Applied Mathematics Lab

ETMA 252

Faculty: **Mrs. Anita**

Name: **Syeda Reeha Quasar**

Roll No.: **14114802719**

Semester: **4**

Maharaja Agrasen Institute of Technology, PSP Area,

Sector – 22, Rohini, New Delhi – 110085

MAHARAJA AGRASEN INSTITUTE OF TECHNOLOGY

COMPUTER SCIENCE & ENGINEERING DEPARTMENT

VISION

To Produce "Critical thinkers of Innovative Technology"

MISSION

To provide an excellent learning environment across the computer science discipline to inculcate professional behavior, strong ethical values, innovative research capabilities and leadership abilities which enable them to become successful entrepreneurs in this globalized world.

1. To nurture an **excellent learning environment** that helps students to enhance their problem solving skills and to prepare students to be lifelong learners by offering a solid theoretical foundation with applied computing experiences and educating them about their **professional, and ethical responsibilities**.
2. To establish **Industry-Institute Interaction**, making students ready for the industrial environment and be successful in their professional lives.
3. To promote **research activities** in the emerging areas of technology convergence.
4. To build engineers who can look into technical aspects of an engineering solution thereby setting a ground for producing successful **entrepreneur.**

VISION

To nurture young minds in a learning environment of high academic value and im-bibe spiritual and ethical values with technological and management competence.

MISSION

The Institute shall endeavor to incorporate the following basic missions in the teaching methodology:

Engineering Hardware - Software Symbiosis

Practical exercises in all Engineering and Management disciplines shall be carriedout by Hardware equipment as well as the related software enabling deeper un- derstanding of basic concepts and encouraging inquisitive nature.

Life - Long Learning

The Institute strives to match technological advancements and encourage studentsto keep updating their knowledge for enhancing their skills and inculcating theirhabit of continuous learning.

Liberalization and Globalization

The Institute endeavors to enhance technical and management skills of students so that they are intellectually capable and competent professionals with Industrial Aptitude to face the challenges of globalization.

Diversification

The Engineering, Technology and Management disciplines have diverse fields of studies with different attributes. The aim is to create a synergy of the above at- tributes by encouraging analytical thinking.

Digitization of Learning Processes

The Institute provides seamless opportunities for innovative learning in all En- gineering and Management disciplines through digitization of learning processes using analysis, synthesis, simulation, graphics, tutorials and related tools to createa platform for multi-disciplinary approach.

INDEX

S.No.	Program Name	Date	Marks	Signature	Remarks
1.	Algebra of Matrices a) To find transpose of a matrix. b) To find addition of two matrices. c) To find multiplication of two matrices	18-03-2021			
2.	To find the inverse of a square matrix using Gauss-Jordan method.	25-03-2021			
3.	To find Eigen values and Eigen vectors of a square matrix.	15-04-2021			
4.	To find the solution of algebraic and transcendental equations using (a) Bisection method (b) Newton- Raphson method.	22-04-2021			
5.	To find the value of a definite integral using (a) Trapezoidal rule (b) Simpsons 1/3 rule (c) Simpsons 3/8 rule.	29-04-2021			
6.	To solve ordinary differential equations using Runge- Kutta Method.	20-05-2021			
7.	To solve ordinary differential equations using Euler's method.	20-05-2021			
8.	To plot unit step function and square wave function.	27-05-2021			
9.	To find mean, standard deviation and moments about mean of a given frequency data.	03-06-2021			

Date 18-03-2021

Expt. No. 1 Page No. 1.

Experiment-1

Aim: To perform following algebra on matrices with and without function using Scilab a) Addition of matrices
b) Finding transpose
c) Multiplication of matrices

A.1 Matrix Addition using function

```
function [C] = addition (m, n, A, B)
    C = zeros (m, n);
    C = A + B ;
    disp('First matrix is', A)
    disp('Second matrix is', B)
    disp('Sum of matrices is')
endfunction
```

A.2

Amit
1/4/21

Teacher's Signature : _____

OUTPUT

Scilab 6.1.0 Console

File Edit Control Applications ?

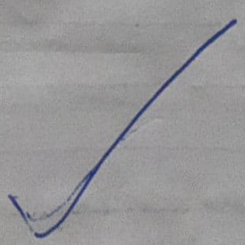
```
--> A = ones(2, 2)
 A  =

    1.    1.
    1.    1.

--> B = [2 3; 4 5]
 B  =

    2.    3.
    4.    5.

--> C = addition(2, 2, A, B)

 "First matrix is"

    1.    1.
    1.    1.

 "Second matrix is"

    2.    3.
    4.    5.

 "Sum of matrices is"
 C  =

    3.    4.
    5.    6.

-->
```

Scilab 6.1.0 Console

File Edit Control Applications ?

```
--> exec('C:\Users\DELL\Desktop\addition.sce', -1)

--> C = addition(2, 2, [1 2; 3 4], [5 6; 7 8])

 "First matrix is"

    1.    2.
    3.    4.

 "Second matrix is"

    5.    6.
    7.    8.

 "Sum of matrices is"
 C  =

    6.    8.
    10.   12.
```

ADDTION USING FUNCTIONS



Date _____

Expt. No. _____ Page No. 2

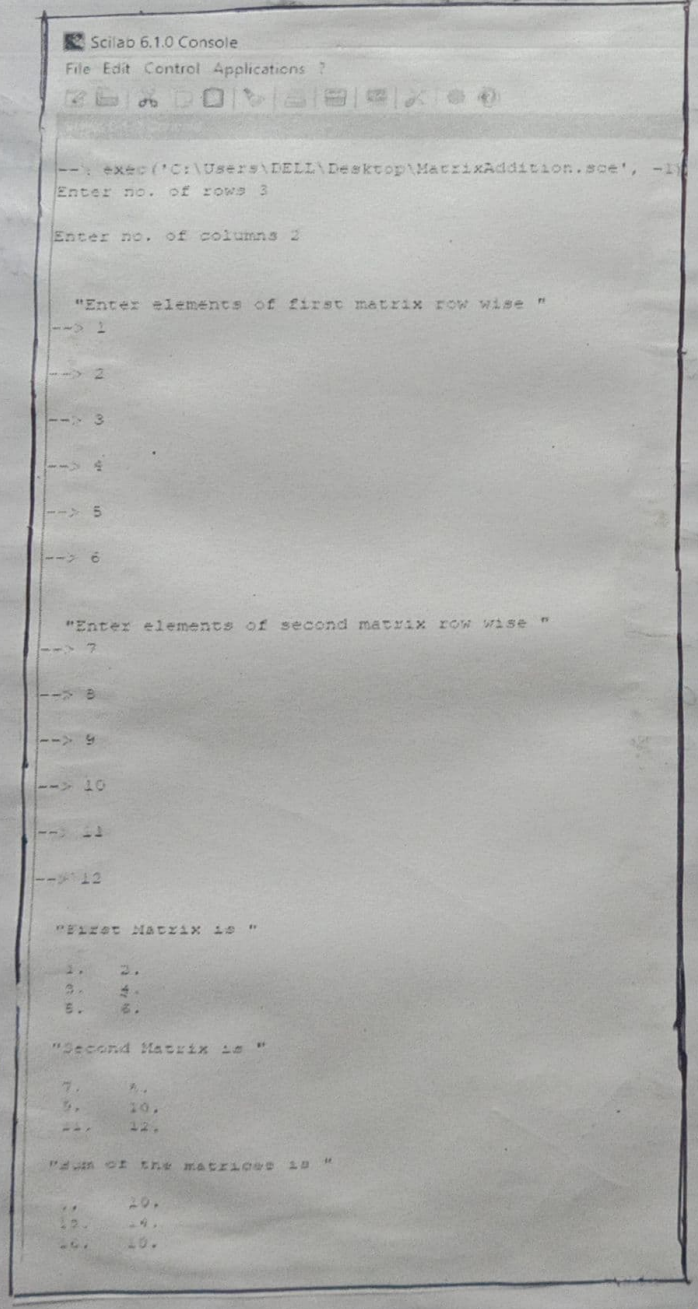
A.2 Matrix addition without Function

```
m = input ('Enter no. of rows')
n = input ('Enter no of columns')
A = zeros (m, n)
B = zeros (m, n)
C = zeros (m, n)
disp ('Enter elements of first matrix row wise')
for i = 1:m
    for j = 1:n
        A(i, j) = input ('')
    end
end
disp ('Enter elements of second matrix row wise')
for i = 1:m
    for j = 1:n
        B(i, j) = input ('')
    end
end
for i = 1:m
    for j = 1:n
        C(i, j) = A(i, j) + B(i, j)
    end
end
disp ('First Matrix is', A)
disp ('Second Matrix is', B)
disp ('Sum of matrices is', C)
```

Arvita
1/4/21

Teacher's Signature : _____

OUTPUT



Scilab 6.1.0 Console

File Edit Control Applications ?

```
--> exec('C:\Users\DELL\Desktop\MatrixAddition.sce', -1)
Enter no. of rows 3

Enter no. of columns 2


  "Enter elements of first matrix row wise "
--> 1

--> 2

--> 3

--> 4

--> 5

--> 6


  "Enter elements of second matrix row wise "
--> 7

--> 8

--> 9

--> 10

--> 11

--> 12

 "First Matrix is "

   1.    2.
   3.    4.
   5.    6.

 "Second Matrix is "

   7.    8.
   9.    10.
   11.   12.

 "Sum of the matrices is "

   8.    10.
   12.   14.
   16.   18.
```

MATRIX ADDITION WITHOUT FUNCTION

Date _____

Expt. No. _____ Page No. 3

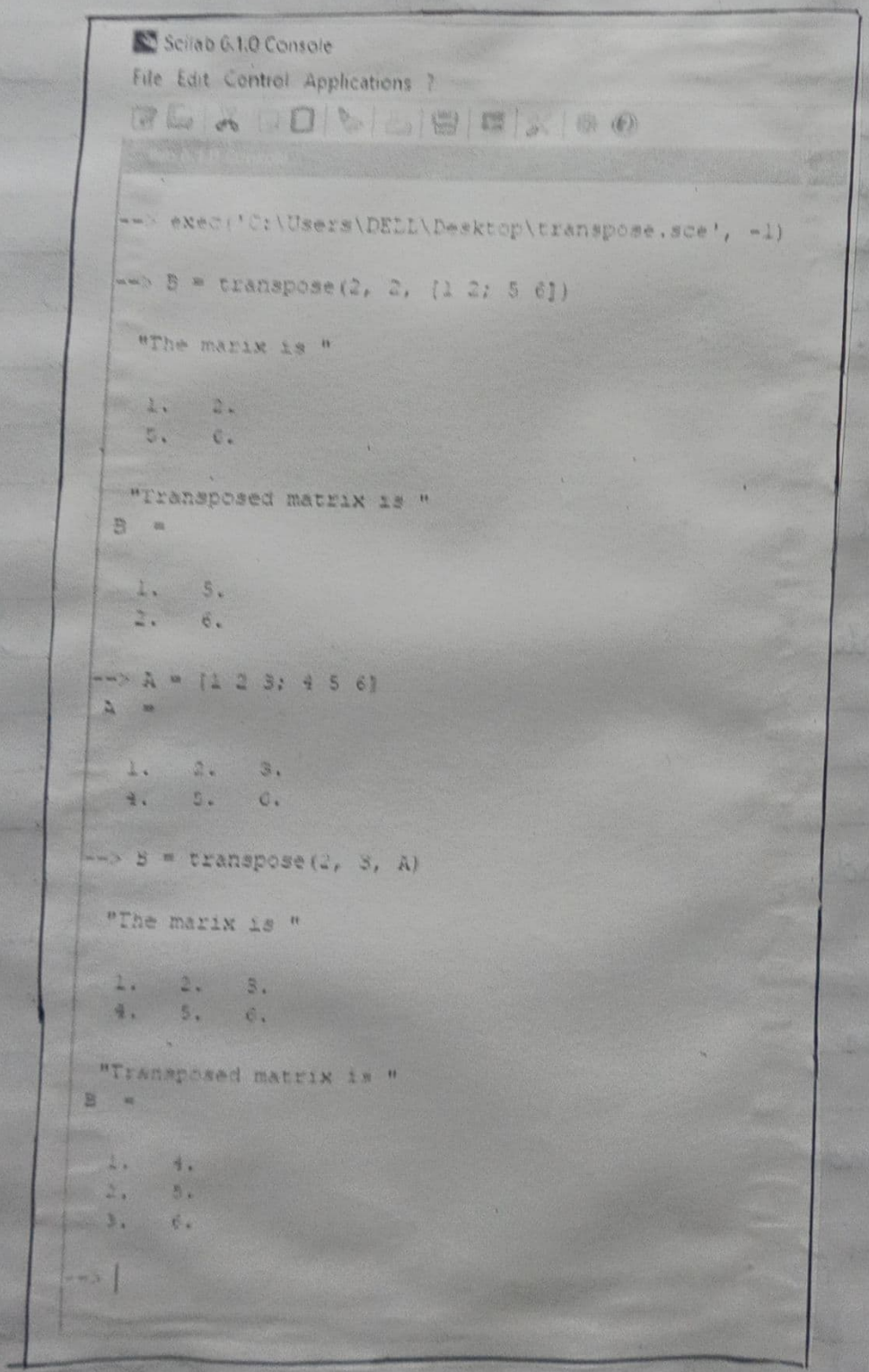
B.1 Matrix transpose with functions

```
function [B] = transpose (m, n, A)
        B = zeros (m, n)
        B = A'
        disp ('The matrix is', A)
        disp ('Transposed matrix is')
end function
```

Amit
1/4/21

Teacher's Signature : _____

OUTPUT



TRANSPOSE OF MATRIX

Date

Expt. No. _____ Page No. 4

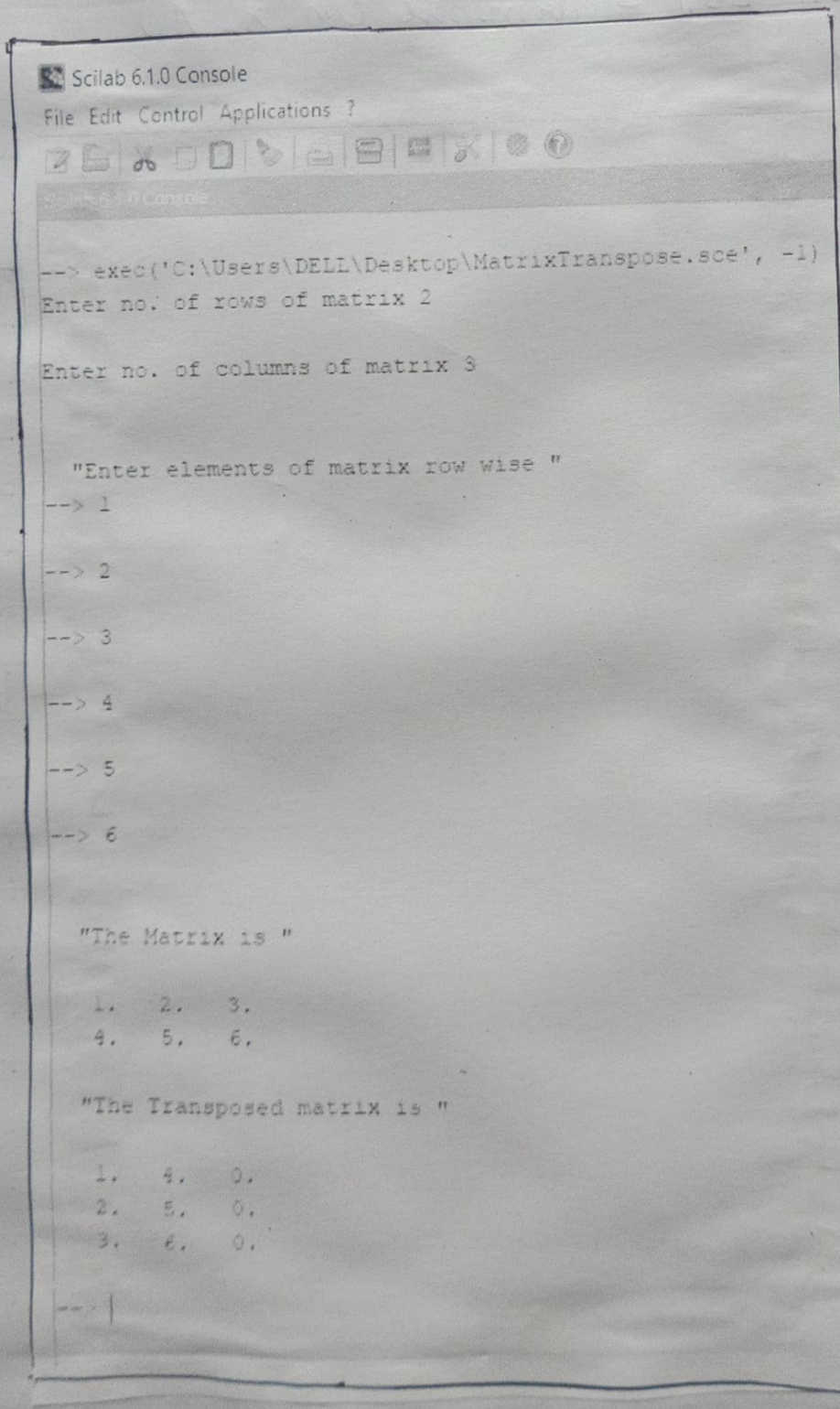
B.2 Matrix transpose without function

```
m = input ('Enter no. of rows of matrix')
n = input ('Enter no. of columns of matrix')
A = zeros(m, n);
B = zeros(m, n);
disp('Enter elements of matrix row-wise')
for i = 1:m
    for j = 1:n
        A(i, j) = input('');
    end
end
for i = 1:n
    for j = 1:m
        B(i, j) = A(j, i);
    end
end
disp('The Matrix is', A)
disp('The transposed Matrix is', B)
```

Anita
1/4/21

Teacher's Signature : _____

OUTPUT



TRANSPOSE OF MATRIX WITHOUT FUNCTION

Date _____

Expt. No. _____ Page No. 5

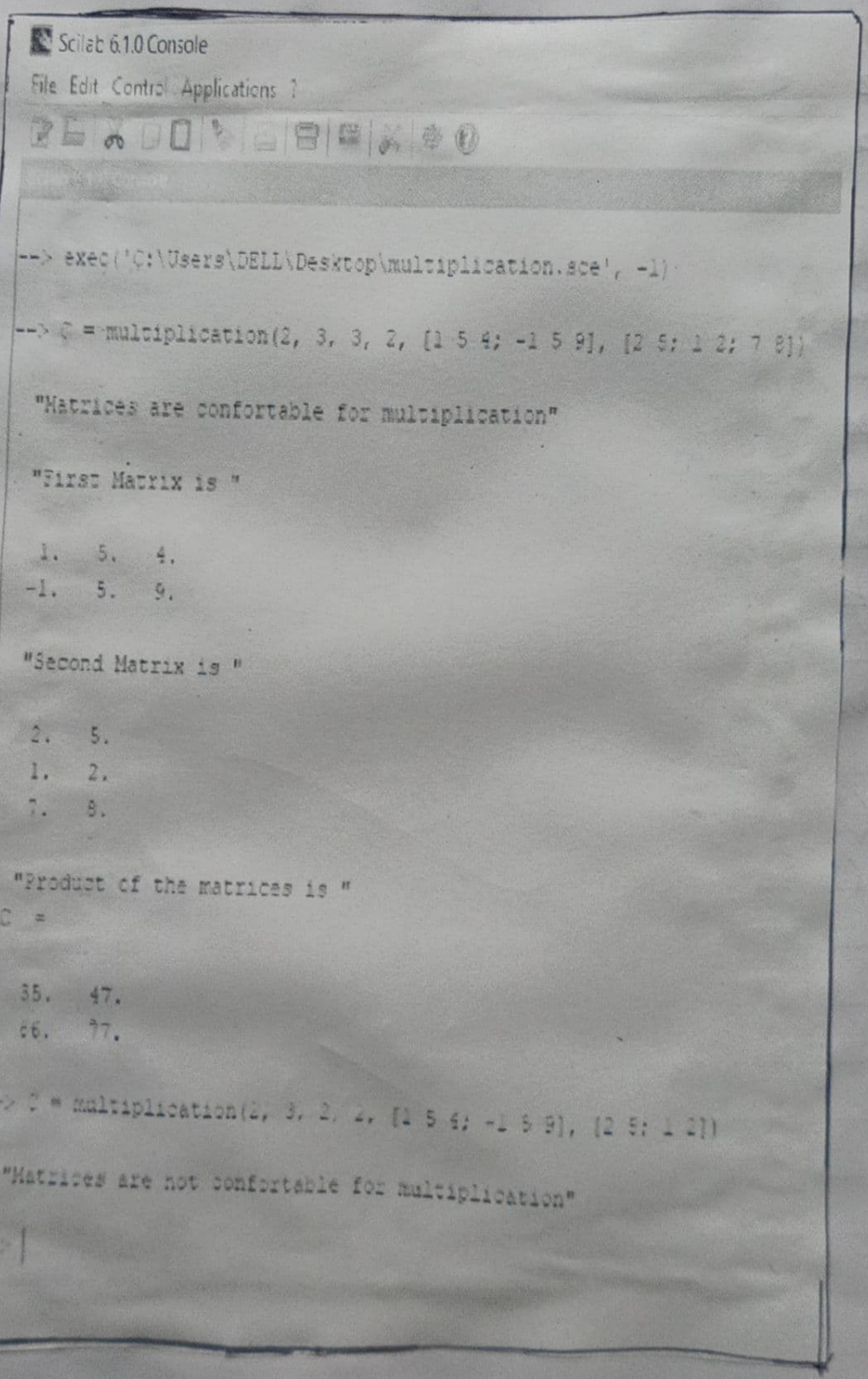
C.1 Matrix multiplication with function

```
function [C] = multiplication (m, n, p, q, A, B)
        C = zeros (m, n)
        if n == p then
            disp ('Matrices are comfortable for multiplication')
        else
            disp ('Matrices are not comfortable for multiplication')
            abort
        end
        C = A * B
        disp ('First Matrix is', A)
        disp ('Second Matrix is', B)
        disp ('Product of matrices is')
endfunction
```

