- **What is the Scilab?**

Scilab is **free and open-source software for numerical computation providing a powerful computing environment** for engineering and scientific applications.

OR

Scilab is **a programming language associated with a rich collection of numerical algorithms covering many** aspects of scientific computing problems.

It consists of two words scientific and laboratory. (**sci: - scientific and lab: - laboratory**).

- **Simple numerical calculations: -**

Operations are written with "+ "for addition, "– "for subtraction, "* "for multiplication, " / "for division, "^ "for exponents.

-->2+3.4

ans = 5.4

--> 57/4

ans = 14.25

--> (2+9) ^5

ans = 161051.

The case is sensitive. It is thus necessary to respect uppercase and lowercase for the calculations to be performed properly.

For example, with sqrt command (which calculates the square root):

-->sqrt (9)

ans = 3.

while:

-->SQRT (9)

! --error 4 Undefined variable: SQRT

- **%e and %pi represents respectively e and π**

--> %e

%e =

2.7182818

--> %pi

%pi =

3.1415927

- **%i represents the ' i 'of complexes in input and is displayed 'i 'in output:**

--> 2+3*%i

ans =

2. + 3.i

- **For not displaying the results**

In adding a semi colon "; "at the end of a command line, the calculation is done but the result is not displayed.

--> (1+sqrt (5))/2;

--> (1+sqrt (5))/2

ans =

1.618034

- **To remind the name of a function**

--> exp (10)/factorial (10)

ans =

0.0060699

- Matrices are defined using square brackets. As mentioned before, matrix computation is the basis of calculations in Scilab. A space or comma is used to separate columns and semicolons are used to separate rows.

To define a column vector and obtain a column display:

-->v= [3; -2;5]

v =

3.

-2.

5.

To define a row vector and obtain a row display:

-->v= [3, -2, 5]

Mention: - This command can also be typed under the form:

v= [3 -2 5]

v =

3. - 2. 5.

To define a matrix and obtain a tabular display:

-->m= [1 2 3;4 5 6;7 8 9]

Mention: - This command can also be typed under the form:

m= [1,2,3;4,5,6;7,8,9]

m =

1. 2. 3.

4. 5. 6.

7. 8. 9.

disp function :-

disp must always be used with parentheses.

With the vector v previously defined:

```
-->v (2)

 ans =

    - 2.
```

- To display a string (usually a sentence), put it between quotes:

```
-->disp("Bob won")

   Bob won
```

- To display a combination of words and values use the **string** command which converts values to character strings using a "**+** "between the different parts:

