Trabajo práctico nro. 7

UTN FACULTAD REGIONAL MENDOZA	**Asignatura: Programación I**	
	Cursado: Primer Trimestre	**Horas semanales**: 4
	Carrera: *Tecnicatura Universitaria en Programación*	**Nivel (Año):**
	Ciclo Lectivo: 2023	☐ 1° ☐ 2° ☐ 3°

Integrantes de la Cátedra:

- **DOCENTES:**

Nombre del Profesor	Periodo	Cantidad horas materia
		6 horas

1. Escribe un programa que tome una lista de números como entrada y la ordene en orden ascendente utilizando el método de ordenamiento de burbuja.

```python
# Función para ordenar una lista en orden ascendente usando el método de burbuja
def order_bubble(lista):
    n = len(lista)
    for i in range(n):
        change = False
        for j in range(0, n - i - 1):
            if lista[j] > lista[j + 1]:
                lista[j], lista[j + 1] = lista[j + 1], lista[j]  # Intercambiar elementos
                change = True
        if not change:
            break  # Si no hay intercambios, la lista está ordenada

# Solicitar al usuario una lista de números
intos = input("Ingresa una lista de números separados por espacios: ")
numbers = [int(x) for x in intos.split()]  # Convertir la entrada a una lista de números

# Llamar a la función de ordenamiento de burbuja
order_bubble(numbers)

# Mostrar la lista ordenada
print("Lista ordenada en orden ascendente:", numbers)
```

2. Crea un programa que tome una lista de palabras como entrada y las ordene alfabéticamente en orden ascendente utilizando el método de ordenamiento de selección.

```python
# Función para ordenar una lista de palabras en orden ascendente usando el método de sele
def order_selection(lista):
    n = len(lista)
    for i in range(n - 1):
        min_idx = i
        for j in range(i + 1, n):
            if lista[j] < lista[min_idx]:
                min_idx = j
        lista[i], lista[min_idx] = lista[min_idx], lista[i]  # Intercambiar elementos
# Solicitar al usuario una lista de palabras
into = input("Ingresa una lista de palabras separadas por espacios: ")
words = into.split()  # Dividir la entrada en una lista de palabras
# Llamar a la función de ordenamiento de selección
order_selection(words)
# Mostrar la lista de palabras ordenada
print("Lista de palabras ordenada en orden ascendente:", " ".join(words))
```

3. Crea una lista de diccionarios, donde cada diccionario contiene información sobre un libro (título, autor, año de publicación, etc.). Luego, escribe un programa que ordene la lista de libros en función de un campo específico, como el año de publicación o el autor.

```python
# Crear una lista de diccionarios con información sobre libros
books = [
    {'titulo': 'Libro A', 'autor': 'Autor X', 'año_publicacion': 2010},
    {'titulo': 'Libro B', 'autor': 'Autor Y', 'año_publicacion': 2005},
    {'titulo': 'Libro C', 'autor': 'Autor Z', 'año_publicacion': 2019},
    # Agrega más libros según sea necesario
]
# Función para ordenar la lista de libros en función de un campo específico
def order_books(lista, field):
    return sorted(lista, key=lambda x: x[field])
# Solicitar al usuario el campo por el que desea ordenar los libros
order_field = input("Ingresa el campo por el que deseas ordenar los libros (titulo/autor/año_publicacion): ").lower()
# Verificar si el campo es válido
if order_field in books[0]:
    # Llamar a la función para ordenar la lista de libros
    ordered_books = order_books(books, order_field)
    # Mostrar la lista de libros ordenada
    for book in ordered_books:
        print(f"Título: {book['titulo']}, Autor: {book['autor']}, Año de Publicación: {book['año_publicacion']}")
else:
    print("El campo especificado no es válido.")
```