Ans
11/4/21

Teacher's Signature : _____

OUTPUT



MULTIPLICATION OF MATRICES

Date _____

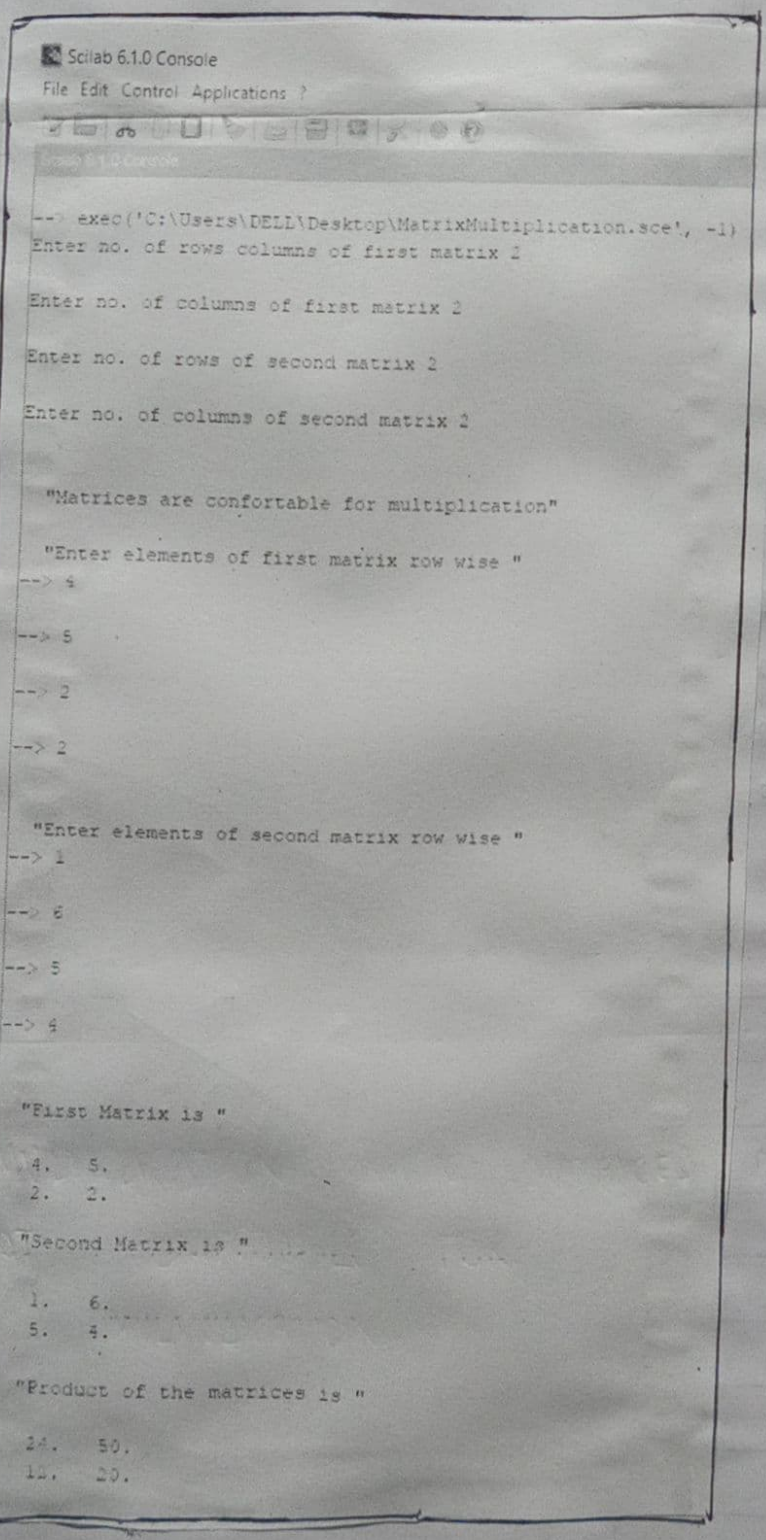
Expt. No. _____ Page No. 6

C.2 Matrix multiplication without function

```
m = input('Enter no. of rows of first Matrix')
n = input('Enter no. of columns of first Matrix')
p = input('Enter no. of rows of second Matrix')
q = input('Enter no. of columns of second Matrix')
if n==p then
    disp('Matrices are comfortable for multiplication')
else
    disp('Matrices are not comfortable for multiplication')
    abort
end
A = zeros(m, n)
B = zeros(p, q)
C = zeros(m, q)
disp('Enter elements of first matrix row wise')
for i = 1:m
    for j = 1:n
        A(i, j) = input('')
    end
end
disp('Enter elements of second matrix row wise')
for i = 1:p
    for j = 1:q
        B(i, j) = input('')
    end
end
for i = 1:m
    for j = 1:q
```

Teacher's Signature : _____

OUTPUT

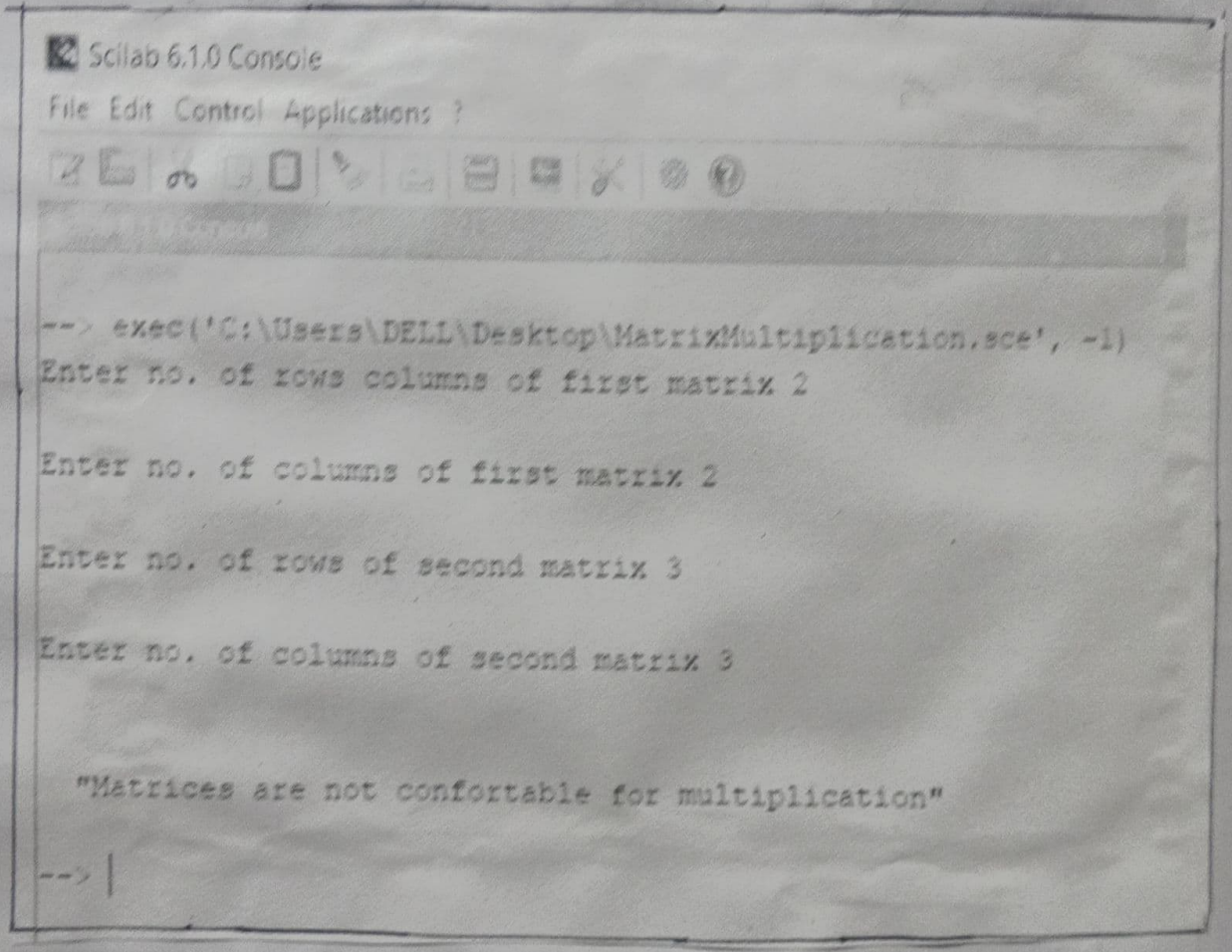


MATRIX MULTIPLICATION WITHOUT FUNCTION

```
            for k = 1:n
                C(i,j) = (A(i,j) * B(k,j)) + C(i,j)
            end
        end
    end
    disp ('First matrix is', A)
    disp ('Second matrix is', B)
    disp ('Product of matrices is', C)
```

Amit
1/4/21

OUTPUT



MATRIX MULTIPLICATION WHEN MATRICES ARE NOT COMPATABLE FOR MULTIPLICATION

Date 25-03-2021

Expt. No. 2 Page No. 8

Experiment-2

Aim: To find the inverse of any 3 by 3 matrix by Gauss Jordan Method

Source Code:

```
function [B] = invr(A)
    B = eye(3,3)
    disp('Given Matrix A is :- ', A)
    if (det(A)==0) then
        disp('Matrix is singular, Inverse does not exist')
        abort
    end
    Aug = [A, B]
    if (Aug(1,1)==0 & Aug(2,1)~=0) then
        C(1,:) = Aug(1,:)
        Aug(1,:) = Aug(2,:)
        Aug(2,:) = C(1,:)
    elseif (Aug(1,1)==0 & Aug(3,1)~=0) then
        C(1,:) = Aug(1,:)
        Aug(1,:) = Aug(3,:)
        Aug(3,:) = C(1,:)
    end
    Aug(1,:) = Aug(1,:)/Aug(1,1)
    Aug(2,:) = Aug(2,:) - Aug(2,1) * Aug(1,:)
    Aug(3,:) = Aug(3,:) - Aug(3,1) * Aug(1,:)
    if (Aug(2,2)==0) then
```

Teacher's Signature : _____

CONSOLE OUTPUT

Scilab 6.1.0 Console

File Edit Control Applications ?

```
--> B = inv([2 0 0; 3 0 2; 7 0 3])

 "Given Matrix A is -: "

    2.    0.    0.
    3.    0.    2.
    7.    0.    3.

 "Matrix is singular, Inverse does not exist."

--> B = inv([1 0 4; 2 -2 1; -1 1 -1])

 "Given Matrix A is -: "

    1.    0.    4.
    2.   -2.    1.
   -1.    1.   -1.

 "The inverse of given matrix is:- "
 B  =

    1.    4.    8.
    1.    3.    7.
    0.   -1.   -2.

--> B = inv([2 6 1; 3 9 2; 0 -1 3])

 "Given Matrix A is -: "

    2.    6.    1.
    3.    9.    2.
    0.   -1.    3.

 "The inverse of given matrix is:- "
 B  =
```

INVESE OF A MATRIX - SINGULAR MATRIX CASE PRINTING INVERSE

Date _____

Expt. No. _____ Page No. 9

```
C(2,:) = Aug(2,:)
Aug(2,:) = Aug(3,:)
Aug(3,:) = C(2,:)
end
Aug(2,:) = Aug(2,:)/Aug(2,2)
Aug(1,:) = Aug(1,:) - Aug(1,2) * Aug(2,:)
Aug(3,:) = Aug(3,:) - Aug(3,2) * Aug(2,:)
Aug(3,:) = Aug(3,:)/Aug(3,3)
Aug(1,:) = Aug(1,:) - Aug(1,3) * Aug(3,:)
Aug(2,:) = Aug(2,:) - Aug(2,3) * Aug(3,:)
Aug(:,1:3) = []
B = Aug(:,1:3),°
disp('The inverse of given matrix is :-')
endfunction
```

Ans

1/4/21

Teacher's Signature : _____

OUTPUT

Scilab 6.1.0 Console

File Edit Control Applications ?

```
--> B = inv([2 6 1; 3 9 2; 0 -1 3])

 "Given Matrix A is -: "

   2.    6.    1.
   3.    9.    2.
   0.   -1.    3.

 "The inverse of given matrix is:- "
B  =

   29.   -19.    3.
  -9.     6.   -1.
  -3.     2.    0.

-->
```

INVERSE OF MATRIX

Scilab 6.1.0 Console

File Edit Control Applications ?

```
--> B = inv([2 0 0; 3 0 2; 7 0 3])

 "Given Matrix A is -: "

   2.    0.    0.
   3.    0.    2.
   7.    0.    3.

 "Matrix is singular, Inverse does not exist"

-->
```

INVERSE NOT POSSIBLE

Date 15-04-2021

Expt. No. 3 Page No. 10.

Experiment - 3

Aim : To find Eigen Values and Eigen Vector of a 2x2 matrices

Source Code :

```
clc;
disp("Enter the matrix row wise");
for i = 1:2
    for j = 1:2
        A(i,j) = input('');
    end
end
trace = A(1,1) + A(2,2);
determinant = A(1,1) * A(2,2) - A(1,2) * A(2,1);
disp("The characteristic Equation is : ");
disp('e^2 + ' + string(-trace) + '*e ' + string(determinant) + '=0');
e1 = (trace + sqrt(trace^2 - 4 * determinant)/2;
e2 = (trace - sqrt(trace^2 - 4 * determinant)/2;
if A(1,2) ~= 0 then
    v1 = [A(1,2); e1 - A(1,1)];
    v2 = [A(1,2); e2 - A(1,1)];
elseif A(2,1) ~= 0 then
    v1 = [e1 - A(2,2); A(2,1)];
    v2 = [e2 - A(2,2); A(2,1)];
else
    v1 = [1; 0];
```

Teacher's Signature : _____

Date _____

Expt. No. _____ Page No. 11.

```
        V2 = [0; 1];
end
disp("First Eigen value is: ", e1);
disp("First Eigen vector is: ", V1);
disp("Second Eigen Value is: ", e2);
disp("Second Eigen vector is: ", V2);
```

Teacher's Signature : _____

EXPERIMENT - 1

APPLIED MATHEMATICS LAB

Aim

Algebra of Matrices

a) To find transpose of a matrix.

b) To find addition of two matrices.

c) To find multiplication of two matrices.

Syeda Reeha Quasar
14114802719
4C7

EXPERIMENT – 1

Aim:

Algebra of Matrices

a) To find transpose of a matrix.

b) To find addition of two matrices.

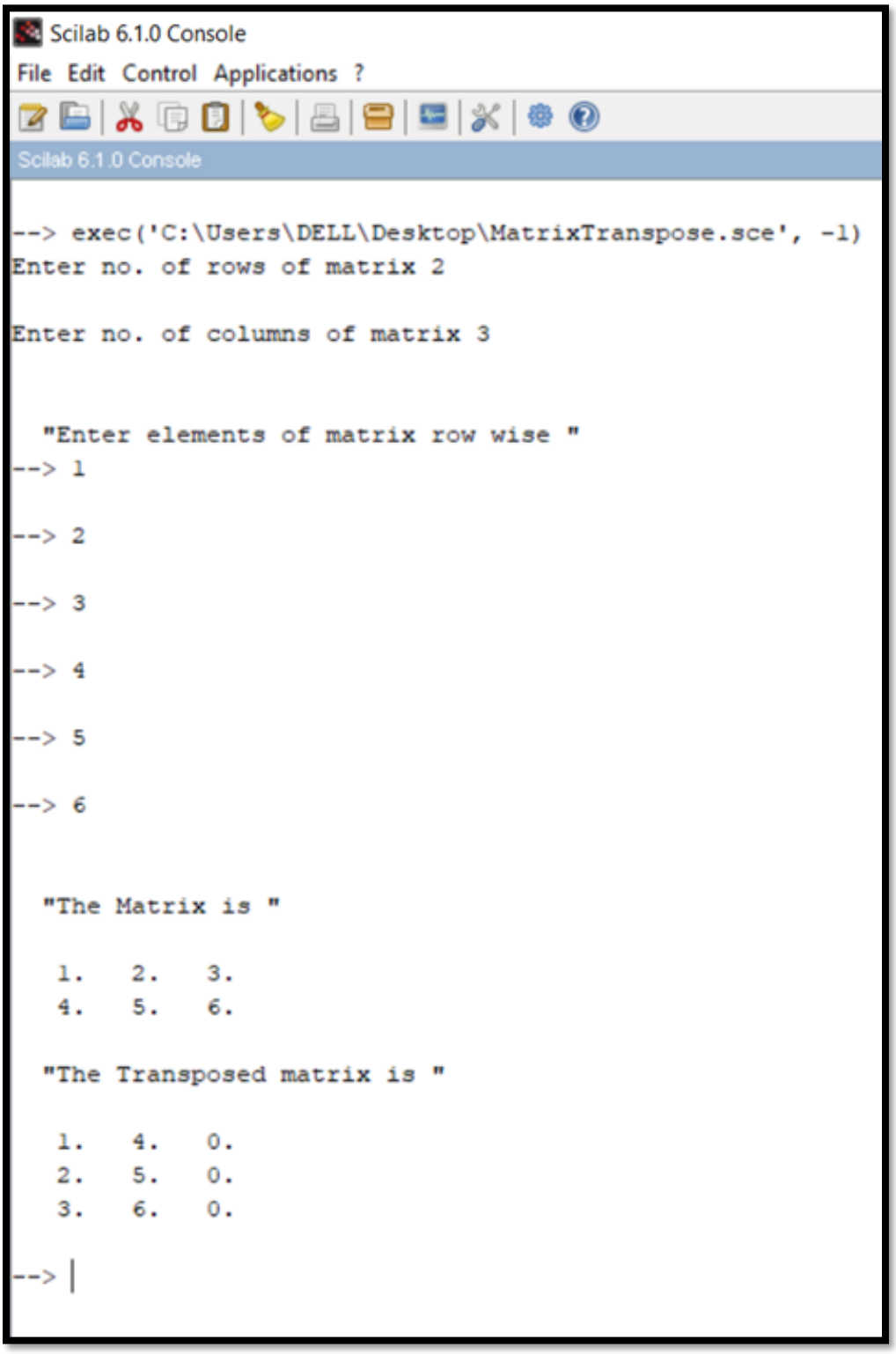
c) To find multiplication of two matrices

