**MATRICES IN MATLAB**

**Lab Report # 02**



**CSE301 - L Signals & Systems Lab**

Submitted by: AWAIS SADDIQUI

Registration No: 21PWCSE1993

Class Section: “A”

Submitted to:

**Dr. Durr-e-Nayab**

**Department of Computer System Engineering**

**UET Peshawar**

**301L: Signals & Systems Lab**

**LAB ASSESSMENT RUBRICS**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Marking Criteria** | **Exceeds expectation (5-4)** | **Meets expectation** **(3-2)** | **Does not meet expectation (1)** | **Score** |
| **1. Realization of Experiment** | Program compiles (noerrors and no warnings).  Program always works correctly and meets the specification(s).  Completed between 71-100% of the requirements. | Program compiles (no errors and some warnings).  Some details of the program specification are violated, program functions incorrectly for some inputs.  Completed between 41-70% of the requirements. | Program fails to or compile with lots of warnings.  Program only functions correctly in very limited cases or not at all.  Completed less than  40% of the requirements. | 30% |
| **2. Ability to apply required code utility or data structure** | Able to apply required data type or data structure and produce correct results. Familiarize and selects proper functions for simulation of given problem using  software tools like  MATLAB. | Able to apply required data type or data structure but does not produce correct results. Need guidance to select proper functions for simulation of given problem using  software tools like  MATLAB. | Unable to identify required data type or data structure.  Incapable of selecting proper functions for simulation of given problem using software tools like MATLAB. | 20% |
| **3. Documentation** | Clearly and effectively documented including descriptions of all variables/functions. Specific purpose is noted for each function, control structure, input requirements and output results. | Basic documentation including descriptions of all variables/functions. Specific purpose is noted for each function and control structure. | No documentation included. | 10% |
| **4. Ability to run/debug** | Executes Matlab codes without errors, excellent user | Executes Matlab codes without errors.  User prompts are | Does not execute Matlab codes due to errors. | 20% |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | prompts, good use of symbols, spacing in output.  Thorough and organized testing has been completed and output from test cases is included. | understandable, minimum use of symbols or spacing in output.  Some testing has been completed. | User prompts are misleading or nonexistent.  No testing has been completed. |  |
| **5. Results compilation** | Show processed results effectively by conducting simple computations and plotting using collected data | Show processed results effectively by conducting simple computations and plotting using collected data with minor error | Unable to show processed results effectively by conducting simple computations and plotting using collected data with minor error | 10% |
| **6. Efficiency** | Excellent use of CPU and Memory. | Good but not smart use of CPU and Memory. | Inefficient use of CPU and Memory. | 10% |
| **7. Lab**  **Performance (Team work and Lab**  **etiquettes)** | Actively engages and cooperates with other group members in an effective manner. Respectfully and carefully observes safety rules and procedures | Cooperates with other group members in a reasonable manner. Observes safety rules and procedures with minor deviation. | Distracts or discourages other group members from conducting the experiment. Disregards  safety rules and procedures. | 10% |

**Instructor:**

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Signals & Systems Laboratory**

**MATRICES:**

MATLAB works with essentially only one kind of object, a rectangular numerical matrix possibly, with complex entries. Every MATLAB variable refers to a matrix [a number is a 1 by 1 matrix]. In some situations, 1‐by‐1 matrices are interpreted as scalars, and matrices with only one row or one column are interpreted as vectors

**Matrices in MATLAB:**

MATLAB is designed to make definition of matrices and matrix manipulation as simple as possible. Matrices can be introduced into MATLAB in several different ways.

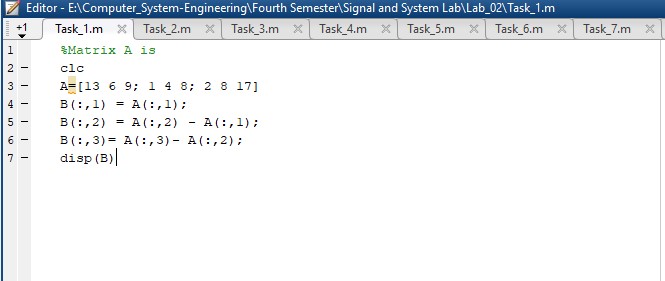
**Objectives of the Lab:**

* Built in Matrix Functions
* Indexing Matrices
* Sub Matrices
* Matrix element level operations
* Round Floating-Point numbers to Integers

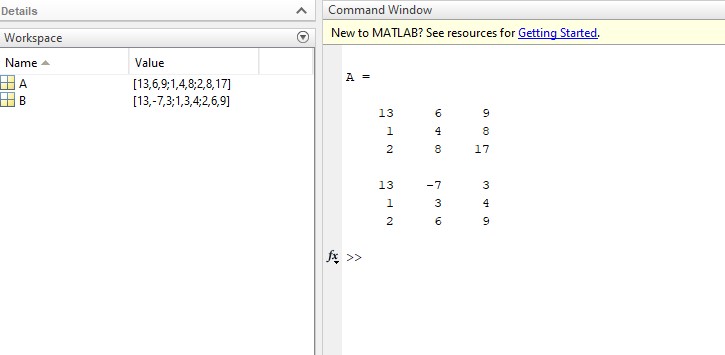
**Task 01:**

Write a program to generate a new matrix B from the matrix A given below such that each column in the new matrix except the first one is the result of subtraction of that column from the previous one i.e. 2nd new column is the result of subtraction of 2nd column and 1st column and so on. Copy the first column as it is in the new matrix.

