Linear Algebra

Laboratory Activity No. 6

Matrices

|  |  |
| --- | --- |
| *Submitted by:* | *Instructor:* |
| Sustento, Myke Alvin E. | Engr. Dylan Josh D. Lopez |

November 02, 2020

# Objectives

This laboratory activity aims to implement the principles and techniques of matrices and matrix operations using Python as the programming language.

# Methods

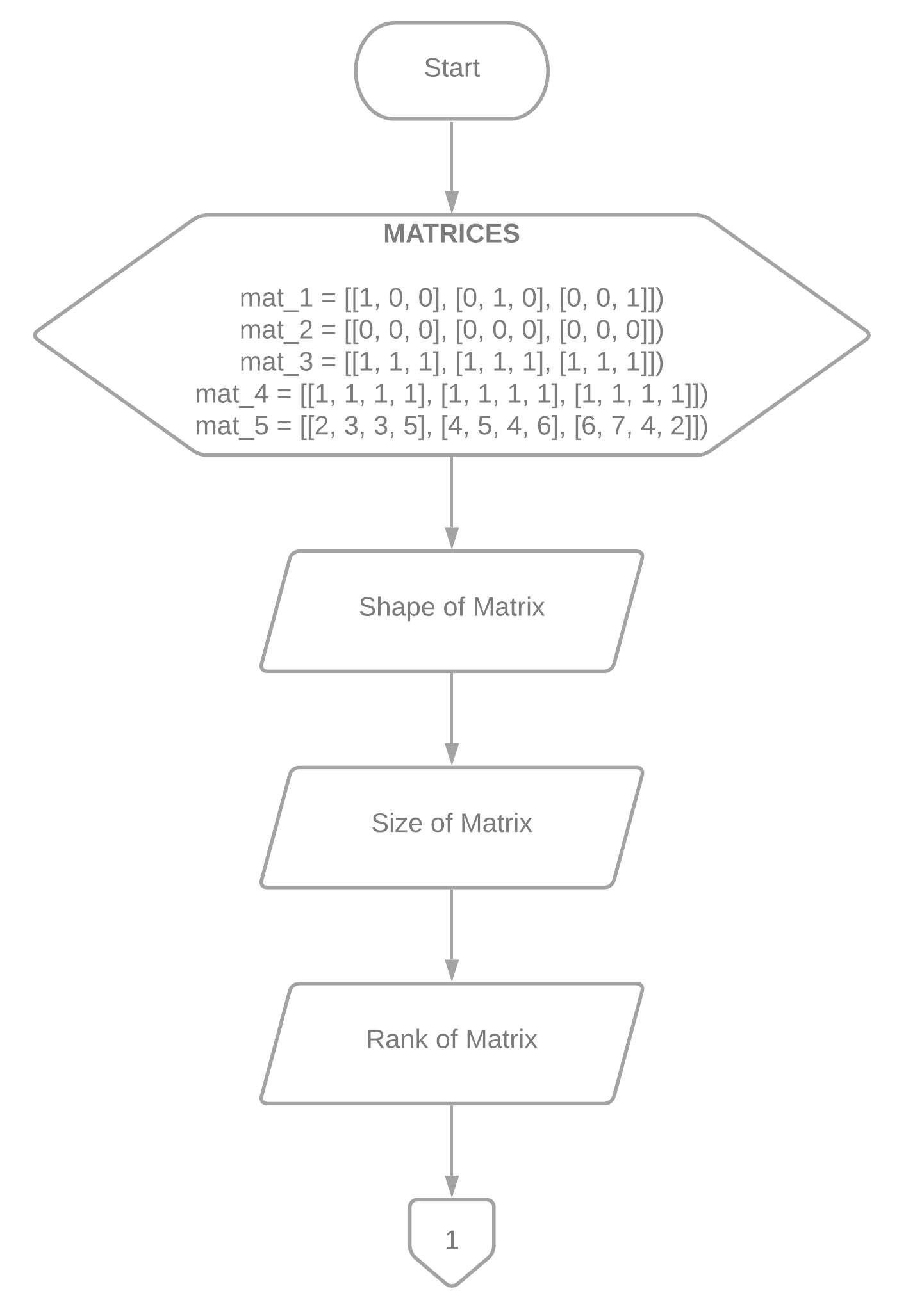


Figure 1 The Flowchart for Task 1 part 1

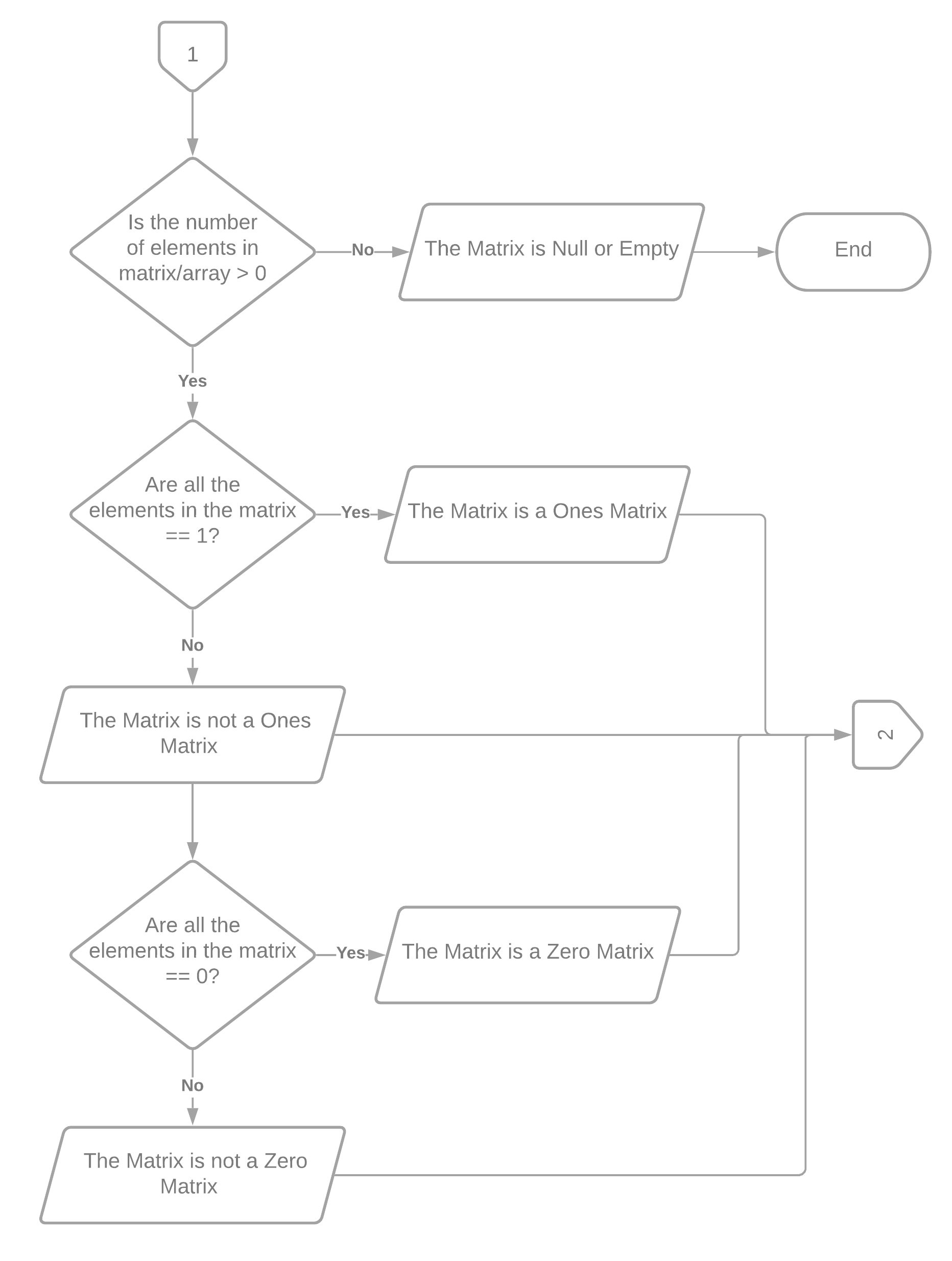


Figure 2 The Flowchart for Task 1 part 2

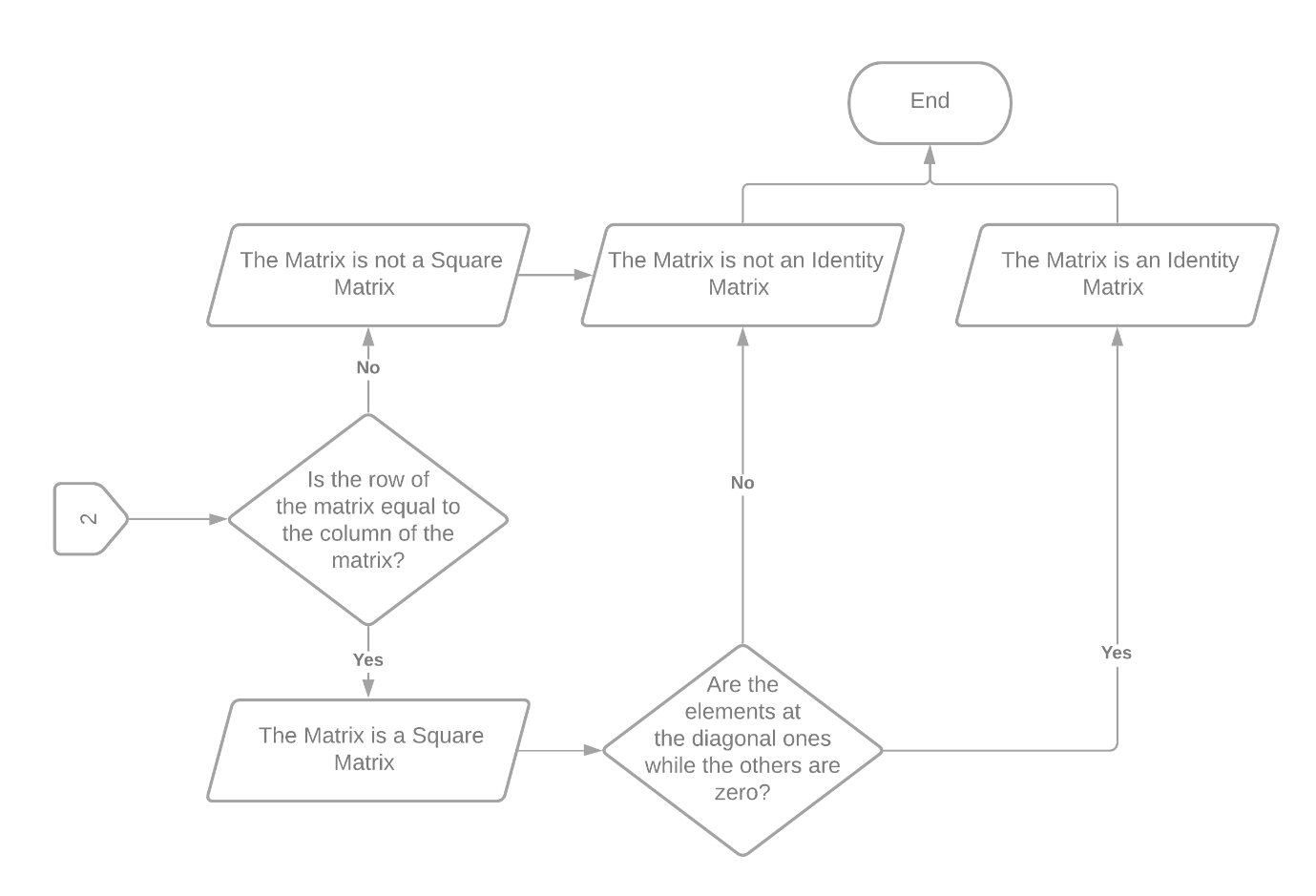


Figure 3 The Flowchart for Task 1 part 3

Figures 1 to 3 show the flowchart that was used to create the program in task 1 which describes a matrix. The program starts by displaying the shape, size, and rank of the matrix and then determine if there are elements in the matrix to determine if the matrix is null. Afterward, determine if the matrix is a one, zero, square, or identity matrix. If all the elements in the matrix are 1, the matrix is a ones matrix but if all the elements in the matrix are 0, the matrix is a zero matrix. If the rows and columns of the matrix are equal, the matrix is square, and if the elements on the diagonal are all 1 while 0 elsewhere, then the matrix is also an identity matrix.

