



Activity Sheet 04B
Implementation of Array Operations using Java

findLocationDelete()

- Search for the position in which ITEM will be deleted
- Return the position

Note: You are to decide what parameters are to be passed to the called method.

main()

- Display an option similar to this
 1. Insert a value
 2. Delete a value
 3. Traverse array
 4. Exit
- Provide a switch statement in which cases for the shown options are written
- Also the array must be checked if it is already full or not

The screenshot displays the NetBeans IDE environment. The main editor window shows the source code for a Java application named 'Activity04B_NW2C_Perillo.java'. The code includes a package declaration, imports, and a public class with a main method. The main method uses a Scanner to read user input and a switch statement to handle four options: inserting a value, deleting a value, traversing the array, and exiting. The array is implemented as an ArrayList. The output window on the right shows the execution of the program, displaying the menu options and the results of user interactions, such as inserting and deleting elements, and traversing the array.

```
import java.util.Scanner;

public class Activity04B_NW2C_Perillo {
    public static void main(String[] args) {
        Act04BSortedArray_NW2C_Perillo sortedLinearArray = new Act04BSortedArray();
        Scanner in = new Scanner(System.in);
        String choice;
        while (true) {
            System.out.println("\n1. Insert a value"
                               + "\n2. Delete a value"
                               + "\n3. Traverse the array"
                               + "\n4. Exit");
            System.out.print("Enter your choice: ");
            choice = in.nextLine();
            switch (choice) {
                case "1":
                    System.out.print("Enter element to inserted: ");
                    sortedLinearArray.insertItem(Integer.parseInt(in.nextLine()));
                    break;
                case "2":
                    System.out.print("Enter element to deleted: ");
                    sortedLinearArray.deleteItem(Integer.parseInt(in.nextLine()));
                    break;
                case "3":
                    sortedLinearArray.printArray();
                    break;
                case "4":
                    System.exit(0);
            }
        }
    }
}
```

Output - assignment (run) #11 X

```
1. Insert a value
2. Delete a value
3. Traverse the array
4. Exit
Enter your choice: 1
Enter element to inserted: 25

1. Insert a value
2. Delete a value
3. Traverse the array
4. Exit
Enter your choice: 1
Enter element to inserted: 45

1. Insert a value
2. Delete a value
3. Traverse the array
4. Exit
Enter your choice: 3
The elements of the array: 25 45

1. Insert a value
2. Delete a value
3. Traverse the array
4. Exit
Enter your choice: 2
Enter element to deleted: 45

1. Insert a value
2. Delete a value
3. Traverse the array
4. Exit
Enter your choice: 3
The elements of the array: 25

1. Insert a value
2. Delete a value
3. Traverse the array
4. Exit
Enter your choice: 4
BUILD SUCCESSFUL (total time: 59 seconds)
```



The screenshot displays the NetBeans IDE environment. The main editor window shows the source code for the file `Act04BSortedArray_NW2C_Perillo.java`. The code is as follows:

```
package dsaa;

public class Act04BSortedArray_NW2C_Perillo {
    private int[] array;
    private int currentSize;

    public Act04BSortedArray_NW2C_Perillo() {
        this(10);
    }

    public Act04BSortedArray_NW2C_Perillo(int size) {
        array = new int[size];
        currentSize = 0;
    }

    public void insertItem(int item) {
        if (currentSize < array.length) {
            int insertLocation = findLocationInsert(item);
            for (int i = currentSize - 1; i >= insertLocation; i--) {
                array[i + 1] = array[i];
            }
            array[insertLocation] = item;
            currentSize++;
        }
    }

    private int findLocationInsert(int item) {
        for (int i = 0; i < currentSize; i++) {
            if (array[i] > item) {
                return i;
            }
        }
    }
}
```

The right-hand pane shows the output of the program, titled "Output - assignment (run) #11". The output consists of several test cases, each starting with a menu of options: "1. Insert a value", "2. Delete a value", "3. Traverse the array", "4. Exit". The user's choice is indicated by "Enter your choice: [number]".

- Test case 1: Choice 1, "Enter element to inserted: 25".
- Test case 2: Choice 1, "Enter element to inserted: 45".
- Test case 3: Choice 3, "The elements of the array: 25 45".
- Test case 4: Choice 2, "Enter element to deleted: 45".
- Test case 5: Choice 3, "The elements of the array: 25".
- Test case 6: Choice 4.

The output concludes with the message "BUILD SUCCESSFUL (total time: 59 seconds)".