**Code:**



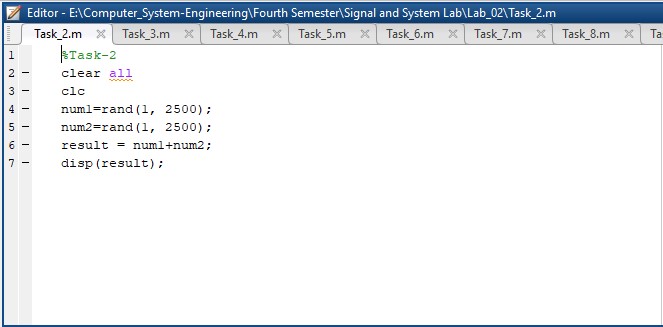
**Output:**



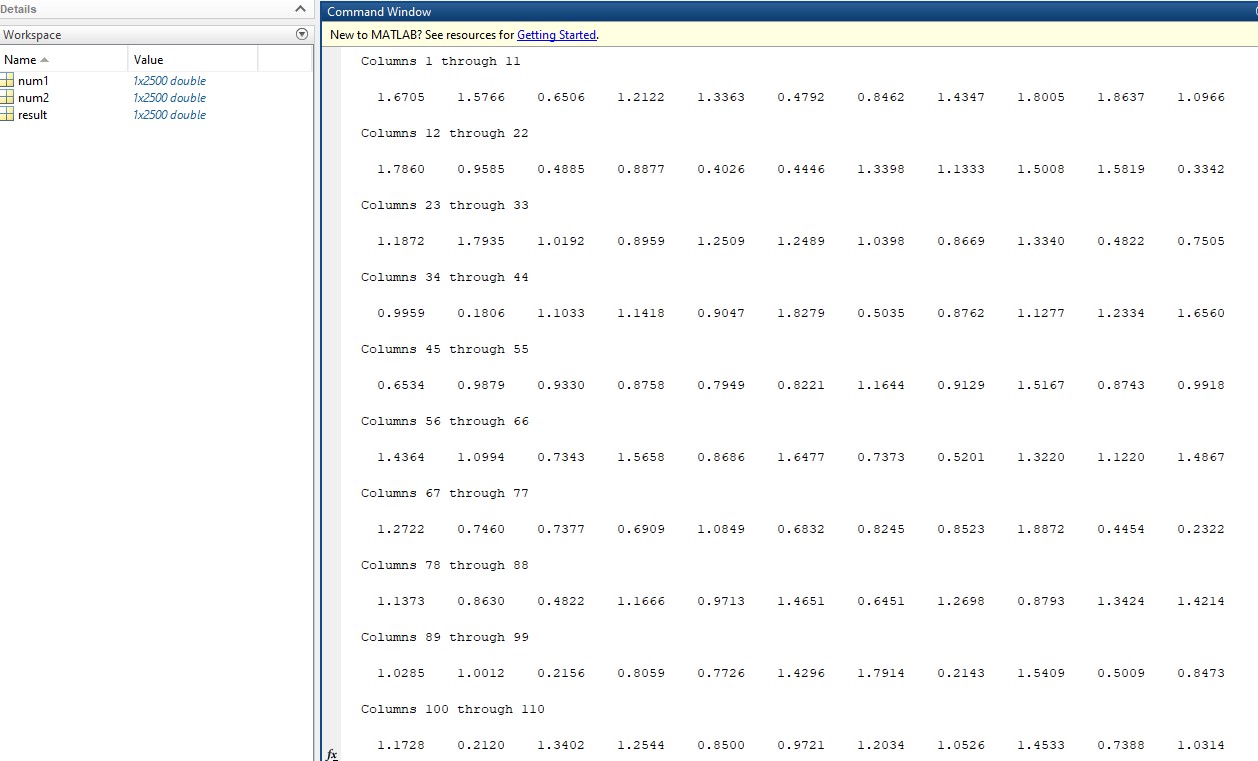
**Task 02:**

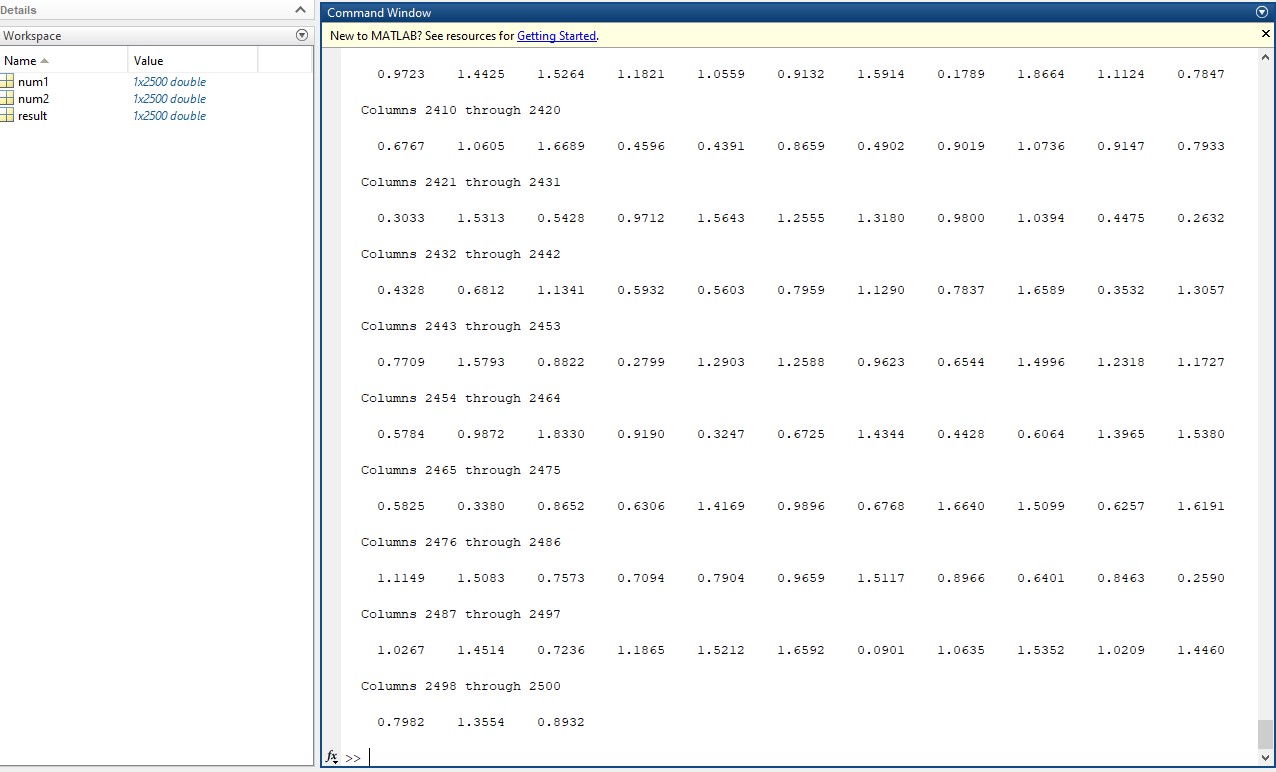
Generate two 2500 sampled random discrete time signals (1 dimensional) using rand() function i.e. rand(1, 2500). Write a program to add the two such random signals together using simple vector addition.

