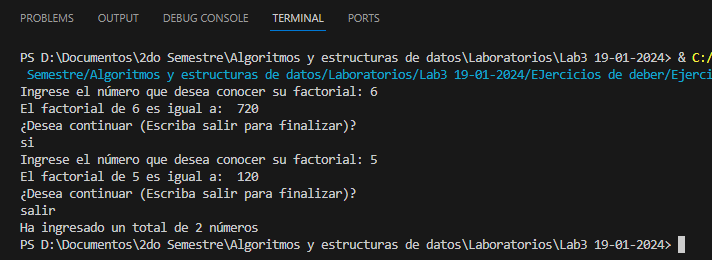
Nota: El correo se considerará válido si tiene el símbolo "@".

Ejecución:

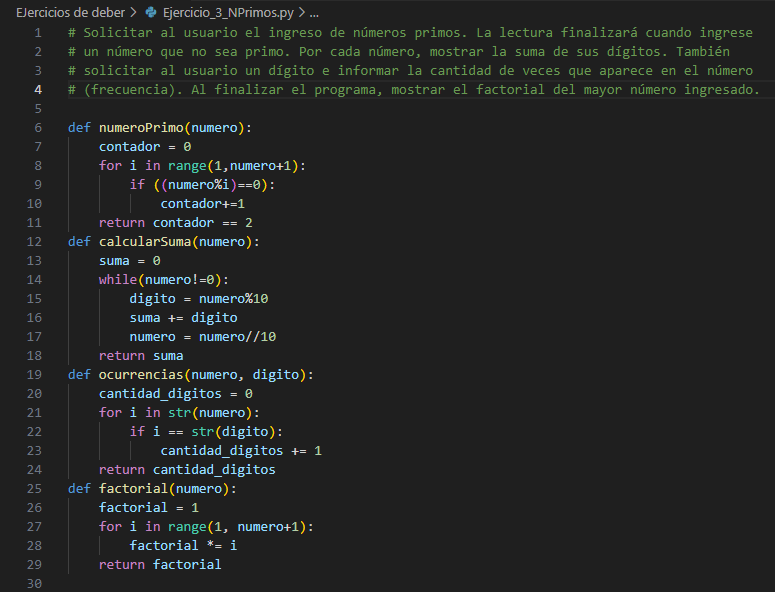
Dado el código se verifica que el usuario inserte el carácter especial (@), sin eso el programa seguirá pidiendo que inserte un correo valido al momento de ingresar con todo y el carácter especial el programa terminara enseguida y en pantalla mostrara dirección valida

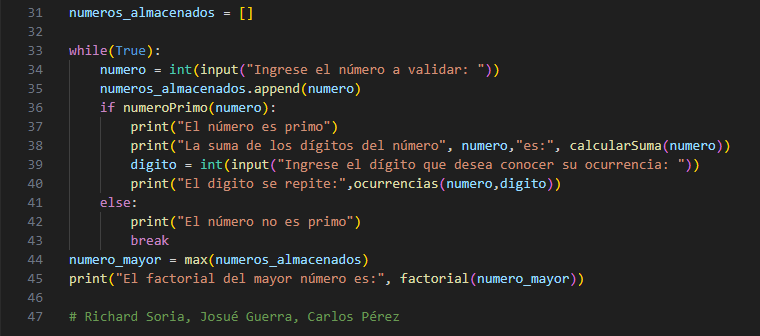
1. Escribir un programa que pida números al usuario, mostrar el factorial de cada uno y, al finalizar, la cantidad total de números leídos en total.

****Nota: Utilice una o más funciones, si es necesario.

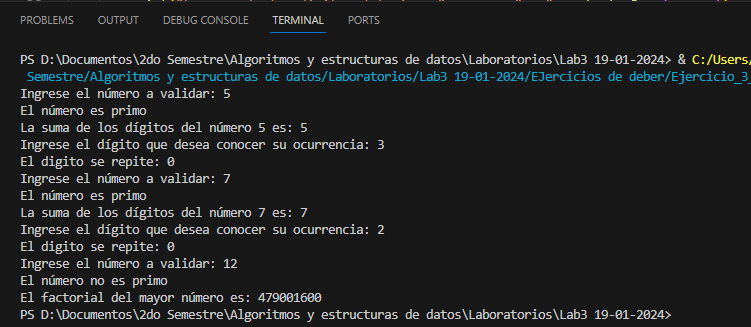
Ejecución:

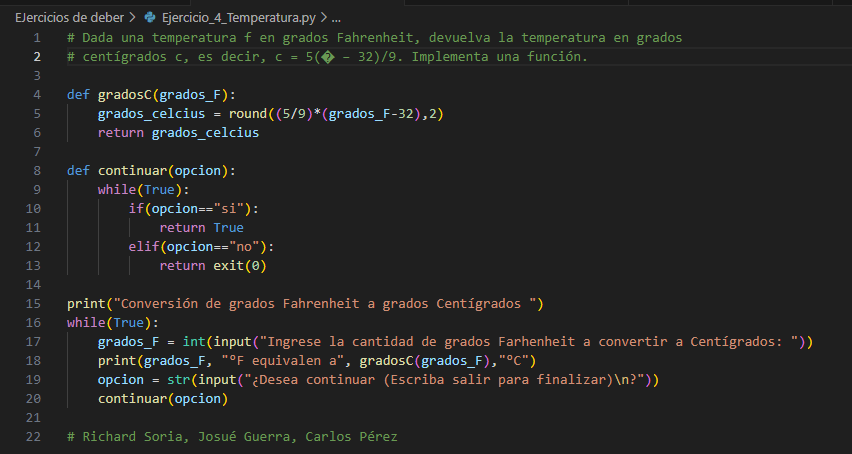
1. Solicitar al usuario el ingreso de números primos. La lectura finalizará cuando ingrese un número que no sea primo. Por cada número, mostrar la suma de sus dígitos. También solicitar al usuario un dígito e informar la cantidad de veces que aparece en el número (frecuencia). Al finalizar el programa, mostrar el factorial del mayor número ingresado.

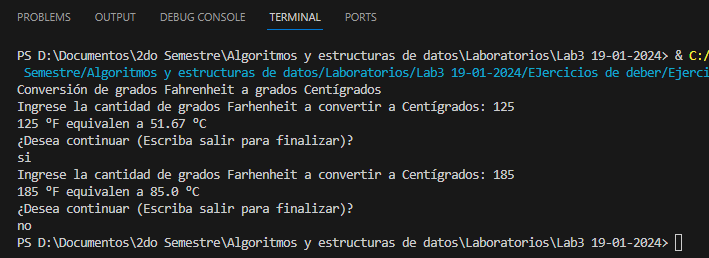




Ejecución:

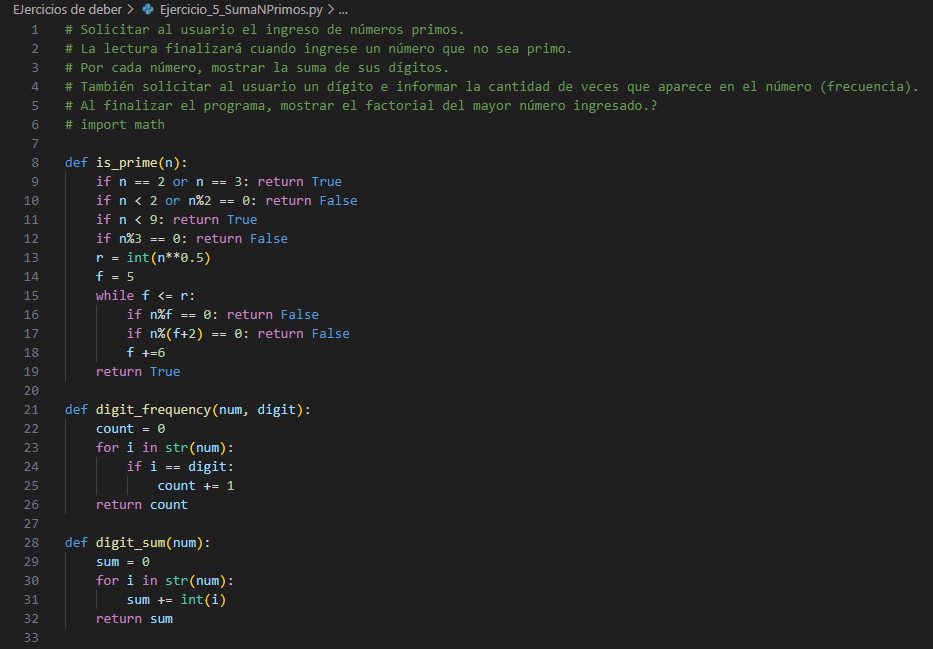
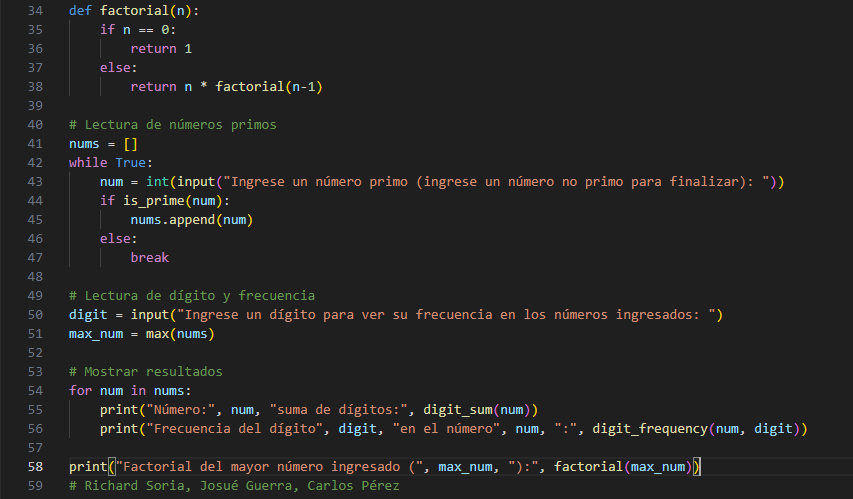