4. Escribe un programa que tome una lista de cadenas como entrada y las ordene en orden ascendente según su longitud. Puedes utilizar el método de ordenamiento de inserción.

```python
# Función para ordenar una lista de cadenas por longitud usando el método de inserción
def order_for_length(lista):
    for i in range(1, len(lista)):
        actual_word = lista[i]
        j = i - 1
        while j >= 0 and len(actual_word) < len(lista[j]):
            lista[j + 1] = lista[j]
            j -= 1
        lista[j + 1] = actual_word

# Solicitar al usuario una lista de cadenas
into = input("Ingresa una lista de cadenas separadas por espacios: ")
strings = into.split()  # Dividir la entrada en una lista de cadenas

# Llamar a la función de ordenamiento por longitud
order_for_length(strings)

# Mostrar la lista de cadenas ordenada por longitud
print("Lista de cadenas ordenada por longitud:", strings)
```

5. Modifica uno de los ejercicios anteriores para ordenar la lista de números en orden descendente en lugar de ascendente.

```python
# Solicitar al usuario una lista de números
into = input("Ingresa una lista de números separados por espacios: ")
numbers = [int(x) for x in into.split()]  # Convertir la entrada a una lista de números

# Ordenar la lista de números en orden descendente
ordered_numbers = sorted(numbers, reverse=True)

# Mostrar la lista de números ordenada en orden descendente
print("Lista de números ordenada en orden descendente:", ordered_numbers)

```

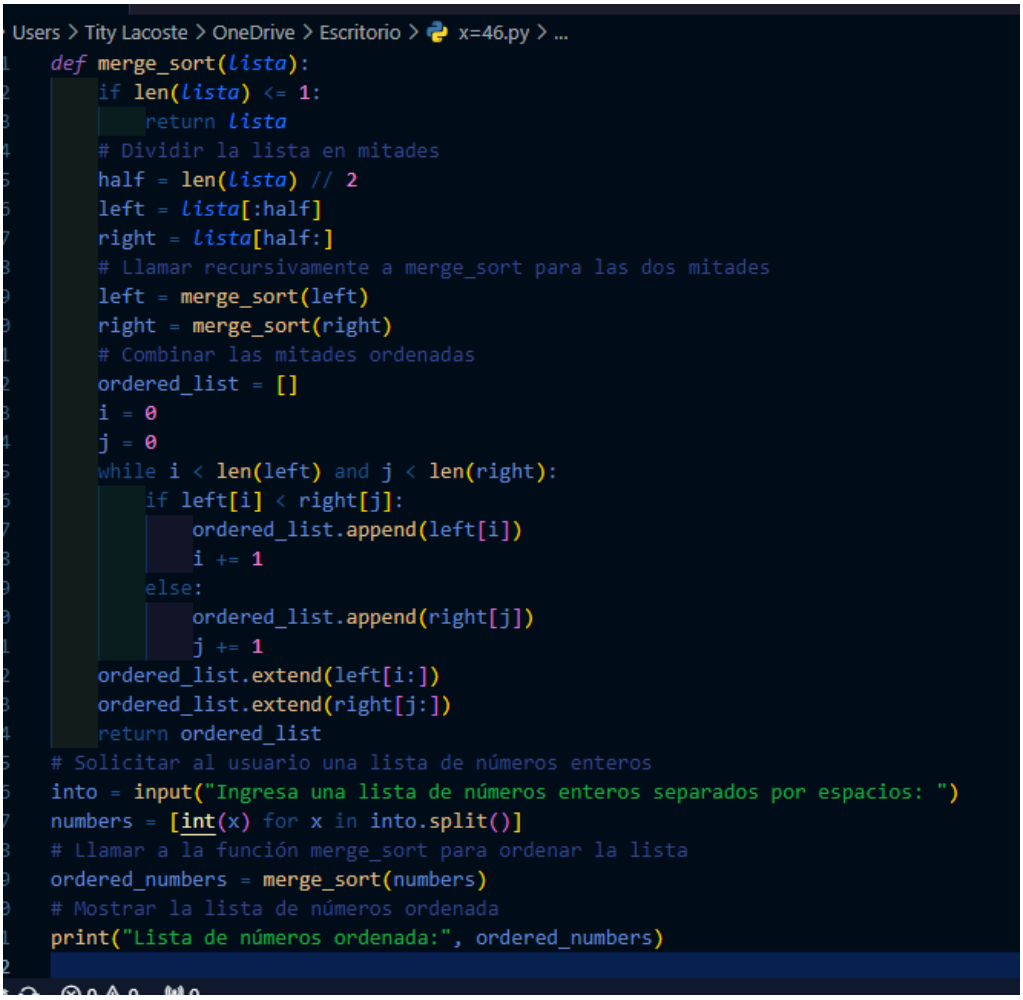
6. Escribe un programa que tome una lista de números enteros y ordene la lista utilizando el algoritmo de ordenamiento por conteo.

```
> Users > Tity Lacoste > OneDrive > Escritorio > x=46.py > ...
def count_order(lista):
    # Encontrar el valor máximo y mínimo en la lista
    max_value = max(lista)
    min_value = min(lista)

    # Crear una lista de conteo con un tamaño igual al rango de valores
    rango = max_value - min_value + 1
    count = [0] * rango

    # Contar la frecuencia de cada número en la lista
    for numb in lista:
        count[numb - min_value] += 1

    # Reconstruir la lista ordenada a partir de la lista de conteo
    ordered_list = []
    for i in range(rango):
        ordered_list.extend([i + min_value] * count[i])

    return ordered_list

# Solicitar al usuario una lista de números enteros
into = input("Ingresa una lista de números enteros separados por espacios: ")
numbers = [int(x) for x in into.split()]

# Llamar a la función de ordenamiento por conteo
ordered_numbers = count_order(numbers)

# Mostrar la lista de números ordenada
print("Lista de números ordenada:", ordered_numbers)

