

LAB WORK

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)

Code:

```
#include <iostream>
using namespace std;
class Matrix
{
private:
    int M[20][20];
public:
    void input_matrix()
    {
        cout<<"\nEnter the 3*3 matrix: \n";
        for(int i=0;i<3;i++)
        {
            for(int j=0;j<3;j++)
            {
                cin>>M[i][j];
            }
        }
    }
}
```

```

}
int operator +( Matrix X)
{
    int L[10][10];
    for(int i=0;i<3;i++)
    {
        for(int j=0;j<3;j++)
        {
            L[i][j]=M[i][j]+X.M[i][j]; //sum of matrices//
        }
    }
    cout<<"\n\nAddition: \n";
    for(int i=0;i<3;i++)
    {
        for(int j=0;j<3;j++)
        {
            cout<<L[i][j]<<' ';
        }
        cout<<endl;
    }
    return 0;
}
int operator -( Matrix Y)
{
    int L[10][10];
    for(int i=0;i<3;i++)

```

```

{
    for(int j=0;j<3;j++)
    {
        L[i][j]=M[i][j]-Y.M[i][j];  //Difference of matrices//
    }
}
cout<<"\n\nDifference: \n";
for(int i=0;i<3;i++)
{
    for(int j=0;j<3;j++)
    {
        cout<<L[i][j]<<' ';
    }
    cout<<endl;
}
return 0;

}

int operator *(Matrix Z)
{
    int L[10][10];
    for(int i=0;i<3;i++)
    {
        for(int j=0;j<3;j++)
        {
            L[i][j]=0;
            for(int k=0;k<3;k++)

```

```

        {
            L[i][j]+=M[i][k]*(Z.M[k][j]); //Multiplication of matrices//
        }

    }

}

cout<<"\n\nMultiplication: \n";
for(int i=0;i<3;i++)
{
    for (int j=0;j<3;j++)
    {
        cout<<L[i][j]<<' ';
    }
    cout<<endl;
}

return 0;
}

int operator !()    //Transpose of matrix//
{
    int L[10][10],i,j;
    for(int i=0;i<3;i++)
    {
        for(int j=0;j<3;j++)
        {
            L[j][i]=M[i][j];
        }
    }
}

```

```

cout<<"\n\nTranspose: \n";
for(int i=0;i<3;i++)
{
    for(int j=0;j<3;j++)
    {
        cout<<L[i][j]<<' ';
    }
    cout<<endl;
}
return 0;
}

int operator ~ ()          //Inverse//
{
    float L[10][10];
    float determinant = 0;
    for (int i = 0; i <3; i++)
    {
        determinant = determinant+ (M[0][i] * (M[1][(i + 1)% 3 ] * M[2][(i +
2)% 3 ] - M[1][(i + 2) % 3] * M[2][(i + 1)% 3]));
    }
    cout << "\nDeterminant: " << determinant;
    cout << "\n\nInverse: \n";
    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 3; j++)
        {
            cout << ((M[(j + 1) % 3][(i + 1) % 3] * M[(j + 2) % 3][(i + 2) % 3]) -
(M[(j + 1) % 3][(i + 2) % 3] * M[(j + 2) % 3]

```

```

        [(i + 1) % 3])) / determinant << "\t";

    }

    cout<<endl;

}

return 0;

}

int operator ++ ()          //Rank of matrix//
{
    float determinant = 0;
    for (int i = 0; i < 3; i++)
    {
        determinant = determinant + (M[0][i] * (M[1][(i + 1) % 3] * M[2][(i + 2) %
3] - M[1][(i + 2) % 3] * M[2][(i + 1) % 3]));
    }
    if (determinant!=0)
    {
        cout<<"\n\nRank is 3"<<endl;
    }
    else
    {
        int p,q,r,s;
        p=M[0][0]*M[1][1]-M[0][1]*M[1][0];
        q=M[0][1]*M[1][2]-M[0][2]*M[1][1];
        r=M[1][0]*M[2][1]-M[1][1]*M[2][0];
        s=M[1][1]*M[2][2]-M[1][2]*M[2][1];
        if(p!=0||q!=0||r!=0||s!=0)

