**Chapter 7 Project 1 Test Plan**

**Program Goals and Objectives**

The purpose of this program is to read an unspecified number of input integers between 1 and 100 and output how many times each integer occurs. The program has to end with the input 0.

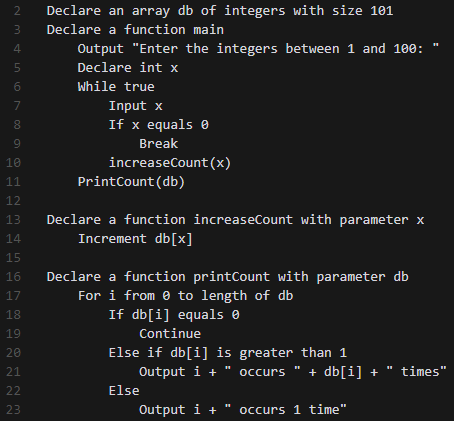
**Program Functional Requirements**

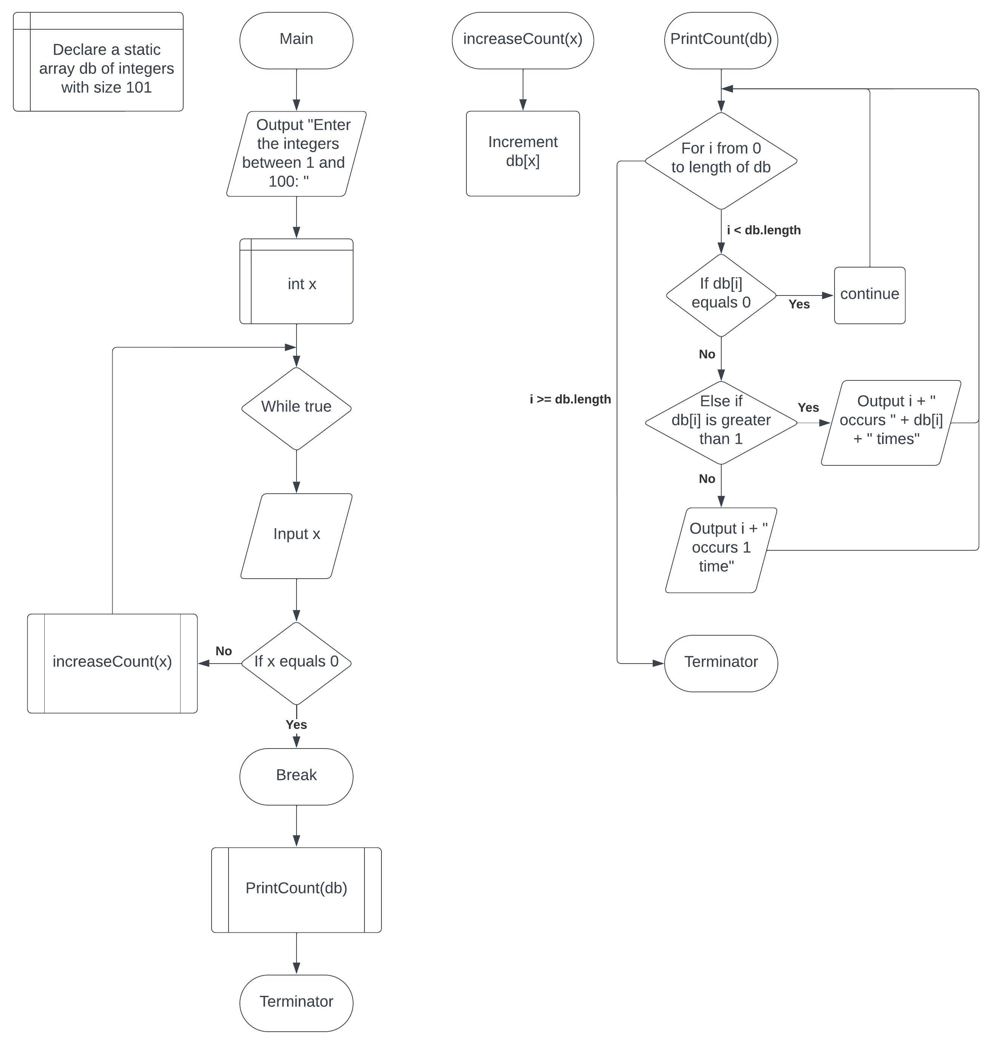
1. The user needs to be prompted for integer inputs between 1 and 100.
2. The input must be repeated infinitely until 0 is entered.
3. When 0 is finally entered, the program must output the number of times each input integer occurred.
4. If an integer appears more than once, the program must say “integer occurs x times”, otherwise it must say “integer occurs 1 time”.
5. The program must display the output in increasing order of the input integers.

