Actividad funciones algoritmos

1. Escriba una función para encontrar el máximo de tres números.

```python
def max_de_tres(num1, num2, num3):
    # Compara los tres números y devuelve el mayor
    return max(num1, num2, num3)

# Ejemplo de uso
numero1 = 10
numero2 = 25
numero3 = 15

resultado = max_de_tres(numero1, numero2, numero3)
print(f"El número máximo es: {resultado}")
```

2. Escriba un programa para calcular las áreas de las figuras geométricas utilizando una función para cada área.

```python
import math

def area_circulo(radio):
    """Calcula el área de un círculo dado su radio."""
    return math.pi * radio ** 2

def area_cuadrado(lado):
    """Calcula el área de un cuadrado dado el tamaño de un lado."""
    return lado ** 2

def area_triangulo(base, altura):
    """Calcula el área de un triángulo dado su base y altura."""
    return (base * altura) / 2

# Ejemplo de uso
if __name__ == "__main__":
    radio = 5
    lado = 4
    base = 6
    altura = 3

    print(f"El área del círculo es: {area_circulo(radio):.2f}")
    print(f"El área del cuadrado es: {area_cuadrado(lado):.2f}")
    print(f"El área del triángulo es: {area_triangulo(base, altura):.2f}")
```

3. **Escriba** una función para sumar todos los números de una lista. *Lista de muestras*: (8, 2, 3, 0, 7)*Resultado esperado*: 20.

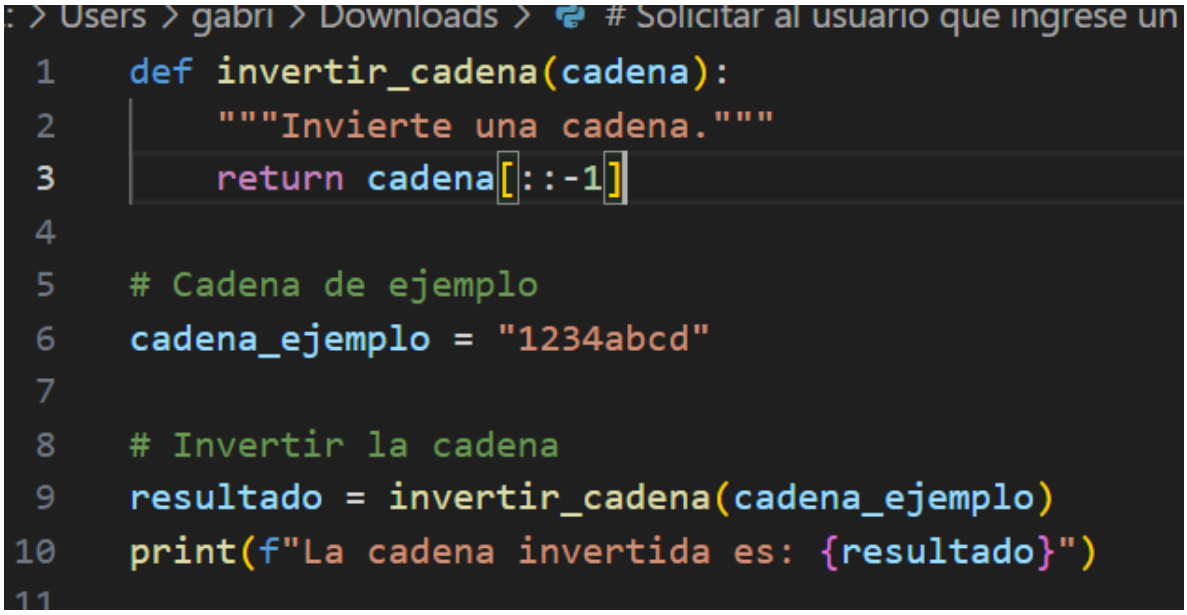
```python
def suma_lista(numeros):
    """Suma todos los números en una lista."""
    return sum(numeros)

# Lista de muestra
lista_muestra = [8, 2, 3, 0, 7]

# Calcular la suma
resultado = suma_lista(lista_muestra)
print(f"La suma de la lista es: {resultado}")
```