**Code:**



**Output:**

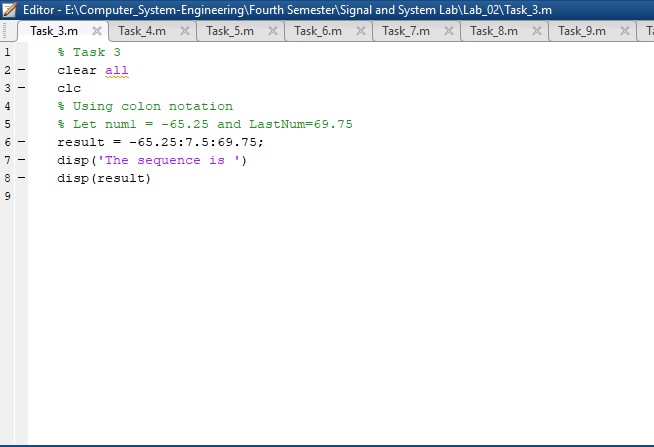




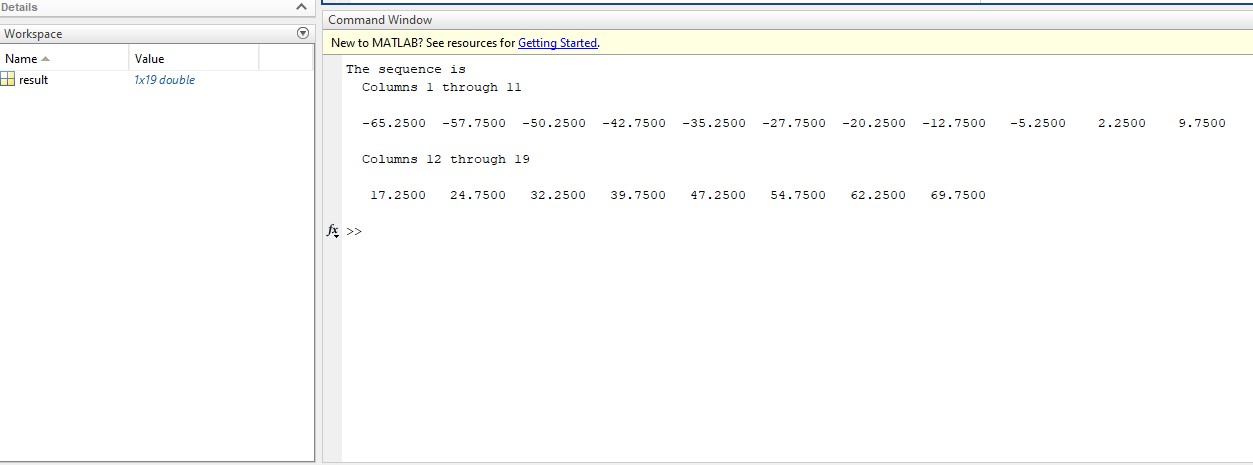
**Task 03:**

Using colon notation, generate the following sequence: -65.25, -57.75, -50.25. . . . . . . . . . ., 54.75,62.25, 69.75

**Code:**



**Output:**



**Task 04:**

**Given the matrices:**

**A=[-12,34,61,-9;65,78,90,12; 14,78,45,12; 60,25,3,8]**

**B= [34,67,08,09; 12, -91,12,9; 89, -8,0,02; 16,09,23,67]**

**Find the following:**

1) Array addition; store the result in matrix C

2) Array subtraction; store the result in matrix D

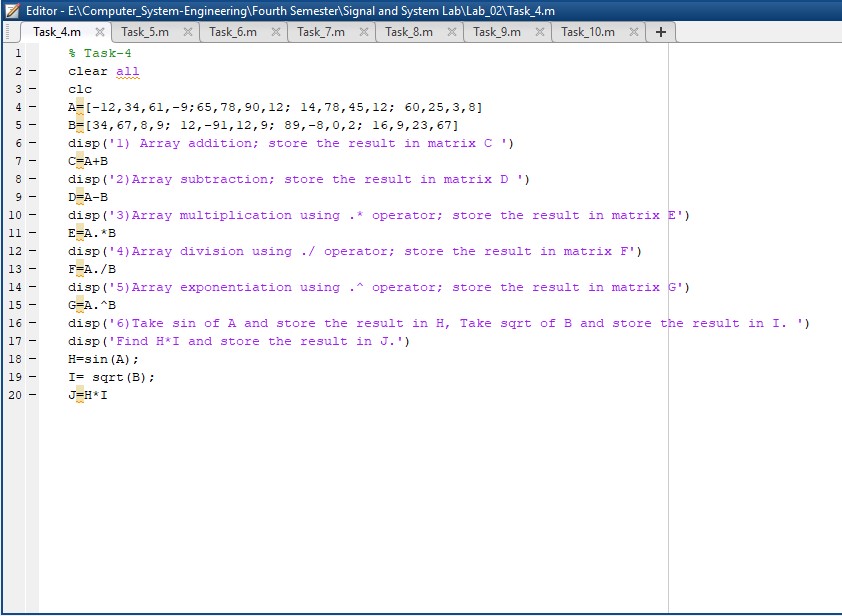
3) Array multiplication using .\* operator; store the result in matrix E

