Project ITC

**Project name**: MAT Tricks (Matrix Calculator).

## Aims:

1. Perform arithmetic operations on matrices (Except division).
2. Find echelon and reduce echelon form.
3. Solves systems of linear equations.
4. Calculate SPAN of matrix.
5. Apply and find transformations.

## Challenges:

1. Building complex logics.
2. Deal with every sort off input.
3. Making program simple to use and easy to operate.

## Modules:

1. Function of converting reduced echelon form.
2. Function for multiplication of matrices.
3. Function for Transformation operations.
4. Function for Addition and Subtraction of matrices.
5. Function for Analysis of echelon form.

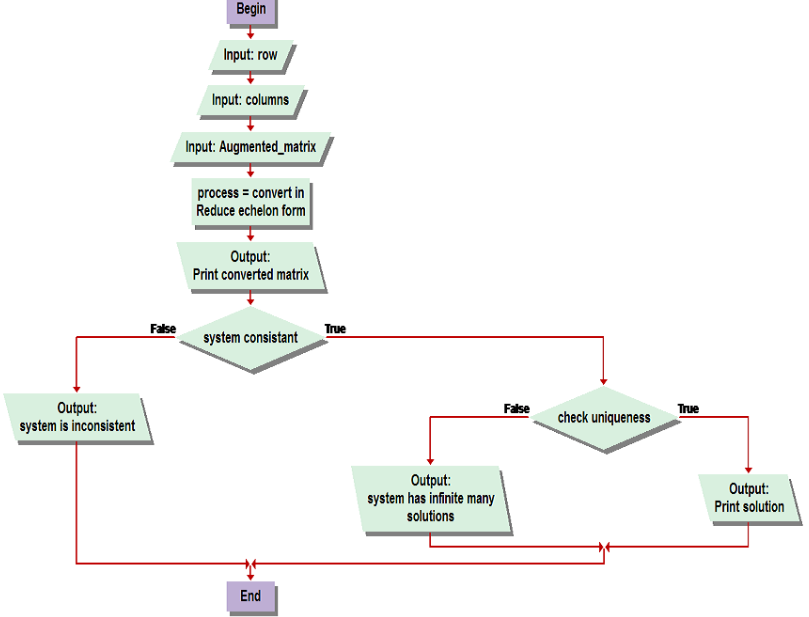
## Module 1

## (R.Echelon form)

### Algorithm:

1. Input number of row.
2. Input number of Column.
3. Input matrix with given row and columns.
4. Convert Matrix into Reduced Echelon Form.
5. Print Reduced Echelon Form.
6. If System is consistent, then go to step 7, else print “system is inconsistent” and go to step 8.
7. If the solution is unique then, print solution, else System has infinite many solutions.
8. end

### Flowchart:



## Module 2

## (Multiplication)

### Algorithm:

1. Start
2. Input Row1, Column1
3. Print Matrix1
4. Input Row2, Column2
5. Print Matrix2

[ If Column1==Row2

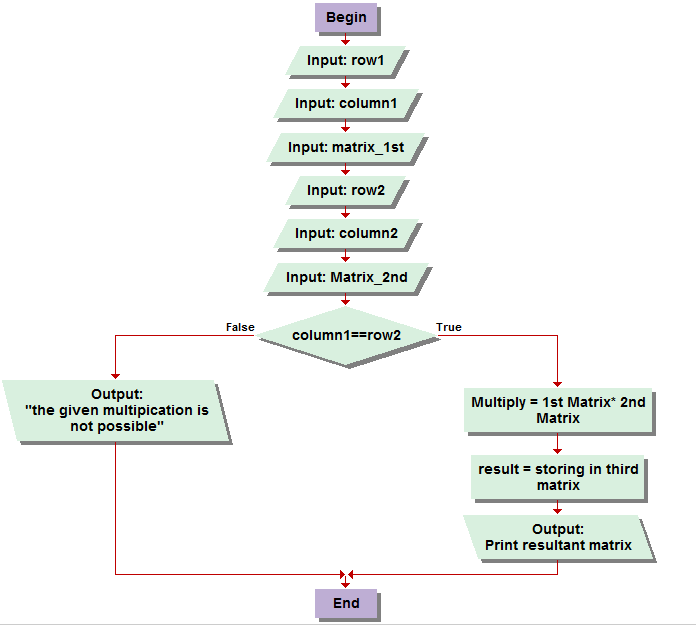
Multiply = Matrix1 \* Matrix2

Store Multiply in Matrix3

Print Matrix3 ]