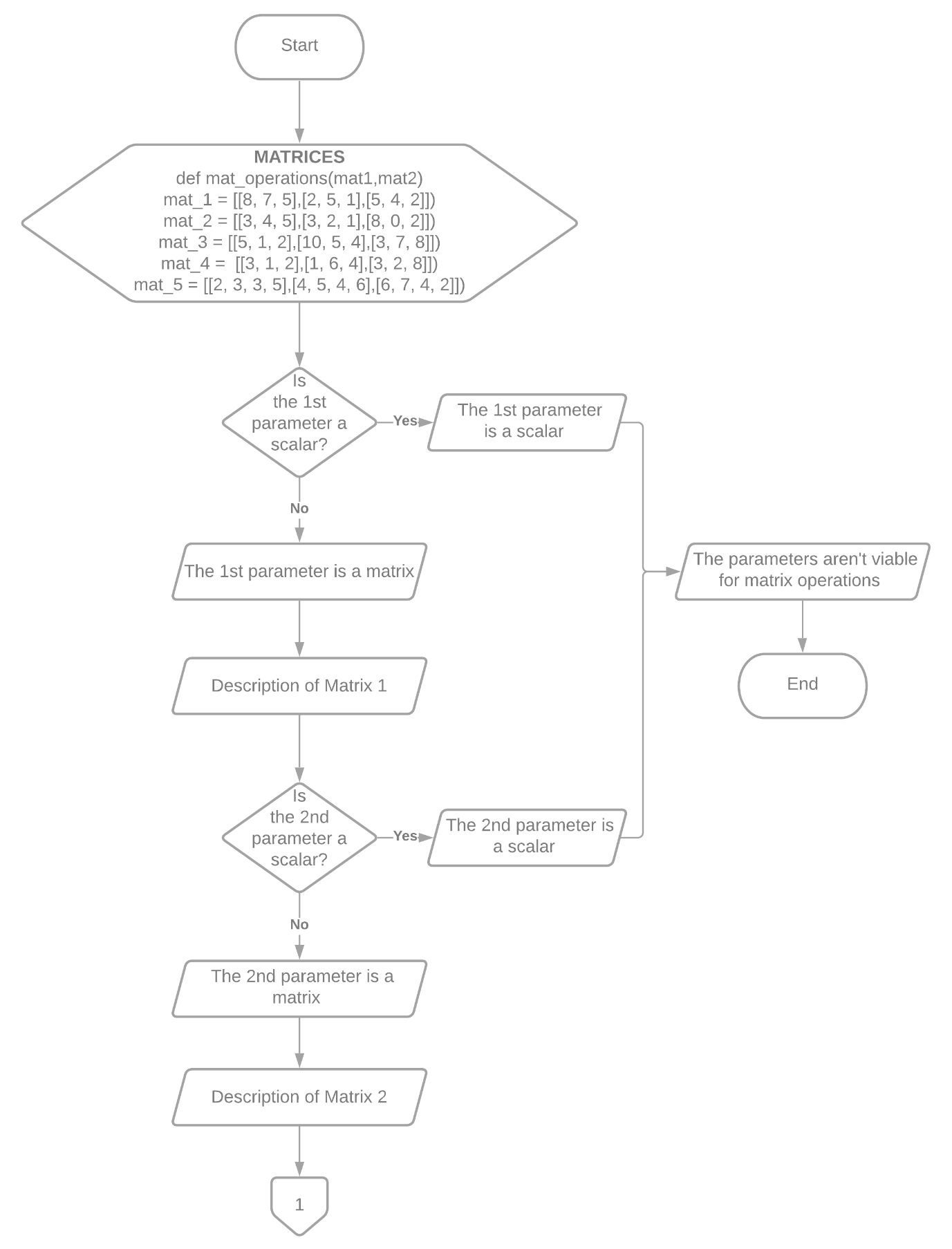


Figure 4 The Flowchart for Task 2 part 1

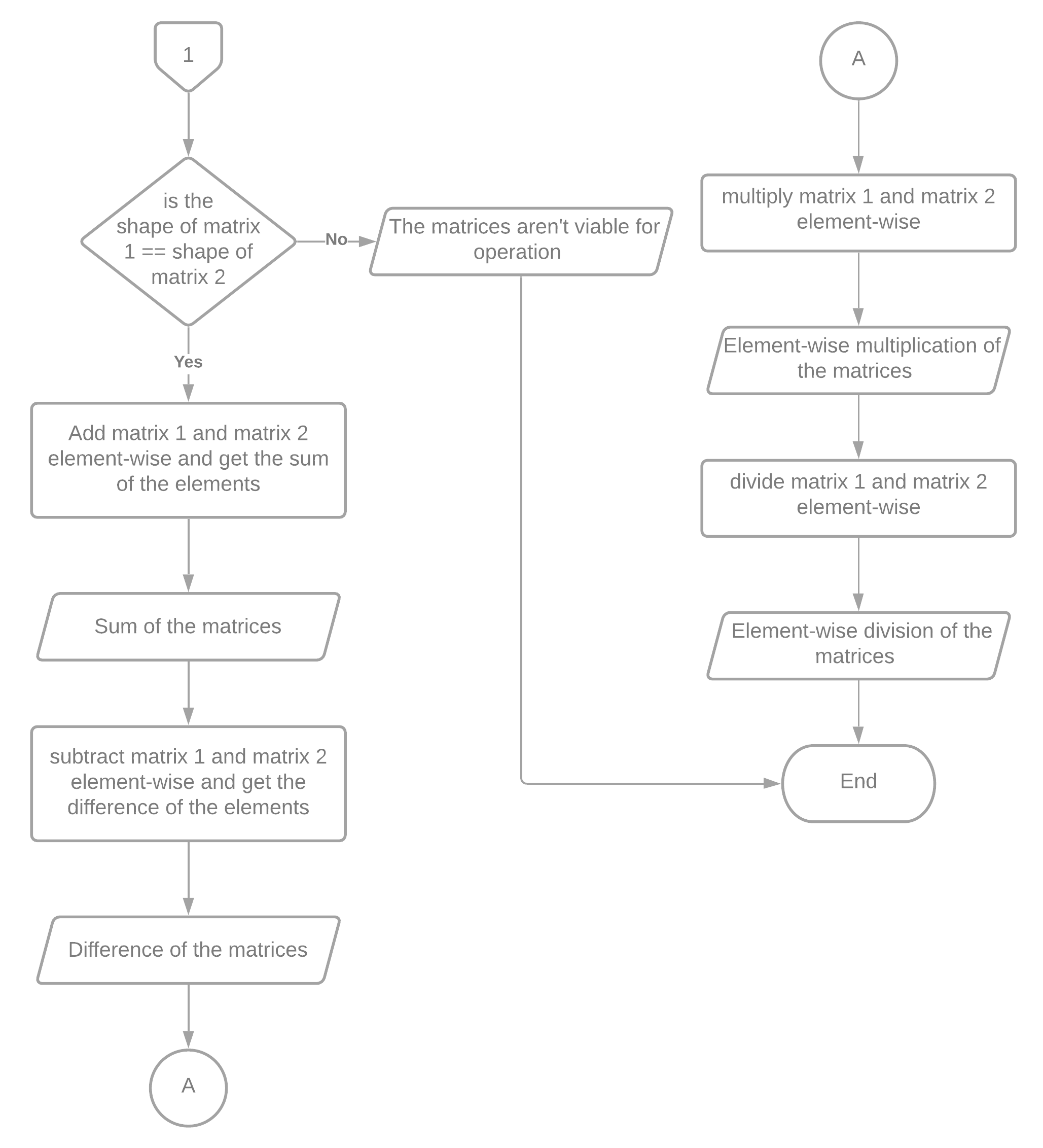


Figure 5 The Flowchart for Task 2 part 2

Figure 4 to 5 shows the flowchart that was used to create the program in task 2 which takes in 2 matrices or scalars and uses matrix operations to get the sum, difference, element-wise multiplication, and element-wise division of the matrices. The program starts by checking if the 1st and 2nd parameters are either matrices or scalars. If one of the parameters is a scalar, the program would show that it is a scalar and that it isn’t viable for matrix operations and then end the program. If both the parameters are matrices, the program would show the description of both the matrices like it did in task 1. Afterward, it would check if the shape of the 1st matrix is equal to the shape of the 2nd matrix. If the shape wasn’t the same, the program would output the matrices aren’t viable for operation since they aren’t broadcastable. However, if the shape of the 1st matrix is equal to the shape of the 2nd matrix then it would get the sum, difference, element-wise multiplication, and element-wise division of the matrices and end the program.

