**ASSIGNMENT NO:4** **DATE:11/07/2015**

**PROGRAM TITLE:Write a Program to perform the following Matrix operations**

**1.Add two Matrices.**
**2.Multiply two Matrices.**
**3.Transpose given Matrix.**
**4.Find Determinant of given Matrix.**

**THEORY:**Matrices are 2-Dimensional Arrays. There are certain conditions to check for before executing operations on matrices.
1. Both matrices must be of same dimension.
2. Column size of 1st matrix must be equal to row size of 2nd matrix.
3. Row and Column sizes are interchanged and the elements reflect this change.
4. The matrix must be square for us to find it's determinant.

**PROGRAM ALGORITHM:**

```
Algo_matrixadd(a,b)//a and b are the input matrices
{
      check if both are of same dimension;
      Create new matrix c of same dimension;
      Add corresponding elements and store in corresponding position in c;
      Return c;
}

Algo_matrixmul(a,b)//a and b are the input matrices
{
      if(column size of a not equal to row size of b)
      {
            print "Multiplication not possible";
            stop program;
      }
      create matrix c;
      for(i=1 to row size of a)
      {
            for(j=1 to column size of b)
            {
                  set s to zero;
                  for(k=1 to column size of a)
                  {
                        add a[i][k] multiplied with b[k][j] to s;
                  }
                  put s into c[i][j];
            }
      }
      return c;
}

Algo_determinant(a)//a is the given matrix
{
      check if given matrix is square or not;
```

```
     create new matrix of type double and copy elements of a to it;
     perform gauss elimination on the new matrix;
     multiply the elements of the left diagonal to get the determinant;
}
```

## **PROGRAM CODE:**

```
/*C++ Program to implement Matrix functions*/
#include<iostream>
#include<cstdlib>
using namespace std;

/*Class Matrix and associated functions*/
class Matrix
{
     int m,n;
     int **mat;
     public:
          Matrix(int,int);
          ~Matrix();
          void input();
          Matrix operator+(Matrix);
          Matrix operator*(Matrix);
          Matrix transpose();
          int det();
          void display();
};

/*Parameterised constructor*/
Matrix::Matrix(int p,int q)
{
     int i;
     m=p;
     n=q;
     mat=new int*[m];
     for(i=0;i<m;i++)
          mat[i]=new int [n];
}

/*Destructor*/
Matrix::~Matrix()
{
     int i;
     for(i=0;i<m;i++)
          delete []mat[i];
     delete []mat;
}

/*Function to take input*/
void Matrix::input()
{
     int i,j;
     cout<<"\tEnter the elements::"<<endl;
     for(i=0;i<m;i++)
     {
```

```
                  for(j=0;j<n;j++)
                  {
                        cin>>mat[i][j];
                  }
            }
}

/*Fucntion using operator overload to add two matrices*/
Matrix Matrix::operator+(Matrix a)
{
            int i,j;

            /*Checking if size of both matrices are same*/
            if((m!=a.m)||(n!=a.n))
            {
                  cout<<"\tThe Matrices cannot be added."<<endl;
                  exit(0);
            }
            Matrix b(m,n);
            for(i=0;i<m;i++)
            {
                  for(j=0;j<n;j++)
                  {
                        b.mat[i][j]=mat[i][j]+a.mat[i][j];
                  }
            }
            return b;
}

/*Function using operator overload to multiply two matrices*/
Matrix Matrix::operator*(Matrix a)
{
            int i,j,k,s;

            /*Checking if column size of frist matrix is equal to row size of
second matrix*/
            if(n!=a.m)
            {
                  cout<<"\tThe Matrices cannot be multiplied."<<endl;
                  exit(0);
            }
            Matrix b(m,a.n);
            for(i=0;i<m;i++)
            {
                  for(j=0;j<a.n;j++)
                  {
                        s=0;
                        for(k=0;k<n;k++)
                        {
                              s=s+(mat[i][k]*a.mat[k][j]);
                        }
                        b.mat[i][j]=s;
                  }
            }
            return b;
```

```
}

/*Function to transpose the matrix*/
Matrix Matrix::transpose()
{
      int i,j;
      Matrix b(n,m);
      for(i=0;i<m;i++)
      {
            for(j=0;j<n;j++)
            {
                  b.mat[j][i]=mat[i][j];
            }
      }
      return b;
}

/*Fucntion to find determinant of a square matrix*/
int Matrix::det()
{
      if(m!=n)
      {
            cout<<"\tDeterminant cannot be found. Not a square
Matrix."<<endl;
            exit(0);
      }
      int i,j,k;
      double d=1,x=0;

      /*Create new matrix of double type and copy the contents*/
      double **b=new double*[m];
      for(i=0;i<m;i++)
            b[i]=new double [n];
      for(i=0;i<m;i++)
      {
            for(j=0;j<n;j++)
                  b[i][j]=mat[i][j];
      }

      /*Perform Gauss elimination*/
      for(i=0;i<n-1;i++)
      {
            for(j=i+1;j<n;j++)
            {
                  x=-b[j][i]/b[i][i];
                  for(k=i+1;k<n;k++)
                        b[j][k]=b[j][k]+(x*b[i][k]);
            }
      }

      /*Multiply the elements in the left diagonal*/
      for(i=0;i<m;i++)
      {
            d=d*b[i][i];
      }
```

```
    cout<<"\tThe Determinant is::"<<d<<endl;
    return 0;
}

/*Function to display the matrix*/
void Matrix::display()
{
    int i,j;
    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
        {
            cout<<"\t"<<mat[i][j];
        }
        cout<<endl;
    }
}
int main()
{
    int p,q,r,ch;
    cout<<"\tEnter size of matrix::";
    cin>>r>>q;
    Matrix a(r,q);
    a.input();
    cout<<"\tThe Matrix entered is::"<<endl;
    a.display();
    while(ch)
    {


cout<<"\tMenu::\n\t1.Add\n\t2.Multiply\n\t3.Transpose\n\t4.Determinant\n\
t0.Exit\n\tEnter Choice::";
        cin>>ch;
        switch(ch)
        {
            case 1: {
                cout<<"\n\tEnter size of second matrix::";
                cin>>p>>q;
                Matrix b(p,q);
                b.input();
                cout<<"\tThe Matrix entered is::"<<endl;
                b.display();
                Matrix c(p,q);
                c=a+b;
                cout<<"\tThe Final Matrix is::\n";
                c.display();
                }
                break;
            case 2: {
                cout<<"\n\tEnter size of second matrix::";
                cin>>p>>q;
                Matrix d(p,q);
                d.input();
                cout<<"\tThe Matrix entered is::"<<endl;
                d.display();
```

```
                        Matrix e(r,q);
                        e=a*d;
                        cout<<"\tThe Final Matrix is::\n";
                        e.display();
                        }
                        break;
                case 3:    a=a.transpose();
                        cout<<"\tThe Final Matrix is::\n";
                        a.display();
                        break;
                case 4: a.det();
                        break;
                case 0: exit(0);
                        break;
                default:cout<<"\n\tRe-enter Choice.\n";
        }
    }
    return 0;
}
```

