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 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. Return [row] and [column] of the largest element
6. Output the [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 of the array: “

Declare rows as integer from input

Declare cols as integer from input

Declare double array a with sizes of rows and cols

Output “Enter the array: “

FOR each row element

FOR each column element

Add input as double

Close input

Declare int array b as method locateLargest

Output “The location of the largest element is at (“ x “,” y “)”

End

Function locateLargest( requires double array a )

Declare largestNumber as first element of a

Declare rowOfLargestNumber as integer 0

Declare colOfLargestNumber as integer 0

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 a[i][j] as largestNumber

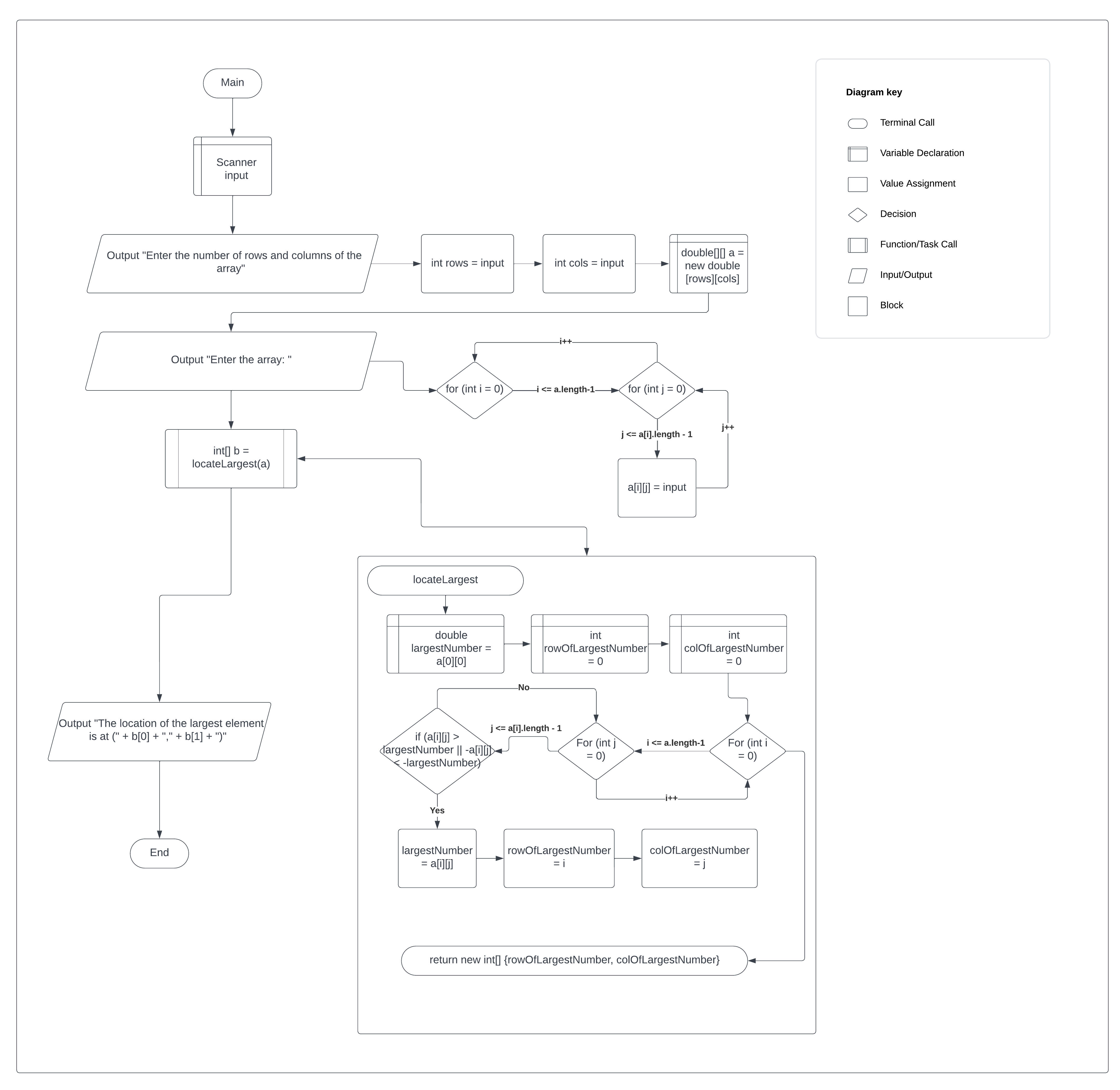
Assign rowOfLargestNumber as i

Assign colOfLargestNumber as j

Return int array with rowOfLargestNumber and colOfLargestNumber as elements

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 and cols | 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 (rows \* cols) 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 | Int[] b = locateLargest(a) | Pass |
| 5a | Return [row] and [column] of the largest element | Method iterates through a and locates the largest element of the array, and returns int array p back to method 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 || -a[i][j] < -largestNumber) {largestNumber = a[i][j]; rowOfLargestNumber = i; colOfLargestNumber = j; } } }  return p | Pass |
| 6a | Output the [row] and [column] of the largest element | The largest element’s row and column index is output to the user | Output “The location of the largest element is at (“ + b[0] + “, “ + b[1] + “)” | 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 location of the largest element is (1, 2) | The location of the largest element is (1, 2) | Pass |
| 2 | (1a) 1 5  (2a) -1 0 1 2 3 | The location of the largest element is (0, 4) | The location of the largest element is (0, 4) | Pass |
| 3 | (1a) 3 3  (2a) -1 -2 -3 -0.5 -1 -4 -6 -8 -0.25 | The location of the largest element is (2, 2) | The location of the largest element is (2, 2) | Pass |
| 4 | (1a) 2 2  (2a) 54 45 61 2 | The location of the largest element is (1, 0) | The location of the largest element is (1, 0) | Pass |

A screenshot of a computer program

Description automatically generated