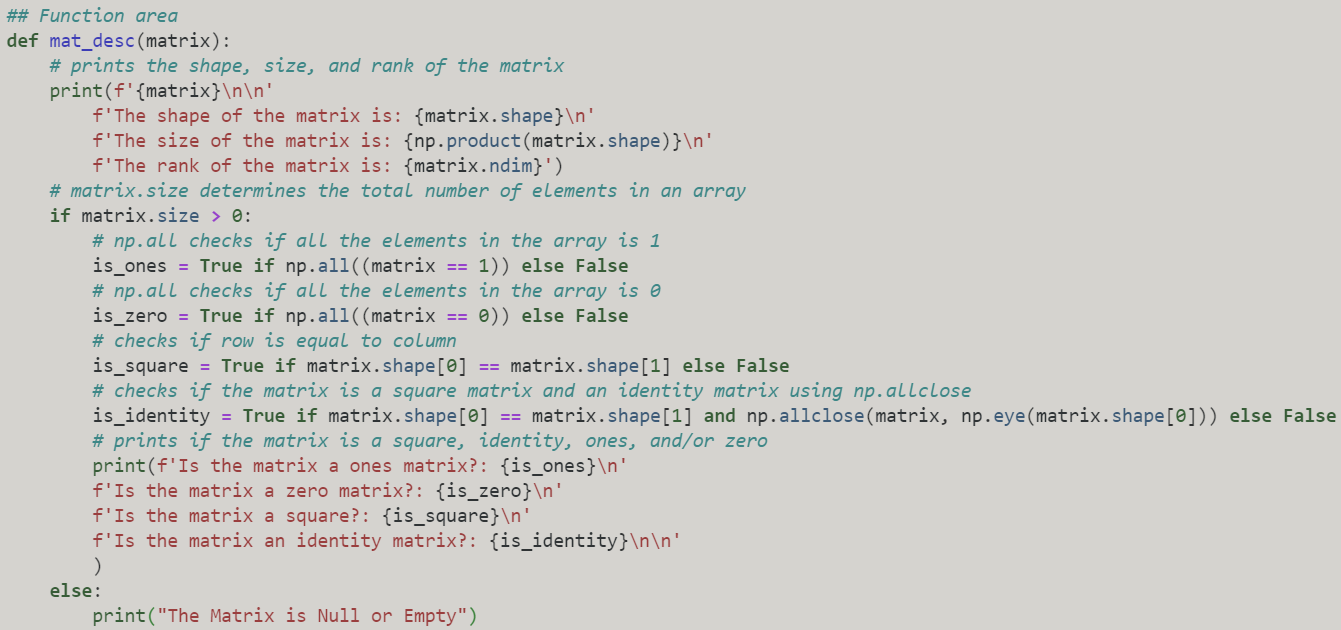


Figure 6 The mat\_desc() method code in Task 1

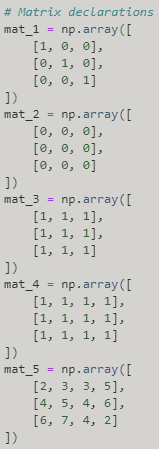


Figure 7 The Sample Matrix Declarations in Task 1

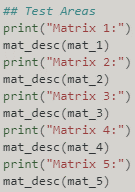


Figure 8 Calling the mat\_desc() Method

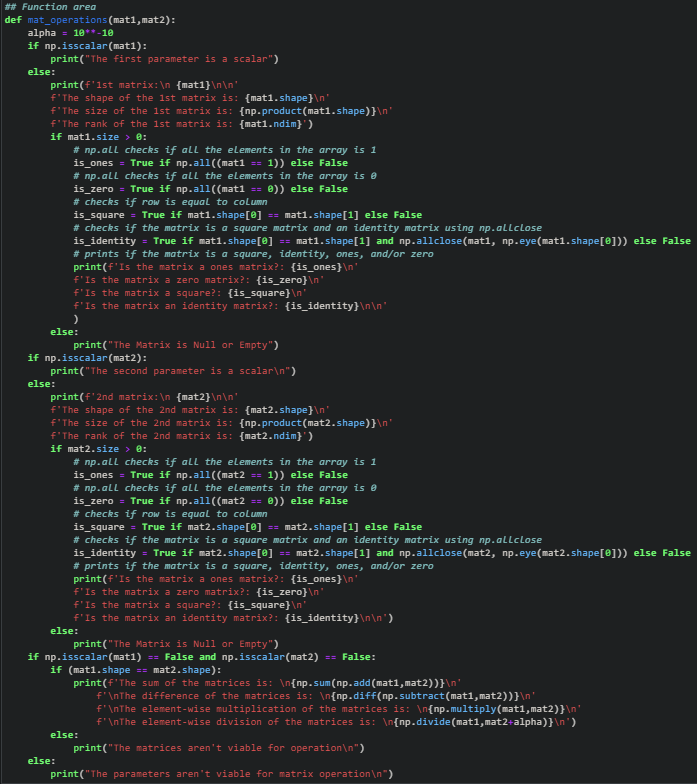


Figure 9 The mat\_operations() method code in Task 2

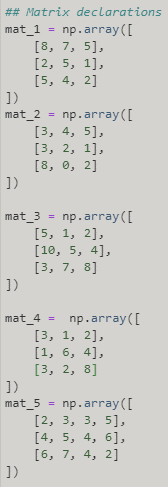


Figure 10 The Sample Matrix Declarations in Task 2

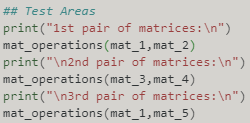


Figure 8 Calling the mat\_operations() Method

The practices of the activity consist of using the functions in the Numpy library. The methods that were used were mat\_desc() and mat\_operations(). The method mat\_desc() was used in task 1 to show the description of the matrices while the method mat\_operations() were used in task 2 to perform matrix operations. The functions that were used were np.array() to create an array [1] that would represent the matrix/matrices, np.ndarray.shape to get the current shape of the matrix [2], np.prod() to get the product of the elements in the matrix [3], np.ndarray.ndim to get the dimension or rank of an matrix [4], np.ndarray.size to get the number of elements in the matrix [5], np.allclose() to check if two matrix are equal element-wise [6], np.eye() to create a matrix with ones on the diagonal and zeros elsewhere or an identity matrix [7] to use as basis for the np.allclose() function, np.all() to test whether all matrix elements are equal to either 0 or 1 to determine if it is a zero matrix or ones matrix [8], np.isscalar() to check if the elements are scalar [9], np.add() to add the two matrices element-wise [10], np.sum() to get the sum of matrix elements [11], np.subtract() to subtract the two matrices element-wise [12], np.diff() was used to get the difference of the matrix [13], np.multiply() to multiply the two matrices element-wise [14], and np.divide() to divide the two matrices element-wise [15]. These practices were used to achieve the deliverables of the laboratory activity which is to be familiar with matrices and perform matrix operations using Python.

# Results

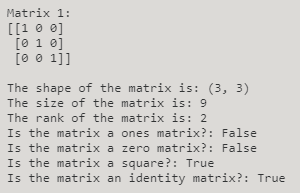


Figure 9 1st Matrix in Task 1

Figure 9 shows the 1st matrix in task 1. The shape of the matrix is (3,3) since the rows are 3 while the columns are also 3. The size of the matrix is 9 for the reason that . The rank of the matrix is 2 for it is a 2-dimensional array. The matrix isn’t a ones matrix because not all the elements are 1. Also, it isn’t a zero matrix seeing that not all the elements are 0. The matrix is a square matrix seeing that the rows and columns are of the same value. Since the diagonal values are 1 while the others are 0, the matrix is an identity matrix.

