

Lab 9

1. Given two matrix M and N , use function overloading and operator overloading to perform the following operations:

- 1. $M+N$**
- 2. $M-N$**
- 3. $Transpose(M)$**
- 4. $M*N$**
- 5. $Inverse(M)$**
- 6. $Rank(M)$**

A. code:-

```
#include<iostream>
using namespace std;
class Matrix
{
    int a[10][10];
public:
    void accept();
    void display();

    int operator + (Matrix x);
    int operator - (Matrix y);
    int operator * (Matrix z);
    int operator ! ();
    int operator ~ ();
    int operator ++ ();
};
void Matrix::accept()
{
    cout<<"Enter the 3*3 Matrix: ";
```

```

for (int i = 0; i < 3; i++)
{
for (int j = 0; j < 3; j++)
{
cin >> a[i][j];
}
}
}
void Matrix::display()
{
for (int i = 0; i < 3; i++)
{
cout << " ";
for (int j = 0; j < 3; j++)
{
cout << a[i][j] << "\t";
}
cout << "\n";
}
}
int Matrix::operator + (Matrix x)
{
int mat[3][3];
int i;
for (i = 0; i < 3; i++)
{
for (int j = 0; j < 3; j++)
{
mat[i][j] = a[i][j] + x.a[i][j];
}
}
cout << "\nAddition of matrix is:" << endl;
for (int i = 0; i < 3; i++)
{
cout << " ";
for (int j = 0; j < 3; j++)
{
cout << mat[i][j] << "\t";
}
cout << "\n";
}
}

```

```

    }
    return 0;
}
int Matrix::operator - (Matrix y) //Subtraction using function overloading and operator
overloading
{
    int mat[3][3];
    int i;
    for (i = 0; i < 3; i++)
    {
        for (int j = 0; j < 3; j++)
        {
            mat[i][j] = a[i][j] - y.a[i][j];
        }
    }
    cout << "\nSubtraction of Matrix is:" << endl;
    for (int i = 0; i < 3; i++)
    {
        cout << " ";
        for (int j = 0; j < 3; j++)
        {
            cout << mat[i][j] << "\t";
        }
        cout << "\n";
    }
    return 0;
}
int Matrix::operator * (Matrix z)
{
    int mat[3][3];
    int i;
    for (i = 0; i < 3; i++)
    {
        for (int j = 0; j < 3; j++)
        {
            mat[i][j] = 0;
            for (int k = 0; k < 3; k++)
            {
                mat[i][j] += a[i][k] * z.a[k][j];
            }
        }
    }
}

```

```

}
}
cout << "\nMultiplication of Matrix is: \n";
for (int i = 0; i < 3; i++)
{
    cout << " ";
    for (int j = 0; j < 3; j++)
    {
        cout << mat[i][j] << "\t";
    }
    cout << "\n";
}
return 0;
}

int Matrix::operator ! () //Transposing using function overloading and operator
overloading
{
    int mat[3][3];
    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 3; j++)
        {
            mat[j][i] = a[i][j];
        }
    }
    cout << "Transpose of Matrix is: " << endl;
    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 3; j++)
        {
            cout << mat[i][j] << "\t";
        }
        cout << "\n";
    }
    return 0;
}

int Matrix::operator ~ () //Inversion using function overloading and operator overloading
{
    float determinant = 0;
    for (int i = 0; i < 3; i++)

```

```

{
    determinant = determinant + (a[0][i] * (a[1][(i + 1) % 3] * a[2][(i + 2) % 3] - a[1][(i + 2) %
3] *
a[2][(i + 1) % 3]));
}
cout << "\nDeterminant is: " << determinant;
cout << "\nInverse of matrix is: \n";
for (int i = 0; i < 3; i++)
{
    for (int j = 0; j < 3; j++)
    {
        cout << ((a[(j + 1) % 3][(i + 1) % 3] * a[(j + 2) % 3][(i + 2) % 3]) - (a[(j + 1) % 3][(i + 2) %
3] *
a[(j + 2) % 3]
[(i + 1) % 3])) / determinant << "\t";
        cout << "\n";
    }
}
return 0;
}
int Matrix::operator ++ ()
{
    float determinant = 0;
    for (int i = 0; i < 3; i++)
    {
        determinant = determinant + (a[0][i] * (a[1][(i + 1) % 3] * a[2][(i + 2) % 3] - a[1][(i + 2) %
3] *
a[2][(i + 1) % 3]));
    }
    if (determinant)
    {
        return 1;
    }
    else
    {
        return 0;
    }
}
int main()
{

```

```
Matrix M,N;  
int y;  
M.accept();  
N.accept();  
cout << "First Matrix is: \n";  
M.display();  
cout << "Second Matrix is: \n";  
N.display();
```

```
//operators  
M + N;  
M - N;  
M *N;  
!M;  
~M;  
y = ++M;  
if (y == 1)  
cout << "Rank=2" << endl;  
if (y == 0)  
cout << "Rank=0" << endl;  
return 0;  
}
```

SAMPLE INPUT AND SAMPLE OUTPUT:

```
Enter the 3*3 Matrix: 0 -3 -2 1 -4 -2 -3 4 1
Enter the 3*3 Matrix: 0 -3 -2 1 -4 -2 -3 4 1
First Matrix is:
0      -3      -2
1      -4      -2
-3      4       1
Second Matrix is:
0      -3      -2
1      -4      -2
-3      4       1

Addition of matrix is:
0      -6      -4
2      -8      -4
-6      8       2

Subtraction of Matrix is:
0      0       0
0      0       0
0      0       0

Multiplication of Matrix is:
3      4       4
2      5       4
1      -3      -1
```

```
Transpose of Matrix is:
0      1      -3
-3      -4      4
-2      -2      1

Determinant is: 1
Inverse of matrix is:
4
-5
-2
5
-6
-2
-8
9
3
Rank=2
```