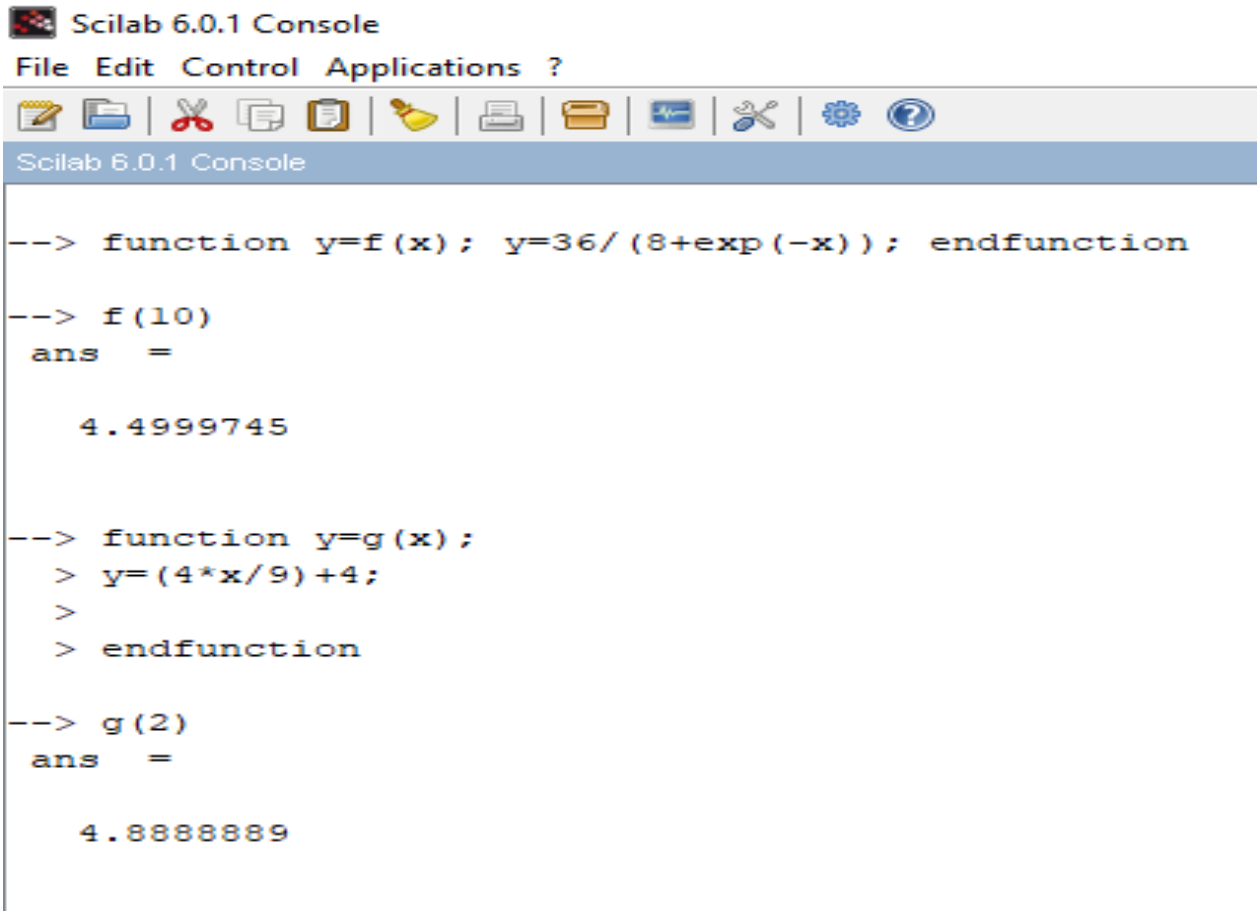
```
-->d=500;

-->disp("Bob won "+string(d)+" dollars")

Bob won 500 dollars
```

- If the sentence contains an single quote, the latter needs to be doubled in the string to be displayed properly:

```
-->disp(" It''s  fair ")

  It's fair
```

- **Functions**

Functions are the easiest and most natural way to make computations from variables and for obtaining results from variables. A function definition begins with **function** and ends with **endfunction**. For example



Question:- make function :- 1) h(X) = $\frac{9+x}{x} + 10$ **2) f(x)=exp(x+3).**

Q.1-- Aim: Write the program to find out even or odd:-

SOURCE CODE:

```
a=input("enter the value:")
if modulo(a,2)==0 then
    disp("it''s even number")
else
    disp("it''s odd number")
end
```

OUTPUT:

```
--> exec('C:\Users\user\bri.sce', -1)
enter the value:5
it's odd number
```

Q.2-- I planted a Christmas tree in 2005 measuring 1.20m. it grows by 30cm per year. i decided to cut it when it exceeds 7m. In what year will I cut the tree?

SOURCE CODE:

```
h=1.2;
y=2005;
while h<7
 h=h+0.3;
 y=y+1;
end
disp("I will cut the tree in "+string(y))
```

OUTPUT:

```
--> exec('C:\Users\user\bri.sce', -1)
I will cut the tree in 2025
```

PRACTICAL:- 01

A) CARDINALITY

AIM: Write a program in Scilab to find cardinality of the following sets

(i) A = {1, 3, 5 7, 9}

(ii) B = {x, y, z}

(iii) C = {Apple, Acer, Sony, Dell, Lenovo}

SOURCE CODE:

```
A=[1,3,5,7,9];
a=length(A);
disp(a,'Cardinality of set A is:')
x=1;
y=2;
z=3;
B=[x,y,z];
b=length(B);
disp(b,'Cardinality of set B is:');
Apple=2;
Acer=4;
Sony=6;
Dell=8;
Lenovo=10;
C=[Apple, Acer, Sony, Dell, Lenovo];
c=length(C);
disp(c,'Cardinality of set C is:');
```

OUTPUT:

```
exec('E:\Scilab program files\cardinality.sce');
Cardinality of set A is:
5.
Cardinality of set B is:
3.
Cardinality of set C is:
5.
```

B) POWER SETS

AIM: Write a program in Scilab to find number of proper subsets of A where

(i) |A| = 5.