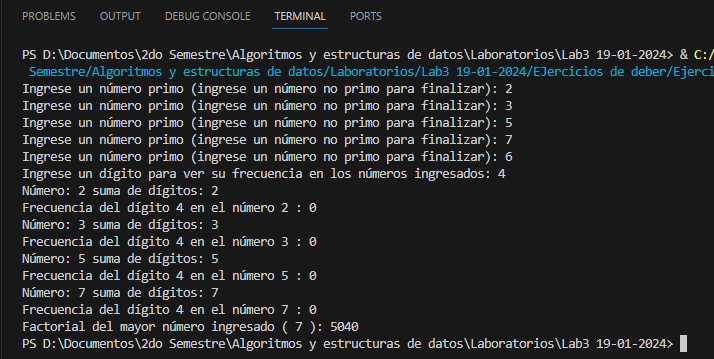
1. Dada una temperatura f en grados Fahrenheit, devuelva la temperatura en grados centígrados c, es decir, c = 5(� − 32) /9. Implementa una función que,

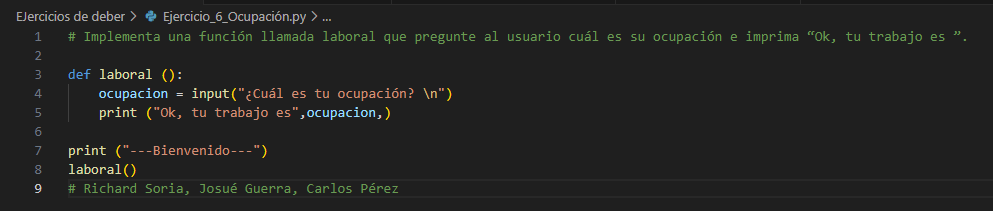
Ejecución:

El código como tal realiza la conversión de grados Fahrenheit a grados centígrados, lo que el usuario inserta los números a convertir y el programa arroja en pantalla el resultado, también pide si el usuario desea continuar o no dependiendo de eso el programa o continuara o finalizara.

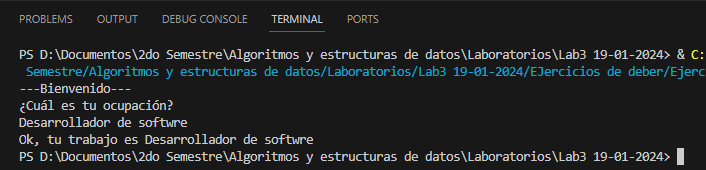
La resolución del siguiente ejercicio se encuentra en la siguiente página.

1. Solicitar al usuario el ingreso de números primos. La lectura finalizará cuando ingrese un número que no sea primo. Por cada número, mostrar la suma de sus dígitos. También solicitar al usuario un dígito e informar la cantidad de veces que aparece en el número (frecuencia). Al finalizar el programa, mostrar el factorial del mayor número ingresado.

