NAME-JYOTI
ROLL NO. -MAT/19/96

MATRICES

In mathematics,a matrix is a rectangular array or table of numbers, symbols,or expressions,arranged in rows and columns. For example , the matrix which has dimension 2× 3 has two rows and three columns.

Matrices are entered in "row form", such that

In[1]:=
```
mat1 = {{2, 1}, {-1, 2}}
```
Out[1]=
```
{{2, 1}, {-1, 2}}
```

gives the following matrix (the // and "MatrixForm" displays the result so it looks like a matrix)

In[2]:=
```
mat1 // MatrixForm
```
Out[2]//MatrixForm=

$$\begin{pmatrix} 2 & 1 \\ -1 & 2 \end{pmatrix}$$

The command below will request Mathematica to provide every output in MatrixForm

In[3]:=
```
$Post := If[MatrixQ[#], MatrixForm[#], #] &
```
In[4]:=
```
mat1
```
Out[4]//MatrixForm=

$$\begin{pmatrix} 2 & 1 \\ -1 & 2 \end{pmatrix}$$

$Post is a global variable whose value ,if set , is a function that will be applied to every output generated in the current session.

The command IfMatrixQ[#],MatrixForm[#],#]& isan example of pure function.
The symbol # represents the argument of the function.
The symbol & is used to seperate the definition of the function from the argument.
The effect of the function will be to put matrix output into MatrixForm, but to leave non matrix output alone.
this is accomplished with the if command ,which takes 3 arguments.

1- The first is a condition.
2- The second is what is returned if the condition is true.
3- The third is what is returned if the condition is false.

In[5]:=
```
MatrixQ[mat1]
```
Out[5]=
```
True
```

In[6]:=
```
MatrixQ[x]
```
Out[6]=
```
False
```

Hence the condition is checked .

the Dimensions command returns a list containing the number of rows and columns in the matrix,respectively.

In[7]:=
```
Dimensions[mat1]
```
Out[7]=
```
{2, 2}
```

Other commands that produces matrices quickly.

below command is used to get 3×4 matrix with random integer entries between 0 to 10.

In[8]:=
```
RandomInteger[10, {3, 4}]
```
Out[8]//MatrixForm=

$$\begin{pmatrix} 4 & 7 & 1 & 1 \\ 1 & 2 & 1 & 9 \\ 3 & 4 & 5 & 1 \end{pmatrix}$$

The next command gives a 3×4 matrix whose i th , j th entry is i+5j

In[9]:=
```
Table[i + 5 j, {i, 3}, {j, 4}]
```

Out[9]//MatrixForm=

$$\begin{pmatrix} 6 & 11 & 16 & 21 \\ 7 & 12 & 17 & 22 \\ 8 & 13 & 18 & 23 \end{pmatrix}$$

Zero matrix

In[10]:=
```
Table[0, {5}, {5}]
```

Out[10]//MatrixForm=

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Constant matrix

In[11]:=
```
ConstantArray [2, {3, 4}]
```

Out[11]//MatrixForm=

$$\begin{pmatrix} 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 \end{pmatrix}$$

lower triangular matrix.

In[12]:=
```
Table[If[i > j, i + 2 j, 0], {i, 4}, {j, 4}]
```

Out[12]//MatrixForm=

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 4 & 0 & 0 & 0 \\ 5 & 7 & 0 & 0 \\ 6 & 8 & 10 & 0 \end{pmatrix}$$

upper triangular matrix.

In[13]:=
```
Table[If[i < j, i + 2 j, 0], {i, 4}, {j, 4}]
```

Out[13]//MatrixForm=

$$\begin{pmatrix} 0 & 5 & 7 & 9 \\ 0 & 0 & 8 & 10 \\ 0 & 0 & 0 & 11 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Array command

It works much like the Table command but uses a function (either built-in or user defined) rather than an expression to compute the entries.
using the built-in function Min for f produces a matrix with each entry is the minimum of the row number and column number of that entry's position:

In[14]:=
```
Array[Min, {3, 4}]
```

Out[14]//MatrixForm=

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 2 & 2 \\ 1 & 2 & 3 & 3 \end{pmatrix}$$

Max command:- entry will be maximum of the entry position. for eg-21 is the entry position then maximum value is 2 so output is 2.

In[15]:=
```
Array[Max, {3, 4}]
```

Out[15]//MatrixForm=

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 2 & 3 & 4 \\ 3 & 3 & 3 & 4 \end{pmatrix}$$

user-defined function:

In[16]:=
```
f[i_, j_] := i + 2 j;
Array[f, {3, 3}]
```

Out[17]//MatrixForm=

$$\begin{pmatrix} 3 & 5 & 7 \\ 4 & 6 & 8 \\ 5 & 7 & 9 \end{pmatrix}$$

Identity Matrix:-

In[18]:=
```
IdentityMatrix [3]
```

Out[18]//MatrixForm=

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Diagonal Matrix:-

In[19]:=
```
DiagonalMatrix [{1, 2, 3, 4}]
```

Out[19]//MatrixForm=

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 4 \end{pmatrix}$$

Superdiagonal Matrix

In[20]:=
```
DiagonalMatrix [{1, 2, 3}, 1]
```

Out[20]//MatrixForm=

$$\begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Subdiagonal Matrix

In[21]:=
```
DiagonalMatrix[{1, 2, 3}, -1]
```

Out[21]//MatrixForm=

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 3 & 0 \end{pmatrix}$$

Matrix operations

In[22]:=
```
mat1 = {{1, 2, 3}, {4, 5, 6}, {2, 0, 1}}
```

Out[22]//MatrixForm=

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 2 & 0 & 1 \end{pmatrix}$$

In[23]:=
```
mat2 = {{2, 3, 4}, {5, 5, 5}, {6, 7, 8}}
```

Out[23]//MatrixForm=

$$\begin{pmatrix} 2 & 3 & 4 \\ 5 & 5 & 5 \\ 6 & 7 & 8 \end{pmatrix}$$

Addition

In[24]:=
```
mat1 + mat2
```

Out[24]//MatrixForm=

$$\begin{pmatrix} 3 & 5 & 7 \\ 9 & 10 & 11 \\ 8 & 7 & 9 \end{pmatrix}$$

Difference

In[25]:=
```
mat1 - mat2
```

Out[25]//MatrixForm=

$$\begin{pmatrix} -1 & -1 & -1 \\ -1 & 0 & 1 \\ -4 & -7 & -7 \end{pmatrix}$$

Multiplication

In[26]:=
```
mat1.mat2
```

Out[26]//MatrixForm=

$$\begin{pmatrix} 30 & 34 & 38 \\ 69 & 79 & 89 \\ 10 & 13 & 16 \end{pmatrix}$$

Scalar multiplication

In[27]:=
```
7 * mat1
```

Out[27]//MatrixForm=

$$\begin{pmatrix} 7 & 14 & 21 \\ 28 & 35 & 42 \\ 14 & 0 & 7 \end{pmatrix}$$

In[28]:=
```
7 * mat2
```

Out[28]//MatrixForm=

$$\begin{pmatrix} 14 & 21 & 28 \\ 35 & 35 & 35 \\ 42 & 49 & 56 \end{pmatrix}$$

Transpose

In[29]:=
```
Transpose[mat1]
```

Out[29]//MatrixForm=

$$\begin{pmatrix} 1 & 4 & 2 \\ 2 & 5 & 0 \\ 3 & 6 & 1 \end{pmatrix}$$

In[30]:=
```
Transpose[mat2]
```

Out[30]//MatrixForm=

$$\begin{pmatrix} 2 & 5 & 6 \\ 3 & 5 & 7 \\ 4 & 5 & 8 \end{pmatrix}$$

Inverse

In[31]:=
```
Inverse[mat1]
```

Out[31]//MatrixForm=

$$\begin{pmatrix} -\frac{5}{9} & \frac{2}{9} & \frac{1}{3} \\ -\frac{8}{9} & \frac{5}{9} & -\frac{2}{3} \\ \frac{10}{9} & -\frac{4}{9} & \frac{1}{3} \end{pmatrix}$$

Determinant

In[32]:=
```
Det[mat1]
```

Out[32]= -9

In[33]:=
```
Det[mat2]
```

Out[33]= 0

Row Reduction

Mathematica does row reduction for you. Technically this gives the "reduced row echelon form ",with as many off-diagonal zeroes as possible.

In[34]:=
```
RowReduce[mat1]
```

Out[34]//MatrixForm=

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

A simple example is discussed below:

In[35]:=
```
mat3 = {{2, 2}, {1, 1}}
```

Out[35]//MatrixForm=

$$\begin{pmatrix} 2 & 2 \\ 1 & 1 \end{pmatrix}$$

In[36]:=
```
RowReduce[mat3]
```

Out[36]//MatrixForm=

$$\begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix}$$

IMPORTANCE AND APPLICATION OF MATRICES IN DAY TO DAY LIFE

Matrices or matrix is commonly used its mathematics, but have you thought about how important it is or where you can use it? Ever wondered where the word matrix came from? Matrix is actually a Latin word used for the womb it is also used to express something that is formed or produced. Matrices are used a lot in daily life but its applications are usually not known.Below are the importance and the application of maths through matrices in a simple form.

Matrices have the following uses:

■Encryption

In encryption, we use it to scramble data for security purpose to encode and to decode this data we need matrices. There is a key which helps encode and decode data which is generated by matrices.

■Games especially 3D

They use it to alter the object, in 3d space. They use the 3d matrix to 2d matrix to convert it into the different objects as per requirement.

■Economics and business

To study the trends of a business, shares and more. To create business models etc.

■Construction

Have you seen some buildings are straight but sometimes architects try to change the outer structure of the building like the famous Burj Khalifa etc. This can be done with matrices.

A matrix is made of rows and columns you can change the number of rows and columns within a matrix.

Matrices can help support various historical structures

■Dance – contra dance

It is used to organise complicated group dances.

■Animation:

It can help make animations in a more precise and perfect.

■Physics:

Matrices are applied in the study of electrical circuits, quantum mechanics and optics. It helps in the calculation of battery power outputs, resistor conversion of electrical energy into another useful energy. Therefore, matrices play a major role in calculations. Especially in solving the problems using Kirchoff's laws of voltage and current. It helps in studying quantum physics and in using it.

■Geology:

Matrices are used for taking seismic surveys.

Everything we learn has its own application. Knowning the applications makes the things more interesting and understandable.

Thank you