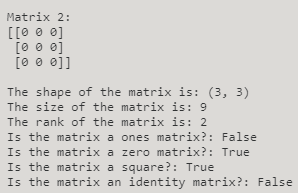


Figure 10 2nd Matrix in Task 1

Figure 10 shows the 2nd matrix in task 1. The shape of the matrix is (3,3) since the rows are 3 while the columns are also 3. The size of the matrix is 9 for the reason that . The rank of the matrix is 2 for it is a 2-dimensional array. The matrix isn’t a ones matrix because not all the elements are 1 but it is a zero matrix because all the elements are 0. The matrix is a square matrix seeing that the rows and columns are of the same value. The matrix is not an identity matrix since the diagonal values aren’t 1.

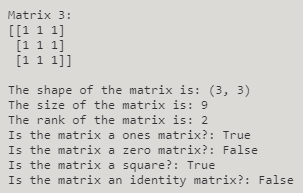


Figure 11 3rd Matrix in Task 1

Figure 11 shows the 3rd matrix in task 1. The shape of the matrix is (3,3) since the rows are 3 while the columns are also 3. The size of the matrix is 9 for the reason that . The rank of the matrix is 2 for it is a 2-dimensional array. The matrix is a ones matrix because all the elements are 1 but it is not a zero matrix because not all the elements are 0. The matrix is a square matrix seeing that the rows and columns are of the same value. The matrix is not an identity matrix since the diagonal values aren’t 1 while the others are 0.

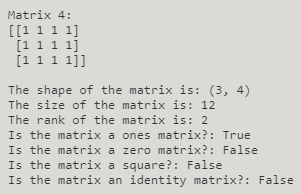


Figure 12 4th Matrix in Task 1

Figure 12 shows the 4th matrix in task 1. The shape of the matrix is (3,4) since the rows are 3 while the columns are 4. The size of the matrix is 12 for the reason that . The rank of the matrix is 2 for it is a 2-dimensional array. The matrix is a ones matrix because all the elements are 1 but it is not a zero matrix because not all the elements are 0. The matrix is not a square matrix seeing that the rows and columns are not the same value. The matrix is not an identity matrix since it is not a square matrix and the diagonal values aren’t 1 while the others are 0.

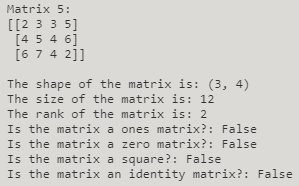


Figure 13 5th Matrix in Task 1

Figure 13 shows the 5th matrix in task 1. The shape of the matrix is (3,4) since the rows are 3 while the columns are 4. The size of the matrix is 12 for the reason that . The rank of the matrix is 2 for it is a 2-dimensional array. The matrix is not a ones matrix because not all the elements are 1. Also, it is not a zero matrix because not all the elements are 0. The matrix is not a square matrix seeing that the rows and columns are not the same value. The matrix is not an identity matrix since it is not a square matrix and the diagonal values aren’t 1 while the others are 0.

