Goals & Objectives

The goal of this program is to accept user input and build a two-dimensional array and add elements to the array from the inputs. The program will then iterate through the array and locate the largest element, and output the largest element, including the array index (x and y) to the user.

Functional Requirements

1. Prompt user input [row] and [column] sizes of the array
2. Prompt user input for the elements of the array
3. Add elements to the array
4. Pass array to method locateLargest
5. Store maximum value, row, and column of the largest element in class Location
6. Output the maximum value, row, and column of the largest element

Pseudocode

Import Scanner Utility

Function Main {

Declare input as new Scanner

Output “Enter the number of rows and columns in the array: “

Declare x as integer from input

Declare y as integer from input

Declare double array a with sizes of x and y

Output “Enter the array: “

FOR each x element

FOR each y element

Add input as double

Close input

Declare Location l as function locateLargest(a)

Output “the largest element is “ + l.maxValue + “, located at (“ +l.x + “,” + l.y +”)”

End

Function locateLargest( requires double array a )

Declare location as new Location

FOR (int I = 0; I <= a.length – 1; i++)

FOR (int j = 0; j <= a[i].length – 1; j++)

IF (a[i][j] > largestNumber OR inverse of [ai][j] < inverse of largestNumber)

Assign location.maxValue as a[i][j]

Assign location.x as i

Assign location.y as j

Return location

End

Class Location {

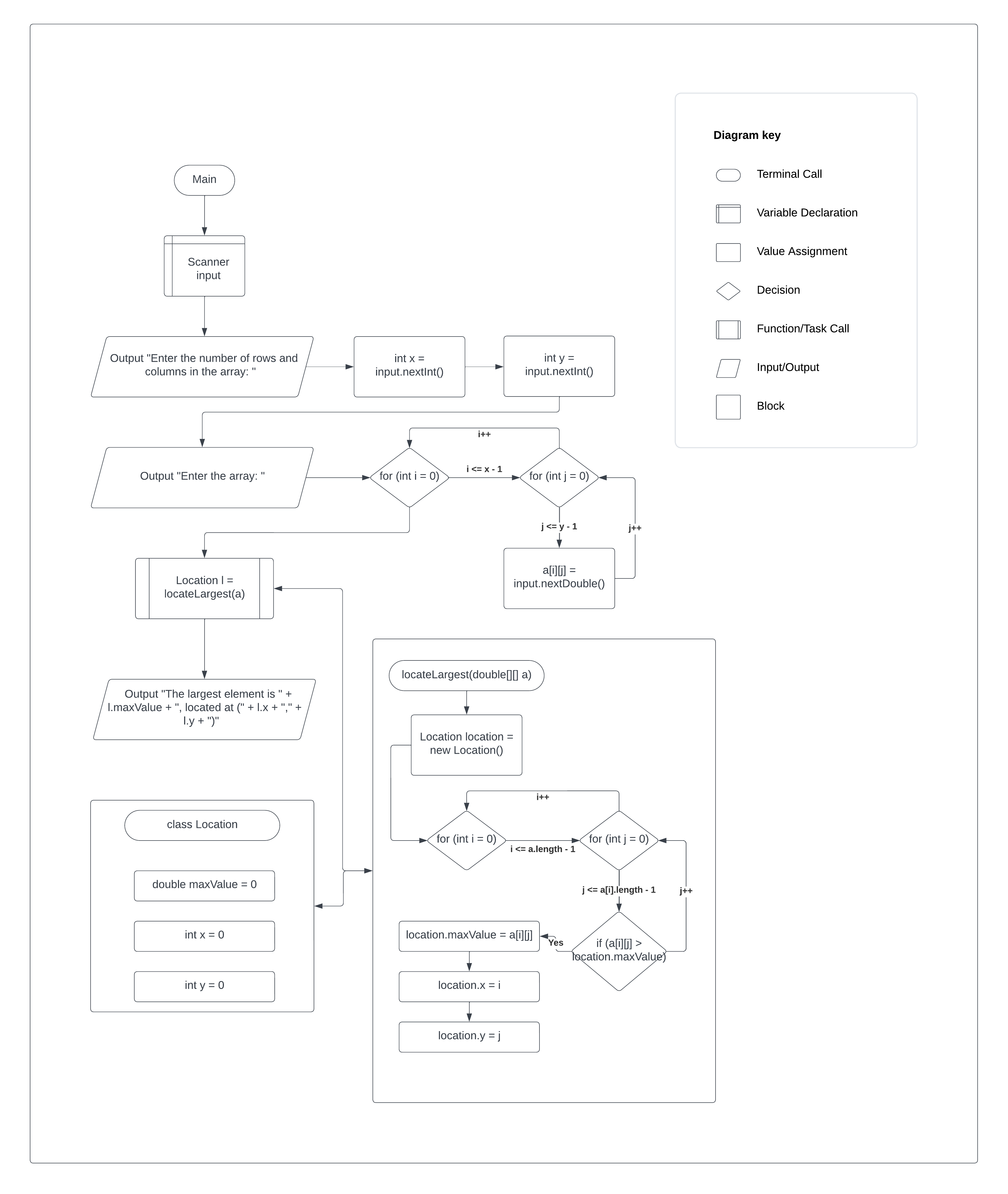
Declare double maxValue as 0

Declare int x as 0

Declare int y as 0

End

Flowchart



Test Plan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case** | **Input/Output** | **Expected Result** | **Actual Result** | **Outcome (Pass/Fail)** |
| 1a | Prompt user input [row] and [column] sizes of the array | User inputs two numbers for rows (x) and cols (y) | Output “Enter the number of rows and columns of the array: “  Int rows = input.nextInt()  Int cols = input.nextInt() | Pass |
