

It is required to design and implement an extension to the calculator created in the first milestone to support more functionalities to be done on matrices.

Operations that will be supported in this second milestone are: 1- Matrix Transpose 2- Matrix Inverse 3- Matrix Determinant

The matrix should be entered by the user in the following format: [1 2 5, -1 8 14.2, 7.01 -18 99.3] which is the format equivalent to a 3x3 matrix as shown:

$$\begin{bmatrix} 1 & 2 & 5 \\ -1 & 8 & 14.2 \\ 7.01 & -18 & 99.3 \end{bmatrix}$$

### Input Format

The matrix should be entered by the user in the following format: [1 2 5, -1 8 14.2, 7.01 -18 99.3] which is the format equivalent to a 3x3 matrix as shown:

$$\begin{bmatrix} 1 & 2 & 5 \\ -1 & 8 & 14.2 \\ 7.01 & -18 & 99.3 \end{bmatrix}$$

User can choose between the different operations as follows:

- In case of Transpose: user types one matrix only then enters capital 'T' letter as shown:

```
[1 2 4, 9 3 7]
T
[1 9, 2 3, 4 7]
```

- In case of Inverse: user types one matrix only then enters capital 'I' letter as in the case of transpose.
- In case of Determinant: user types one matrix only then enters capital 'D' letter, and the output should be shown as a number.

## Constraints

User will input string as shown previously: Example for a 3x4 matrix, it should be written as follows [-0.1 2 5, -1 8 14.2, 7.01 -18 99.3, 15.0 -99.24 18] User will input the matrix of size nxm; where 0

## Important Note

- If the user enters a matrix in an incorrect format (other than the stated input format) your program must detect that and output the word "ERROR!" (MUST BE WRITTEN AS SHOWN, don't forget the exclamation mark, and note that all letters are capitals) For Example:

```
[1 2 4, 4 7, 8]
+
[-5 3 5, 7 6 1]
ERROR!
```

- Your program should validate the mathematical conditions to do the different operations, for example: In the determinant, you should check if the matrix given can have a determinant or not. If not, you should output "ERROR!" to the user. Also, in the inverse case, you should detect the validity of the inversion. If no validity, your program should output an "ERROR!"

## Output Format

The output should be a matrix of the result printed on a new line as shown previously.

In case of the determinant, the output should be the value of the determinant of the matrix only.

Also, the error message should be in the following format: "ERROR!" on a new line as shown in the examples above.

**NOTE: In the case of decimal point outputm you should set the precision of the output to 2 decimal point places**

## Sample Input 0

```
[5 9 -8 -9 -1, -2 1 7 8 2]
T
```

## Sample Output 0

```
[5 -2, 9 1, -8 7, -9 8, -1 2]
```

### Sample Input 1

```
[1 2 3, 6 7 0, 0 1 2]
```

```
I
```

### Sample Output 1

```
[1.75 -0.12 -2.62, -1.5 0.25 2.25, 0.75 -0.12 -0.62]
```