**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **13-July-2020** | **Name:** | **Bhuvanesh M** |
| **Course:** | **Matlab Onramp** | **USN:** | **4AL16EC015** |
| **Topic:** | **Logical arrays in Matlab** | **Semester & Section:** | **8th sem & ‘A’ section** |
| **Github Repository:** | **Bhuvan** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| **Image of session** |
| **Report –**  In my first session today I have studied about –Logical arrays in Matlab **Reduce Logical Arrays to Single Value** Example  This example shows how to use the any and all functions to reduce an entire array to a single logical value.  The any and all functions are natural extensions of the logical | (OR) and & (AND) operators, respectively. However, rather than comparing just two elements, the any and all functions compare all of the elements in a particular dimension of an array. It is as if all of those elements are connected by & or | operators and the any or all functions evaluate the resulting long logical expressions. Therefore, unlike the core logical operators, the any and all functions reduce the size of the array dimension that they operate on so that it has size 1. This enables the reduction of many logical values into a single logical condition.  First, create a matrix A that contains random integers between 1 and 25. Reset the random number generator to the default state for reproducibility.  rng default  A = randi(25,5)  A = 5×5  21 3 4 4 17  23 7 25 11 1  4 14 24 23 22  23 24 13 20 24  16 25 21 24 17  Next, use the mod function along with the logical NOT operator, ~, to determine which elements in A are even.  A = ~mod(A,2)  A = 5x5 logical array  0 0 1 1 0  0 0 0 0 0  1 1 1 0 1  0 1 0 1 1  1 0 0 1 0  The resulting matrices have values of logical 1 (true) where an element is even, and logical 0 (false) where an element is odd.  Since the any and all functions reduce the dimension that they operate on to size 1, it normally takes two applications of one of the functions to reduce a 2–D matrix into a single logical condition, such as any(any(A)). However, if you use the notation A(:) to regard all of the elements of A as a single column vector, you can use any(A(:)) to get the same logical information without nesting the function calls.  Determine if any elements in A are even.  any(A(:))  ans = logical  1  You can perform logical and relational comparisons within the function call to any or all. This makes it easy to quickly test an array for a variety of properties.  Determine if all elements in A are odd.  all(~A(:))  ans = logical  0  Determine whether any main or super diagonal elements in A are even. Since the vectors returned by diag(A) and diag(A,1) are not the same size, you first need to reduce each diagonal to a single scalar logical condition before comparing them. You can use the short-circuit OR operator || to perform the comparison, since if any elements in the first diagonal are even then the entire expression evaluates to true regardless of what appears on the right-hand side of the operator.  any(diag(A)) || any(diag(A,1))  ans = logical  1 |