**Test Matrix**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case | Input | Expected Result | Actual Result | Outcome (pass/fail) |
| 1 | 2 5 6 5 4 3 23 43 2 0 | 2 occurs 2 times  3 occurs 1 time  4 occurs 1 time  5 occurs 2 times  6 occurs 1 time  23 occurs 1 time  43 occurs 1 time | 2 occurs 2 times  3 occurs 1 time  4 occurs 1 time  5 occurs 2 times  6 occurs 1 time  23 occurs 1 time  43 occurs 1 time | pass |
| 2 | -5 -10 -10 -1 10 10 -1 0 | Error message | Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: Index -5 out of bounds for length 101 | fail |
| 3 | Five 5 1 1 0 | Error message | Exception in thread "main" java.util.InputMismatchException | fail |
| 4 | 2 5 6 5 4 110 0 | Error message | Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: Index 110 out of bounds for length 101 | fail |

**Pseudocode / Flowchart**





**Chapter 7 Project 2 Test Plan**

**Program Goals and Objectives**

The purpose of this program is to evaluate whether a user input list is sorted in increasing order using a Boolean method.

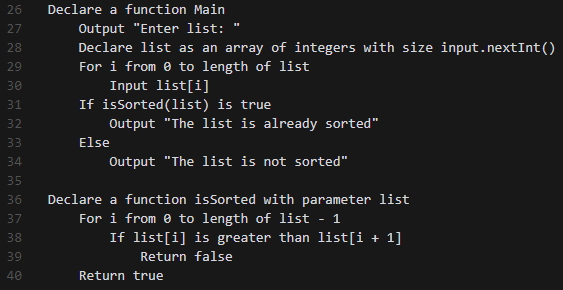
**Program Functional Requirements**

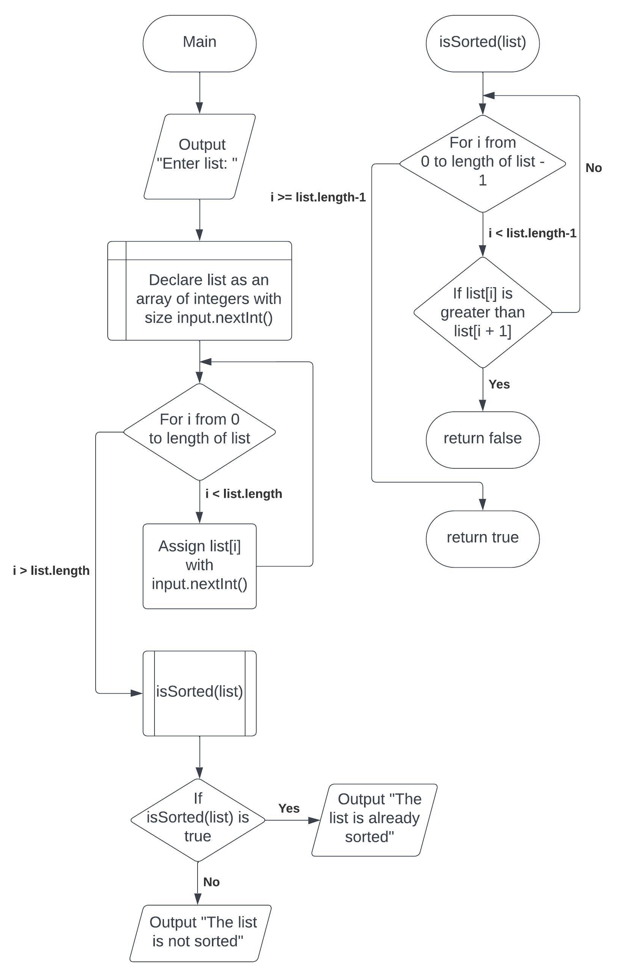
1. The user needs to be prompted for integer inputs to make up a list.
2. The first input from the user will represent the length of the list and will not be part of the list.
3. The program will then invoke a method using “public static boolean isSorted(int[] list)” header to determine if the input list is sorted in increasing order.
4. If the method evaluates the list to be sorted, the program will output “The list is already sorted”, otherwise it will output “The list is not sorted”