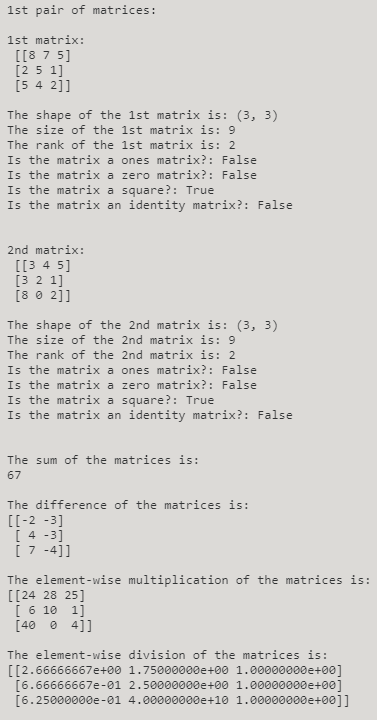


Figure 14 1st Pair of Matrices in Task 2

Figure 14 shows the 1st pair of matrices in task 2. The shape of the 1st matrix is (3,3) since the rows are 3 while the columns are also 3. The size of the matrix is 9 for the reason that . The rank of the matrix is 2 for it is a 2-dimensional array. The matrix is not a ones matrix because not all the elements are 1. Also, it is not a zero matrix because not all the elements are 0. The matrix is a square matrix seeing that the rows and columns are the same value. The matrix is not an identity matrix since it is not a square matrix and the diagonal values aren’t 1 while the others are 0.

The shape of the 2nd matrix is (3,3) since the rows are 3 while the columns are also 3. The size of the matrix is 9 for the reason that . The rank of the matrix is 2 for it is a 2-dimensional array. The matrix is not a ones matrix because not all the elements are 1. Also, it is not a zero matrix because not all the elements are 0. The matrix is a square matrix seeing that the rows and columns are the same value. The matrix is not an identity matrix since it is not a square matrix and the diagonal values aren’t 1 while the others are 0.

The sum of the matrices is 67 while the difference of the matrices is . The element-wise multiplication of the matrices is while the element-wise division of the matrices is rounded off to the nearest hundredths.

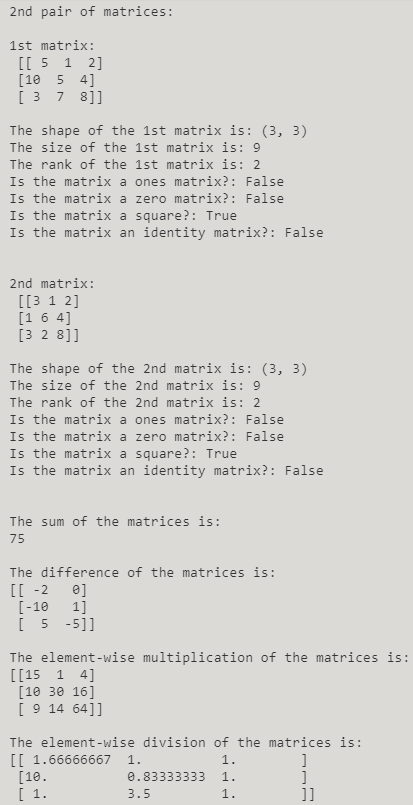


Figure 15 2nd Pair of Matrices in Task 2

Figure 15 shows the 2nd pair of matrices in task 2. The shape of the 1st matrix is (3,3) since the rows are 3 while the columns are also 3. The size of the matrix is 9 for the reason that . The rank of the matrix is 2 for it is a 2-dimensional array. The matrix is not a ones matrix because not all the elements are 1. Also, it is not a zero matrix because not all the elements are 0. The matrix is a square matrix seeing that the rows and columns are the same value. The matrix is not an identity matrix since it is not a square matrix and the diagonal values aren’t 1 while the others are 0.

The shape of the 2nd matrix is (3,3) since the rows are 3 while the columns are also 3. The size of the matrix is 9 for the reason that . The rank of the matrix is 2 for it is a 2-dimensional array. The matrix is not a ones matrix because not all the elements are 1. Also, it is not a zero matrix because not all the elements are 0. The matrix is a square matrix seeing that the rows and columns are the same value. The matrix is not an identity matrix since it is not a square matrix and the diagonal values aren’t 1 while the others are 0.

The sum of the matrices is 75 while the difference of the matrices is . The element-wise multiplication of the matrices is while the element-wise division of the matrices is rounded off to the nearest hundredths.

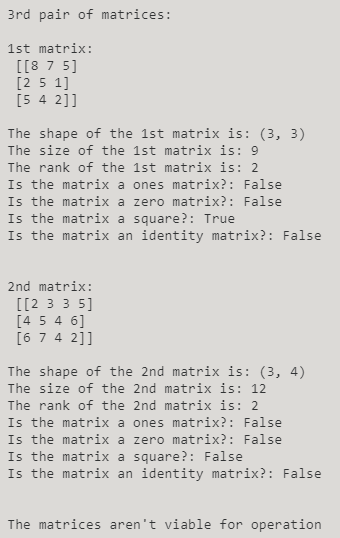


Figure 16 3rd Pair of Matrices in Task 2

Figure 16 shows the 3rd pair of matrices in task 2. The shape of the 1st matrix is (3,3) since the rows are 3 while the columns are also 3. The size of the matrix is 9 for the reason that . The rank of the matrix is 2 for it is a 2-dimensional array. The matrix is not a ones matrix because not all the elements are 1. Also, it is not a zero matrix because not all the elements are 0. The matrix is a square matrix seeing that the rows and columns are the same value. The matrix is not an identity matrix since it is not a square matrix and the diagonal values aren’t 1 while the others are 0.