```

7. Crea una lista que contenga tanto números como cadenas de caracteres. Escribe un programa que ordene esta lista de manera que primero estén los números ordenados de menor a mayor y luego las cadenas de caracteres ordenadas alfabéticamente.

```python
# Crear una lista que contenga números y cadenas de caracteres
my_list = [5, 'manzana', 2, 'banana', 10, 'uva', 'pera']
# Separar los números y las cadenas de caracteres en listas diferentes
numbers = [x for x in my_list if isinstance(x, (int, float))]
strings = [x for x in my_list if isinstance(x, str)]

# Ordenar la lista de números de menor a mayor
ordered_numbers = sorted(numbers)

# Ordenar la lista de cadenas de caracteres alfabéticamente
ordered_strings = sorted(strings)

# Combinar las listas ordenadas
ordered_list = ordered_numbers + ordered_strings

# Mostrar la lista ordenada
print("Lista ordenada:", ordered_list)

```

8. Implementa el algoritmo de ordenamiento Merge Sort y úsalo para ordenar una lista de números.


Users > Tity Lacoste > OneDrive > Escritorio > x=46.py > ...
def merge_sort(lista):
if len(lista) <= 1:
return lista
# Dividir la lista en mitades
half = len(lista) // 2
left = lista[:half]
right = lista[half:]
# Llamar recursivamente a merge_sort para las dos mitades
left = merge_sort(left)
right = merge_sort(right)
# Combinar las mitades ordenadas
ordered_list = []
i = 0
j = 0
while i < len(left) and j < len(right):
if left[i] < right[j]:
ordered_list.append(left[i])
i += 1
else:
ordered_list.append(right[j])
j += 1
ordered_list.extend(left[i:])
ordered_list.extend(right[j:])
return ordered_list
# Solicitar al usuario una lista de números enteros
into = input("Ingresa una lista de números enteros separados por espacios: ")
numbers = [int(x) for x in into.split()]
# Llamar a la función merge_sort para ordenar la lista
ordered_numbers = merge_sort(numbers)
# Mostrar la lista de números ordenada
print("Lista de números ordenada:", ordered_numbers)