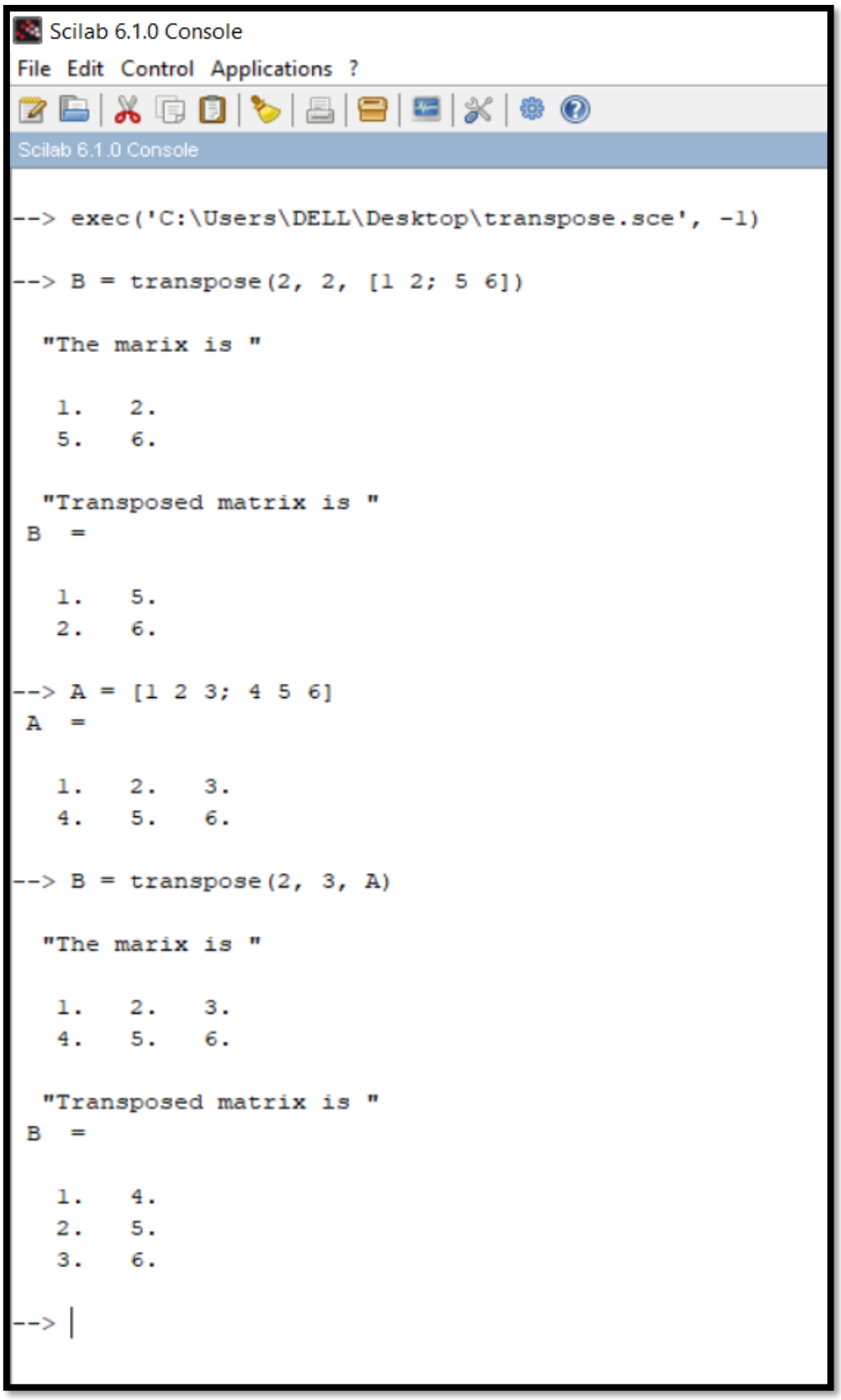
To Find transpose of a Matrix

Source Code:

```
m = input('Enter no. of rows of matrix')
n = input('Enter no. of columns of matrix')
A = zeros(m, n);
B = zeros(m, n);

disp('Enter elements of matrix row wise ')

for i = 1:m
   for j = 1:n
      A(i, j) = input('');
   end
end

for i = 1:n
   for j = 1:m
      B(i, j) = A(j, i);
   end
end

disp('The Matrix is ', A)
disp('The Transposed matrix is ', B)
```

Output:


Scilab 6.1.0 Console
File Edit Control Applications ?
Scilab 6.1.0 Console
--> exec('C:\Users\DELL\Desktop\MatrixTranspose.sce', -1)
Enter no. of rows of matrix 2
Enter no. of columns of matrix 3
"Enter elements of matrix row wise "
--> 1
--> 2
--> 3
--> 4
--> 5
--> 6
"The Matrix is "
1. 2. 3.
4. 5. 6.
"The Transposed matrix is "
1. 4. 0.
2. 5. 0.
3. 6. 0.
-->

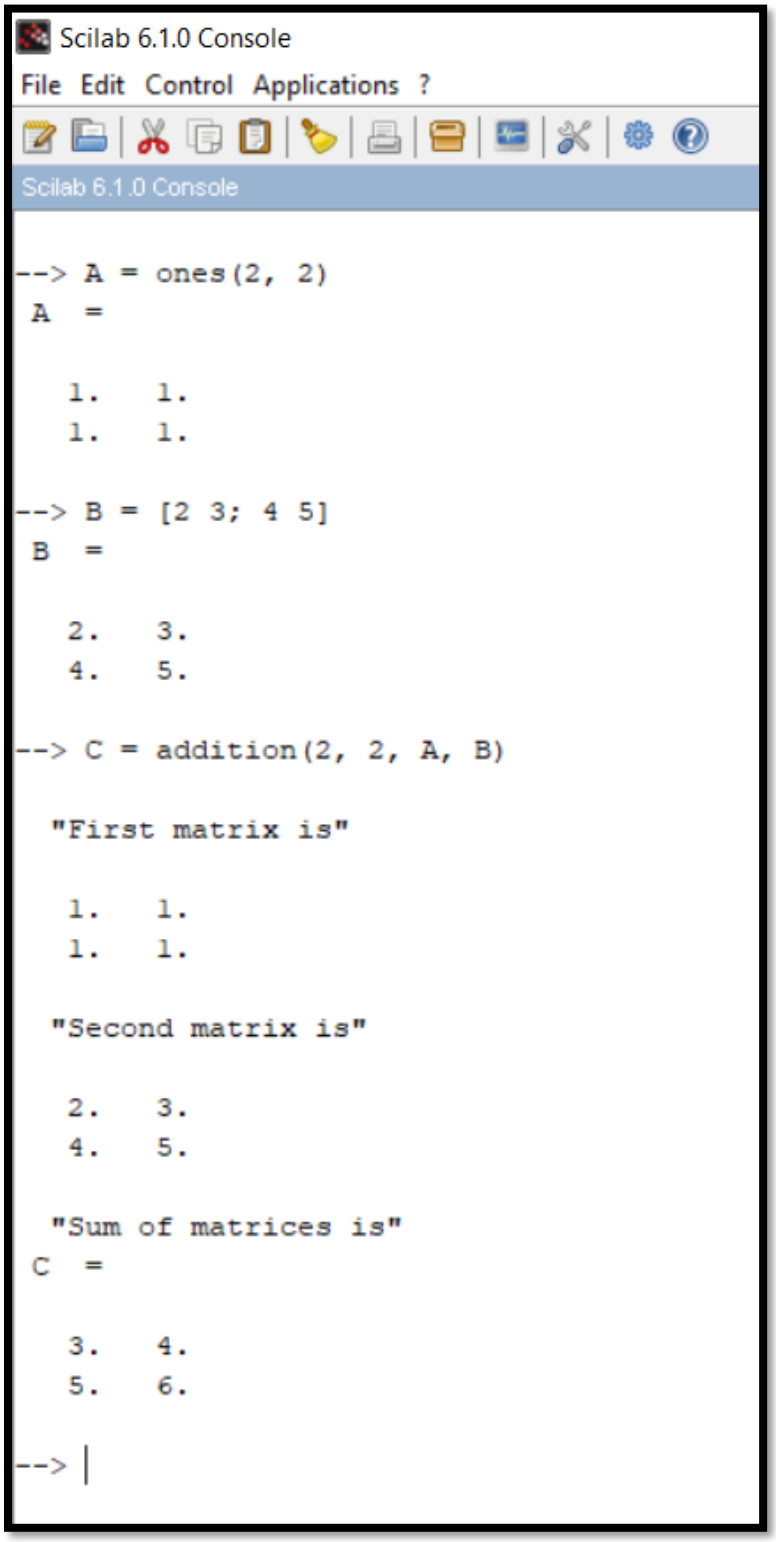

Scilab 6.1.0 Console
File Edit Control Applications ?
Scilab 6.1.0 Console
--> exec('C:\Users\DELL\Desktop\transpose.sce', -1)
--> B = transpose(2, 2, [1 2; 5 6])
"The marix is "
1. 2.
5. 6.
"Transposed matrix is "
B =
1. 5.
2. 6.
--> A = [1 2 3; 4 5 6]
A =
1. 2. 3.
4. 5. 6.
--> B = transpose(2, 3, A)
"The marix is "
1. 2. 3.
4. 5. 6.
"Transposed matrix is "
B =
1. 4.
2. 5.
3. 6.
-->

To Find Addition of two Matrices

Source Code:

```
m = input('Enter no. of rows')
n = input('Enter no. of columns')
A = zeros(m, n)
B = zeros(m, n)
C = zeros(m, n)

disp('Enter elements of first matrix row wise ')

for i = 1:m
   for j = 1:n
      A(i, j) = input('')
   end
end

disp('Enter elements of second matrix row wise ')

for i = 1:m
   for j = 1:n
      B(i, j) = input('')
   end
end

for i = 1:m
   for j = 1:n
      C(i, j) = A(i, j) + B(i, j)
   end
end

disp('First Matrix is ', A);
disp('Second Matrix is ', B);
disp('Sum of the matrices is ', C);
```

Output:


Scilab 6.1.0 Console
File Edit Control Applications ?
Scilab 6.1.0 Console
--> A = ones(2, 2)
A =
1. 1.
1. 1.
--> B = [2 3; 4 5]
B =
2. 3.
4. 5.
--> C = addition(2, 2, A, B)
"First matrix is"
1. 1.
1. 1.
"Second matrix is"
2. 3.
4. 5.
"Sum of matrices is"
C =
3. 4.
5. 6.
-->

Scilab 6.1.0 Console

File Edit Control Applications ?

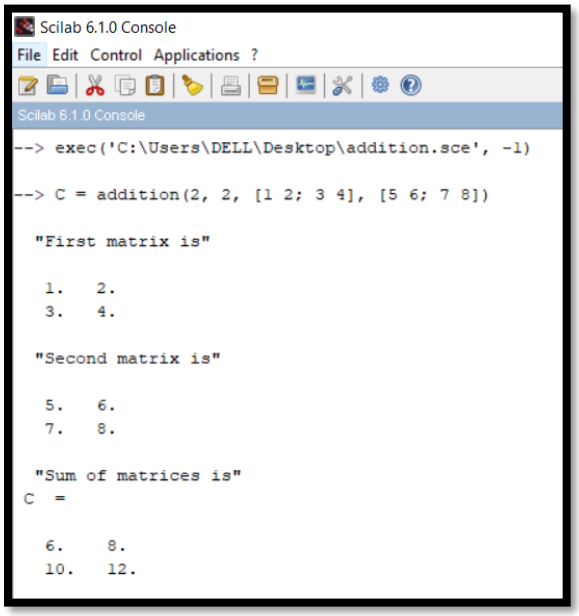
Scilab 6.1.0 Console

```
--> exec('C:\Users\DELL\Desktop\MatrixAddition.sce', -1)
Enter no. of rows 3

Enter no. of columns 2


  "Enter elements of first matrix row wise "
--> 1

--> 2

--> 3

--> 4

--> 5

--> 6


  "Enter elements of second matrix row wise "
--> 7

--> 8

--> 9

--> 10

--> 11

--> 12

  "First Matrix is "

   1.    2.
   3.    4.
   5.    6.

  "Second Matrix is "

   7.    8.
   9.    10.
   11.   12.

  "Sum of the matrices is "

   8.    10.
   12.   14.
   16.   18.
```

Source Code:

```
function [C]=addition(m, n, A, B)
   C = zeros(m, n);
   C = A + B;
   disp('First matrix is', A)
   disp('Second matrix is', B)
   disp('Sum of matrices is')
endfunction
```

Output:


Scilab 6.1.0 Console
File Edit Control Applications ?
Scilab 6.1.0 Console
--> exec('C:\Users\DELL\Desktop\addition.sce', -1)
--> C = addition(2, 2, [1 2; 3 4], [5 6; 7 8])
"First matrix is"
1. 2.
3. 4.
"Second matrix is"
5. 6.
7. 8.
"Sum of matrices is"
C =
6. 8.
10. 12.

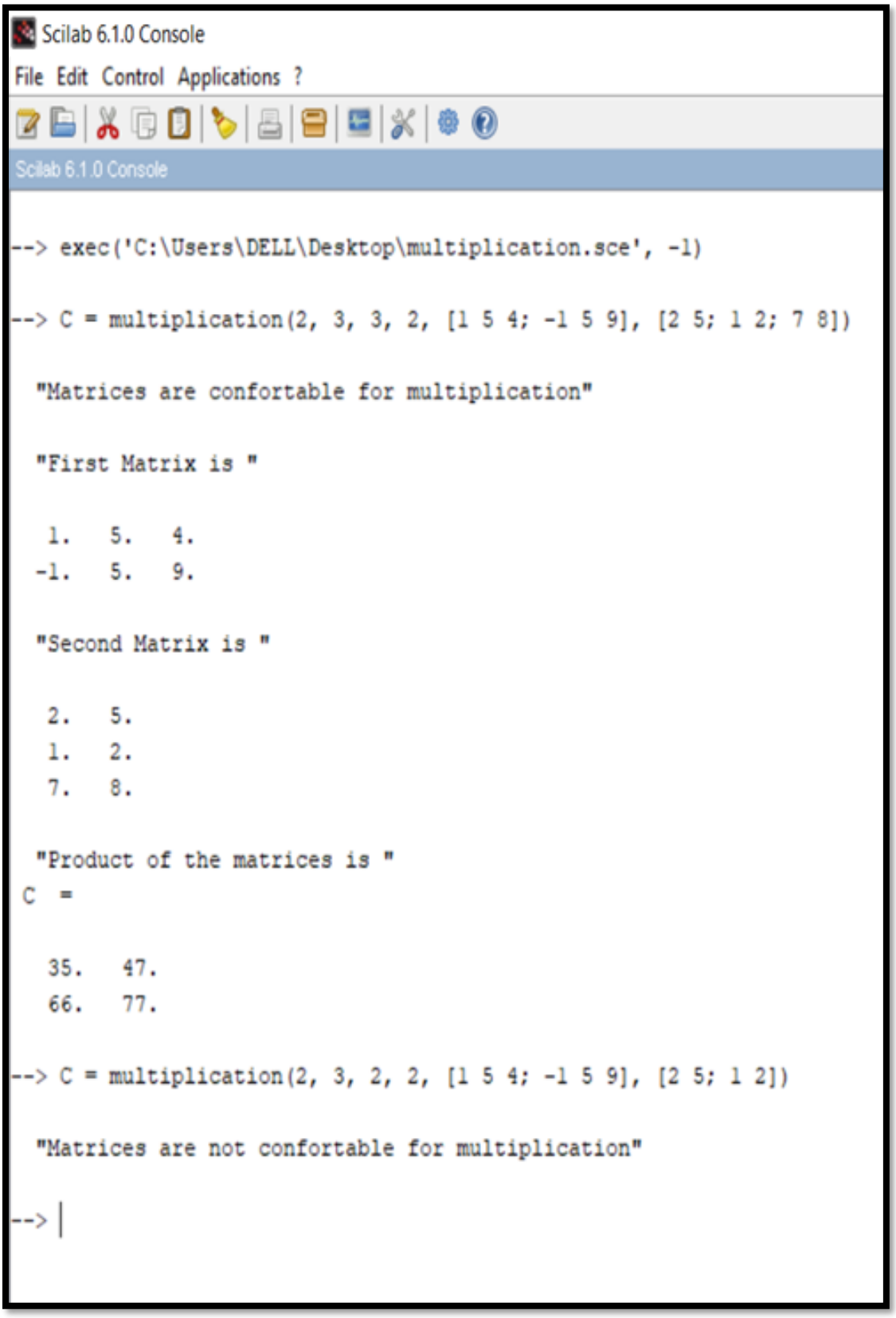
To Find multiplication of two Matrices

Source Code:

```
m = input('Enter no. of rows columns of first matrix')
n = input('Enter no. of columns of first matrix')
p = input('Enter no. of rows of second matrix')
q = input('Enter no. of columns of second matrix')

if n == p then
  disp('Matrices are confortable for multiplication')
else
   disp('Matrices are not confortable for multiplication')
   abort
end

A = zeros(m, n)
B = zeros(p, q)
C = zeros(m, q)

disp('Enter elements of first matrix row wise ')

for i = 1:m
   for j = 1:n
      A(i, j) = input('')
   end
end

disp('Enter elements of second matrix row wise ')

for i = 1:p
   for j = 1:q
      B(i, j) = input('')
   end
end

for i = 1:m
   for j = 1:q
      for k = 1:n
```

```
        C(i, j) = C(i, j) + (A(i, j) * B(k , j))
      end
   end
end

disp('First Matrix is ', A);
disp('Second Matrix is ', B);
disp('Product of the matrices is ', C);
```

Output:


Scilab 6.1.0 Console
File Edit Control Applications ?
Scilab 6.1.0 Console
--> exec('C:\Users\DELL\Desktop\multiplication.sce', -1)
--> C = multiplication(2, 3, 3, 2, [1 5 4; -1 5 9], [2 5; 1 2; 7 8])
"Matrices are confortable for multiplication"
"First Matrix is "
1. 5. 4.
-1. 5. 9.
"Second Matrix is "
2. 5.
1. 2.
7. 8.
"Product of the matrices is "
C =
35. 47.
66. 77.
--> C = multiplication(2, 3, 2, 2, [1 5 4; -1 5 9], [2 5; 1 2])
"Matrices are not confortable for multiplication"
-->

Scilab 6.1.0 Console

File Edit Control Applications ?

