|  |  |
| --- | --- |
| **PRACTICAL NO. 6C** | |
| **NAME OF STUDENT** | Madhukumar Paka |
| **CLASS/BATCH/ROLL NUMBER** | SEIT2/B3/15 |
| **DATE OF PERFORMANCE** | 13/05/2021 |
| **DATE OF SUBMISSION** | 13/05/2021 |
| **PROBLEM STATEMENT** | Write a Python program to implement Different Linear algebra functions using Scipy. |
| **SOFTWARE USED** | Google Colab |
| **PROGRAM CODE** | #multiplying two matrices  multiplication = np.multiply(a,b)  print("\n sum of a+b : " , multiplication)  #determinant  determinant = linalg.det(a)  print("\n determinant: ",determinant)  #finding transpose of a matrix :  transpose = (a.T)  print('\n Transpose of A : ', transpose )  #EigenValues  print("inverse of matrix:\n", linalg.inv(a))  eg\_val, eg\_vect = linalg.eig(a)  print("Eigen values:\n", eg\_val)  print("Eigen vectors:\n", eg\_vect)    # We are trying to solve a linear algebra system which can be given as  #         x + 3y +10z = 10  #         2x + 12y + 7z = 18  #         5x + 8y + 8z = 30  # Creating input array  a = np.array([[1, 3, 10], [2, 12, 7], [5, 8, 8]])  # Solution Array  b = np.array([[10], [18], [30]])  # Solve the linear algebra  x = linalg.solve(a, b)  # Print results  print(x)  # Checking Results  print("\n Checking results, Vectors must be zeros")  print(a.dot(x) - b) |
| **OUTPUT** |  |
| **CONCLUSION** | Thus, we have successfully implemented a Python program to perform different Linear algebra functions using Scipy. |
| **LAB OUTCOME** | Understand the structure, syntax, and semantics of the Python language.  Design and Develop cost-effective robust applications using the latest Python trends and technologies. |