(ii) A = {Triangle, Rectangle, Square, Pentagon, Hexagon, Star}.

SOURCE CODE:

(i)

```
n=5;//number of elements of set A
m=2^n-1;//number of elements of power set of A excluding A itself
disp(m,'Number of proper subsets A =');
```

OUTPUT:

```
exec('E:\Scilab program files\power_set1.sce');
Number of proper subsets A =
31.
```

SOURCE CODE:

(ii)

```
Triangle=1;
Rectangle=2;
Square=3;
Pentagon=4;
Hexagon=5;
Star=6;
A=[Triangle, Rectangle, Square, Pentagon, Hexagon, Star];
n=length(A);//number of members in A
m=2^n-1;//number of elements of power set of A excluding A itself
disp(m,'Number of proper subsets of A =');
```

OUTPUT:

```
exec('E:\Scilab program files\power_set1.sce');
Number of proper subsets of A =
63.
```

PRACTICAL:- 02

A) INCLUSION EXCLUSION PRINCIPLE

Q.1. A survey of 550 T.V. watchers produced the following information: 285 watch football, 195 watch hockey, 115 watch baseball, 45 watch football and baseball, 70 watch football and hockey, 50 watch hockey and baseball, 100 do not watch any of the three games.

(i) How many people in the survey watch all the three games?

(ii) How many people watch exactly one of the three games?

SOURCE CODE:

```
U=550;//total number of people surveyed
F=285;//number of people watch football
H=195;//number of people watch hockey
B=115;//number of people watch baseball
FandB=45;//number of people watch football and baseball
FandH=70;//number of people watch football and hockey
HandB=50;//number of people watch hockey and baseball
None=100;//number of people watch none of the three
//To find number of people watch all the three games
```

```
ForHorB=U-None;//number of people watch at least one of the three
//By Inclusion Exclusion Principle
FandHandB=ForHorB-F-H-B+FandH+FandB+HandB;
disp(FandHandB,'Number of people watch all the three games is:');
//To find number of people watch exactly one of the three games
OnlyF=F-FandH-FandB+FandHandB;
OnlyH=H-HandB-FandH+FandHandB;
OnlyB=B-HandB-FandB+FandHandB;
ExactlyOne=OnlyF+OnlyH+OnlyB;//number of people watch exactly
one of the three games
disp(ExactlyOne,'Number of people watch exactly one of the three
games.is:');
```

OUTPUT:

```
exec('E:\Scilab program files\inc_exc2.sce');
Number of people watch all the three games is:
20.
Number of people watch exactly one of the three games.is:
325.
```

Q.2. Out of 1200 students of a college, 552 took Economics, 627 took Mathematics, 540 took Information Technology, 217 took Economics and Mathematics, 307 took Economics and Information Technology, 240 took Mathematics and Information Technology, 213 took all the three subjects.

(i) How many took at least one of the three subjects?

(ii) How many took none of the three subject?

SOURCE CODE:

```
U=1200;//total number of students
E=552;//number of students taken Economics
M=627;//number of students taken Mathematics
IT=540;//number of students taken Information Technology
EandM=217;//number of students taken Economics and Mathematics
EandIT=307;//number of students taken Economics and Information
Technology
MandIT=240;//number of students taken Mathematics and Information
Technology
EandMandIT=213;//number of students taken all the three subjects
//To find number of students taken at least one of the three subjects
//By Inclusion Exclusion Principle
EorMorIT=E+M+IT-EandM-EandIT-MandIT+EandMandIT;
disp(EorMorIT,'Number of students taken at least one of the three
subjects is:');
//To find number of students not taken any of the three subjects
None=U-EorMorIT;
```

```
disp(None,'Number of students not taken any of the three subjects
is:');
```

OUTPUT:

```
exec('E:\Scilab program files\inc_exc1.sce');
Number of students taken at least one of the three subjects is:
1168.
Number of students not taken any of the three subjects is:
32
```

B) SEQUENCE

Example: Calculation of 20 terms of a sequence defined by recurrence by:

$$U_1 = 4$$

$$U_{n+1} = U_n + 2n + 3$$

SOURCE CODE:

```
u(1)=4;
for n=1:20
 u(n+1)= u(n)+2*n+3;
disp([n,u(n)])
 end
```