Scilab 6.1.0 Console

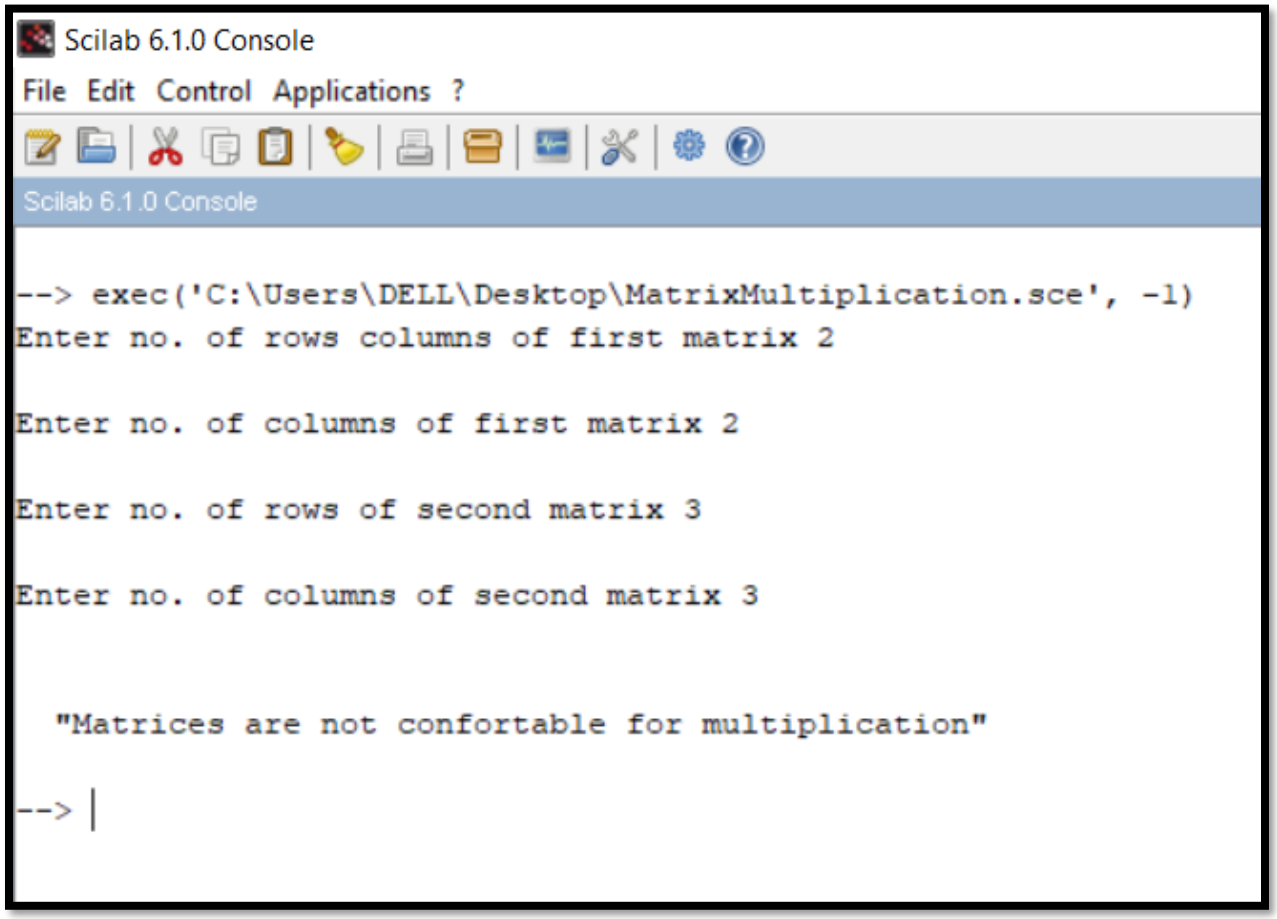
```
--> exec('C:\Users\DELL\Desktop\MatrixMultiplication.sce', -1)
Enter no. of rows columns of first matrix 2

Enter no. of columns of first matrix 2

Enter no. of rows of second matrix 2

Enter no. of columns of second matrix 2


  "Matrices are confortable for multiplication"

  "Enter elements of first matrix row wise "
--> 4

--> 5

--> 2

--> 2


  "Enter elements of second matrix row wise "
--> 1

--> 6

--> 5

--> 4


  "First Matrix is "

   4.    5.
   2.    2.

  "Second Matrix is "

   1.    6.
   5.    4.

  "Product of the matrices is "

   24.    50.
   12.    20.
```

To Find multiplication of two Matrices

Source Code:

```
function [C]=multiplication(m, n, p, q, A, B)
   C = zeros(m, n)
   if n == p then
      disp('Matrices are confortable for multiplication')
   else
      disp('Matrices are not confortable for multiplication')
      abort
   end
   C = A * B
   disp('First Matrix is ', A);
   disp('Second Matrix is ', B);
   disp('Product of the matrices is ')
endfunction
```

Output:


Scilab 6.1.0 Console
File Edit Control Applications ?
Scilab 6.1.0 Console
--> exec('C:\Users\DELL\Desktop\MatrixMultiplication.sce', -1)
Enter no. of rows columns of first matrix 2
Enter no. of columns of first matrix 2
Enter no. of rows of second matrix 3
Enter no. of columns of second matrix 3
"Matrices are not confortable for multiplication"
-->

EXPERIMENT - 2

APPLIED MATHEMATICS LAB

Aim

To find the inverse of a square matrix using Gauss-Jordan method.

Syeda Reeha Quasar
14114802719
4C7

EXPERIMENT – 2

Aim:

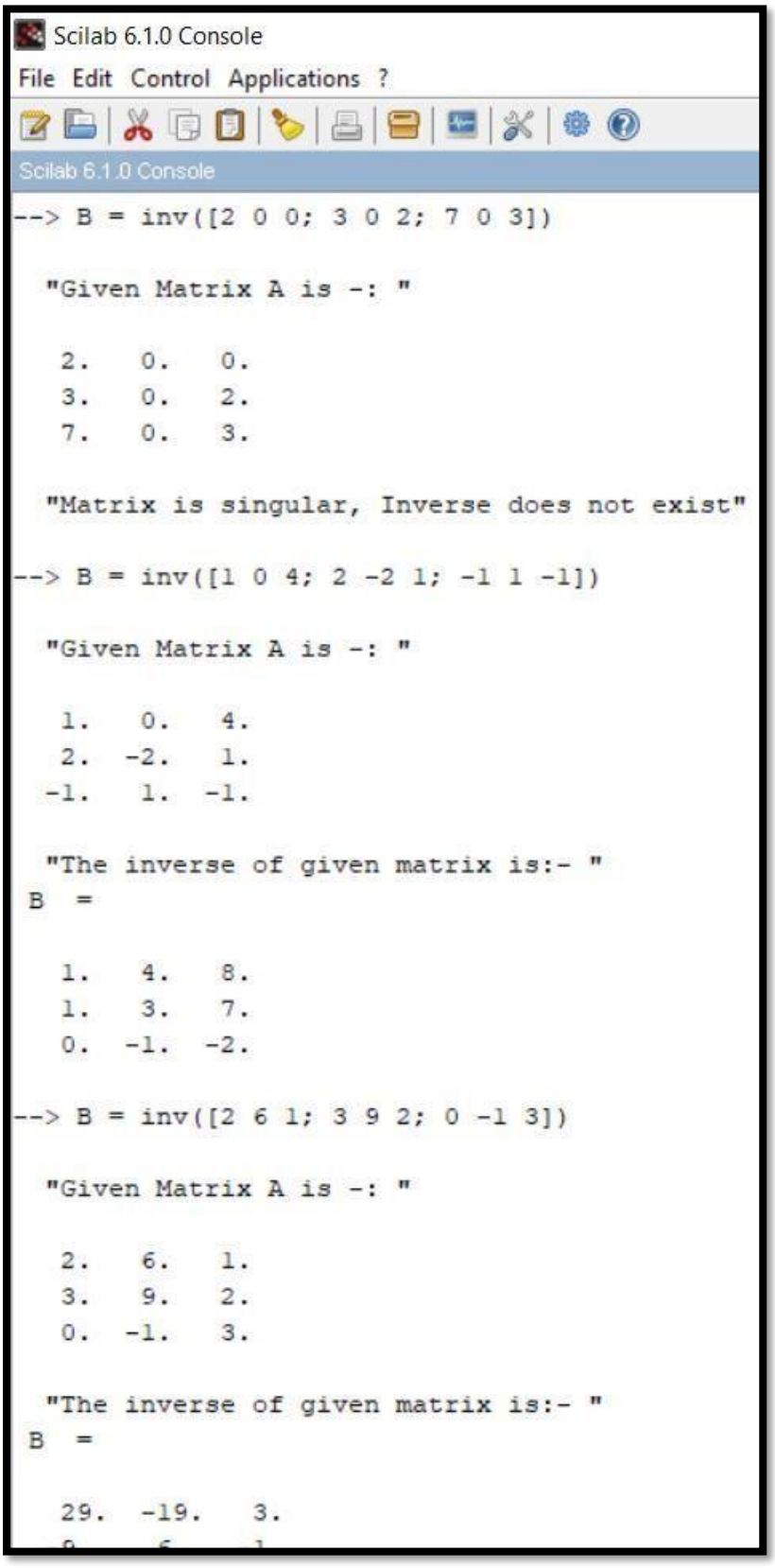
To find the inverse of a square matrix using Gauss-Jordan method.

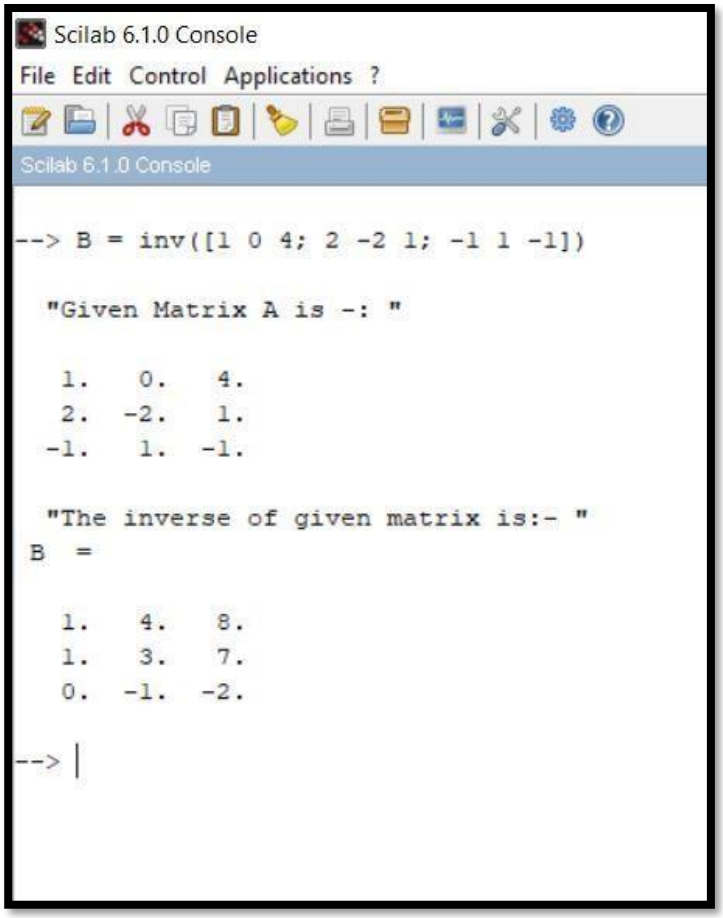
Source Code:

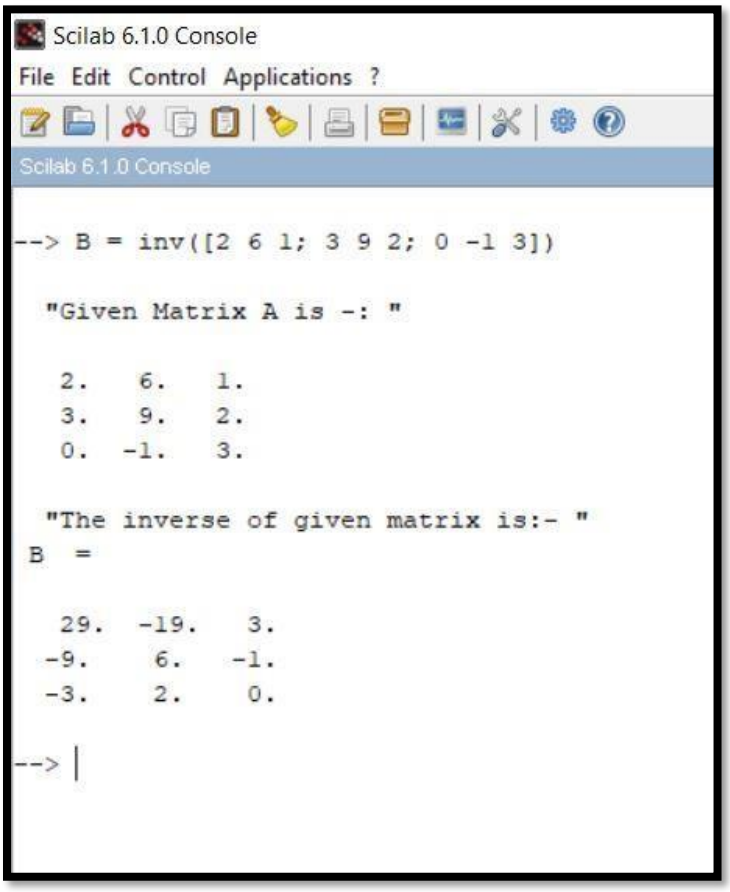
```
function [B]=inv(A)
  B = eye(3, 3)

  disp('Given Matrix A is -: ', A)

  if (det(A) == 0) then
    disp('Matrix is singular, Inverse does not exist')
    abort
  end

  Aug = [A, B]

  if (Aug(1, 1) == 0 & Aug(2, 1) ~= 0)  then
    C(1, :) = Aug(1, :)
    Aug(1, :) = Aug(2, :)
    Aug(2, :) = C(1, :)
  elseif (Aug(1, 1) == 0 & Aug(3, 1) ~= 0) then
    C(1, :) = Aug(1, :)
    Aug(1, :) = Aug(3, :)
    Aug(3, :) = C(1, :)
  end

  Aug(1, :) = Aug(1, :)/Aug(1, 1)
  Aug(2, :) = Aug(2, :) - Aug(2, 1) * Aug(1, :)
  Aug(3, :) = Aug(3, :) - Aug(3, 1) * Aug(1, :)

  if (Aug(2, 2) == 0) then
    C(2, :) = Aug(2, :)
    Aug(2, :) = Aug(3, :)
    Aug(3, :) = C(2, :)
```

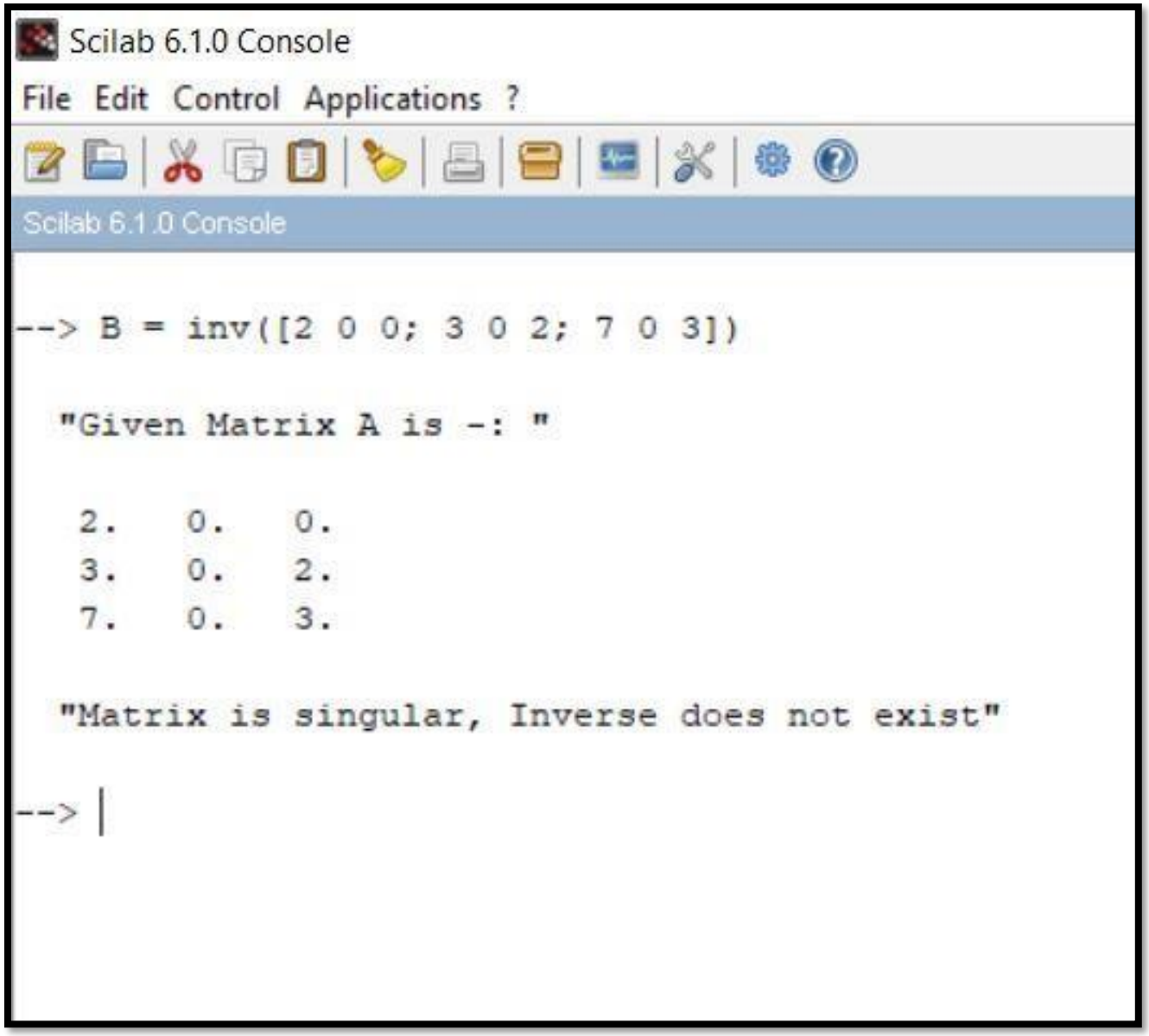
```
    end

    Aug(2, :) = Aug(2, :)/Aug(2, 2)
    Aug(1, :) = Aug(1, :) - Aug(1, 2) * Aug(2, :)
    Aug(3, :) = Aug(3, :) - Aug(3, 2) * Aug(2, :)

    Aug(3, :) = Aug(3, :)/Aug(3, 3)
    Aug(1, :) = Aug(1, :) - Aug(1, 3) * Aug(3, :)
    Aug(2, :) = Aug(2, :) - Aug(2, 3) * Aug(3, :)

    Aug(:, 1:3) = []
    B = Aug(:, 1: 3);

    printf('\n\n Name - Syeda Reeha Quasar \n Enrolment No. - 14114802719
\n Group - C7 \n\n')

    disp('The inverse of given matrix is:- ')
endfunction
```

Output:


Scilab 6.1.0 Console
File Edit Control Applications ?
Scilab 6.1.0 Console
--> B = inv([2 0 0; 3 0 2; 7 0 3])
"Given Matrix A is -: "
2. 0. 0.
3. 0. 2.
7. 0. 3.
"Matrix is singular, Inverse does not exist"
--> B = inv([1 0 4; 2 -2 1; -1 1 -1])
"Given Matrix A is -: "
1. 0. 4.
2. -2. 1.
-1. 1. -1.
"The inverse of given matrix is:- "
B =
1. 4. 8.
1. 3. 7.
0. -1. -2.
--> B = inv([2 6 1; 3 9 2; 0 -1 3])
"Given Matrix A is -: "
2. 6. 1.
3. 9. 2.
0. -1. 3.
"The inverse of given matrix is:- "
B =
29. -19. 3.


Scilab 6.1.0 Console
File Edit Control Applications ?
Scilab 6.1.0 Console
--> B = inv([1 0 4; 2 -2 1; -1 1 -1])
"Given Matrix A is -: "
1. 0. 4.
2. -2. 1.
-1. 1. -1.
"The inverse of given matrix is:- "
B =
1. 4. 8.
1. 3. 7.
0. -1. -2.
-->


Scilab 6.1.0 Console
File Edit Control Applications ?
Scilab 6.1.0 Console
--> B = inv([2 6 1; 3 9 2; 0 -1 3])
"Given Matrix A is -: "
2. 6. 1.
3. 9. 2.
0. -1. 3.
"The inverse of given matrix is:- "
B =
29. -19. 3.
-9. 6. -1.
-3. 2. 0.
-->


Scilab 6.1.0 Console
File Edit Control Applications ?
Scilab 6.1.0 Console
--> B = inv([2 0 0; 3 0 2; 7 0 3])
"Given Matrix A is -: "
2. 0. 0.
3. 0. 2.
7. 0. 3.
"Matrix is singular, Inverse does not exist"
-->

EXPERIMENT - 3

APPLIED MATHEMATICS LAB

Aim

To find Eigen values and Eigen vectors of a square matrix.

