LAB211 Assignment

Type:	**Short Assignment**
Code:	**J1.S.P0053**
LOC:	**42**
Slot(s):	**1**

Title

Sort one-dimensional array with bubble sort algorithm.

Background

N/A

Program Specifications

Design a program for to input requirement

- Number of items (elements) of the one-dimensional array.
- The value of items in the array must be integer.

Menu of program as the following:

1. Input items of the array
2. Sort the array in ascending order
3. Sort the array in descending order
4. Exit

Function details:

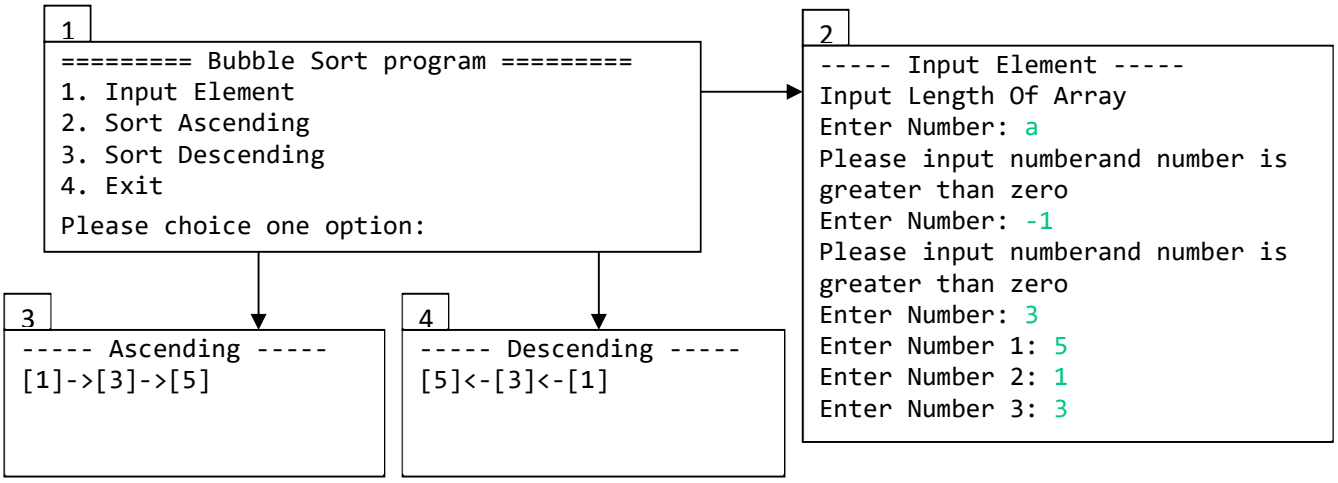
Function 1: Display a menu and ask a user to select an option.

- User runs the program. The program prompts users to select an option.
- User selects an option, performs **Function 2**.

Function 2: Perform function based on the selected option.

- Option 1: Input Element(s)
  a. Requirement for inputs task information including "length of array, number".
  b. Check the valid data with the following conditions:
     i. Length of the array must be more than 0.
  c. Save element(s) of array.
  d. Return to the home screen.
- Option 2: Sort in ascending order
  a. Display element of the array in ascending order
  b. Return to the home screen
- Option 3: Sort in Descending order
  a. Display element of the array in a descending order
  b. Return to the home screen
- Option 4: Exit the program

Expectation of User interface:



Guidelines

Suggestion:

Implement methods

- checkln
- sortAscending
- sortDescending

in startup code.

Bubble sort is a simple sorting algorithm that repeatedly steps through the list to be sorted, compares each pair of adjacent items and swaps them if they are in the wrong order. The pass through the list is repeated until no swaps are needed, which indicates that the list is sorted. The algorithm, which is a comparison sort, is named for the way smaller elements "bubble" to the top of the list.

Format 1: Sort from the beginning to the end. Start from the beginning of the row, compare and change positions of each pair to the end the row to move the biggest elements to the end. Then check the remaining elements in the row and repeat the process for arranging.

Format 2: Sort from the end to the beginning. Start from the end of the row & change positions of each pair to move smaller element of the element pair to the start position of the existing row. Then do not check it in the next step. Repeat above steps until no any pair remained.

Function 1: Initiate an array and input array's items.

- The program will asks about number of elements and requires inputting elements.
- Check the input through the "public integer chechIn(String inputVal)" function.
- Implement the function: public Integer checkIn(String inputVal).
  - InputVal: input value
  - Return value : the number or null

Function 2: Sort in ascending array.

- The program to change positions of items so the result is an array containing the elements sorted in ascending order.
- Implement function: int[] sortAscending(int[] arrayNeedSort)
  - Input:

- arrayNeedSort: array needs to be sorted

  - Return value: it is the array containing the elements sorted in ascending order

Function 3:.Sort in descending array

- The program to change positions of elements resulting an array containing the elements sorted in descending order

- Implement the function: int[] sortDescending(int[] arrayNeedSort)

  - Input:

    - arrayNeedSort: the array needs to be sorted

  - Return value: it is the array containing the elements sorted in descending order