1. Else Print “The given multiplication is not possible”
2. End

### Flowchart:



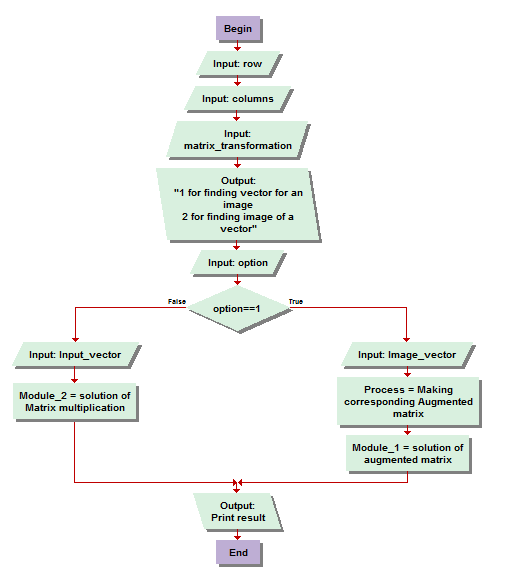
## Module 3

## (Transformations)

### Algorithm:

1. Input no. of rows.
2. Input no. of columns.
3. Input transformation (Matrix).
4. Input Command.
5. If command is finding image then go to step 6 else go to step 8.
6. Input Vector with entries equal to columns.
7. Multiply Matrix with vector by using Module 2 (Multiplication module) and go to step 11.
8. Input image vector with entries equal to rows.
9. Make a corresponding augmented matrix.
10. Solve the system using module 1(R.Echelon form).
11. Print results.

### Flowchart:



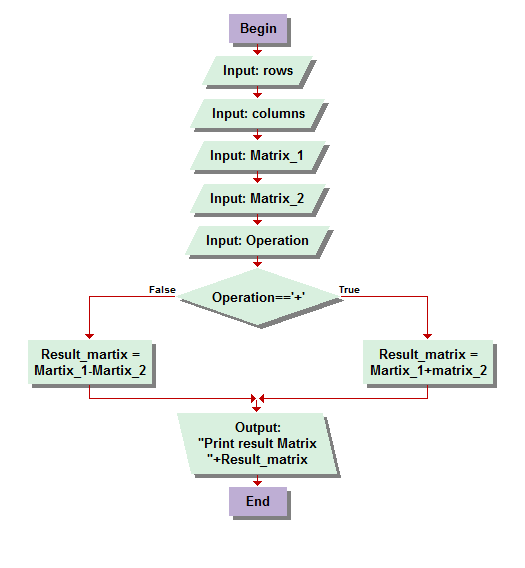
## Module 4

## (Addition and Subtraction or matrices)

### Algorithm:

1. Input no. of rows.
2. Input no. of columns.
3. Input 1st matrix with given rows and columns.
4. Input 2nd matrix with given rows and columns.
5. Input operation.
6. If operation is addition then add 1st matrix and 2nd matrix, else subtract 2nd matrix from 1st matrix.
7. Print results.

### Flowchart:



## Module 5

## (Analysis of Matrix)

### Algorithm:

1. Input number of rows.
2. Input number of columns.
3. Input a matrix with given rows and column.
4. Convert matrix into Reduce echelon form.
5. If number of columns is equal to 1, then matrix is vector go to step 13, else go to step 6.
6. If number of rows is equal to the number of pivot positions in Reduce echelon form then go to step 7 else go to step 8.
7. If number of columns is equal to number of pivot positions in Reduce echelon form, then the matrix is both one to one and onto function and go to step 9, else matrix is one-to-one and go to step 9.
8. If number of columns is equal to number of pivot positions in Reduce echelon form, then the matrix is one to one function, else matrix is not one-to-one or onto.
9. If number of pivot positions is equal to number of rows, then span is Rrows and go to step 12, else go to step 10.
10. If number of pivot positions is equal to 1, then span is a line in Rrows and go to step 12, else go to step 11.
11. If number of pivot positions is equal to 2, then span is a plane in Rrows else span is space in Rrows.
12. If number of pivot positions is equal to columns then column of matrix are independent linearly else they are dependent linearly.
13. Print results

## Flowchart:

## wadood.bmp