PRACTICAL No. 3 (a)

POLYNOMIAL EVALUATION

AIM: Write a program in Scilab to evaluate the following polynomials:

(i) $f(x) = x\ .\ 3 - 2x + 1\ at\ x = 2$

(ii) $g(x) = x\ .\ 2 - 1\ at\ x = 3$

(iii) $h(x) = 2x\ .\ 3 - 7x2 + 4x - 15\ at\ x = 5$

SOURCE CODE:

```
f=poly([1,-2,0,1],'x','c');
disp(f,'the polynomial f is ');
k=horner(f,2);
disp(k,'value of polynomial f at x = 2 is ');
g=poly([1,-1],'x','r');
disp(g,'the polynomial g is ');
l=horner(g,3);
disp(l,'value of polynomial g at x = 3 is ');
x = poly(0, 'x');
h = 2*x^3-7*x^2+4*x-15;
disp(h,'the polynomial h is')
m=horner(h,5);
disp(m,'value of the polynomial h at x = 5 is ');
```

PRACTICAL No. 3 (B)

GREATEST COMMON DIVISOR

AIM: Write a program in Scilab to compute Greatest Common Divisor.

SOURCE CODE:

```
Function                                                    thegcd1(a,b)

A=int32([a,b]);

ans=gcd(A);

disp(ans,'the    gcd    of    the    two    numbers    is')

endfunction
```

OUTPUT:

```
--> thegcd1(50,25);
 the gcd of the two numbers is
  25
```

SOURCE CODE:

```
function                                                    g=gcd(a,b)
    r=b
    while(r>0)
        r=modulo(a,b)
        a=b
        b=r
    end
    g=a
endfunction
```

OUTPUT:

```
--> exec('C:\Users\user\bri.sce', -1)


--> gcd(25,60)
ans  =
 5.
```

SOURCE CODE:

```
function thegcd2(a, b, c)
A=int32([a,b,c]);
ans=gcd(A);
disp(ans,'the gcd of the three numbers is')
endfunction
```

OUTPUT:

```
exec('E:\Scilab program files\thegcd1.sci');
thegcd2(50,25,250);
the gcd of the three numbers is
25
```

PRACTICAL No. 4(A)

FACTORIAL NOTATION

AIM: Write a program in Scilab for the following

Q.1 Find first six factorial values by using $n! = n \times (n-1)!$ without using inbuilt factorial function.

Q.2 Compute (i) 6! (ii) 6!/4! (iii) 10!/7! , by using inbuilt factorial function.

SOURCE CODE:

```
//finding some initial factorial values without using inbuilt factorial function

//calculation by using n!=n*(n-1)!

f0=1;

disp(f0,'0!=');

f1=1*f0;

disp(f1,'1!=');

f2=2*f1;

disp(f2,'2!=');

f3=3*f2;
```

```
disp(f3,'3!=');
f4=4*f3;
disp(f4,'4!=');
f5=5*f4;
disp(f5,'5!=');
//6! by using inbuilt factorial function
f6=factorial(6);
disp(f6,'6!=');
//finding 6!/4! and 10!/7!
k1=factorial(6)/factorial(4);
disp(k1,'value of 6!/4! is:');
k2=factorial(10)/factorial(7);
disp(k2,'value of 10!/7! is:');
```

OUTPUT:

```
exec('E:\Scilab program files\counting_factorial.sce');
0!=
 1.
1!=
 1.
```

2!=

2.

3!=

6.

4!=

24.

5!=

120.

6!=

720.

value of 6!/4! is:

30.

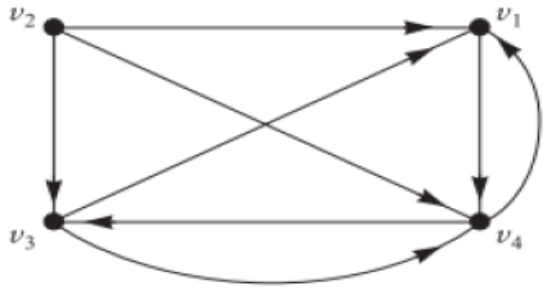
value of 10!/7! is:

720.

