(1) Write a function that takes as input an integer and calculate if it is prime number.

(2) A Palindrome is a string that is same when read backward and forward. Write a function that accepts an input from the user (at command prompt) and gives as output True or False. True if the string is a palindrome and False if it is not. A repeated string(n, N) is a substring of length n inside a bigger substring of length N that has been repeated n times inside the string without any gaps in between. Write a function that checks for this after accepting the input string from the user (at command prompt) and a threshold repeat parameter n. Now go ahead and define a bigger function that accepts an input string and outputs: “Normal String” (if it is neither a palnidrome nor a repeated string(n, N), “Palindrome” (If it is a Palindrome but not a repeated string(n, N), “Repeated String” (if it is Repeated string (n,N) but not a Palindrome) and “None” (if it is neither).

(3) Without using numpy or any third party library, implement the following.

-Write a function that accepts from user number of rows and number of columns and a data-type (integer or float). Then create a random matrix of number of rows and number of columns after checking that the datatype is valid. If not present a user an error message. Once the matrix is created then implement the calculation of determinant of the matrix. Output the created matrix and the calculated determinant. Also make sure that the program is running in loop and continously doing the above task until the user inputs (‘n’/’N’) at the command prompt.

(4) Without using numpy or any third party library, implement the following.

-Write a function that accepts from user number of rows and number of columns and a data-type (integer or float). Then create two random matrices of number of rows and number of columns after checking that the datatype is valid. If not present a user an error message. Once the matrix is created then implement four tasks: element wise addition, element wise subtraction, element wise product and element wise division of two matrices. For the last task you may want to catch the exception when the denominator is 0 (because x/0 is undefined). Ask the user repeatedly if they want to do addition, subtraction, multiplication or division. Output original matrices and the result. Also make sure that the program is running in loop and continously doing the above task until the user inputs (‘n’/’N’) at the command prompt.

(5) Without using numpy or any third party library, implement the following.

-Write a function that accepts from user number of rows and number of columns and a data-type (integer or float) for two matrices. Then create two random matrices of number of rows and number of columns after checking that the datatype is valid. If not present a user an error message. Once the matrix is created then implement dot product of two matrices (after checking the compatible dimensions). Output original matrices and the result. Also make sure that the program is running in loop and continously doing the above task until the user inputs (‘n’/’N’) at the command prompt.

(6) Without using numpy or any third party library, implement the following.

-Write a function that accepts from user number of rows and number of columns and a data-type (integer or float) for a matrix. Then create a random matrix of number of rows and number of columns after checking that the datatype is valid. If not present a user an error message. Once the matrix is created then implement calculation of Eigen values and right Eigen-vectors. You will have to check that matrix is square and if matrix is symmetrical. Output original matrices and the result. Also make sure that the program is running in loop and continously doing the above task until the user inputs (‘n’/’N’) at the command prompt.

(7) Without using any third party library, implement the following.

-Write a function, that as input an equation containing one independent variable and outputs its derivative computed at a particular point. Assume the equations are continous and well behaved.

-Write a function, that takes as input an equation containing one independent variable and outputs its integral between two points. Assume the equations are continous and well behaved.

- Output the equation, derivative and inetgral to the terminal

(8) Using matplotlib,

- plot a sine wave, cosine wave

- plot a circle

- plot a triangle

- plot a square (one of black fill and one of white fill)

- plot a chess board

(9) Write a function that

- reads the csv file from this url (<https://gist.githubusercontent.com/michhar/2dfd2de0d4f8727f873422c5d959fff5/raw/fa71405126017e6a37bea592440b4bee94bf7b9e/titanic.csv>)

- saves it as Pandas Dataframe

- Create a new column that has A, B, C, D, E as values. A if Age is <18, B if 18 <= Age <30, C if <= 30 Age <45, D if 45 <= Age <65 and E if Age >=65. Call this column “AgeCat”

- Calculate an average of Survived column for different Sex categories? What do you observe?

- Calculate an average of Survived column for different AgeCat categories? What do you observe?

- plot a correlation between Sex and Survived columns using matplotlib. What do you see?

- plot a correlation between Age and Survived columns using matplotlib. What do you see?

- What is your overall observation?

(10) Write an object oriented program using concepts of classes that is about students in a class

* Uses constructor for Name, Age, Class Name and Roll number
* Compares two students and outputs if one student has higher marks than the other, only after checking that they belong to the same Class Name
* Prints the message “My Name is ‘abcdef’ and my age is ‘X’ years” to the console when somebody prints the object. ‘abcdef’ is replaced by Student’s name and ‘X’ is replaced by Student’s age
* Create 50 different random students in a loop, without using any input or hard coded values.
* Rank students by Age and if Age is same then rank the student with higher marks higher in the list.
* Print out ranked list calculated above