Goals & Objectives

The goal of this program is to accepts input from a user in the form of integers and specifies the size of a list, and a list of integers. The program will output a confirmation of the inputs and determine whether the list is sorted or not.

Functional Requirements

1. Prompt user input for size of list
2. Prompt user to enter contents of the list
3. Store each item in int array
4. Pass list to method isSorted
5. Iterate through list and determine if list is sorted
6. Output whether list is sort or list in not sorted

Pseudocode

Import Scanner Utility

Function Main {

Declare input as new Scanner

Output “Enter the size of the list:”

Declare size as input.nextInt()

Declare int array list with a size of size

Output “Enter the contents of the list”

Iterate through input, and store each item as input.nextInt();

Close input

Output “The list has “ x “ integers “ + list

IF method isSorted returns false

Output “The list is not sorted”

ELSE

Output “The list is already sorted”

End

Function isSorted( requires int array list )

FOR I = 0, I <= list.length -2, i++

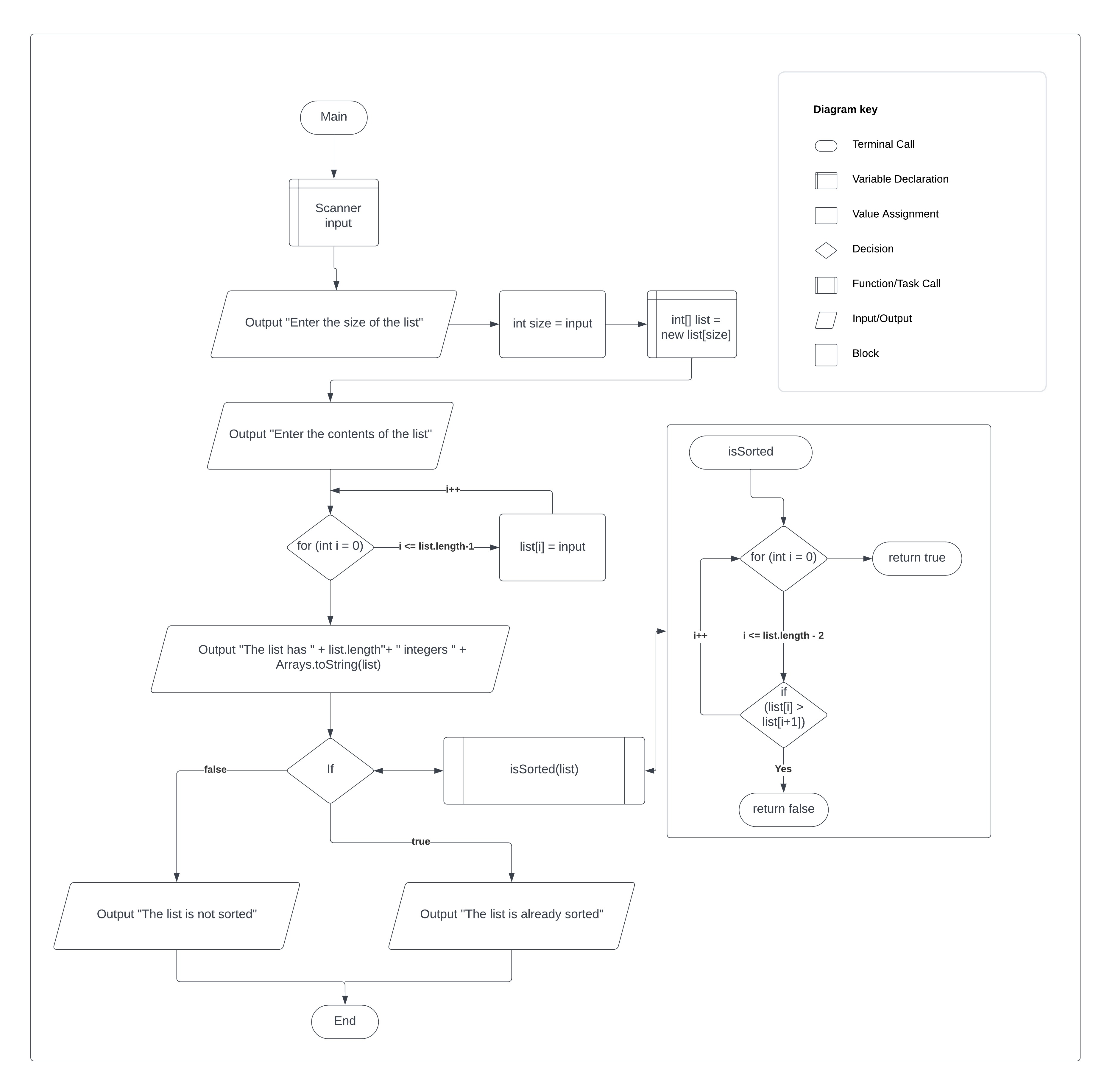
IF list[i] > list[i+1]

Return false

Return true

End

Flowchart



Test Plan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case** | **Input/Output** | **Expected Result** | **Actual Result** | **Outcome (Pass/Fail)** |
| 1a | Prompt user for the size of the list | User inputs an integer for the size of the list | “Enter the size of the list “ | Pass |
| 2a | Prompt user for the list contents | User inputs x integers to fill the list | “Enter the contents of the list: “ | Pass |
| 3a | Store each input item in list | Each input gets stored in the list | for (int I = 0; I <= list.length – 1; i++) { list[i] = input.nextInt(); } | Pass |
| 4a | Pass list to method isSorted | Int array list gets passed to method isSorted | If (isSorted(list) == false) { | Pass |
| 5a | Iterate through list to determine if list is sorted | Method checks current int if greater than the next int. Returns true or false | for (int I = 0; I <= list.length -2; i++) {if (list[i] > list[+1]) { return false} }  return true | Pass |
| 6a | Output whether the list is sorted or not | Outputs “The list is not sorted” if method returns false and outputs “The list is already sorted” if method returns true | If (isSorted(list) == false) { output “The list is not sorted” } else { “The list is already sorted” } | Pass |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case** | **Input/Output** | **Expected Result** | **Actual Result** | **Outcome (Pass/Fail)** |
| 1 | (1a) 4  (2a) 9 3 1 1 | The list has 4 integers [9, 3, 1, 1]  The list is not sorted | The list has 4 integers [9, 3, 1, 1]  The list is not sorted | Pass |
| 2 | (1a) 5  (2a) -1 0 1 2 3 | The list has 5 integers [-1, 0, 1, 2, 3]  The list is already sorted | The list has 5 integers [-1, 0, 1, 2, 3]  The list is already sorted | Pass |
| 3 | (1a) 3  (2a) 4 9 5 | The list has 3 integers [4, 9, 5]  The list is not sorted | The list has 3 integers [4, 9, 5]  The list is not sorted | Pass |
| 4 | (1a) 8  (2a) 10 1 5 16 61 9 11 1 | The list has 8 integers [10, 1, 5, 16, 61, 9, 11, 1]  The list is not sorted | The list has 8 integers [10, 1, 5, 16, 61, 9, 11, 1]  The list is not sorted | Pass |

A screenshot of a computer program

Description automatically generated