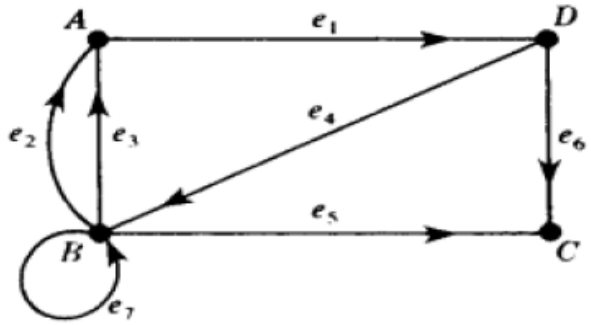
PRACTICAL No. 5

ADJACENCY MATRIX

AIM: Write a program in Scilab to display adjacency matrix of the following digraph and number of edges.



(a)

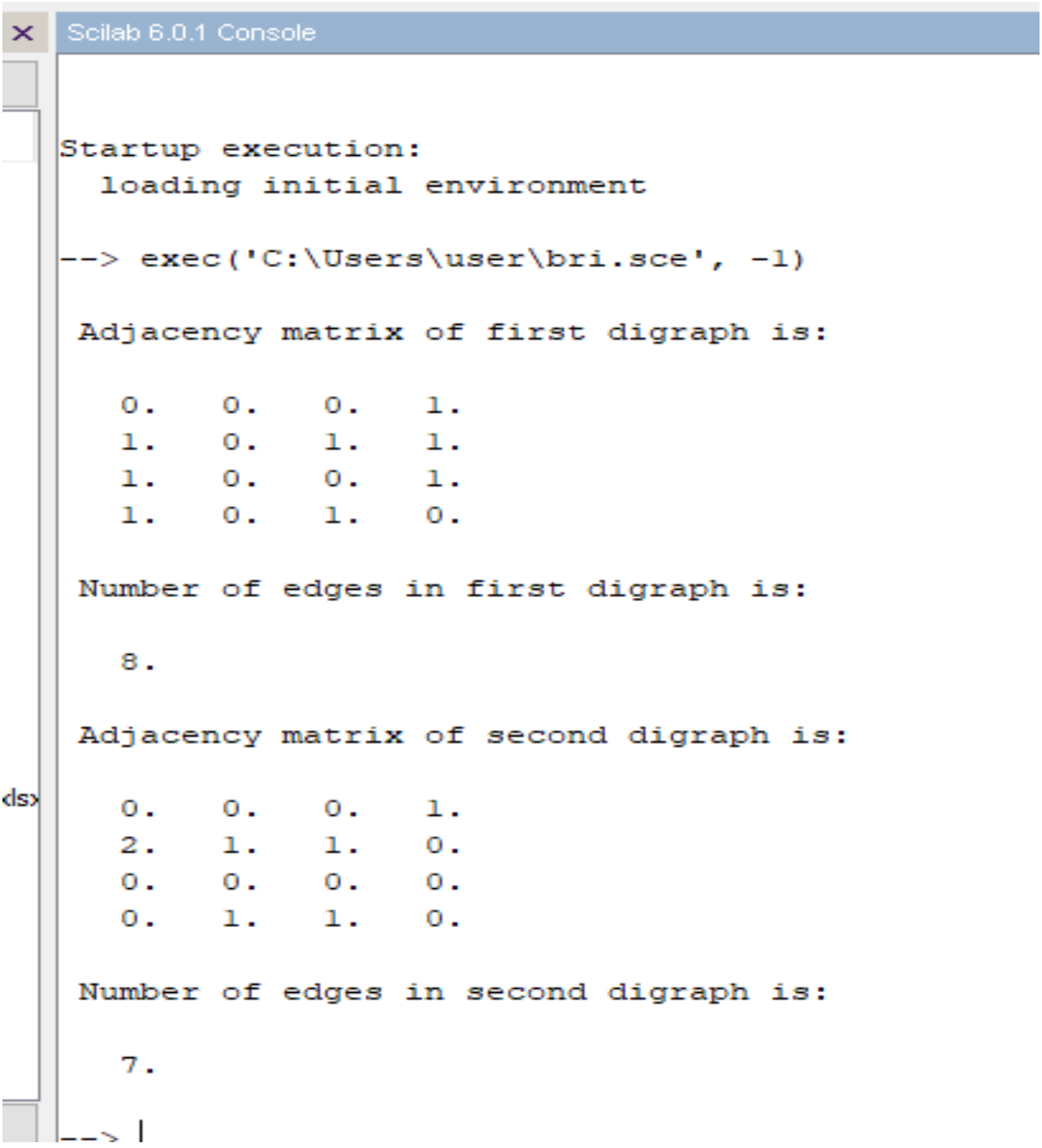


(a)