**Test Matrix**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case | Input | Expected Result | Actual Result | Outcome (pass/fail) |
| 1 | 8 10 1 5 16 61 9 11 1 | The list is not sorted | The list is not sorted | pass |
| 2 | 10 1 1 3 4 4 5 7 9 11 21 | The list is already sorted | The list is already sorted | pass |
| 3 | 4 1.5 3 4.5 6 | Error message | Exception in thread "main" java.util.InputMismatchException | fail |
| 4 | Four 1 4 5 7 | Error message | Exception in thread "main" java.util.InputMismatchException | fail |

**Pseudocode / Flowchart**





**Chapter 8 Project 1 Test Plan**

**Program Goals and Objectives**

The purpose of this program is to return the location of the largest element in a user inputted two-dimensional array. The two-dimensional array size will be provided by user input as well.

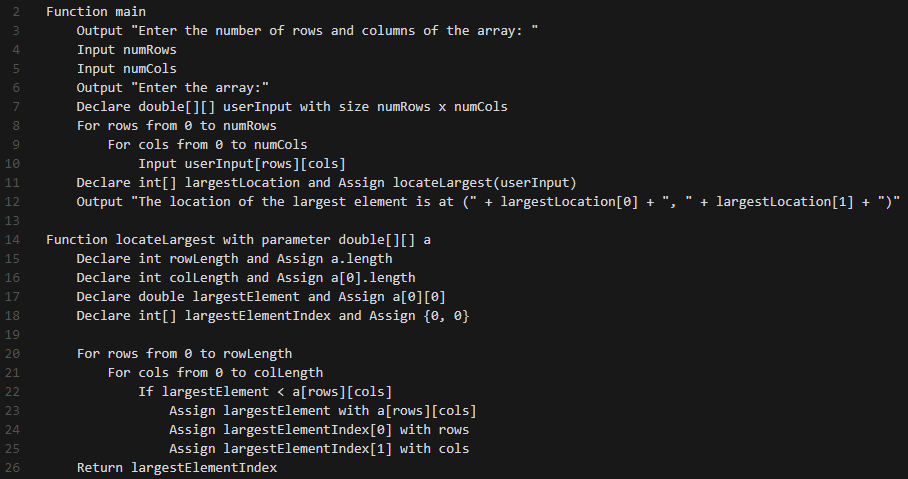
**Program Functional Requirements**

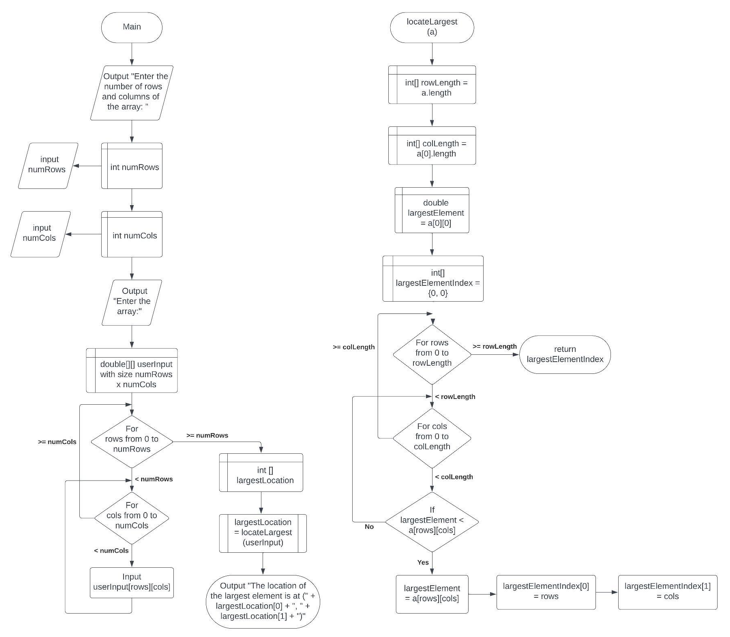
1. The user needs to be prompted for the number of rows and columns the array will have.
2. The user will be prompted to enter a two-dimensional array with as many values as there are rows \* columns.
3. The program will then execute a method with the header “public static int[] locateLargest(double[][] a)” that returns an integer array with 2 values representing the location of the largest element in the two-dimensional array.
4. If there is more than one largest element, the method will return the one with the smallest row and column index.
5. The program will then output the result from the locateLargest method.

**Test Matrix**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case | Input | Expected Result | Actual Result | Outcome (pass/fail) |
| 1 | 3 4 23.5 35 2 10 4.5 3 45 3.5 35 44 5.5 9.6 | The location of the largest element is at (1, 2) | The location of the largest element is at (1, 2) | pass |
| 2 | 3 4 23.5 35 2 45 4.5 3 45 3.5 35 44 5.5 9.6 | The location of the largest element is at (0, 3) | The location of the largest element is at (0, 3) | pass |
| 3 | 2 3  -5 -3 0  1 100 -3 | The location of the largest element is at (1, 1) | The location of the largest element is at (1, 1) | pass |