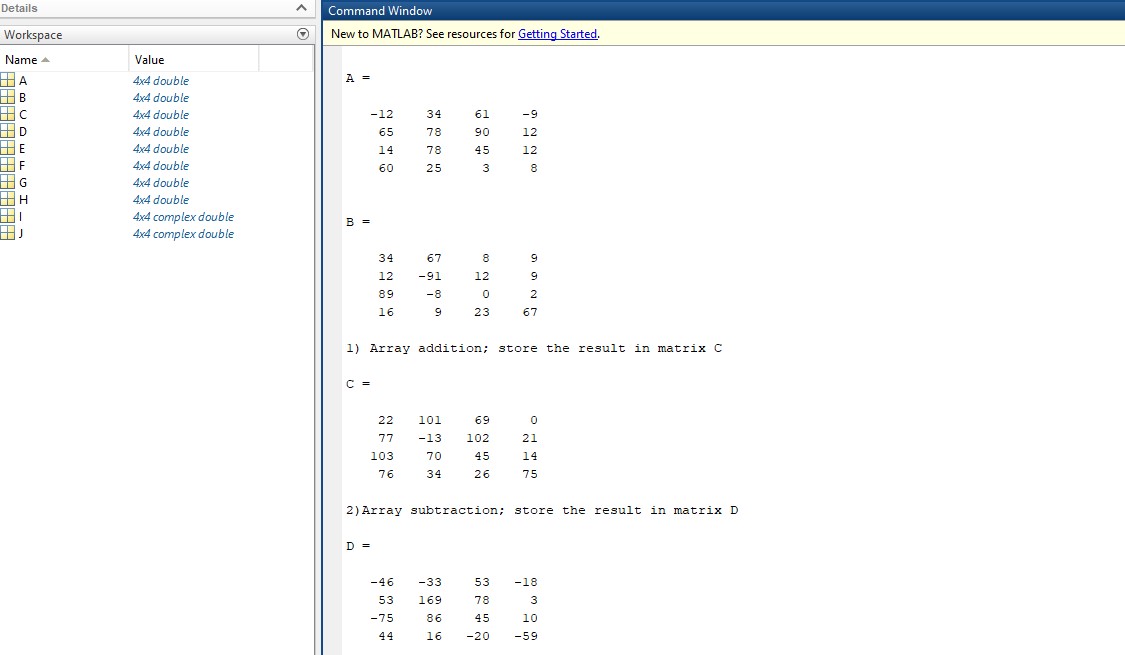
4) Array division using ./ operator; store the result in matrix F

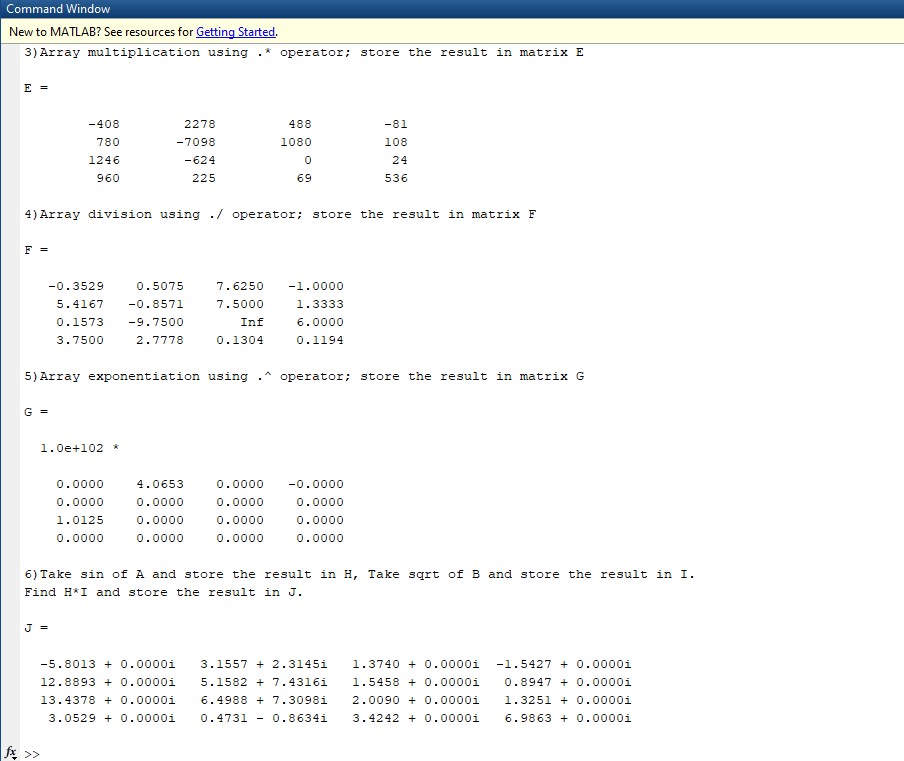
5) Array exponentiation using .^ operator; store the result in matrix G

6) Take sin of A and store the result in H, Take sqrt of B and store the result in I. Find H\*I and store the result in J.

**Code:**



**Output:**



**Task 05:**

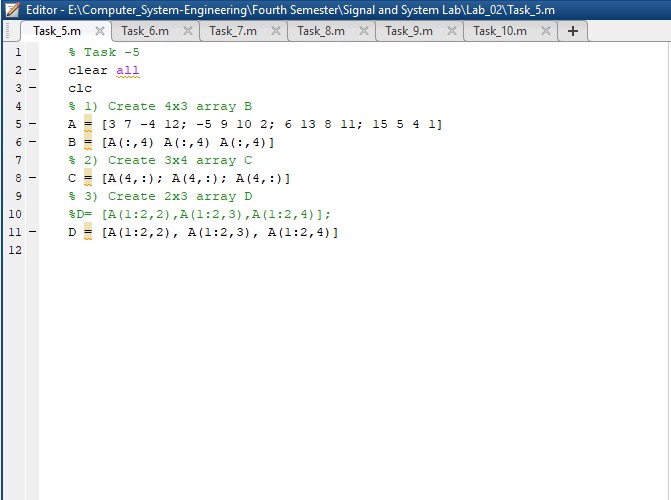
**A=[3 7 -4 12; -5 9 10 2; 6 13 8 11; 15 5 4 1]**