Syeda Reeha Quasar
14114802719
4C7

EXPERIMENT – 3

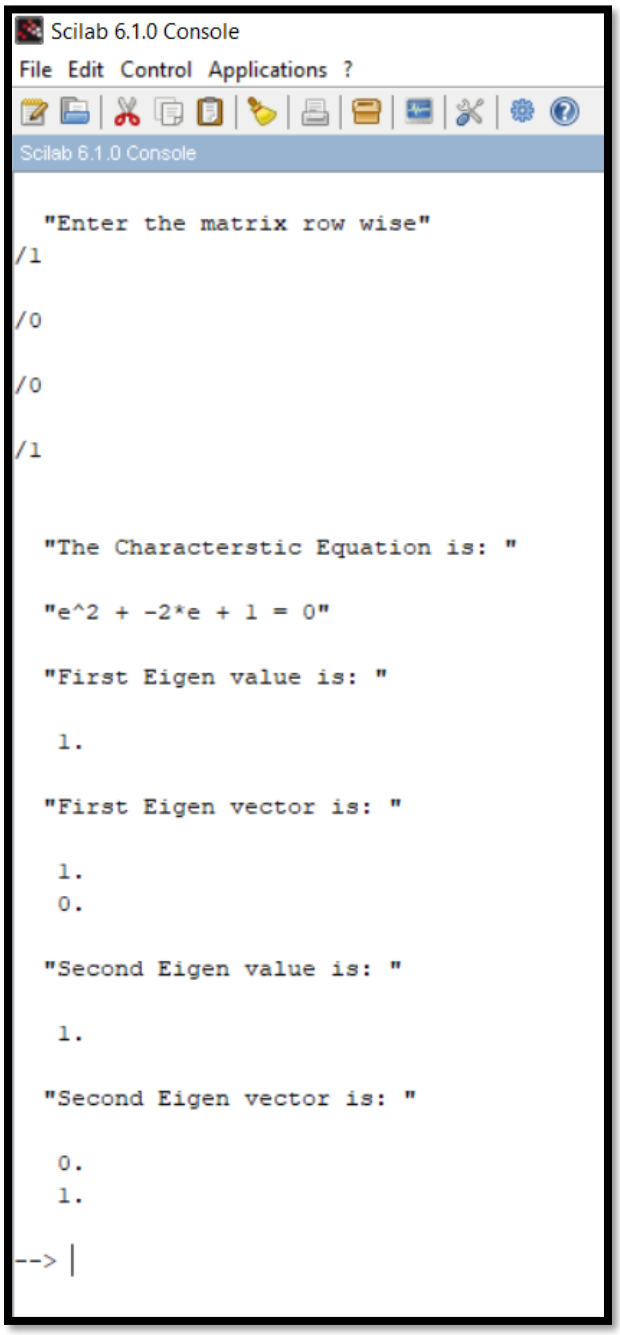
Aim:

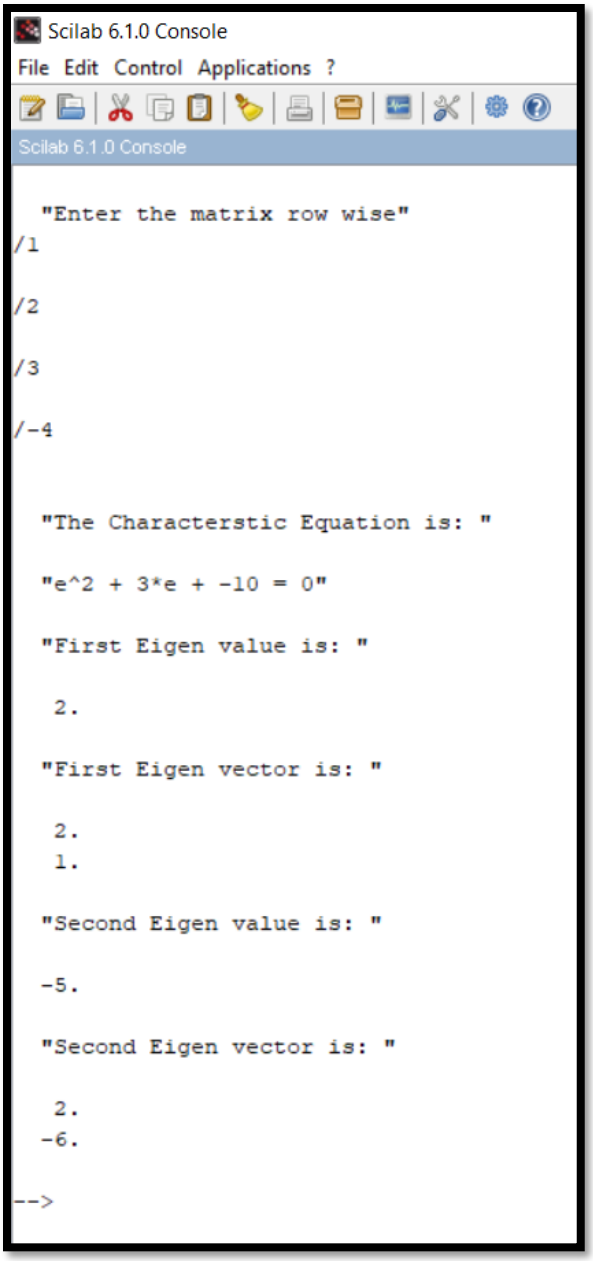
To find Eigen values and Eigen vectors of a square matrix.

Source Code:

```
clc;

printf('\n\n Name - Syeda Reeha Quasar \n Enrolment No. - 14114802719 \n Group - C7 \n\n');
disp("Enter the matrix row wise");

for i = 1:2
   for j = 1:2
      A(i,j) = input('/');
   end
end

trce = A(1, 1) + A(2, 2);
determinant = A(1, 1) * A(2, 2) - A(1, 2) * A(2, 1);

disp("The Characterstic Equation is: ");
disp(['e^2 + ' + string(-trce) + '*e + ' + string(determinant) + ' = 0']);

e1 = (trce + sqrt(trce^2 - 4 * determinant))/2;
e2 = (trce - sqrt(trce^2 - 4 * determinant))/2;

if A(1, 2) ~= 0 then
   v1 = [A(1, 2); e1 - A(1, 1)];
   v2 = [A(1, 2); e2 - A(1, 1)];
elseif A(2, 1) ~= 0 then
   v1 = [e1 - A(2, 2); A(2, 1)];
   v2 = [e2 - A(2, 2); A(2, 1)];
else
   v1 = [1; 0];
   v2 = [0; 1];
end

disp("First Eigen value is: ", e1);
disp("First Eigen vector is: ", v1);
disp("Second Eigen value is: ", e2);
disp("Second Eigen vector is: ", v2);
```

Output:


Scilab 6.1.0 Console
File Edit Control Applications ?
Scilab 6.1.0 Console
"Enter the matrix row wise"
/1
/0
/0
/1
"The Characterstic Equation is: "
"e^2 + -2*e + 1 = 0"
"First Eigen value is: "
1.
"First Eigen vector is: "
1.
0.
"Second Eigen value is: "
1.
"Second Eigen vector is: "
0.
1.
-->


Scilab 6.1.0 Console
File Edit Control Applications ?
Scilab 6.1.0 Console
"Enter the matrix row wise"
/1
/2
/3
/-4
"The Characterstic Equation is: "
"e^2 + 3*e + -10 = 0"
"First Eigen value is: "
2.
"First Eigen vector is: "
2.
1.
"Second Eigen value is: "
-5.
"Second Eigen vector is: "
2.
-6.
-->

EXPERIMENT - 4

APPLIED MATHEMATICS LAB

Aim

To find the solution of algebraic and transcendental equations using
(a) Bisection method
(b) Newton- Raphson method.

Syeda Reeha Quasar
14114802719
4C7

EXPERIMENT – 4

Aim:

To find the solution of algebraic and transcendental equations using

(a) Bisection method

(b) Newton- Raphson method.

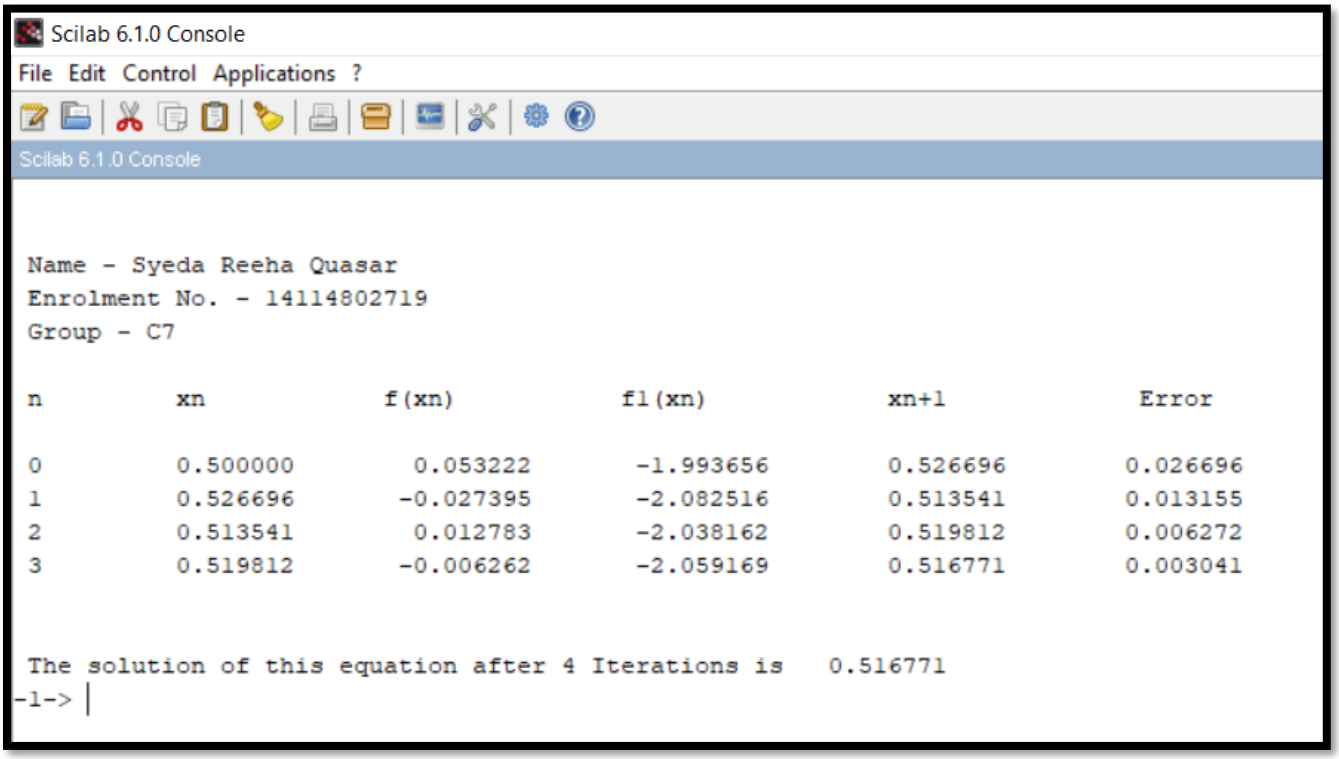
Source Code:

Newton RAPHSON

// Newton Raphson Method

```
clc
printf('\n\n Name - Syeda Reeha Quasar \n Enrolment No. - 14114802719 \n Group - C7 \n\n')

deff ('x = f(x)', 'x = cos(x) - x * exp(x)')
deff ('x = f1(x)', 'x = sin(x) - (x+1) * exp(x)')

x0 = 0.5; e = 0.00001;
printf(' n \t   xn \t\t f(xn) \t\t f1(xn) \t   xn+1 \t    Error \n\n')

for i = 1:4
   x1 = x0 - f(x0)/f1(x0)
   e1 = abs(x0 - x1)
   printf(' %i \t %10f \t %10f \t %10f \t %10f \t %10f \n', i-1, x0, f(x0), f1(x0), x1, e1)
   x0 = x1
   if e1 < e then
      break
   end
end

printf('\n\n The solution of this equation after %i Iterations is %10f', i, x1)
```

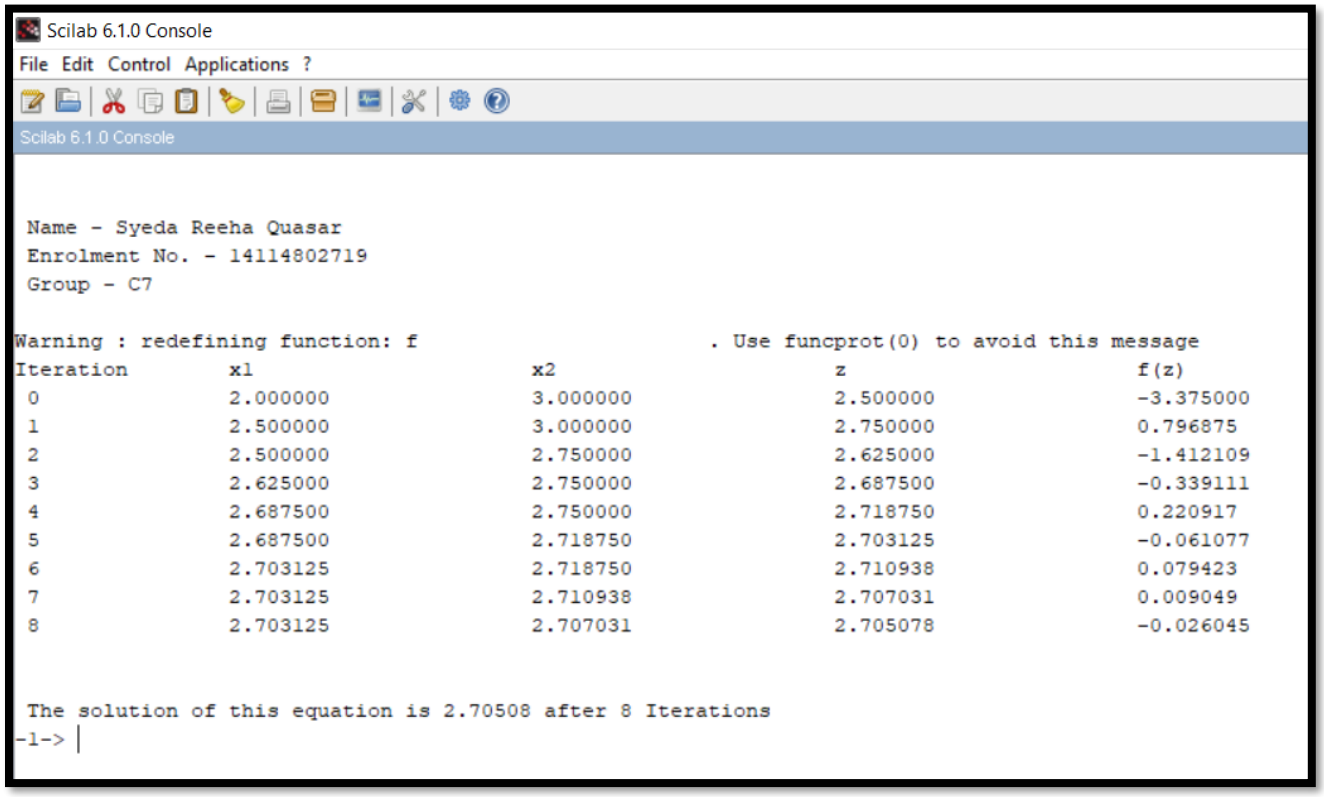
Output:


Scilab 6.1.0 Console
File Edit Control Applications ?
Scilab 6.1.0 Console
Name - Syeda Reeha Quasar
Enrolment No. - 14114802719
Group - C7
n xn f(xn) fl(xn) xn+1 Error
0 0.500000 0.053222 -1.993656 0.526696 0.026696
1 0.526696 -0.027395 -2.082516 0.513541 0.013155
2 0.513541 0.012783 -2.038162 0.519812 0.006272
3 0.519812 -0.006262 -2.059169 0.516771 0.003041
The solution of this equation after 4 Iterations is 0.516771
-1->

Bisection Method

```
// Bisection method

clc
printf('\n\n Name - Syeda Reeha Quasar \n Enrolment No. - 14114802719 \n Group - C7 \n\n')

deff ('y = f(x)', 'y = x^3- 4*x -9')

x1 = 2; x2 = 3; e = 0.001; i = 0;
printf('Iteration \t x1 \t\t\t x2 \t\t\t z \t\t\t f(z) \n')

while abs(x1 - x2) > 2*e
   z = (x1 + x2)/2
   printf(' %i \t\t %f \t\t %f \t\t %f \t\t %f \n', i, x1, x2, z, f(z))
   if f(z) * f(x1) > 0 then
      x1 = z
   else
      x2 = z
   end
   i = i + 1
end

printf('\n\n The solution of this equation is %g after %i Iterations', z, i-1)
```

Output:


Scilab 6.1.0 Console
File Edit Control Applications ?
Scilab 6.1.0 Console
Name - Syeda Reeha Quasar
Enrolment No. - 14114802719
Group - C7
Warning : redefining function: f . Use funcprot(0) to avoid this message
Iteration x1 x2 z f(z)
0 2.000000 3.000000 2.500000 -3.375000
1 2.500000 3.000000 2.750000 0.796875
2 2.500000 2.750000 2.625000 -1.412109
3 2.625000 2.750000 2.687500 -0.339111
4 2.687500 2.750000 2.718750 0.220917
5 2.687500 2.718750 2.703125 -0.061077
6 2.703125 2.718750 2.710938 0.079423
7 2.703125 2.710938 2.707031 0.009049
8 2.703125 2.707031 2.705078 -0.026045
The solution of this equation is 2.70508 after 8 Iterations
-1->

EXPERIMENT - 5

APPLIED MATHEMATICS LAB

Aim

To find the value of a definite integral using
(a) Trapezoidal rule
(b) Simpsons 1/3 rule
(c) Simpsons 3/8 rule.

Syeda Reeha Quasar
14114802719
4C7

EXPERIMENT – 5

Aim:

To find the value of a definite integral using

(a) Trapezoidal rule

(b) Simpsons 1/3 rule

(c) Simpsons 3/8 rule.