**OUTPUT:**

**Set 1:**

```
    Enter size of matrix::2 2
    Enter the elements::
3 4 5 6
    The Matrix entered is::
    3    4
    5    6
    Menu::
    1.Add
    2.Multiply
    3.Transpose
    4.Determinant
    0.Exit
    Enter Choice::3
    The Final Matrix is::
    3    5
    4    6
    Menu::
    1.Add
    2.Multiply
    3.Transpose
    4.Determinant
    0.Exit
    Enter Choice::4
    The Determinant is::-2
    Menu::
    1.Add
    2.Multiply
    3.Transpose
    4.Determinant
    0.Exit
    Enter Choice::1
```

```
     Enter size of second matrix::2 2
     Enter the elements::
1 2 3 4
     The Matrix entered is::
     1    2
     3    4
     The Final Matrix is::
     4    7
     7    10
     Menu::
     1.Add
     2.Multiply
     3.Transpose
     4.Determinant
     0.Exit
     Enter Choice::2

     Enter size of second matrix::2 3
     Enter the elements::
1 2 3 4 5 6
     The Matrix entered is::
     1    2    3
     4    5    6
     The Final Matrix is::
     23   31   39
     28   38   48
     Menu::
     1.Add
     2.Multiply
     3.Transpose
     4.Determinant
     0.Exit
     Enter Choice::0
```

**Set 2:**

```
     Enter size of matrix::2 2
     Enter the elements::
1 2 3 4
     The Matrix entered is::
     1    2
     3    4
     Menu::
     1.Add
     2.Multiply
     3.Transpose
     4.Determinant
     0.Exit
     Enter Choice::1

     Enter size of second matrix::2 3
     Enter the elements::
1 2 3 4 5 6
     The Matrix entered is::
     1    2    3
     4    5    6
     The Matrices cannot be added.
```

**Set 3:**

```
     Enter size of matrix::2 2
     Enter the elements::
1 2 3 4
     The Matrix entered is::
     1    2
     3    4
     Menu::
     1.Add
     2.Multiply
     3.Transpose
     4.Determinant
     0.Exit
     Enter Choice::2

     Enter size of second matrix::3 2
     Enter the elements::
1 2 3 4 5 6
     The Matrix entered is::
     1    2
     3    4
     5    6
     The Matrices cannot be multiplied.
```

## DISCUSSION:

We use operator overloading to add or multiply two matrices.
While finding the determinant we have had to use a double matrix as otherwise there was a loss of precision and there was an error in finding the determinant.