4. **Escriba** una función para multiplicar todos los números de una lista. *Lista de muestra*: (8, 2, 3, -1, 7)*Resultado esperado*: -336

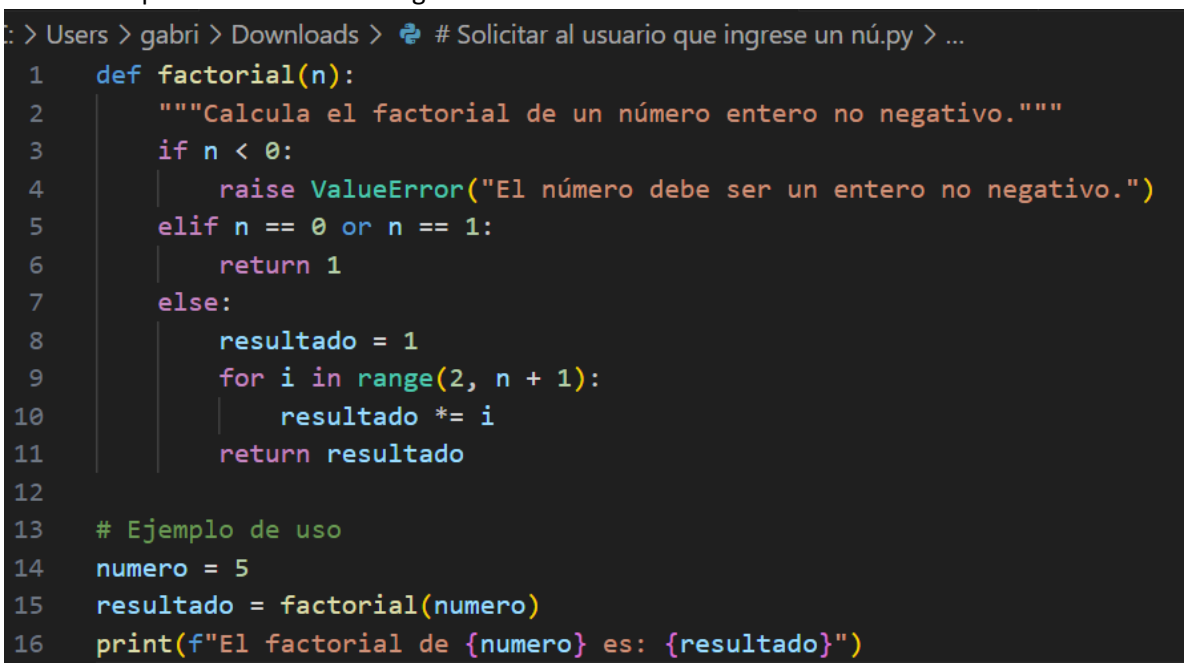
C: > Users > gabri > Downloads > # Solicitar al usuario que ingrese un nú.py > ...

```python
def multiplicar_lista(numeros):
    """Multiplica todos los números en una lista."""
    resultado = 1
    for numero in numeros:
        resultado *= numero
    return resultado

# Lista de muestra
lista_muestra = [8, 2, 3, -1, 7]

# Calcular la multiplicación
resultado = multiplicar_lista(lista_muestra)
print(f"La multiplicación de la lista es: {resultado}")
```

5. **Escriba** un programa para invertir una cadena. *Cadena* de ejemplo: "1234abcd" *Resultado esperado*: "dcba4321"



6. **Escriba** una función para calcular el factorial de un número (un entero no negativo). La función acepta el número como argumento.