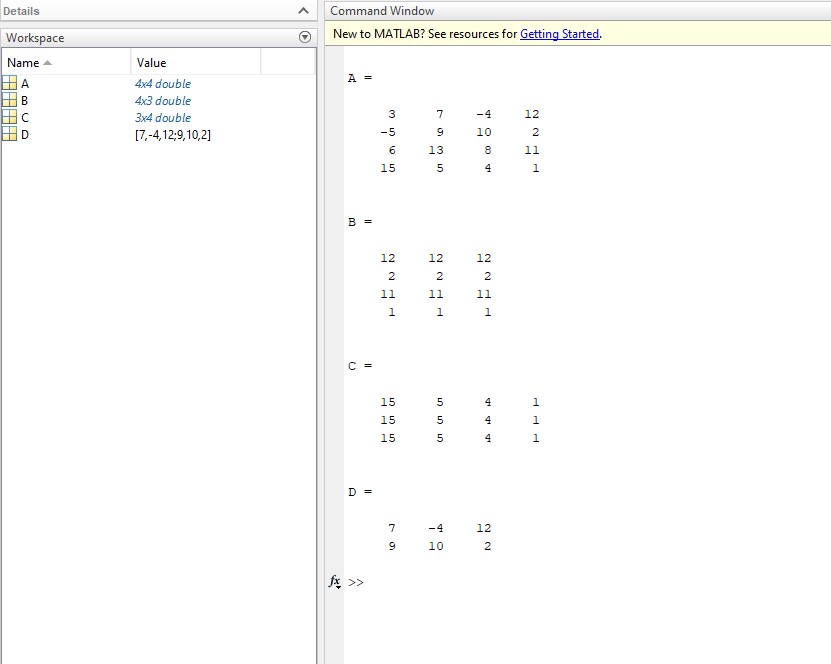
**Find the following:**

1) Create 4x3 array B consisting of all elements in the second through fourth columns of A

2) Create 3x4 array C consisting of all elements in the second through fourth rows of A

3) Create 2x3 array D consisting of all elements in the first two rows and the last three .

**Code:****Output:**



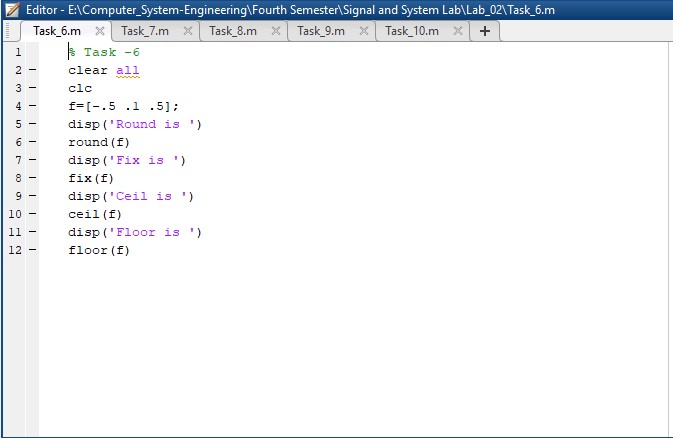
**Task 06:**

MATLAB has functions to round floating point numbers to integers. These are round, fix, ceil, and floor. Test how these functions work. Determine the output of the following:

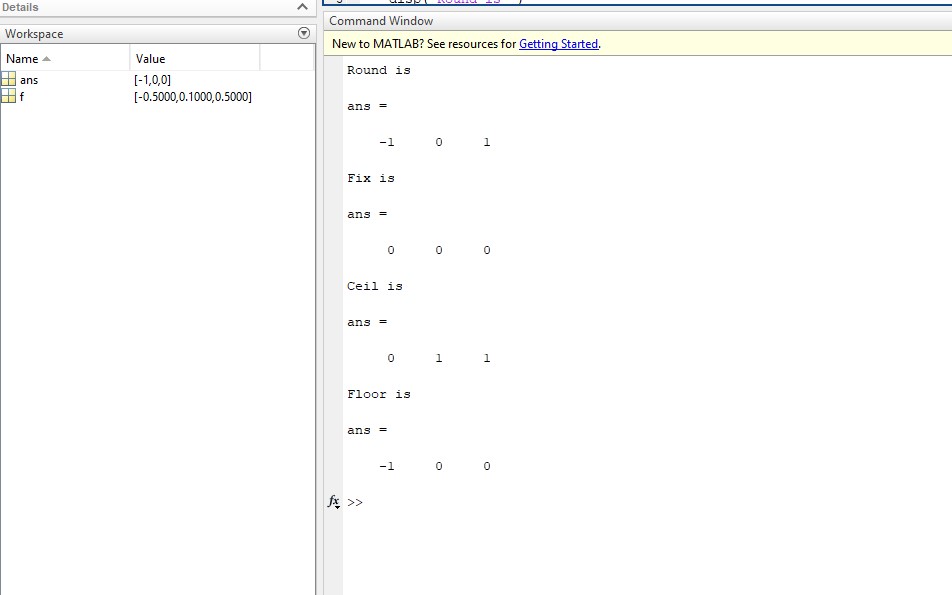
>> f = [‐.5 .1 .5];

1. round(f) 2) fix(f) 3) ceil(f) 4) floor(f)