SOURCE CODE:

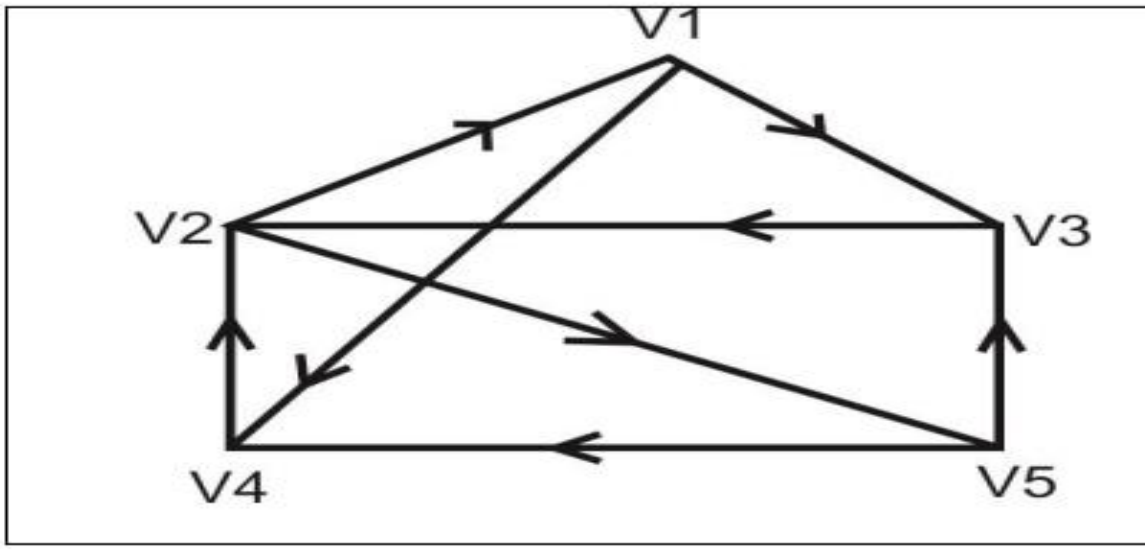
```
A=[0 0 0 1;1 0 1 1;1 0 0 1;1 0 1 0];
disp(A,'Adjacency matrix of first digraph is:')
k=sum(A);
disp(k,'Number of edges in first digraph is:');
B=[0 0 0 1;2 1 1 0;0 0 0 0;0 1 1 0];
disp(B,'Adjacency matrix of second digraph is:');
n=sum(B);
disp(n,'Number of edges in second digraph is:');
```

OUTPUT:


Scilab 6.0.1 Console
Startup execution:
loading initial environment
--> exec('C:\Users\user\bri.sce', -1)
Adjacency matrix of first digraph is:
0. 0. 0. 1.
1. 0. 1. 1.
1. 0. 0. 1.
1. 0. 1. 0.
Number of edges in first digraph is:
8.
Adjacency matrix of second digraph is:
0. 0. 0. 1.
2. 1. 1. 0.
0. 0. 0. 0.
0. 1. 1. 0.
Number of edges in second digraph is:
7.
-->

PRACTICAL No. 6(A)

AIM: Write a program in Scilab to display adjacency matrix of the following Directed graph and number of edges.



SOURCE CODE:

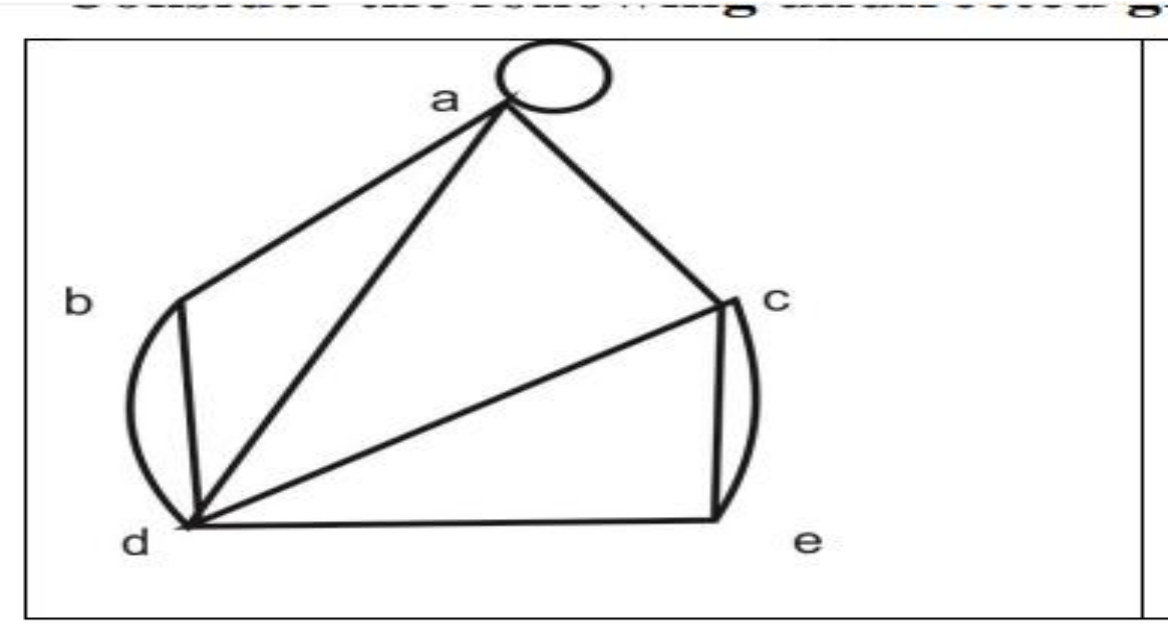
```
A=[0 0 1 1 0;1 0 0 0 1;0 1 0 0 0;0 1 0 0 0;0 0 1 1 0];
disp(A,'Adjacency matrix of first digraph is:')
k=sum(A);
disp(k,'Number of edges in first digraph is:');
```

OUTPUT:

```
   7.


--> exec('C:\Users\user\bri.sce', -1)

 Adjacency matrix of first digraph is:


   0.   0.   1.   1.   0.
   1.   0.   0.   0.   1.
   0.   1.   0.   0.   0.
   0.   1.   0.   0.   0.
   0.   0.   1.   1.   0.

 Number of edges in first digraph is:


   8.


-->
```

PRACTICAL No. 6(B)

AIM: Write a program in Scilab to display adjacency matrix of the following Undirected graph and number of edges.



SOURCE CODE:

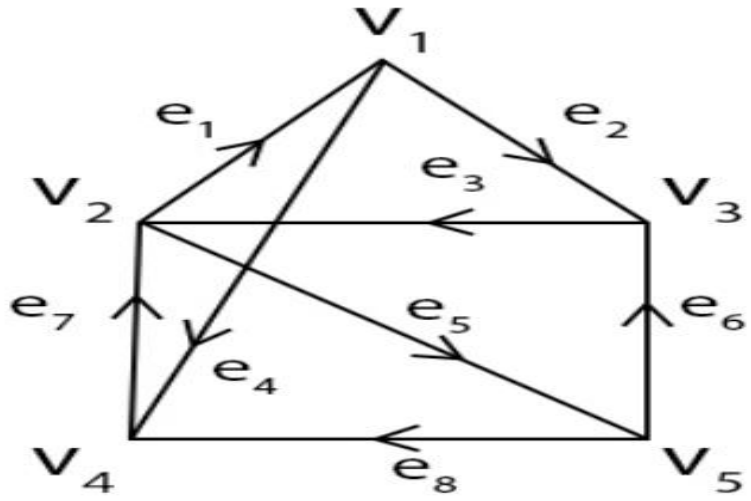
```
B=[2 1 1 1 0;1 0 0 2 0;1 0 0 1 2;1 2 1 0 1;0 0 2 1 0];
disp(B,'Adjacency matrix of first digraph is:')
k=sum(B)/2;
disp(k,'Number of edges in first digraph is:');
```

OUTPUT:

```
    o.


--> exec('C:\Users\user\bri.sce', -1)

 Adjacency matrix of first digraph is:


    2.    1.    1.    1.    0.
    1.    0.    0.    2.    0.
    1.    0.    0.    1.    2.
    1.    2.    1.    0.    1.
    0.    0.    2.    1.    0.

 Number of edges in first digraph is:


    10.


-->
```

PRACTICAL No. 7(A)

AIM: Write a program in Scilab to display incident matrix of the following directed graph.