Ejecución:

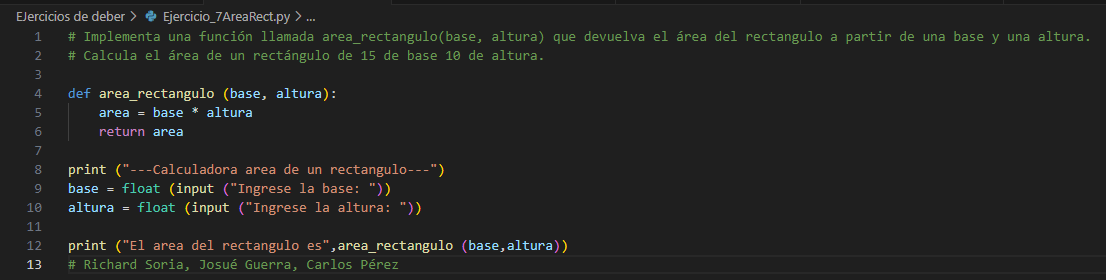
1. Implementa una función llamada laboral que pregunte al usuario cuál es su ocupación e imprima “Ok, tu trabajo es”.

Ejecución:

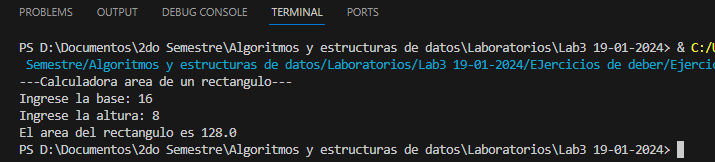


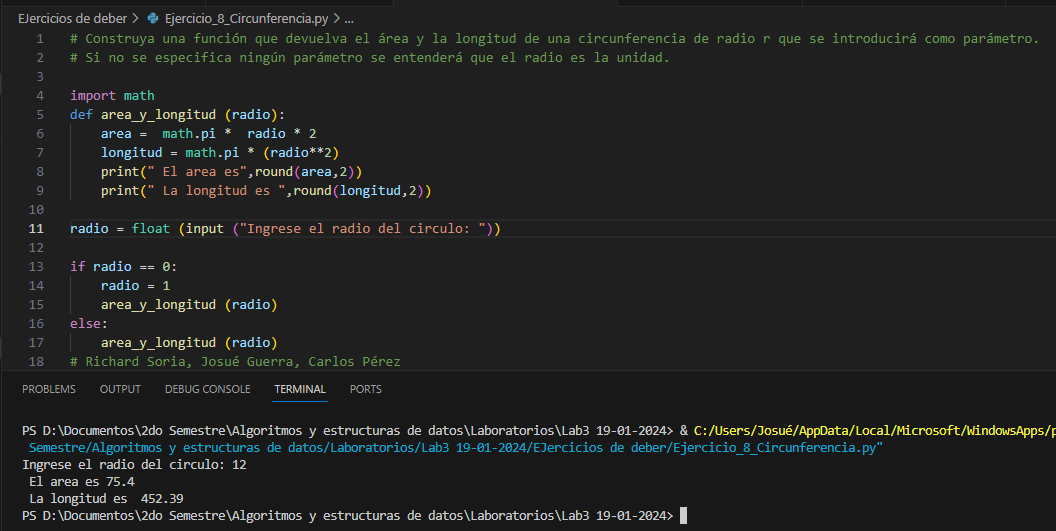
El código solo usa una función en la cual pide al usuario insertar su ocupación y este retorna un texto más junto con la ocupación insertada.

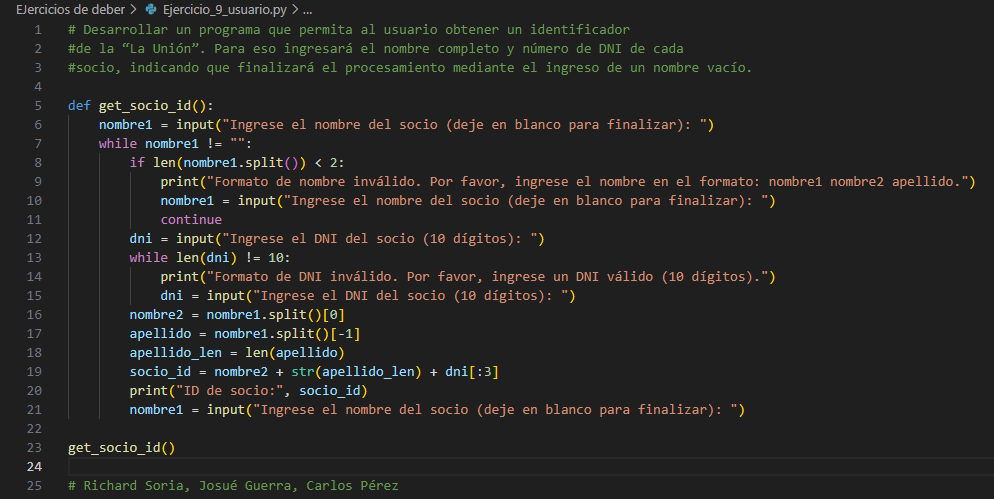
EL SIGUIENTE EJERCICIO SE ENCUENTRA EN LA SIGUIENTE PÁGINA