**Code:**



**Output:**

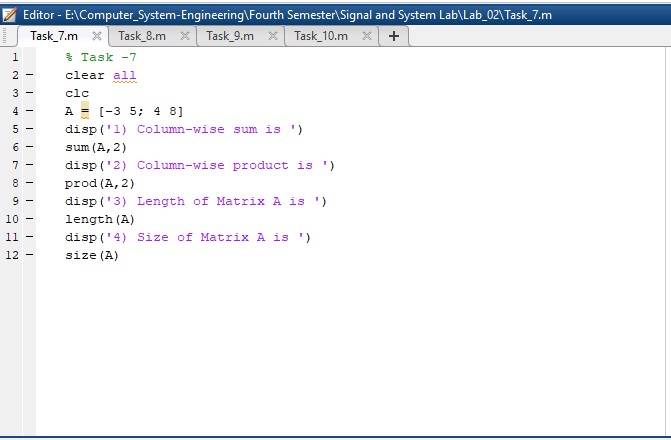


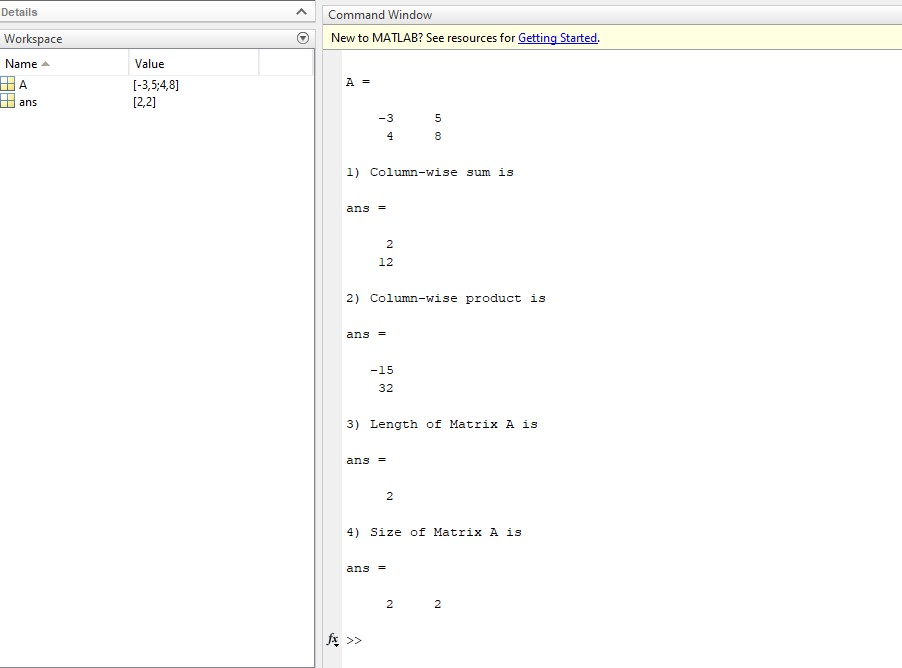
**Task 07:**

A = [-3 5; 4 8]

1) Column‐wise sum of all elements of A . 2) Column‐wise product of all elements of A .

3) Length of matrix A . 4) Size of matrix A

**Code:****Output:**



**Task 08:**

The end command is used to access the last row or column of a matrix. Use the end command to delete and update the last row and column.

Matrix **A = [3 23 34 12 34 5 56 23; 12 34 34 32 23 23 45 1; 67 23 2 4 4 5 6 456; 4 5 1 1 2 34 45 56; 67 67 45 67 78 7 8 5; 6 35 5 3 5 56 7 8]**

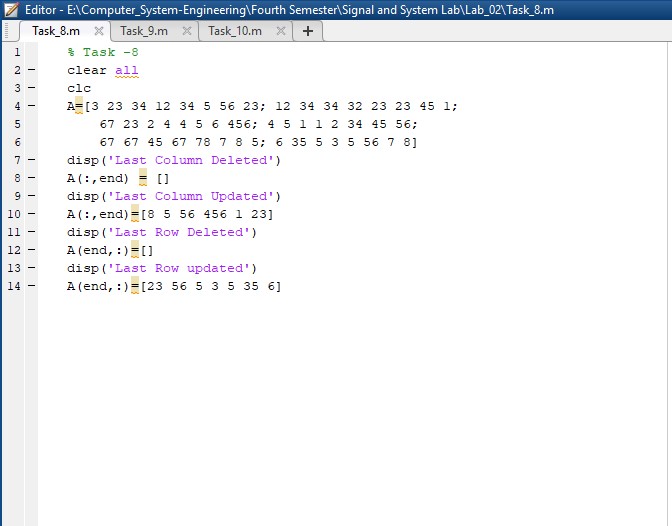
Hint:

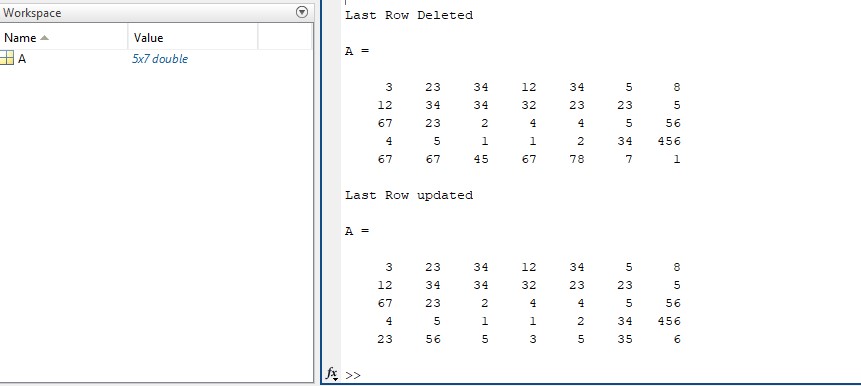
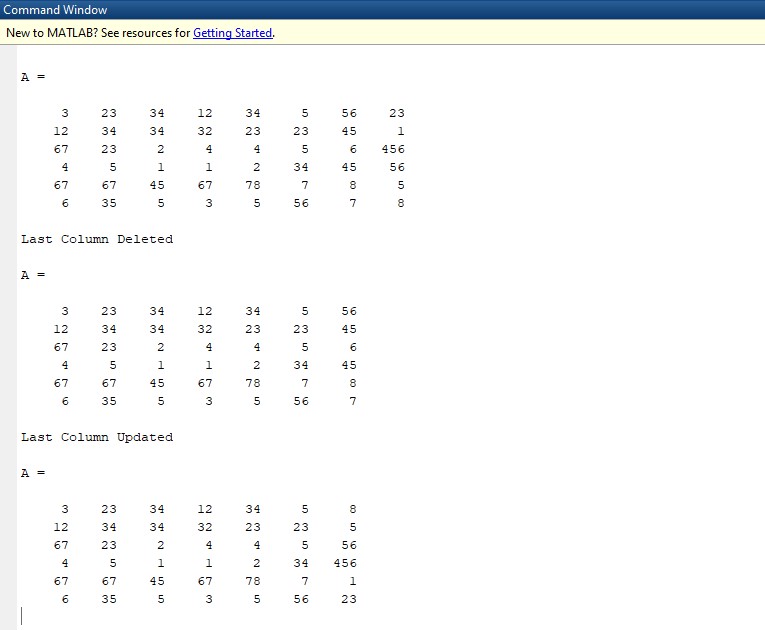
For deleting a column use A(3 , :)=[ ];

