Ramirez Manriquez Luis Fernando

Practica 12

Buttom up

main.py

```python
def fibonacci_bottom_up(n):
    if n == 0:
        return 0
    if n == 1:
        return 1

    f = [0, 1]

    for i in range(2, n + 1):
        f.append(f[i - 1] + f[i - 2])

    return f[n]

if __name__ == "__main__":
    num = 10
    print(f"Fibonacci Bottom-Up de {num} =", fibonacci_bottom_up(num))


```

```
Fibonacci Bottom-Up de 10 = 55


...Program finished with exit code 0
Press ENTER to exit console.
```

Top down

main.py

```python
memoria = {}

def fibonacci_top_down(n):
    if n in memoria:
        return memoria[n]

    if n == 0:
        memoria[0] = 0
        return 0
    if n == 1:
        memoria[1] = 1
        return 1

    valor = fibonacci_top_down(n - 1) + fibonacci_top_down(n - 2)
    memoria[n] = valor
    return valor


if __name__ == "__main__":
    num = 10
    print(f"Fibonacci Top-Down de {num} =", fibonacci_top_down(num))
    print("Memoria usada:", memoria)

```

input

```
Fibonacci Top-Down de 10 = 55
Memoria usada: {1: 1, 0: 0, 2: 1, 3: 2, 4: 3, 5: 5, 6: 8, 7: 13, 8: 21, 9: 34, 10: 55}


...Program finished with exit code 0
Press ENTER to exit console.
```

Divide & Conquer

main.py

```python
def quicksort(lista):
    if len(lista) <= 1:
        return lista

    pivote = lista[len(lista) // 2]
    menores = []
    iguales = []
    mayores = []

    for x in lista:
        if x < pivote:
            menores.append(x)
        elif x == pivote:
            iguales.append(x)
        else:
            mayores.append(x)

    return quicksort(menores) + iguales + quicksort(mayores)


if __name__ == "__main__":
    datos = [5, 2, 9, 1, 7, 3]
    print("Lista original:", datos)
    print("Lista ordenada (QuickSort):", quicksort(datos))

```

```
Lista original: [5, 2, 9, 1, 7, 3]
Lista ordenada (QuickSort): [1, 2, 3, 5, 7, 9]


...Program finished with exit code 0
Press ENTER to exit console.
```