| 2a | Prompt user input for the elements of the array | User enters (x \* y) elements for the array | Output “Enter the array:“ | Pass |
| 3a | Add elements to the array | Each element is added to the double array a | FOR (int I = 0; I <= a.length – 1; i++) {for (int j = 0; j <= a[i].length – 1; j++) {a[i][j] = input.nextDouble() } } | Pass |
| 4a | Pass array to method locateLargest | Array a is passed to method locateLargest | Location l = locateLargest(a) | Pass |
| 5a | Store maximum value, row, and column of the largest element in class Location | Method iterates through array and stores the largest element as maxValue in class Location, and stores row and col index as location.x and location.y and return the class to main | for (int i = 0; i <= a.length - 1; i++) {for (int j = 0; j <= a[i].length - 1; j++) {if (a[i][j] > largestNumber) {location.maxValue = a[i][j]; location.x = i; location.y = j; } } } return location | Pass |
| 6a | Output the maximum value, row, and column of the largest element | The largest element, and it’s index is output to the user | Output "The largest element is " + l.maxValue + ", located at (" + l.x + "," + l.y + ")" | Pass |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case** | **Input/Output** | **Expected Result** | **Actual Result** | **Outcome (Pass/Fail)** |
| 1 | (1a) 3 4  (2a) 23.5 35 2 10 4.5 3 45 3.5 35 44 5.5 9.6 | The largest element is 45.0, located at (1, 2) | The largest element is 45.0, located at (1, 2) | Pass |
| 2 | (1a) 1 5  (2a) -1 0 1 2 3 | The largest element is 3.0, located at (0, 4) | The largest element is 3.0, located at (0, 4) | Pass |
| 3 | (1a) 3 3  (2a) 3 6 9 1 3 4 10 2 8 | The largest element is 10.0, located at (2, 0) | The largest element is 10.0, located at (2, 0) | Pass |
| 4 | (1a) 2 2  (2a) 54 45 61 2 | The largest element is 61.0, located at (1, 0) | The largest element is 61.0, located at (1, 0) | Pass |