For deleting last column use A(: , end)=[ ];

and vice versa.

**Code:**



**Output:**

**Task 09:**

Try the following commands in MATLAB and comment on them:

A(3,end)

(ii) A(:)

(iii) A(: , end)

(iv) Y = linspace(20,100)

(v) Y = linspace(20,100,50)

**Output:**

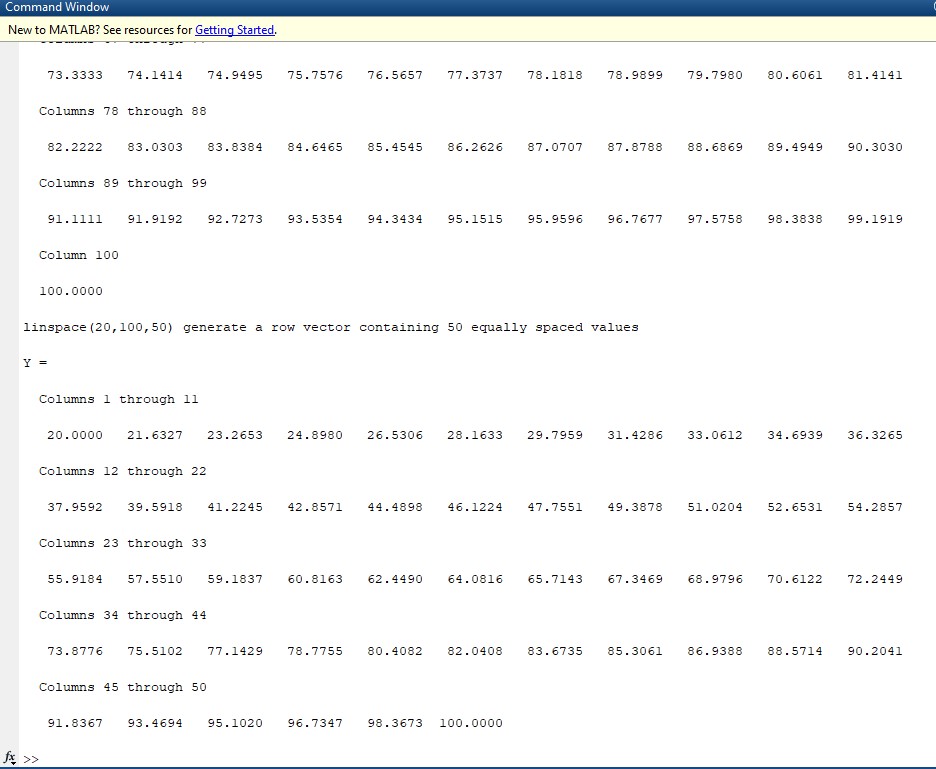
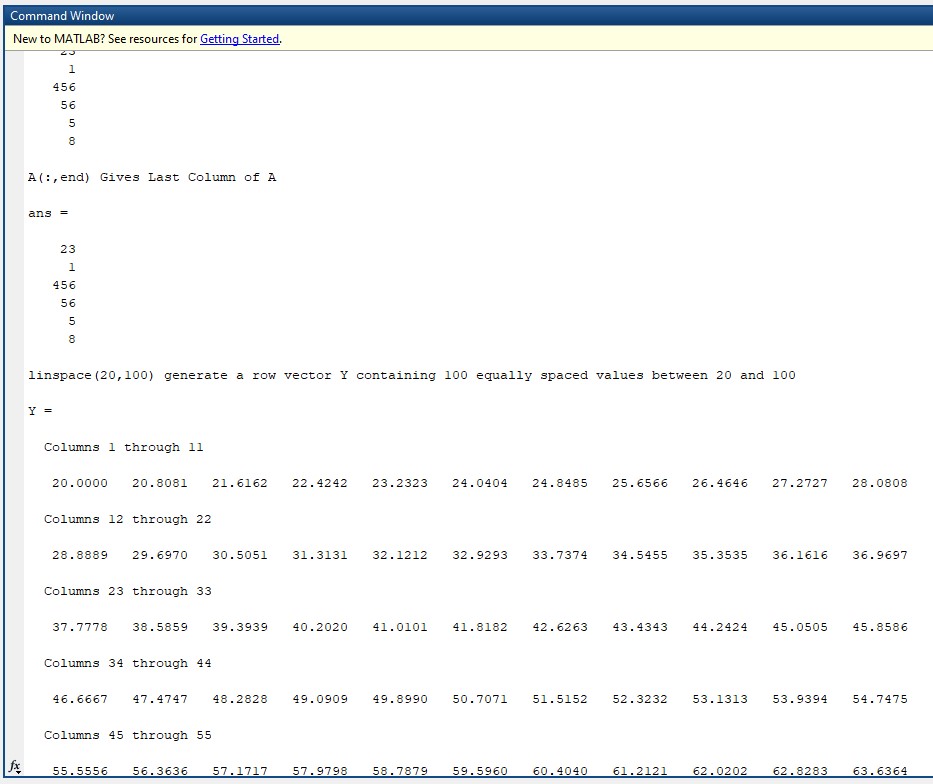
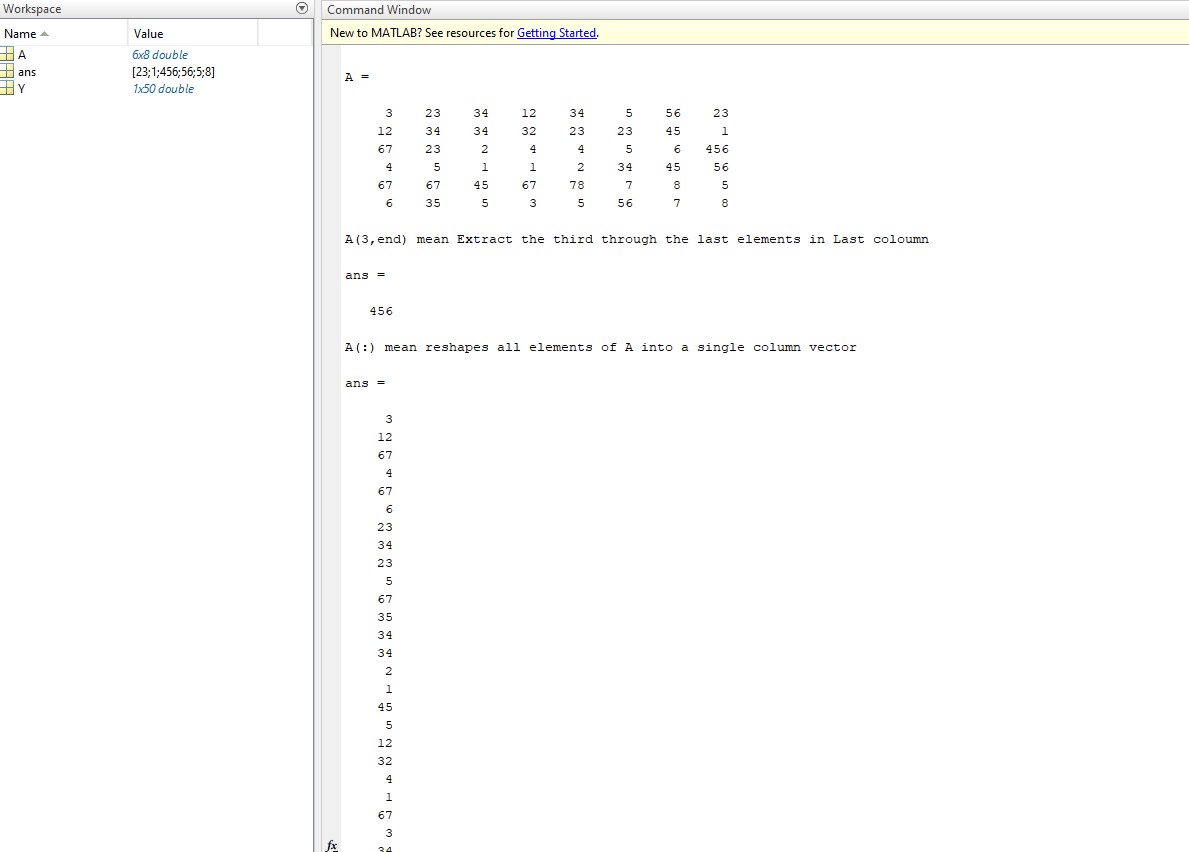
**(i) A(3,end)** =This command accesses the element in the third row and last column of the matrix A. The keyword 'end' is used to refer to the last index in a matrix.