7. **Escriba** una función para comprobar si un número cae en un rango determinado. Defina como parámetros rango de inicio, número y rango final.

```python
def esta_en_rango(inicio, numero, fin):
    """Comprueba si un número cae dentro de un rango determinado."""
    return inicio <= numero <= fin

# Ejemplo de uso
inicio_rango = 10
fin_rango = 20
numero = 15

if esta_en_rango(inicio_rango, numero, fin_rango):
    print(f"{numero} está dentro del rango de {inicio_rango} a {fin_rango}.")
else:
    print(f"{numero} NO está dentro del rango de {inicio_rango} a {fin_rango}.")
```

8. **Escriba** una función que acepte una cadena y calcule el número de letras mayúsculas y minúsculas.

```
> Users > gabri > Downloads > # Solicitar al usuario que ingrese un nú.py > ...
def contar_mayusculas_minusculas(cadena):
    """Cuenta el número de letras mayúsculas y minúsculas en una cadena."""
    mayusculas = 0
    minusculas = 0

    for letra in cadena:
        if letra.isupper():
            mayusculas += 1
        elif letra.islower():
            minusculas += 1

    return mayusculas, minusculas

# Ejemplo de uso
cadena_ejemplo = "Hola Mundo!"
mayusculas, minusculas = contar_mayusculas_minusculas(cadena_ejemplo)
print(f"Número de letras mayúsculas: {mayusculas}")
print(f"Número de letras minúsculas: {minusculas}")
```

9. **Escriba** una función que tome una lista y devuelva una nueva lista con elementos únicos de la primera

lista.

C: > Users > gabri > Downloads > # Solicitar al usuario que ingrese un nú.py > ...

```python
def elementos_unicos(lista):
    """Devuelve una nueva lista con elementos únicos de la lista dada."""
    return list(set(lista))

# Ejemplo de uso
lista_original = [1, 2, 2, 3, 4, 4, 5, 1]
lista_unica = elementos_unicos(lista_original)

print(f"Lista original: {lista_original}")
print(f"Lista con elementos únicos: {lista_unica}")

```

10. **Escriba** una función que tome un número como parámetro y verifique que el número sea primo o no. Un número primo (o primo) es un número natural mayor que 1 y que no tiene divisores positivos aparte de 1 y sí mismo.

```python
def es_primo(numero):
    """Verifica si un número es primo."""
    if numero <= 1:
        return False
    for i in range(2, int(numero**0.5) + 1):
        if numero % i == 0:
            return False
    return True

# Ejemplo de uso
numero_a_verificar = 29

if es_primo(numero_a_verificar):
    print(f"{numero_a_verificar} es un número primo.")
else:
    print(f"{numero_a_verificar} NO es un número primo.")
```

11. **Escriba** un programa para imprimir los números pares de una lista determinada.

C: > Users > gabri > Downloads > # Solicitar al usuario que ingrese un nú.py > ...

```python
def imprimir_pares(lista):
    """Imprime los números pares de la lista dada."""
    pares = [numero for numero in lista if numero % 2 == 0]
    return pares

# Lista de ejemplo
lista_ejemplo = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

# Obtener y mostrar los números pares
numeros_pares = imprimir_pares(lista_ejemplo)
print(f"Números pares en la lista: {numeros_pares}")
```

12. **Escriba** una función que compruebe si una cadena frase o palabra pasada es palíndromo o no. Una palabra o frase que es palíndromo se lee igual de izquierda a derecha que de derecha a izquierda. Por ejemplo: Ana, Anita lava la tina.

C: > Users > gabri > Downloads > # Solicitar al usuario que ingrese un nú.py > ...

```python
def es_palindromo(frase):
    """Verifica si una cadena es un palíndromo."""
    # Normalizar la frase: quitar espacios y convertir a minúsculas
    frase_normalizada = ''.join(frase.lower().split())

    # Comprobar si la frase normalizada es igual a su reverso
    return frase_normalizada == frase_normalizada[::-1]

# Ejemplo de uso
frase_ejemplo = "Ana, Anita lava la tina"
if es_palindromo(frase_ejemplo):
    print(f'"{frase_ejemplo}" es un palíndromo.')
else:
    print(f'"{frase_ejemplo}" NO es un palíndromo.')
```