| 4 | 0 1  1 | Error message | Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: Index 0 out of bounds for length 0 | fail |
| 5 | 3.5 4  1 | Error message | Exception in thread "main" java.util.InputMismatchException | fail |

**Pseudocode / Flowchart**





**Chapter 8 Project 2 Test Plan**

**Program Goals and Objectives**

The purpose of this program is to prompt the user for a number of cities and their respective coordinates and then calculate the central city, in other words the city with the least distance to all other cities. The program must output the coordinates of the central city along with its total distance to all other cities.

**Program Functional Requirements**

1. The user needs to be prompted for an integer number of cities to input
2. The user needs to be prompted to input the coordinates of each city, as many times are there are cities from the previous prompt.
3. The program must evaluate the distance from each city to each other city and sum the total distances.
4. The Program must output the coordinates of the central city with the shortest total distance to each other city and the coordinates of the central city.

**Test Matrix**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case | Input | Expected Result | Actual Result | Outcome (pass/fail) |
| 1 | 5 2.5 5 5.1 3 1 9 5.4 54 5.5 2.1 | The central city is at (2.5, 5.0) The total distance to all other cities is 60.81 | The central city is at (2.5, 5.0) The total distance to all other cities is 60.810516285521615 | pass |
| 2 | 4  1 1 2 2 3 3 4 4 | The central city is at (2.0, 2.0)  The total distance to all other cities is x | The central city is at (2.0, 2.0)  The total distance to all other cities is 5.656854249492381 | pass |
| 3 | Five  1 1 2 2 3 3 4 4 | Error message | Exception in thread "main" java.util.InputMismatchException | fail |
| 4 | 4  -2 -3 -5 -7 1 3 10 10 | The central city is at (-2.0, -3.0)  The total distance to all other cities is x | The central city is at (-2.0, -3.0)  The total distance to all other cities is 29.4000099454535 | pass |

**Pseudocode / Flowchart**

