

- 1) Al guardar el programa presentado a continuación ocurrió un error, que produjo que se perdieran todos los caracteres de tabulación.

Indicar, a la izquierda de cada línea, **cuántos caracteres** de tabulación se deben agregar para que el programa vuelva a funcionar correctamente.

```
def calcular_promedio(lista_numeros):
total = 0
for numero in lista_numeros:
total = total + numero
promedio = total / len(lista_numeros)
return promedio

def procesar_entrada():
numeros = []
continuar_ingreso = True
while continuar_ingreso == True:
entrada = input("Introducir un número (o 'fin' para terminar): ")
if entrada == 'fin':
continuar_ingreso = False
else:
if entrada.isnumeric() == True:
numero = float(entrada)
numeros.append(numero)
else:
print("Por favor, introducir un número válido.")
return numeros

def main():
print("Bienvenido al programa de cálculo de promedio")
numeros = procesar_entrada()
if len(numeros) > 0:
promedio = calcular_promedio(numeros)
print("El promedio de los números introducidos es:" + str(promedio))
else:
print("No se introdujeron números.")

main()
```

2) Para el código presentado en el inciso anterior, listar:

- Todas las **funciones**, indicando su **tipo**. En caso de tener parámetros y/o valor de retorno, indicar **cantidad** y **tipo** de los mismos.
- Todas las **variables**, indicando su **tipo**. Deben estar agrupadas según la función a la que pertenecen.

- 3) Un jardinero está realizando un programa en Python para facilitar los cálculos en su nuevo proyecto de jardinería. En particular, está buscando calcular la cantidad de tierra que va a necesitar para llenar todas sus macetas.

Para esto, ya escribió una función que almacena las dimensiones de cada maceta en una lista de listas, en la que cada elemento tiene el formato `[radio, altura]`. (Ejemplo: `[[10,40],[25,50],[17,48],...]`)

Ahora, necesita ayuda para crear la última función. La misma debe recibir como parámetro la **lista** ya existente en el programa, **calcular el volumen** total y **mostrar el resultado** en pantalla.

$$V = \pi \cdot r^2 \cdot h$$

*Aclaración: El operador potencia (\*\*) puede resultar de utilidad*

```
resultado = base ** exponente
```

Además de la función, se debe desarrollar un **breve script** que utilice la función para **validar** su correcto **funcionamiento**.

This image shows a full page of blank primary-ruled paper. It features ten sets of horizontal blue lines, each set consisting of three parallel lines. These are separated by vertical red margin lines, creating columns for writing. The paper is otherwise completely blank, with no handwriting or other markings.

- 4) El programa presentado a continuación recibe información a través del terminal, opera con la misma y muestra un resultado.

*Aclaraciones:*

- La función 'len' devuelve la cantidad de elementos que conforman un conjunto (lista/str)
- La función 'count' devuelve la cantidad de apariciones de su parámetro.
- La función 'find' devuelve el índice de la primer aparición de su parámetro.
- Al separar un subconjunto de un string ([:]), el caracter con el índice superior no se incluye.

```
entrada_usuario = input("Ingrese el código: ")

if len(entrada_usuario) ≥ 7 and entrada_usuario.count('-') == 2:
    datos_separados = []

    indice_guion = entrada_usuario.find('-')
    dato_1 = entrada_usuario[:indice_guion]
    datos_separados.append(float(dato_1))

    entrada_usuario = entrada_usuario[indice_guion + 1:]
    indice_guion = entrada_usuario.find('-')
    dato_2 = entrada_usuario[:indice_guion]
    datos_separados.append(dato_2)

    dato_3 = entrada_usuario[indice_guion + 1:]
    datos_separados.append(int(dato_3))

    print(datos_separados[0] * datos_separados[2])
    print(datos_separados[1] * datos_separados[2])

else:
    print("Formato incorrecto")
```

Indicar qué veremos en el terminal si se ingresan las siguientes cadenas (escribiendo 'Error' en caso de que el programa falle):

- a) ab-cd-123
- b) 444-44-4
- c) 156.7wat-83
- d) 10.1-meca-7