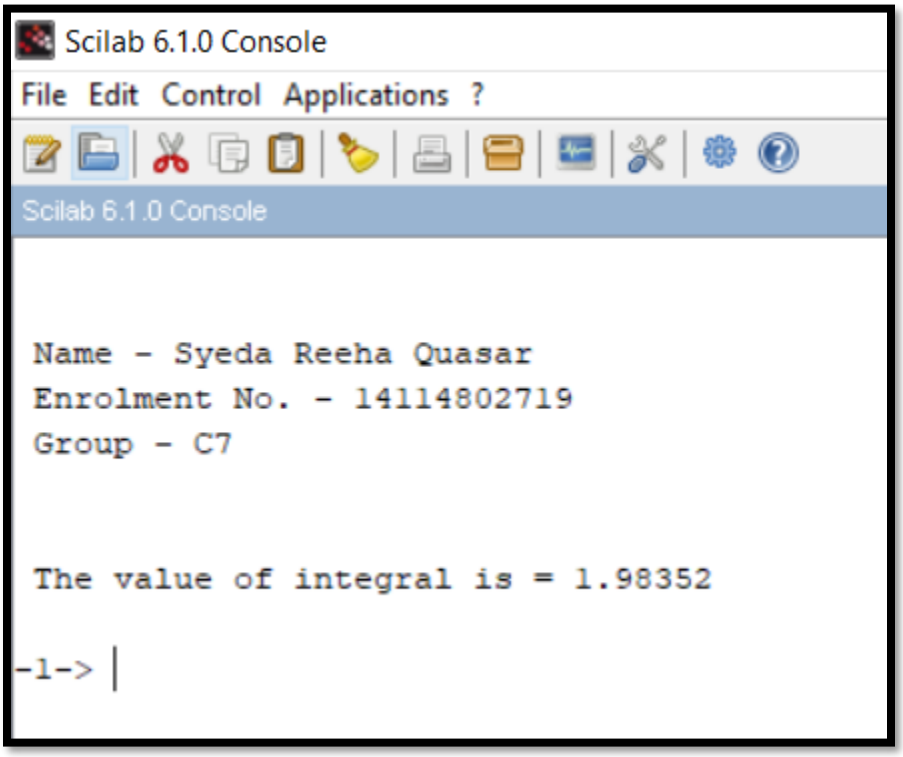
Source Code:

Trapezoidal rule

// Program to find integration by using Trapezoidal rule

```
clc
clear
close

printf('\n\n Name - Syeda Reeha Quasar \n Enrolment No. - 14114802719 \n Group - C7 \n\n')

deff ('y = f(x)', 'y = sin(x)')

x0 = 0
xn = %pi
n = 10
h = (xn - x0)/n
s = 0

for i = 1:n
   s = s + f(x0 + (i - 1)*h) + f(x0 + i * h)
end

integral = (h * s)/2

printf('\n The value of integral is = %g \n', integral)
```

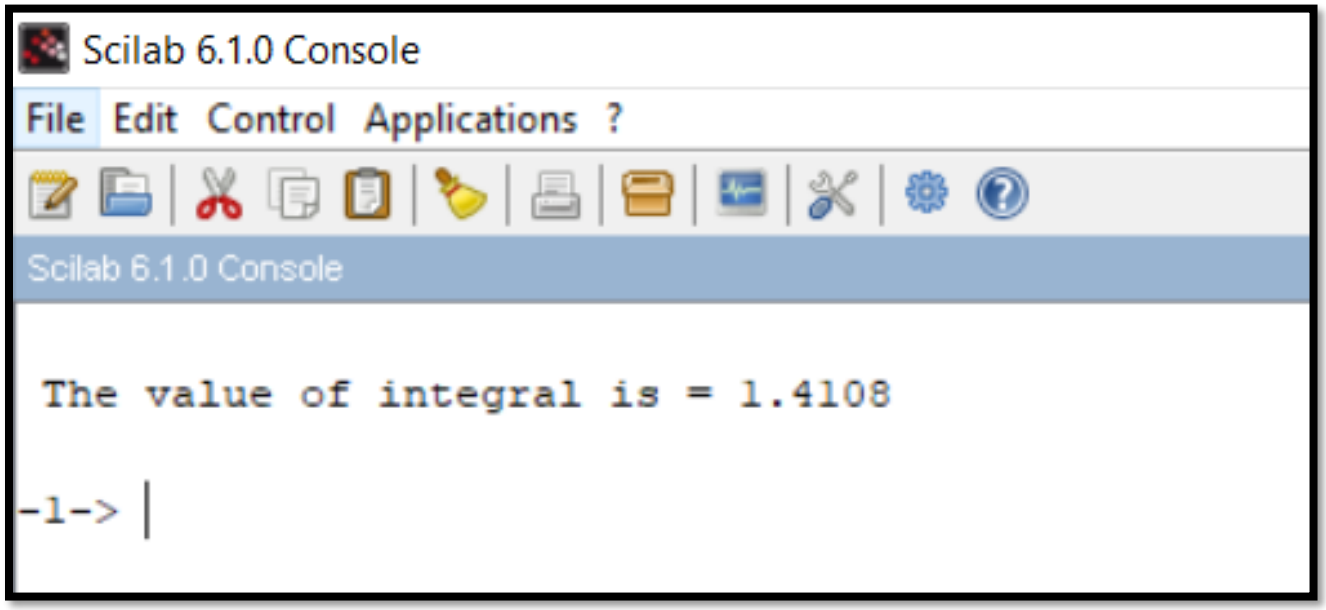
Output:


Scilab 6.1.0 Console
File Edit Control Applications ?
Scilab 6.1.0 Console
Name - Syeda Reeha Quasar
Enrolment No. - 14114802719
Group - C7
The value of integral is = 1.98352
-1->
Variable Browser
Command History
News feed
March 2021: IIT Bombay Workshop
Empower your students
Content

// Program to find integration by using Trapezoidal rule

```
clc
clear
close

deff ('y = f(x)', 'y = 1/(1 + x^2)')

x0 = 0
xn = 6
n = 6
h = (xn - x0)/n
s = 0

for i = 1:n
   s = s + f(x0 + (i - 1)*h) + f(x0 + i * h)
end

integral = (h * s)/2

printf('\n The value of integral is = %g \n', integral)
```

Output:

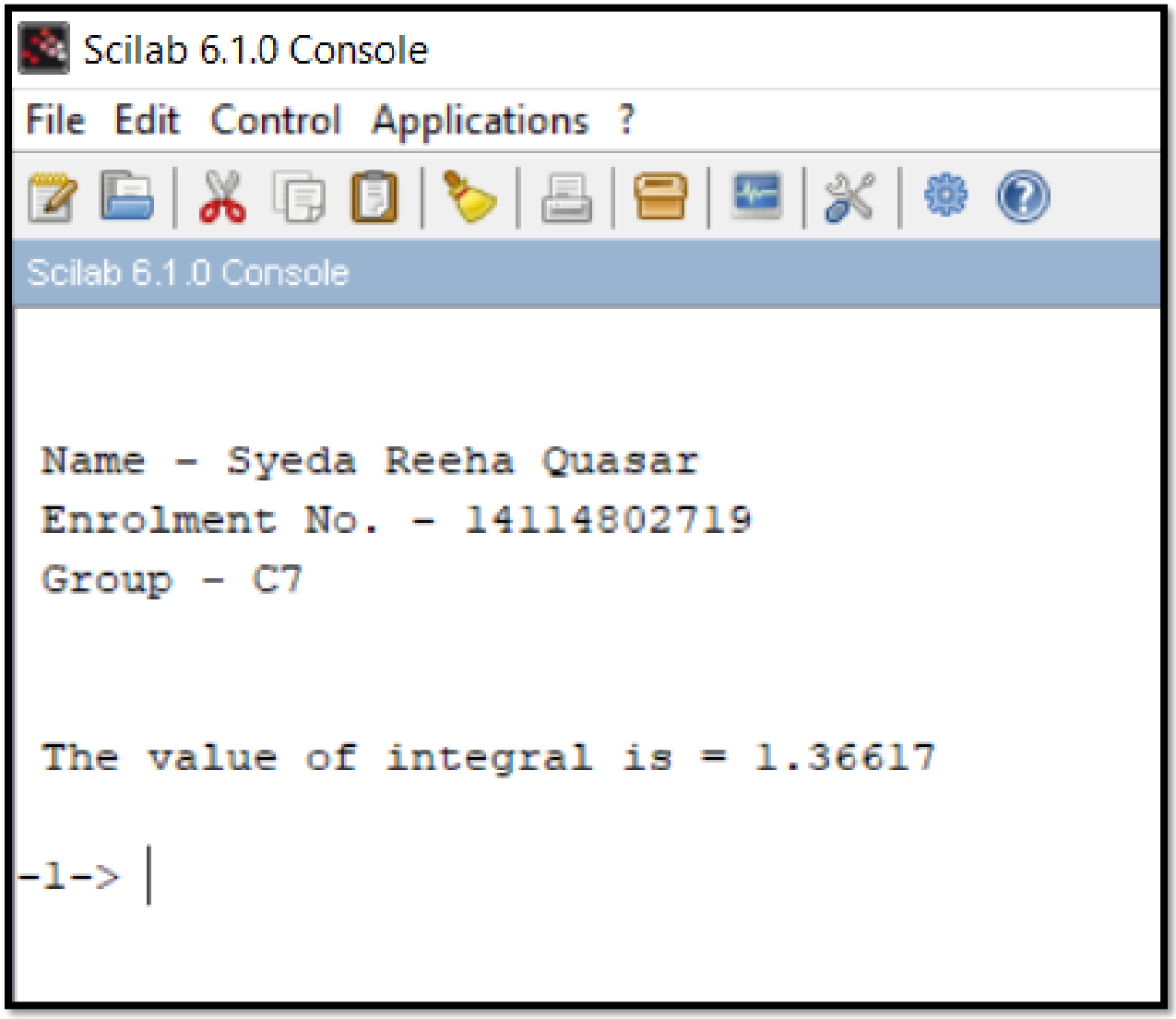

Scilab 6.1.0 Console
File Edit Control Applications ?
Scilab 6.1.0 Console
The value of integral is = 1.4108
-1->

Simpsons 1/3 rule

// Program to find integration by using Simpson's 1/3 rule

```
clc
clear
close

printf('\n\n Name - Syeda Reeha Quasar \n Enrolment No. - 14114802719 \n Group - C7 \n\n')

deff ('y = f(x)', 'y = 1/(1 + x^2)')

x0 = 0
xn = 6
n = 6
h = (xn - x0)/n
s = 0

for i = 1:2:n
   s = s + f(x0 + (i - 1)*h) + 4*f(x0 + i * h) + f(x0 + (i+1) * h)
end

integral = (h * s)/3

printf('\n The value of integral is = %g \n', integral)
```

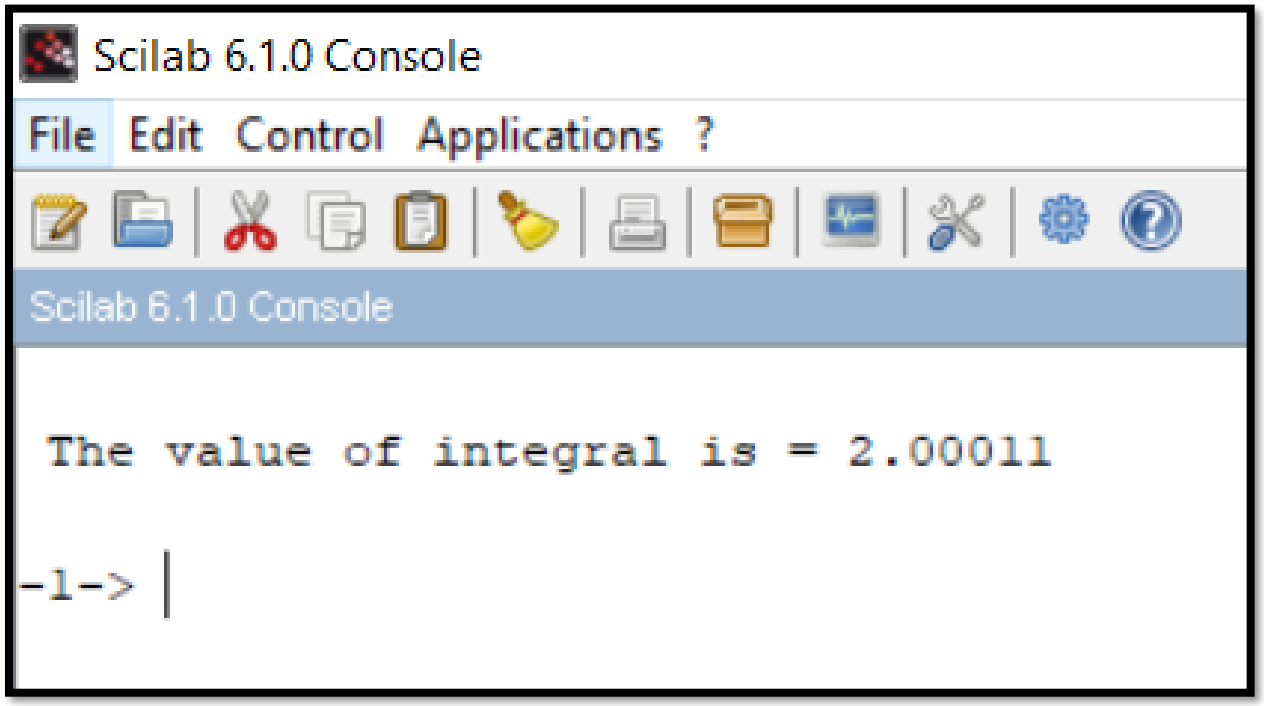
Output:


Scilab 6.1.0 Console
File Edit Control Applications ?
Scilab 6.1.0 Console
Name - Syeda Reeha Quasar
Enrolment No. - 14114802719
Group - C7
The value of integral is = 1.36617
-1->

// Program to find integration by using Simpson's 1/3 rule

```
clc
clear
close

deff ('y = f(x)', 'y = sin(x)')

x0 = 0
xn = %pi
n = 10
h = (xn - x0)/n
s = 0

for i = 1:2:n
   s = s + f(x0 + (i - 1)*h) + 4*f(x0 + i * h) + f(x0 + (i+1) * h)
end

integral = (h * s)/3

printf('\n The value of integral is = %g \n', integral)
```

Output:

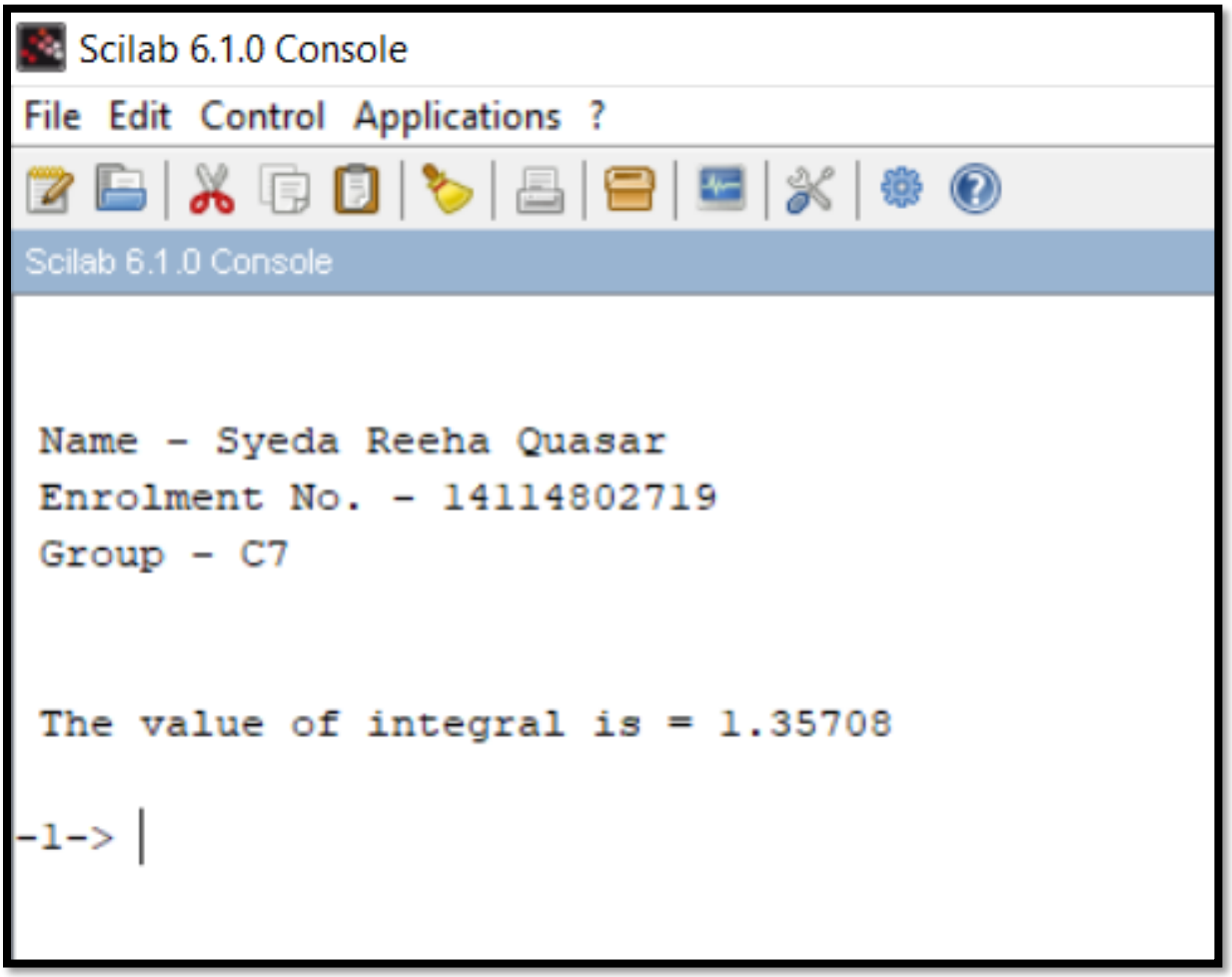

Scilab 6.1.0 Console
File Edit Control Applications ?
Scilab 6.1.0 Console
The value of integral is = 2.00011
-1->
Scilab 6.1.0 Console

Simpsons 3/8 rule

// Program to find integration by using Simpson's 3/8 rule

```
clc
clear
close

printf('\n\n Name - Syeda Reeha Quasar \n Enrolment No. - 14114802719 \n Group - C7 \n\n')

deff ('y = f(x)', 'y = 1/(1 + x^2)')

x0 = 0
xn = 6
n = 6
h = (xn - x0)/n
s = 0

for i = 1:3:n
   s = s + f(x0 + (i - 1)*h) + 3*f(x0 + i * h) + 3 * f(x0 + (i+1) * h) + f(x0 + (i+2) * h)
end

integral = (3 * h * s)/8

printf('\n The value of integral is = %g \n', integral)
```

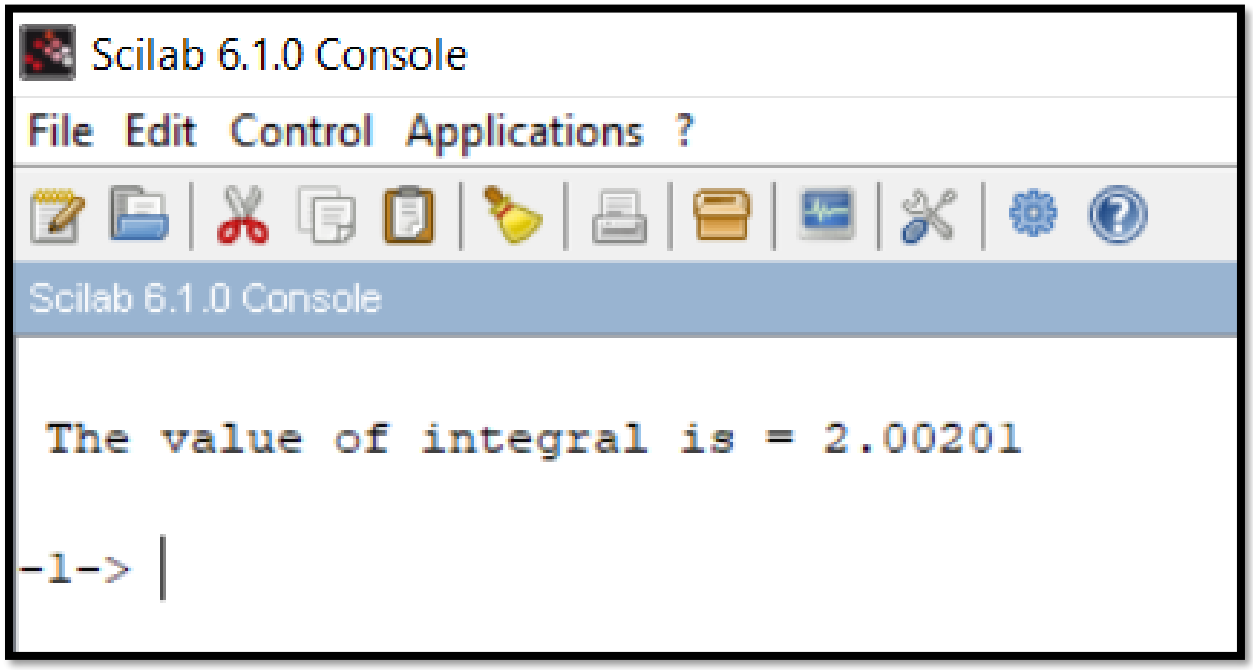
Output:


Scilab 6.1.0 Console
File Edit Control Applications ?
Scilab 6.1.0 Console
Name - Syeda Reeha Quasar
Enrolment No. - 14114802719
Group - C7
The value of integral is = 1.35708
-1->

// Program to find integration by using Simpson's 3/8 rule

```
clc
clear
close

deff ('y = f(x)', 'y = sin(x)')

x0 = 0
xn = %pi
n = 6
h = (xn - x0)/n
s = 0

for i = 1:3:n
   s = s + f(x0 + (i - 1)*h) + 3*f(x0 + i * h) + 3 * f(x0 + (i+1) * h) + f(x0 + (i+2) * h)
end

integral = (3 * h * s)/8

printf('\n The value of integral is = %g \n', integral)
```

Output:


Scilab 6.1.0 Console
File Edit Control Applications ?
Scilab 6.1.0 Console
The value of integral is = 2.00201
-1->

EXPERIMENT - 6

APPLIED MATHEMATICS LAB

Aim

To solve ordinary differential equations using Runge- Kutta Method.