```

```

    {
        cout<<"\n\nRank is 2"<<endl;
    }
else
{
    int c=0;
    for(int i=0;i<3;i++)
    {
        for(int j=0;j<3;j++)
        {
            if(M[i][j]!=0)
            {
                cout<<"\n\nRank is 1"<<endl;
            }
        }
    }
}

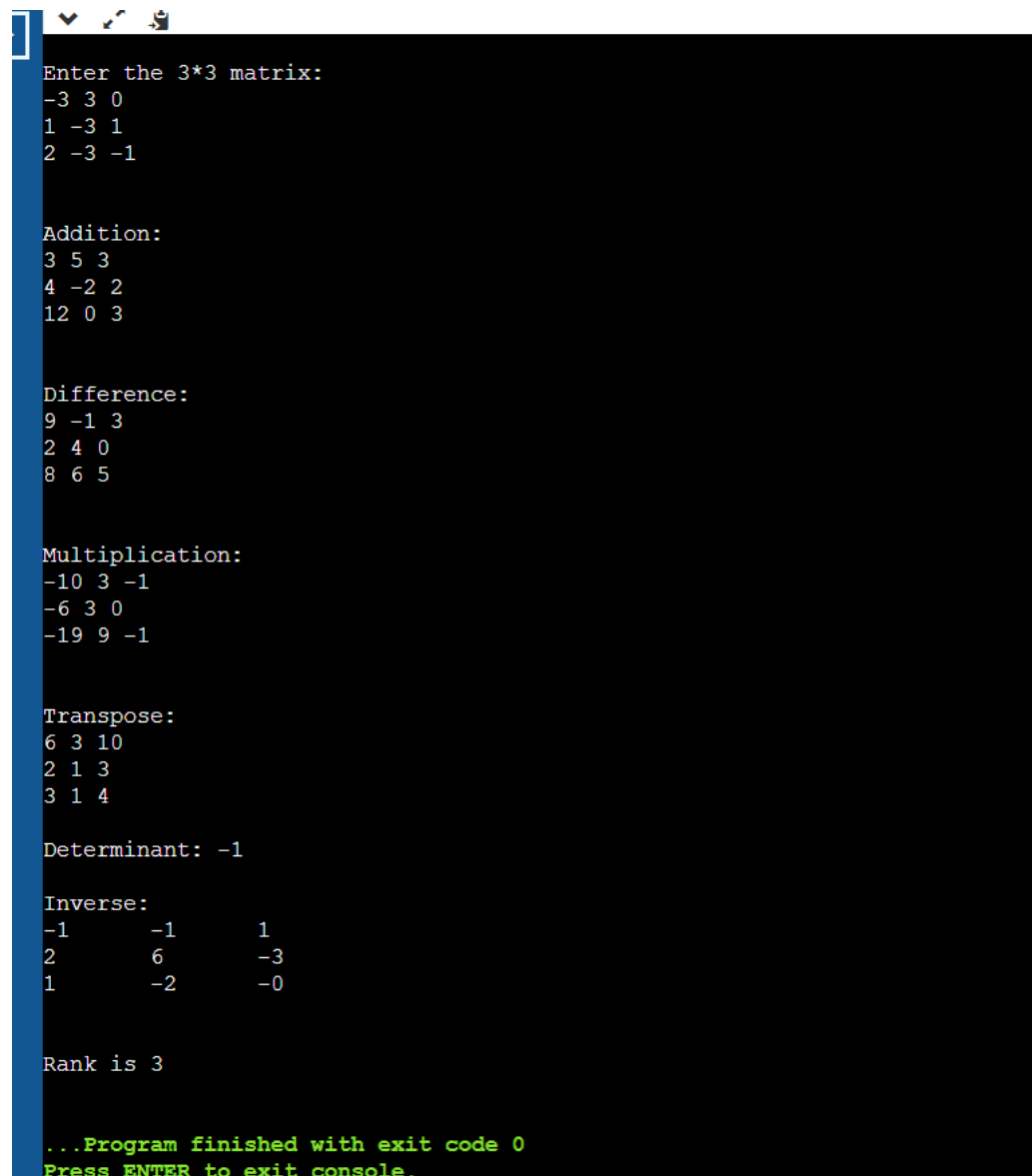
return 0;
}

};

int main()
{
    Matrix x,y;
    x.input_matrix();
    y.input_matrix();
    x+y;

```

```
x-y;  
x*y;  
!x;  
~X;  
++X;  
}
```

A terminal window with a black background and white text. The window has a blue title bar at the top with standard Linux window controls. The text inside the terminal shows a 3x3 matrix being entered, followed by calculations for addition, difference, multiplication, transpose, determinant, inverse, and rank.

```
Enter the 3*3 matrix:  
-3 3 0  
1 -3 1  
2 -3 -1  
  
Addition:  
3 5 3  
4 -2 2  
12 0 3  
  
Difference:  
9 -1 3  
2 4 0  
8 6 5  
  
Multiplication:  
-10 3 -1  
-6 3 0  
-19 9 -1  
  
Transpose:  
6 3 10  
2 1 3  
3 1 4  
  
Determinant: -1  
  
Inverse:  
-1      -1      1  
2       6      -3  
1      -2      -0  
  
Rank is 3  
  
...Program finished with exit code 0  
Press ENTER to exit console.
```