**(ii) A(:)** =This command reshapes the matrix A into a single column vector, where each element of A is listed in column-major order.

**(iii) A(: , end)** = This command accesses all the elements in the last column of the matrix A.

**(iv) Y = linspace(20,100)** =This command generates a row vector Y with 100 equally spaced points between 20 and 100, inclusive. The default number of points is 100, so the resulting vector has 100 elements.

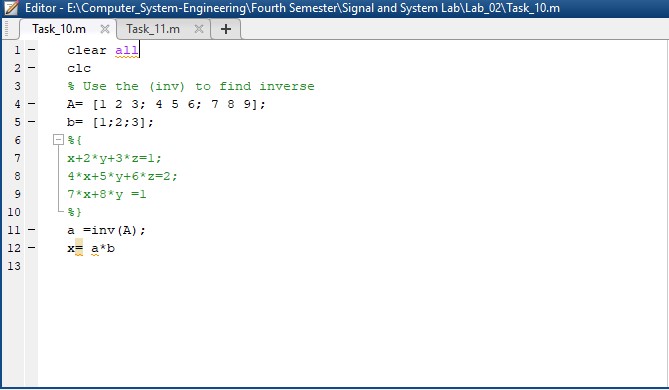
**(v) Y = linspace(20,100,50)** =This command generates a row vector Y with 50 equally spaced points between 20 and 100, inclusive. The resulting vector has 50 elements. The third argument specifies the number of points to generate.



**Task 10:**

Use the inverse (inv(A)) function to find the inverse of A for finding the unknowns for the Linear equation.

**Code:**



**Output:**

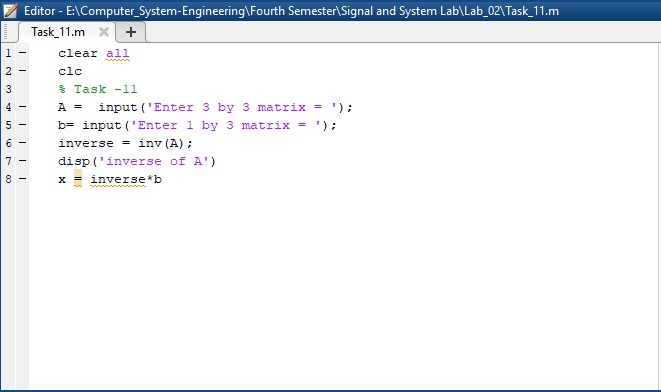


**Task 11:**

Solve Task 10 by taking the equations from user.

Hint: Take the matrix A and b from user.

**Code:**

****

**Output:**

