Recording

Date : 21/06/2025

Problem Specification : Write a program that generates an array of numbers from user input. Implement methods to search for a specific number in the array and return its index, and to sort the array in ascending order using a sorting algorithm.

Assumption : The user will enter valid integer values for the number of elements and the array elements themselves.

Limitation : The program only accepts integer inputs. It will throw an error if non-integer data is entered. The search function as written will only find the first occurrence of the specified number. The "not found" message is displayed incorrectly inside the search loop.

Input :

- An integer representing the number of elements in the array.
- A series of integers to populate the array.
- An integer to be searched for within the array.

Processing :

- The program prompts the user for the size of the array and then the elements of the array.
- It then asks for a number to search for.
- A linear search is performed on the array to find the index of the first instance of the search number.
- A bubble sort algorithm is used to sort the array in ascending order.
- The results of the search and the sorted array are printed to the console.

Output :

- The index of the searched number if found, or a message indicating the number was not found.
- The array sorted in ascending order.

Algorithm :

Search Algorithm (Linear Search)

- **Step 1:** Read the array `numbers`, and the `searchNumber`.
- **Step 2:** Initialize a counter, `i` -> 0.
- **Step 3:** Start a loop that iterates from `i` to the last index of the array.
- **Step 4:** In each iteration, check if `numbers[i]` is equal to `searchNumber`.
- **Step 5:** If they are equal, print the index `i` and terminate the loop.
- **Step 6:** If they are not equal and it is the last element of the array, print a "not found" message.
- **Step 7:** End the programme.

Sort Algorithm (Bubble Sort)

- **Step 1:** Read the array `numbers`.
- **Step 2:** Start an outer loop with a counter `i` from 0 to `n-2`, where `n` is the array size.
- **Step 3:** Start an inner loop with a counter `j` from 0 to `n-i-2`.
- **Step 4:** Check if the element `numbers[j]` is greater than `numbers[j+1]`.
- **Step 5:** If it is greater, swap the elements `numbers[j]` and `numbers[j+1]`.
- **Step 6:** Continue until the inner loop is finished.
- **Step 7:** Continue until the outer loop is finished.
- **Step 8:** End the programme.

Programme listing : Programme file attached.

Test data and expected output :

- **Test data**:
  - Array size: 5
  - Array elements: `[5, 3, 8, 1, 2]`
  - Number to search for: `3`
- **Expected output**:
  - `Index of 3 is 1`
  - `Sorted array: [1, 2, 3, 5, 8]`

Output obtained for test data :

- **Test data**:
  - Array size: 5
  - Array elements: `[5, 3, 8, 1, 2]`
  - Number to search for: `3`
- **Obtained output**:
  - `Index of 3 is 1`
  - `Sorted array: [1, 2, 3, 5, 8]`

Analysis :

- **Search:** The linear search loop involves one comparison (==) per element in the worst-case scenario. The number of operations is proportional to the size of the array, `N`.

- **Sort:** The bubble sort uses nested loops. The outer loop runs N−1 times, and the inner loop runs up to N−1 times. It involves one comparison (>) and three assignment operations for a swap in the inner loop.

Conclusion : This programme successfully accepts an array of integers from a user, searches for a specified number using linear search, and sorts the array in ascending order using bubble sort, displaying the results.

Discussion **:** The algorithms used are functional but not the most efficient. The linear search has a time complexity of O(n). The implemented bubble sort has a time complexity of O(n2), which is inefficient for large arrays. A more efficient sorting algorithm like Merge Sort or Quick Sort, or using the built-in Arrays.sort() method, would provide better performance. Additionally, the logic for handling a number not found in the array is flawed and should be corrected to display the message only once after the search loop has completed without finding the element.