Syeda Reeha Quasar
14114802719
4C7

EXPERIMENT – 6

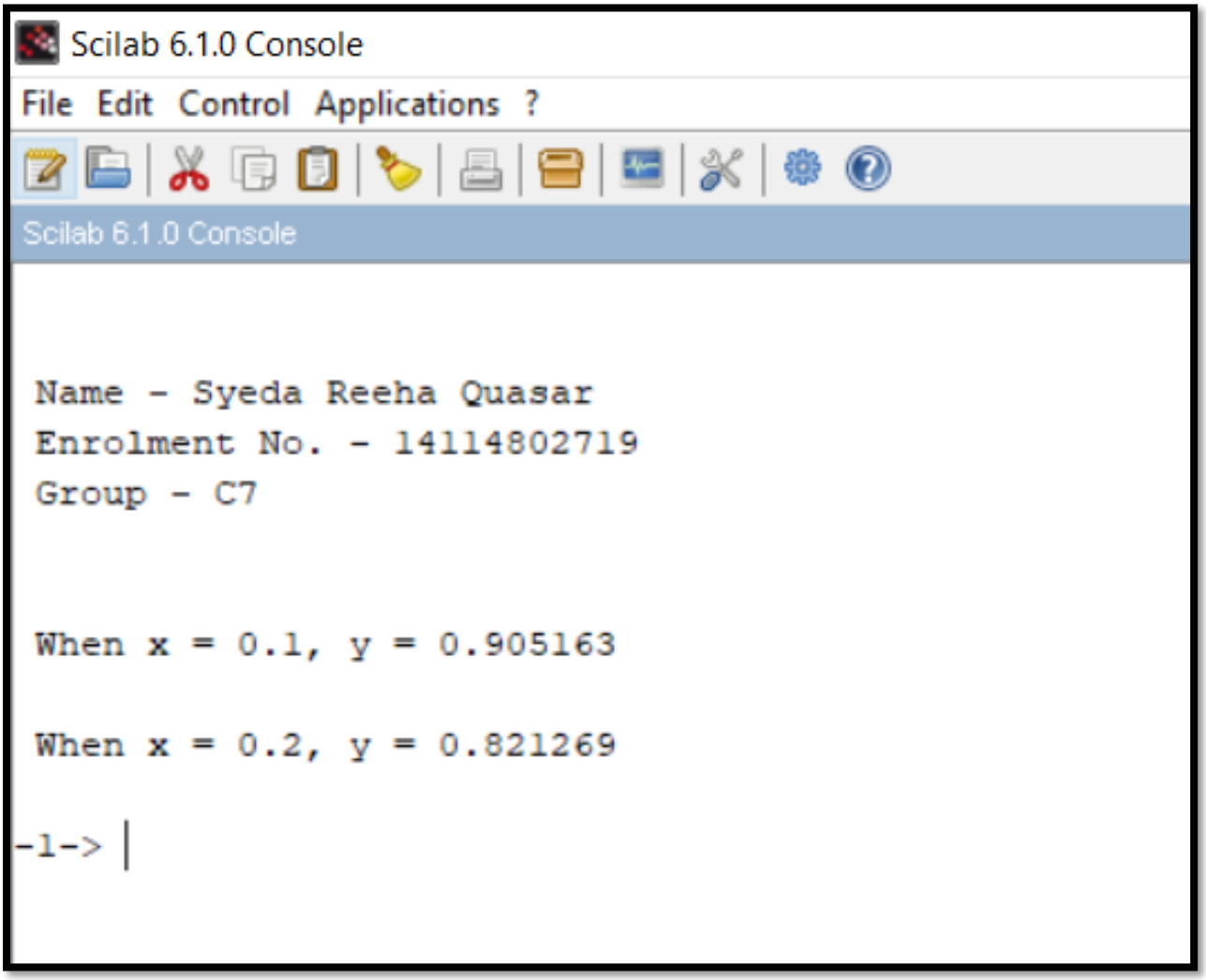
Aim:

To solve ordinary differential equations using Runge- Kutta Method.

Source Code:

```
clc; clear; close;

printf('\n\n Name - Syeda Reeha Quasar \n Enrolment No. - 14114802719 \n Group - C7
\n\n')

deff ('z = f(x,y)', 'z = x*x - y')

x0 = 0; y0 = 1; xn = 0.2; h=0.1;
x = x0;
y = y0;

while x ~= xn
   k1 = h * f(x,y);
   k2 = h * f(x+h/2, y+k1/2);
   k3 = h * f(x+h/2,y+k2/2);
   k4 = h * f(x+h, y+k3);
   k = (k1 + (k2 + k3) * 2 + k4)/6;
   x = x + h;
   y = y + k;
   printf("\n When x = %g, y = %g\n", x, y)
end
```

Output:


Scilab 6.1.0 Console
File Edit Control Applications ?
Scilab 6.1.0 Console
Name - Syeda Reeha Quasar
Enrolment No. - 14114802719
Group - C7
When x = 0.1, y = 0.905163
When x = 0.2, y = 0.821269
-1->

EXPERIMENT - 7

APPLIED MATHEMATICS LAB

Aim

To solve ordinary differential equations using Euler's method.

Syeda Reeha Quasar
14114802719
4C7

EXPERIMENT – 7

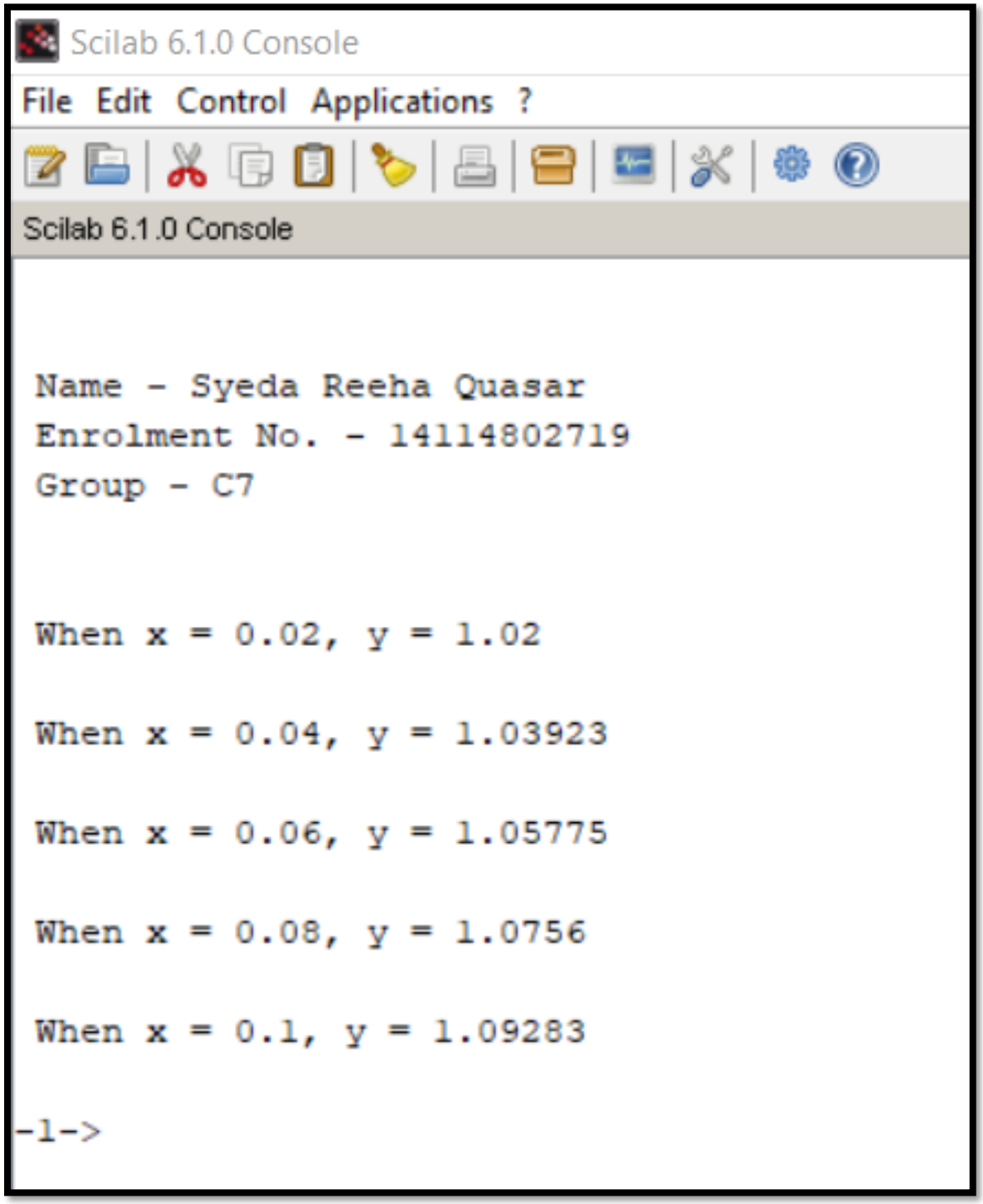
Aim:

To solve ordinary differential equations using Euler's method.

Source Code:

```
clc; clear; close;

printf('\n\n Name - Syeda Reeha Quasar \n Enrolment No. - 14114802719 \n
Group - C7 \n\n')

deff ('z = f(x,y)','z = (y-x)/(y+x)')

x0 = 0;
y0 = 1;
xn = 0.1;
h = 0.02;
x = x0;
y = y0;

while x ~= xn
   y = y + h * f(x,y);
   x=x+h;

printf("\n When x = %g, y = %g\n" ,x ,y)

end
```

Output:


Scilab 6.1.0 Console
File Edit Control Applications ?
Scilab 6.1.0 Console
Name - Syeda Reeha Quasar
Enrolment No. - 14114802719
Group - C7
When x = 0.02, y = 1.02
When x = 0.04, y = 1.03923
When x = 0.06, y = 1.05775
When x = 0.08, y = 1.0756
When x = 0.1, y = 1.09283
-1->

EXPERIMENT - 8

APPLIED MATHEMATICS LAB

Aim

To plot unit step function and square wave function.

Syeda Reeha Quasar
14114802719
4C7

EXPERIMENT – 8

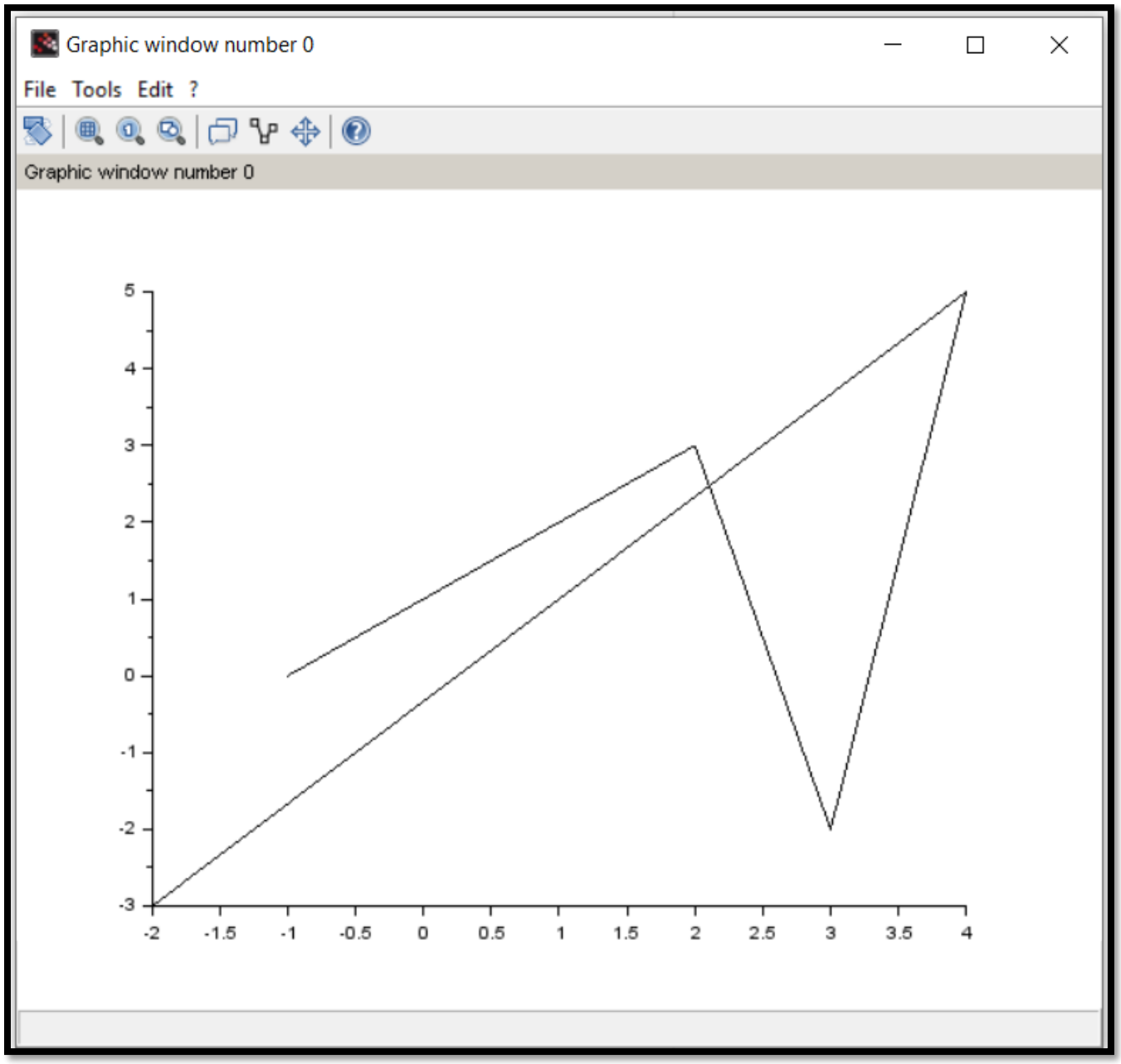
Aim:

To plot unit step function and square wave function.

Source Code:

```
clc;

printf('\n\n Name - Syeda Reeha Quasar \n Enrolment No. - 14114802719 \n
Group - C7 \n\n')

x = [1 -1 2 3 4 -2]
y = [2 0 3 -2 5 -3]

plot2d(x, y)
```

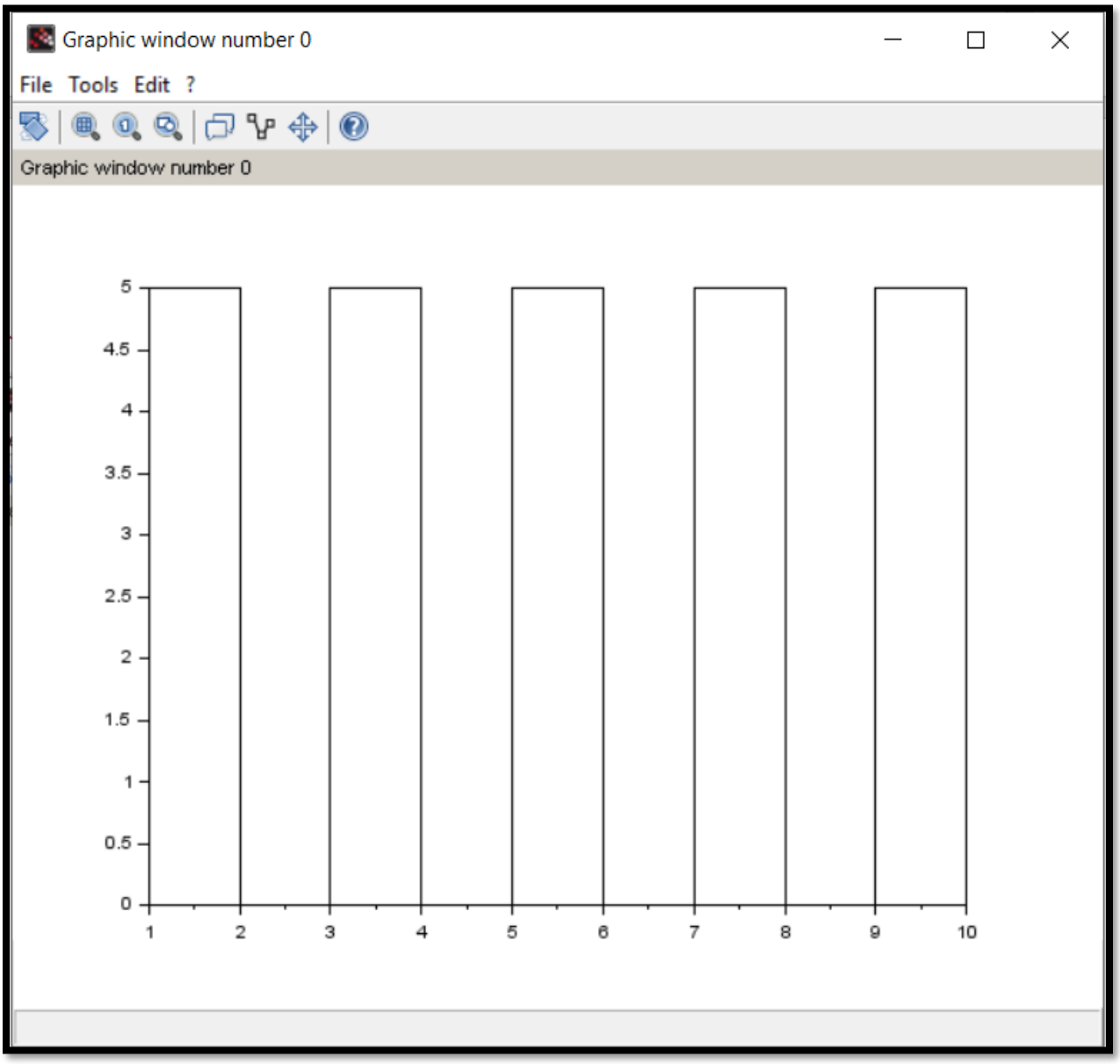
Output:


Graphic window number 0
File Tools Edit ?
Graphic window number 0
5
4
3
2
1
0
-1
-2
-3
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5
3
3.5
4

//generation of square wave

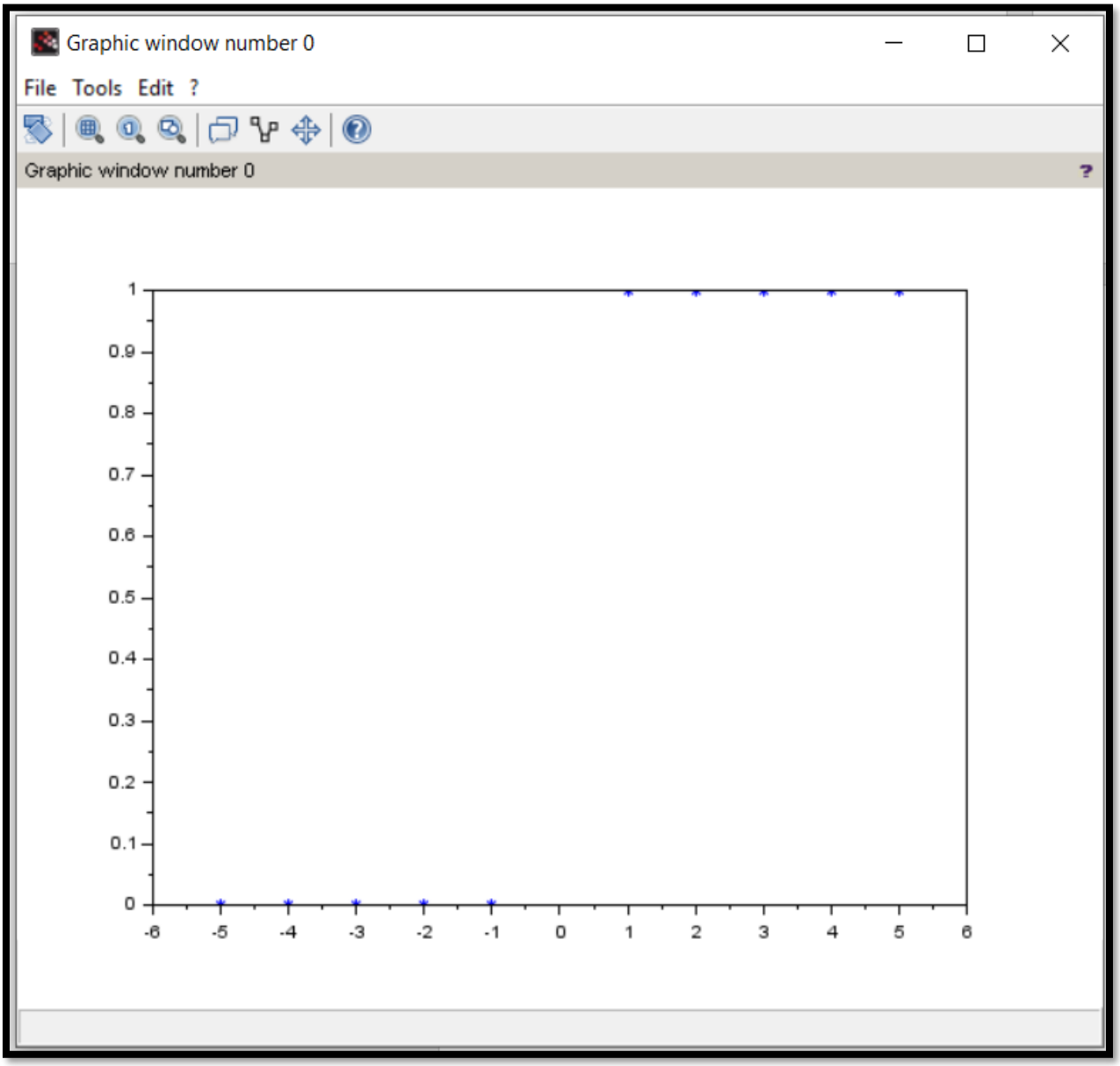
```
clc;

printf('\n\n Name - Syeda Reeha Quasar \n Enrolment No. -
14114802719 \n Group - C7 \n\n')

x = [1 2 3 4 5 6 7 8 9 10];
y = [5 0 5 0 5 0 5 0 5 0];

plot2d2(x, y)
```

