Muhammad Irfan
Reg No. FA21-BSE-027
Section A
Lab task 3

Step 1: Error Checklist Selection

Select the **"An Error Checklist for Inspections"** from *The Art of Software Testing* by Glenford J. Myers. This checklist focuses on fundamental programming errors, making it a good fit for various algorithms. You can find a detailed version of this checklist online.

Step 2: Algorithm Selection

choose the **Bubble Sort** algorithm:

- **Complexity:** It has a reasonable level of complexity for demonstrating checklist application.
- **Error Potential:** Prone to off-by-one errors, incorrect loop termination, and inefficient handling of nearly sorted data. These align well with the checklist.

Step 3: Algorithm Versions

A. Initial 'Flawed' Bubble Sort

```
def bubble_sort_flawed(arr):
    for i in range(len(arr)):  # Off-by-one potential
        for j in range(len(arr) - 1):  # No optimization for nearly sorted
            if arr[j] > arr[j + 1]:
                arr[j], arr[j + 1] = arr[j + 1], arr[j]
```

B. Refined Bubble Sort

```
def bubble_sort_refined(arr):
    swapped = True
    while swapped:
        swapped = False
        for j in range(len(arr) - 1):
            if arr[j] > arr[j + 1]:
                arr[j], arr[j + 1] = arr[j + 1], arr[j]
                swapped = True
```

Step 4: Test Cases

Guideline from Checklist	**Valid Test Case**	**Invalid Test Case**
Are all variables initialized before use?	Input: [6, 3, 1, 8] Expected: [1, 3, 6, 8]	Input: [ ] (empty) Expected: Handle gracefully
Are loop termination conditions correct?	Input: [2, 1, 5] Expected: [1, 2, 5]	Input: [1, 2, 3] (already sorted) Expected: Early termination