Experiment - 1

AIM: Introduction to SCILAB and implementation of basic operations in Scilab.

Theory:

What is Scilab?
Scilab is a free and open-source software for engineers & scientists, with a long history (first release in 1994) and a growing community. Scilab is mainly developed by the Scilab team within ESI Group SCILAB is a numerical, programming, and graphics environment available for free from the French Government's "Institut Nationale de Recherche en Informatique et en Automatique - INRIA (National Institute for Informatics and Automation Research)." It is similar in operation to MATLAB and other existing numerical/graphic environments, and it can be run using a variety of operating systems including UNIX, Windows, Linux, etc. SCILAB is a self-contained package including a large number of intrinsic numeric, programming and graphics functions.

Basic Calculations:

1. Enter simple arithmetic expressions in the Console, like 2 + 3, and press Enter. Scilab will show the result (5) below the expression.
2. Try other calculations, including *, /, -, ^ (exponentiation), and sqrt (square root).

Variables and Data Types

Assign a value to a variable using the = symbol. For example, type a = 10 and press Enter. Now, a stores the value 10.

Vectors and Matrices:

Create a vector (one-dimensional array) by enclosing elements in square brackets, separated by commas. For example, type v = [1, 2, 3, 4] to create a vector named v with four elements.

Output:

```
--> x = 4
 x  =

   4.

--> y = 1
 y  =

   1.

--> x+y
 ans  =

   5.

--> x/y
 ans  =

   4.

--> x*y
 ans  =

   4.

--> sqrt(x)
 ans  =

   2.
```

```
--> n=[1,2,3,4]
 n  =

   1.   2.   3.   4.

--> m=[5,6,7,8]
 m  =

   5.   6.   7.   8.

--> add=n+m
 add  =

   6.   8.   10.   12.

--> diffrence=n-m
 diffrence  =

  -4.  -4.  -4.  -4.
```

Experiment - 2

AIM: Exercises to implement the basic matrix operations in Scilab.

Theory:

1. **Creating Matrices:**
   - Use the zeros() function to create a 3x4 matrix of zeros. Assign it to a variable named A.
   - Use the ones() function to create a 2x3 matrix of ones. Assign it to a variable named B.
   - Use manual entry to create a 4x2 matrix
2. **Matrix Addition and Subtraction:**
   - Perform element-wise addition of matrices A and C. Assign the result to a variable named D.
   - Perform element-wise subtraction of matrix B from matrix C. Assign the result to a variable named E.
3. **Matrix Multiplication:**
   - Perform matrix multiplication of A and C. Assign the result to a variable named F.
4. **Matrix Transpose:**
   - Find the transpose of matrix B. Assign the result to a variable named G.
5. **Matrix Inverse:**
   - Find the inverse of matrix C. Assign the result to a variable named H.

Output:

```
--> x=[1,2;3,4]
 x  =

   1.    2.
   3.    4.

--> y=[5,6;7,8]
 y  =

   5.    6.
   7.    8.

--> sum=x+y
 sum  =

   6.    8.
   10.   12.

--> diffrence=x-y
 diffrence  =

  -4.  -4.
  -4.  -4.

--> mul=x*y
 mul  =

   19.   22.
   43.   50.
```

```
--> transpose=x'
 transpose  =

   1.    3.
   2.    4.

--> inverse=inv(x)
 inverse  =

  -2.    1.
   1.5  -0.5

--> detrminant=det(x)
 detrminant  =

  -2.
```

```
--> zeros(3,4)
 ans  =

   0.    0.    0.    0.
   0.    0.    0.    0.
   0.    0.    0.    0.

--> a=zeros(2,3)
 a  =

   0.    0.    0.
   0.    0.    0.

--> b=ones(2,3)
 b  =

   1.    1.    1.
   1.    1.    1.

--> p=[1 5; 2 6; 3 7; 4 8]
 p  =

   1.    5.
   2.    6.
   3.    7.
   4.    8.
```