Output:


Graphic window number 0
File Tools Edit ?
Graphic window number 0
5
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
1
2
3
4
5
6
7
8
9
10

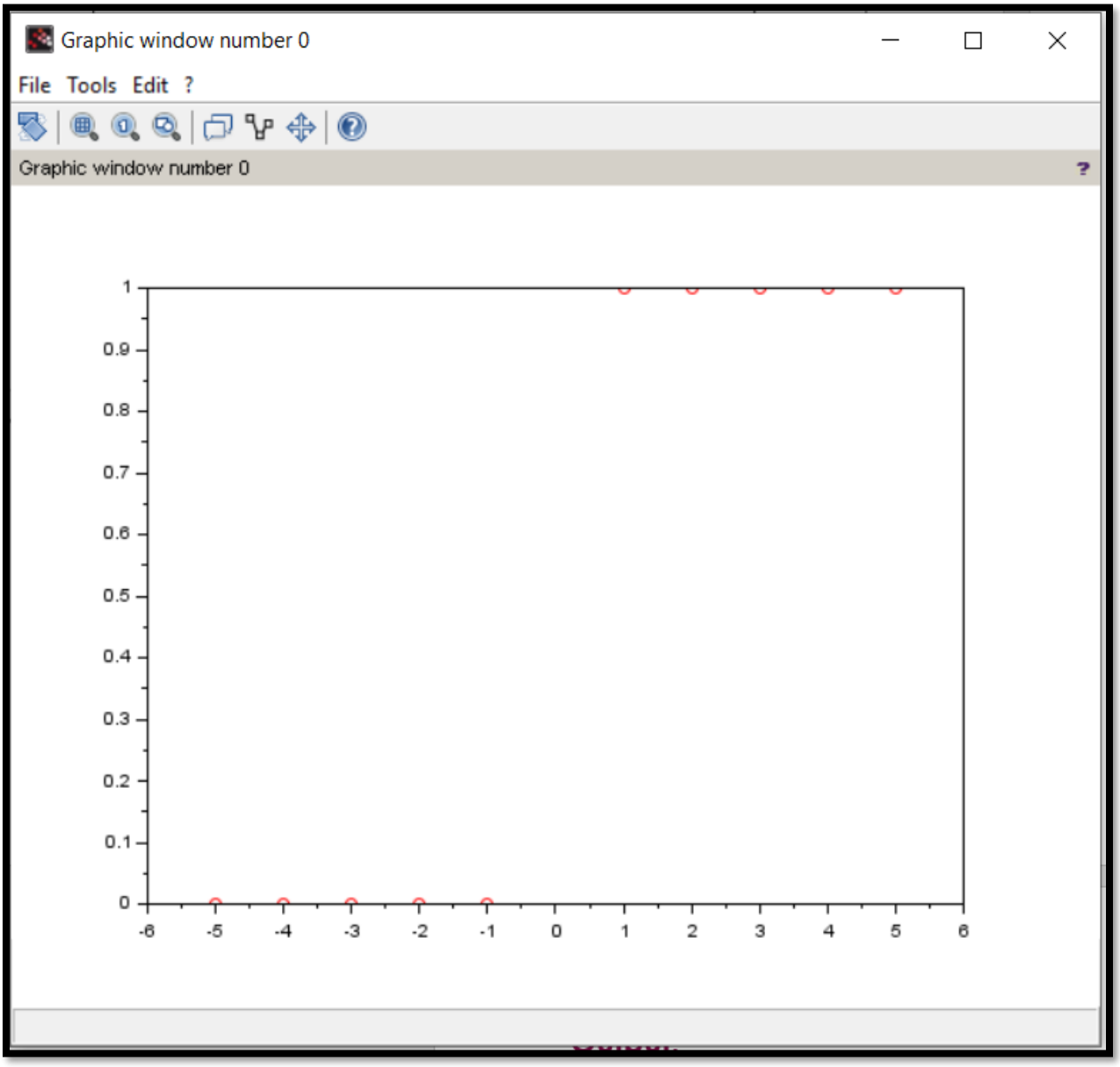
```
// unit step function

clc;

x = [-1 -2 -3 -4 -5 1 2 3 4 5];
y = [0 0 0 0 0 1 1 1 1 1];

//plot(x, y, 'ro');

plot(x, y, 'd*');
```

Output:


Graphic window number 0
File Tools Edit ?
Graphic window number 0
?
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-6
-5
-4
-3
-2
-1
0
1
2
3
4
5
6

```
// unit step function

clc;

x = [-1 -2 -3 -4 -5 1 2 3 4 5];
y = [0 0 0 0 0 1 1 1 1 1];

plot(x, y, 'ro');

//plot(x, y, 'd*');
```

Output:


Graphic window number 0
File Tools Edit ?
Graphic window number 0
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-6
-5
-4
-3
-2
-1
0
1
2
3
4
5
6

EXPERIMENT - 9

APPLIED MATHEMATICS LAB

Aim

To find mean, standard deviation and moments about mean of a given frequency data.

Syeda Reeha Quasar
14114802719
4C7

EXPERIMENT – 9

Aim:

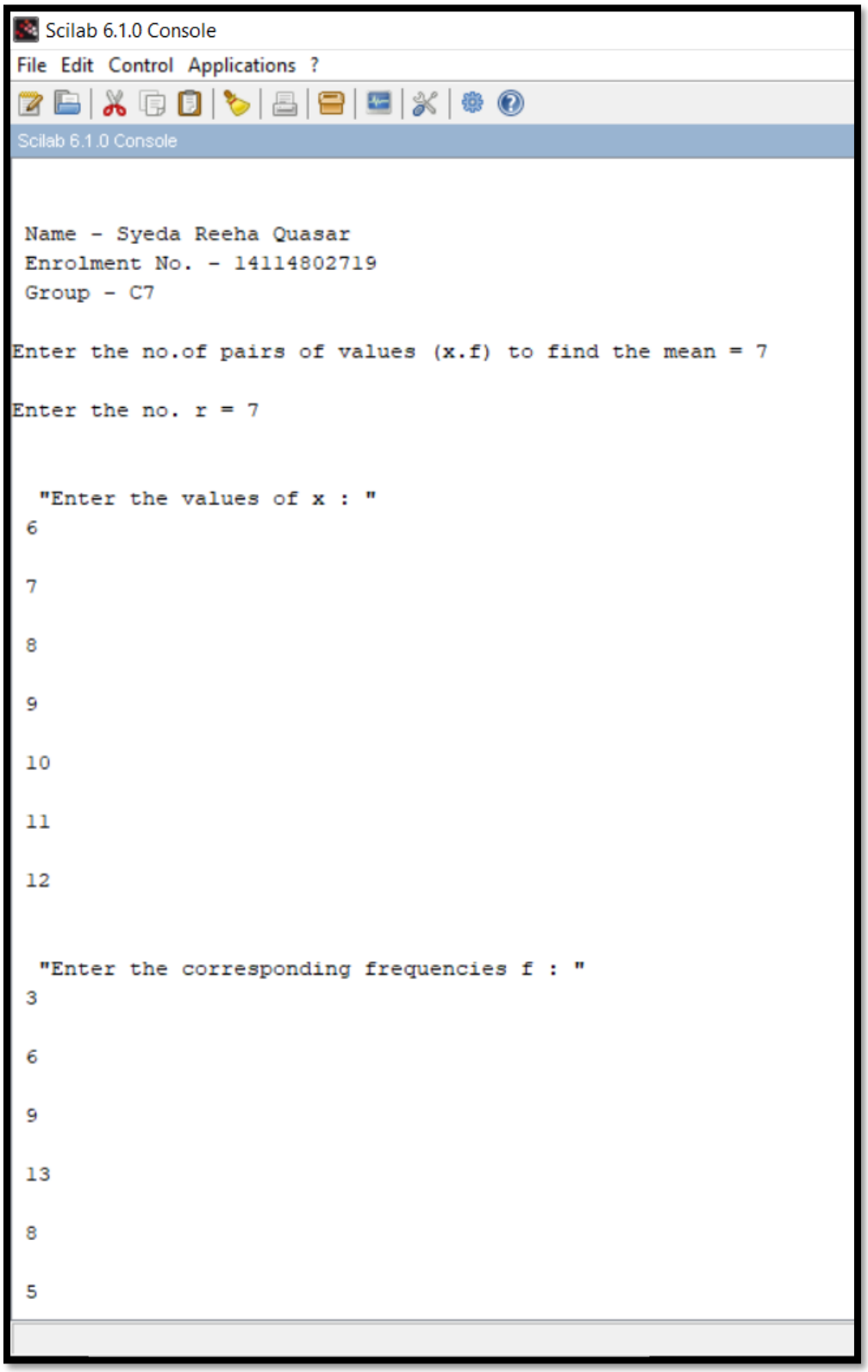
To find mean, standard deviation and moments about mean of a given frequency data.

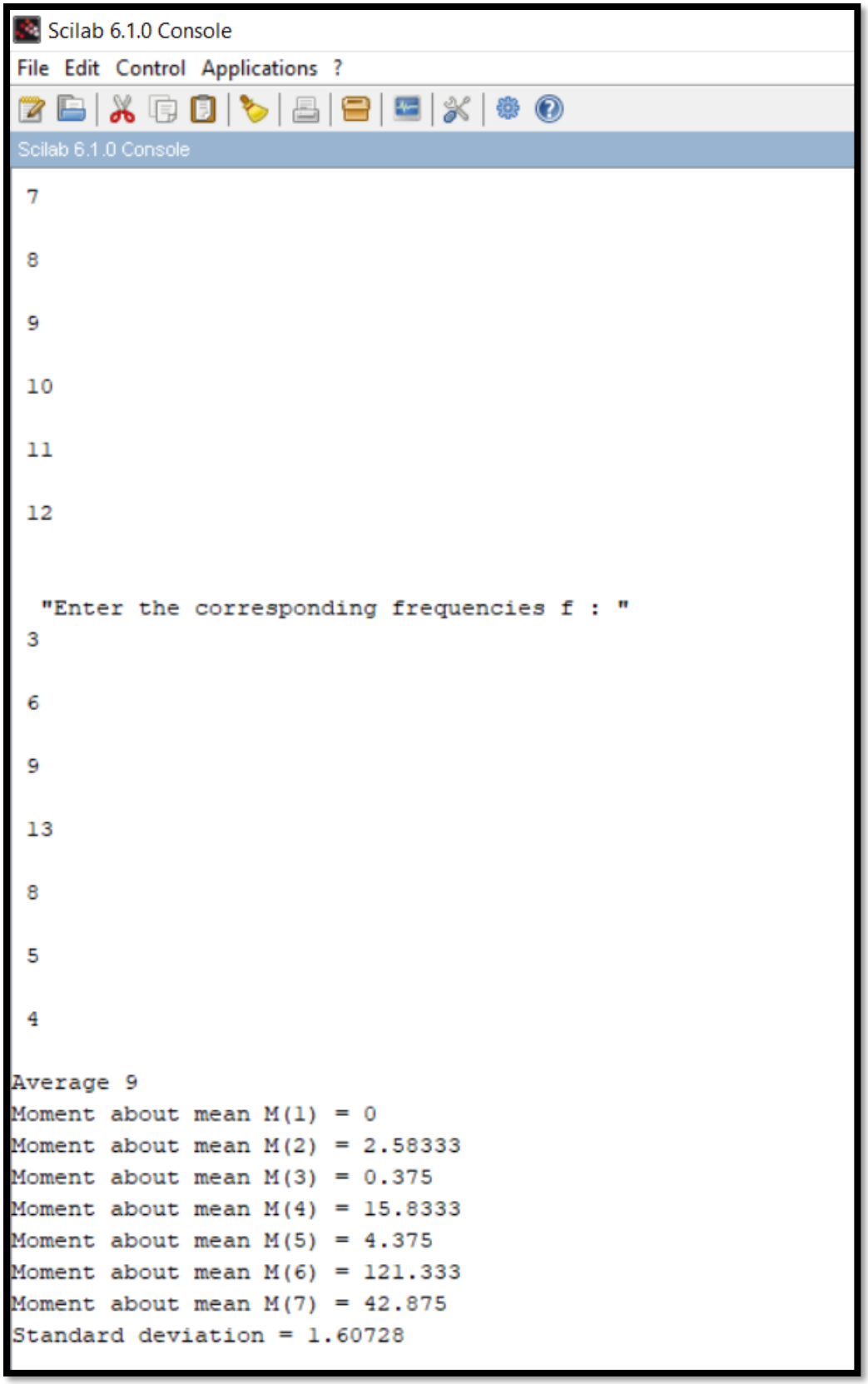
Source Code:

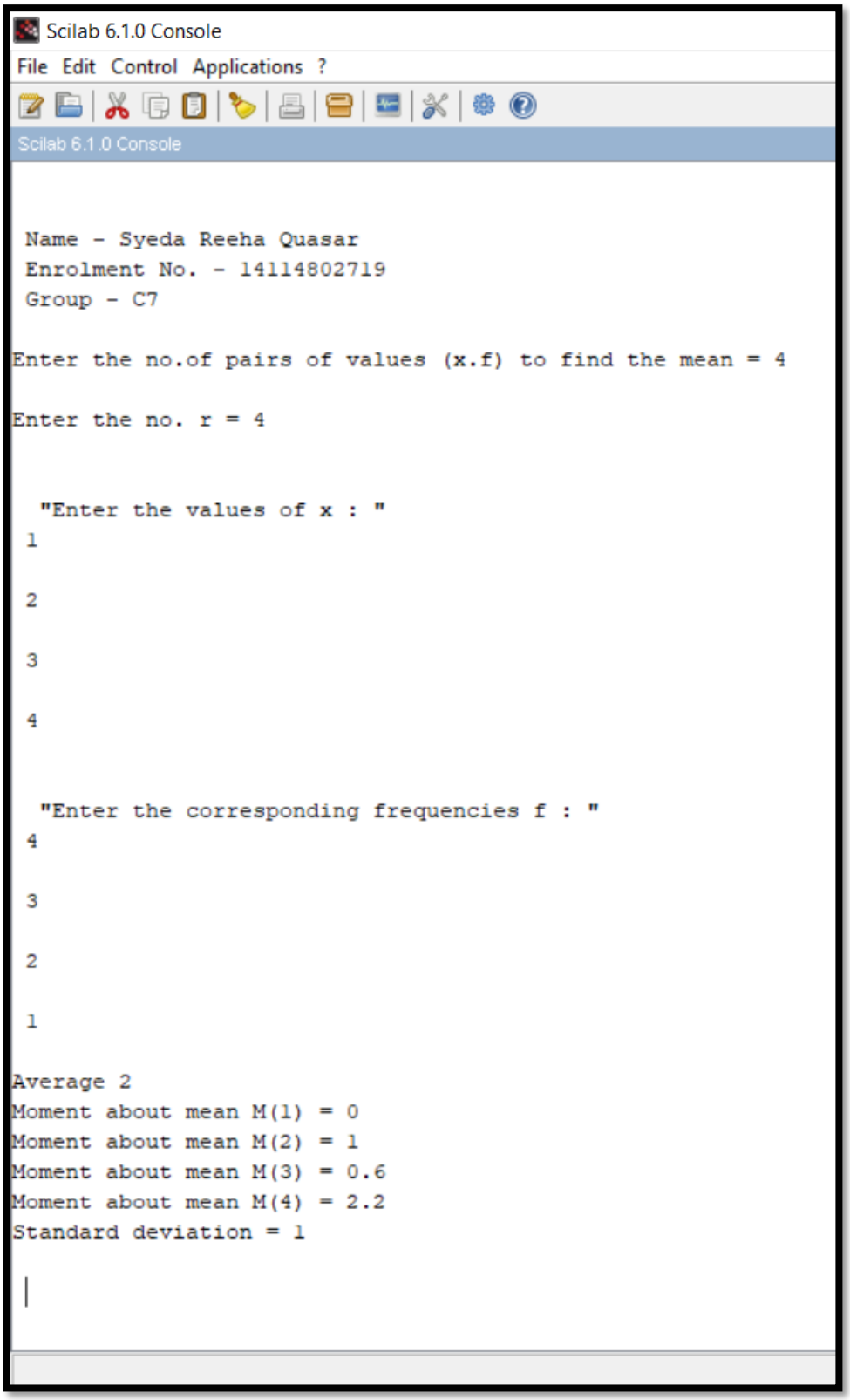
```
//Program to find mean,S.D. and first r moments about mean of given n pairs (x)
clc;clear;close;

clc

printf('\n\n Name - Syeda Reeha Quasar \n Enrolment No. - 14114802719 \n
Group - C7 \n\n')

n = input('Enter the no.of pairs of values (x.f) to find the mean = ')
m = input('Enter the no. r = ')

disp('Enter the values of x : ')

for i = 1:n
   x(i) = input(' ')
end

disp('Enter the corresponding frequencies f : ')

for i=1:n
   f(i) = input(' ')
end

s = 0
s1 = 0

for i=1:n
   s = s + f(i) //Calculate the sum of all frequencies
```

```
   s1 = s1 + f(i)*x(i) // Calculate the sum of all f(i)x(i)
end

A = s1/s  //Calculate the average

printf('Average %g\n', A);

for j = 1:m
   s2 = 0
   for i=1:n
      y(i) = f(i)*(x(i)-A)^j
      s2 = s2 + y(i)
   end;
   M(j) = (s2/s) //Calculate the moments
   printf('Moment about mean M(%i) = %g\n', j, M(j))
end

sd = sqrt(M(2)) //Calculate the standard deviation

printf('Standard deviation = %g\n', sd);
```

Output:


Scilab 6.1.0 Console
File Edit Control Applications ?
Scilab 6.1.0 Console
Name - Syeda Reeha Quasar
Enrolment No. - 14114802719
Group - C7
Enter the no.of pairs of values (x.f) to find the mean = 7
Enter the no. r = 7
"Enter the values of x : "
6
7
8
9
10
11
12
"Enter the corresponding frequencies f : "
3
6
9
13
8
5


Scilab 6.1.0 Console
File Edit Control Applications ?
Scilab 6.1.0 Console
7
8
9
10
11
12
"Enter the corresponding frequencies f : "
3
6
9
13
8
5
4
Average 9
Moment about mean M(1) = 0
Moment about mean M(2) = 2.58333
Moment about mean M(3) = 0.375
Moment about mean M(4) = 15.8333
Moment about mean M(5) = 4.375
Moment about mean M(6) = 121.333
Moment about mean M(7) = 42.875
Standard deviation = 1.60728


Scilab 6.1.0 Console
File Edit Control Applications ?
Scilab 6.1.0 Console
Name - Syeda Reeha Quasar
Enrolment No. - 14114802719
Group - C7
Enter the no.of pairs of values (x.f) to find the mean = 4
Enter the no. r = 4
"Enter the values of x : "
1
2
3
4
"Enter the corresponding frequencies f : "
4
3
2
1
Average 2
Moment about mean M(1) = 0
Moment about mean M(2) = 1
Moment about mean M(3) = 0.6
Moment about mean M(4) = 2.2
Standard deviation = 1