NUMPY

1. We will use the intersect1d function of the numpy library which takes 2 arrays as input and returns an array consisting of common values.
2. We will use the array\_equal method of the numpy library to check weather the given 2 arrays are equal or not. This method will return a boolean value of True if the 2 arrays are equal.
3. We will simply write (ar1+ar2)\*(-0.5\*ar1). This expression is doing the job.
4. We will first extract x and y coordinates separately from the matrix using the splice operator. Then we will find out the r coordinate and theta coordinate as follows: r=np.sqrt(x\*\*2+y\*\*2) theta=np.arctan(y,x)

PANDAS

1. I used the nunique function of the pandas library to count the number of unique rows with respect to different columns.
2. I used the df.mean(axis=1) method to compute the mean and stored in a variable mean. Further used: df[0]=df.iloc[ : , 0 ]-mean to subtract the mean row values for each row in the first column, and repeated this task for the second as well as the third column. Alternatively we can also use a loop for keeping count of the column number. In this case since there were only 3 columns, it was not that difficult.
3. Firstly I filtered the dataframe based on the groups ( i.e. , based on weather the given value belongs to group ‘a’,’b’ or ‘c’). Then I used the sort\_values function of Pandas Library to sort the grouped dataframes in descending order based on the column ‘values’. Then I used the sum function to calculate the sum of top three values in each grouped dataframe.