SOURCE CODE:

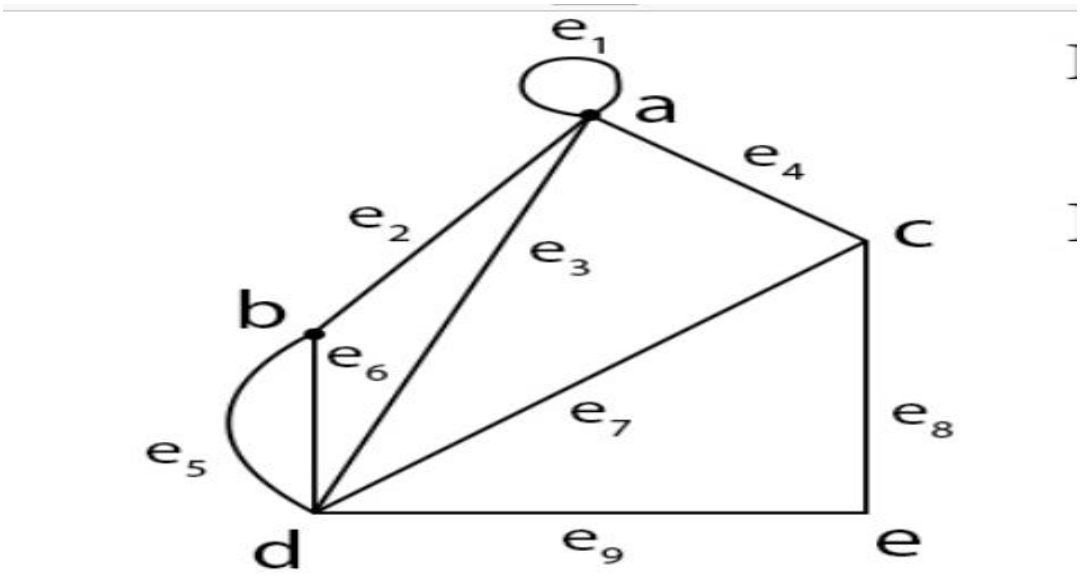
```
B=[1 0 0 0 0 0 0 0;0 0 1 0 0 0 1 0;0 1 0 0 0 1 0 0;0 0 0 1 0
0     0     1;0     0     0     0     1     0     0     0];

disp(B,'Incident      matrix      of      digraph      is:')

k=sum(B);

disp(k,'Number of edges in digraph is:');
```

OUTPUT:

```
--> exec('C:\Users\user\bri.sce', -1)

 Incident matrix of digraph is:

   1.   0.   0.   0.   0.   0.   0.   0.
   0.   0.   1.   0.   0.   0.   1.   0.
   0.   1.   0.   0.   0.   1.   0.   0.
   0.   0.   0.   1.   0.   0.   0.   1.
   0.   0.   0.   0.   1.   0.   0.   0.

 Number of edges in digraph is:

   8.

-->
```

PRACTICAL No. 7(B)

AIM: Write a program in Scilab to display incident matrix of the following Undirected graph.



SOURCE CODE:

```
D=[2 1 1 1 0 0 0 0 0;0 1 0 0 1 1 0 0 0;0 0 0 1 0 0 1 1 0;0 0
1   0   1   1   1   0   1;0   0   0   0   0   0   0   1   1];

disp(D,'Incident   matrix   of   first   digraph   is:')

k=sum(D)/2;

disp(k,'Number of edges in digraph is:');
```

OUTPUT:

```
--> exec('C:\Users\user\bri.sce', -1)

 Incident matrix of first digraph is:

   2.    1.    1.    1.    0.    0.    0.    0.    0.
   0.    1.    0.    0.    1.    1.    0.    0.    0.
   0.    0.    0.    1.    0.    0.    1.    1.    0.
   0.    0.    1.    0.    1.    1.    1.    0.    1.
   0.    0.    0.    0.    0.    0.    0.    1.    1.

 Number of edges in digraph is:

   9.

-->
```