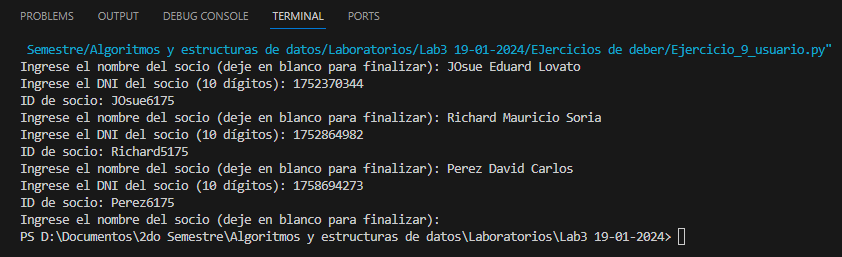
1. Implementa una función llamada area\_rectangulo (base, altura) que devuelva el área del rectángulo a partir de una base y una altura. Calcula el área de un rectángulo de 15 de base 10 de altura.

Ejecución:

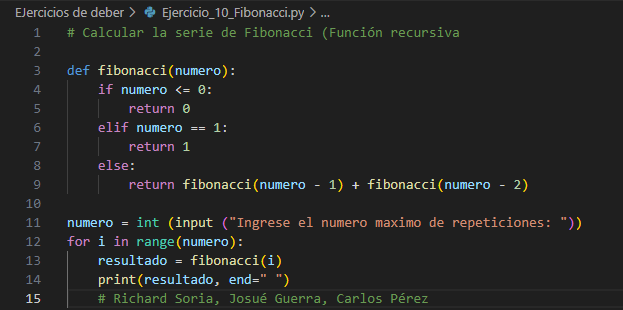


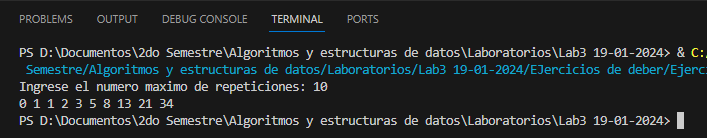
1. Construya una función que devuelva el área y la longitud de una circunferencia de radio r que se introducirá como parámetro. Si no se especifica ningún parámetro se entenderá que el radio es la unidad.
2. Desarrollar un programa que permita al usuario obtener un identificador para cada uno de los socios de un club la “La Unión”. Para eso ingresará el nombre completo y número de DNI de cada socio, indicando que finalizará el procesamiento mediante el ingreso de un nombre vacío.

- Precondición: el formato del nombre de los socios será: nombre apellido. Podría ingresarse más de un nombre, en cuyo caso será: nombre1 nombre2 apellido. Si un socio tuviera más de un apellido, el usuario sólo ingresará uno.  
- Se debe validar que el número de DNI tenga 10 dígitos. En caso contrario, el programa debe dejar al usuario en un bucle hasta que ingrese un DNI correcto.  
- Por cada socio se debe imprimir su identificador único, el cual estará formado por: el primer nombre, la cantidad de letras del apellido y los primeros 3 dígitos de su DNI. Ejemplo:  
Nombre: Alba María Linares  
DNI: 25834910  
Alba7258

Ejecución:

1. Calcular la serie de Fibonacci (Función recursiva)



Ejecución:

**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 “S9-Laboratorio1
* 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

**ENLACES**

**CONCLUSIONES**

En resumen, las funciones y la recursividad en Python tienen como objetivo proporcionar soluciones sofisticadas y efectivas para abordar problemas que implican repeticiones o cálculos complejos. La recursividad, al permitir que una función se llame a sí misma, busca resolver problemas mediante la subdivisión en subproblemas similares al original, con el propósito de reemplazar estructuras repetitivas por una solución más elegante y eficiente. No obstante, es de vital importancia aplicar la recursividad con cuidado, definiendo casos base, gestionando la asignación de memoria y evitando bucles infinitos.

**RECOMENDACIONES**

Es crucial abordar el uso de la recursividad en Python con un enfoque analítico y preciso, estableciendo claramente los casos base para cada problema. Además, se debe manejar con cuidado la asignación de memoria y realizar pruebas exhaustivas para prevenir la ocurrencia de bucles infinitos. La recursividad, a pesar de ser una herramienta poderosa, debe utilizarse con prudencia, considerando alternativas no recursivas cuando sea factible.