10. Функциональный Паттерн "Стратегия" для Сортировки-Разработать функцию, которая принимает стратегию сортировки (например, сортировка пузырьком, сортировка слиянием) как аргумент.

**def** bubble\_sort(*arr*):

*n* = len(*arr*)

    for *i* in range(*n*):

        for *j* in range(0, *n*-*i*-1):

            if *arr*[*j*] > *arr*[*j*+1]:

*arr*[*j*], *arr*[*j*+1] = *arr*[*j*+1], *arr*[*j*]

**def** merge\_sort(*arr*):

    if len(*arr*) > 1:

*mid* = len(*arr*) // 2

*L* = *arr*[:*mid*]

*R* = *arr*[*mid*:]

        merge\_sort(*L*)

        merge\_sort(*R*)

*i* = *j* = *k* = 0

        while *i* < len(*L*) and *j* < len(*R*):

            if *L*[*i*] < *R*[*j*]:

*arr*[*k*] = *L*[*i*]

*i* += 1

            else:

*arr*[*k*] = *R*[*j*]

*j* += 1

*k* += 1

        while *i* < len(*L*):

*arr*[*k*] = *L*[*i*]

*i* += 1

*k* += 1

        while *j* < len(*R*):

*arr*[*k*] = *R*[*j*]

*j* += 1

*k* += 1

**def** sort(*strategy*, *arr*):

    if *strategy* == "bubble":

        bubble\_sort(*arr*)

    elif *strategy* == "merge":

        merge\_sort(*arr*)

    else:

        print("Unknown sorting strategy")

*arr* = [64, 34, 25, 12, 22, 11, 90]

print("Before sorting:", *arr*)

sort("bubble", *arr*)

print("After bubble sort:", *arr*)

*arr* = [64, 34, 25, 12, 22, 11, 90]

print("Before sorting:", *arr*)

sort("merge", *arr*)

print("After merge sort:", *arr*)