The shape of the 2nd matrix is (3,4) since the rows are 3 while the columns are 4. The size of the matrix is 12 for the reason that . The rank of the matrix is 2 for it is a 2-dimensional array. The matrix is not a ones matrix because not all the elements are 1. Also, it is not a zero matrix because not all the elements are 0. The matrix is not a square matrix seeing that the rows and columns are not the same value. The matrix is not an identity matrix since it is not a square matrix and the diagonal values aren’t 1 while the others are 0.

The matrices aren’t viable for operation since the matrices are not broadcastable to a common shape hence the output “The matrices aren’t viable for operation”.

# Conclusion

This laboratory proves that the matrix can be declared, categorized, and solved in Python. In this laboratory activity, it was shown that matrices can be thoroughly described using the different functions that can be found using the Numpy Library. It can describe the shape, size, and rank. It can also determine whether it is a square, empty, identity, ones, or zero matrices by analyzing the elements of the matrices. Since matrix algebra exists, the laboratory also showed that matrices can be solved using the functions in the Numpy library. It can return the sum, difference, element-wise multiplication, and element-wise division of the matrices and also determine whether or not the matrices are viable for operation.

Matrix operations can be used in solving agricultural problems by creating an awareness-adoption matrix strategic decision making in agricultural development projects [16]. It can be a tool to identifies weak and strong aspects of research and development projects to attain the adoption of technologies and contribute to strategic decision-making for developing effective cropping technologies in agriculture.

**References**

[1]"numpy.array — NumPy v1.19 Manual", Numpy.org, 2020. [Online]. Available: https://numpy.org/doc/stable/reference/generated/numpy.array.html. [Accessed: 01- Nov- 2020]. [2]"numpy.ndarray.shape — NumPy v1.19 Manual", Numpy.org, 2020. [Online]. Available: https://numpy.org/doc/stable/reference/generated/numpy.ndarray.shape.html. [Accessed: 01- Nov- 2020]. [3]"numpy.prod — NumPy v1.19 Manual", Numpy.org, 2020. [Online]. Available: https://numpy.org/doc/stable/reference/generated/numpy.prod.html. [Accessed: 01- Nov- 2020]. [4]"numpy.ndarray.ndim — NumPy v1.19 Manual", Numpy.org, 2020. [Online]. Available: https://numpy.org/doc/stable/reference/generated/numpy.ndarray.ndim.html. [Accessed: 01- Nov- 2020]. [5]"numpy.ndarray.size — NumPy v1.19 Manual", Numpy.org, 2020. [Online]. Available: https://numpy.org/doc/stable/reference/generated/numpy.ndarray.size.html. [Accessed: 01- Nov- 2020]. [6]"numpy.allclose — NumPy v1.19 Manual", Numpy.org, 2020. [Online]. Available: https://numpy.org/doc/stable/reference/generated/numpy.allclose.html. [Accessed: 01- Nov- 2020]. [7]"numpy.eye — NumPy v1.19 Manual", Numpy.org, 2020. [Online]. Available: https://numpy.org/doc/stable/reference/generated/numpy.eye.html. [Accessed: 01- Nov- 2020]. [8]"numpy.all — NumPy v1.19 Manual", Numpy.org, 2020. [Online]. Available: https://numpy.org/doc/stable/reference/generated/numpy.all.html. [Accessed: 01- Nov- 2020]. [9]"numpy.isscalar — NumPy v1.19 Manual", Numpy.org, 2020. [Online]. Available: https://numpy.org/doc/stable/reference/generated/numpy.isscalar.html. [Accessed: 01- Nov- 2020]. [10]"numpy.add — NumPy v1.19 Manual", Numpy.org, 2020. [Online]. Available: https://numpy.org/doc/stable/reference/generated/numpy.add.html. [Accessed: 01- Nov- 2020]. [11]"numpy.sum — NumPy v1.19 Manual", Numpy.org, 2020. [Online]. Available: https://numpy.org/doc/stable/reference/generated/numpy.sum.html. [Accessed: 01- Nov- 2020]. [12]"numpy.subtract — NumPy v1.19 Manual", Numpy.org, 2020. [Online]. Available: https://numpy.org/doc/stable/reference/generated/numpy.subtract.html. [Accessed: 01- Nov- 2020]. [13]"numpy.diff — NumPy v1.19 Manual", Numpy.org, 2020. [Online]. Available: https://numpy.org/doc/stable/reference/generated/numpy.diff.html. [Accessed: 01- Nov- 2020]. [14]"numpy.multiply — NumPy v1.19 Manual", Numpy.org, 2020. [Online]. Available: https://numpy.org/doc/stable/reference/generated/numpy.multiply.html. [Accessed: 01- Nov- 2020]. [15]"numpy.divide — NumPy v1.19 Manual", Numpy.org, 2020. [Online]. Available: https://numpy.org/doc/stable/reference/generated/numpy.divide.html. [Accessed: 01- Nov- 2020].

[16] M. Subedi, T. Hocking, M. Fullen, and A. Mccrea, "An Awareness-Adoption Matrix for Strategic Decision Making in Agricultural Development Projects: A Case Study in Yunnan Province, China", ResearchGate, 2020. [Online]. Available: https://www.researchgate.net/publication/222888040\_An\_Awareness-Adoption\_Matrix\_for\_Strategic\_Decision\_Making\_in\_Agricultural\_Development\_Projects\_A\_Case\_Study\_in\_Yunnan\_Province\_China. [Accessed: 02- Nov- 2020].

**Appendix**

Github Link - https://github.com/Sus102/LA\_Lab/tree/master/LA\_LAB\_6