# Program on Various operations on matrix

(Addition, Substraction, Multiplication, Upper Triangular or not, Transpose of matrix, sum of Principal Diagonal elements)

```
#include <iostream>

using namespace std;


int row, col, i, j;

    int y,  m1[10][10], m2[10][10], sum[10][10], diff[10][10], x;

    int product[10][10], alter_m1[10][10], alter_m2[10][10] ;

   bool flag, success;


   void menu () {

    cout<<"\nChoose option from the menu: ";

    cout<<"\n 1. Matrix Addition";

    cout<<"\n 2. Matrix Subtraction";

    cout<<"\n 3. Matrix Multiplication";

    cout<<"\n 4. To find matrix is Upper Triangular or not";

    cout<<"\n 5. To find Transpose of matrix";

    cout<<"\n 6. To find sum of Principal Diagonal elements";

    cout<<"\n 7. ";


    cout<<"\nEnter your choice: ";

    cin>>x;

    }


int add() {

  for (i = 0; i < row; i++)

    {

        for (j = 0; j < col; j++)
    {
      sum[i][j] = m1[i][j] + m2[i][j];
      cout << sum[i][j] << "  ";
    }

    cout << endl;
  }
  cout << "The addition matrix is "<< sum[i][j];
}

int subtract() {
 for (i = 0; i < row; i++)
  {
    for (j = 0; j < col; j++)
    {
      diff[i][j] = m1[i][j] - m2[i][j];
      cout << diff[i][j] << "  ";
    }

    cout << endl;
  }
  cout << "The subtraction matrix is "<< diff[i][j];
}

int multiply() {
 for (i = 0; i < row; i++)
  {
    for (j = 0; j < col; j++)
    {
      product[i][j] = m1[i][j] * m2[i][j];
      cout << product[i][j] << "  ";
    }
    cout << endl;
  }
}

int transpose() {
 for (i = 0; i < row; i++)
  {
    for (j = 0; j < col; j++)
    {
      alter_m1[j][i] = m1[i][j];
      alter_m2[j][i] = m2[i][j];

    }
  }
  cout << "The transpose of 1st matrix is\n";
  for (i = 0; i < row; i++)
  {
    for (j = 0; j < col; j++)
    {

      cout << alter_m1 [i][j]<<" ";
    }
   cout << "\n";
  }
  cout << "\nThe transpose of 2nd matrix is\n";
  for (i = 0; i < row; i++)
  {
    for (j = 0; j < col; j++)
    {
      cout << alter_m2 [i][j]<<" ";
```

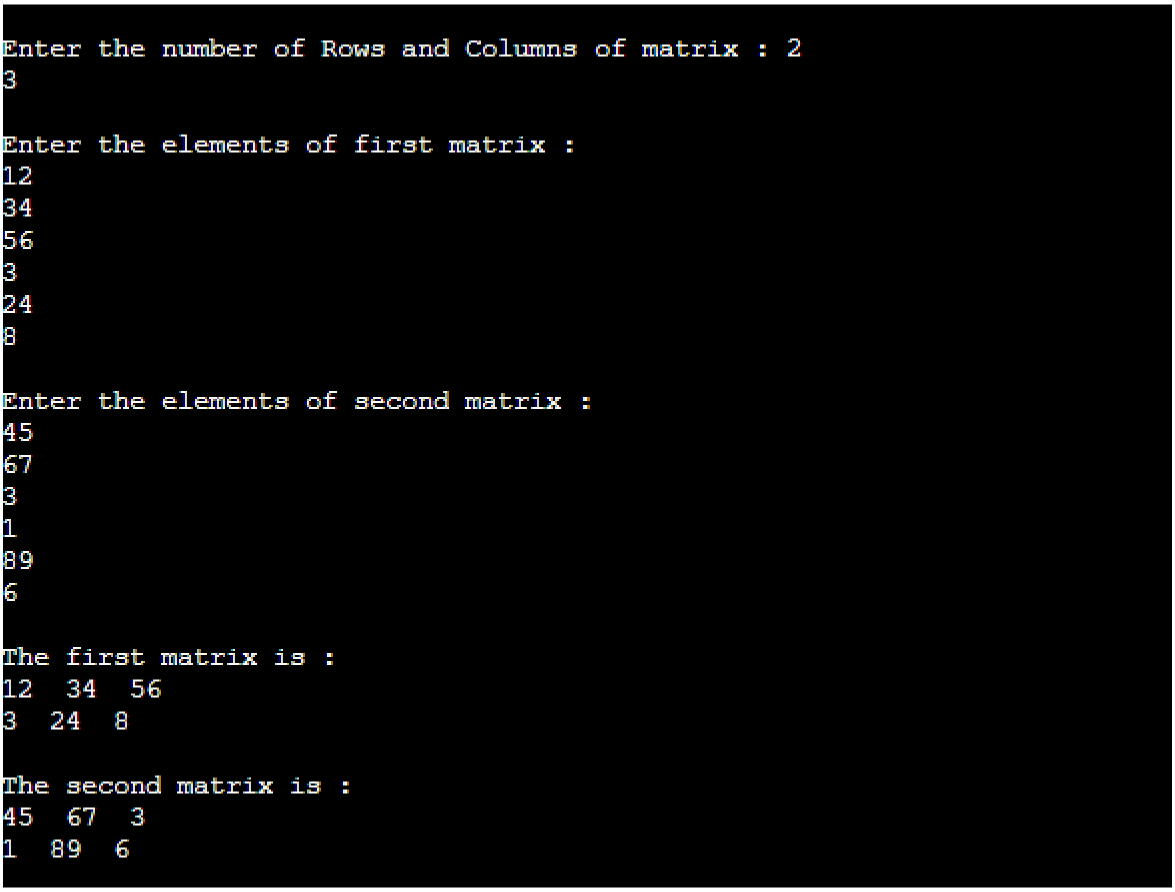
```
        }
       cout << endl;
    }
}

int  isUpper()
{
  for (i = 1; i < row; i++) {
    for (j = 0; j < i; j++) {
      if (m1[i][j] != 0)
        flag = 0;
      else
        flag = 1;

     if (m2[i][j] != 0) {
         success = 0;
         }
     else {
         success = 1;
        }
     }
  }
  if (flag == 1 && success == 1)
    cout << "Both matrices A and B are upper triangular\n";
  else if (flag == 1)
    cout << "Matrix A is upper triangular";
  else if (success == 1)
    cout << "Matrix B is upper triangular\n";
  else
    cout << "Neither A or B is upper triangular\n";
  }
```

```
int diagonalSum() {
 int sum1=0, sum2=0;
  for(i=0;i<row;++i) {
     for(j=0;j<col;++j) {

        if(i==j) {
          sum1= sum1+m1[i][j];
          sum2 = sum2 + m2[i][j];
}
}
}
cout << "Sum of diagonal elements of matrix A is " << sum1 << "\n";
cout << "Sum of diagonal elements of matrix B is " << sum2 << "\n";
}

int main()
{

    cout << "\nEnter the number of Rows and Columns of matrix : ";
    cin >> row >> col;

    cout << "\nEnter the elements of first matrix : \n";

    for (i = 0; i < row; i++) //row=2  i=1
    {
        for (j = 0; j < col; j++)   //col=2
        {
            cin >> m1[i][j];
        }
    }
```

```
cout << "\nEnter the elements of second matrix : \n";

for (i = 0; i < row; i++)
{
    for (j = 0; j < col; j++)
    {
        cin >> m2[i][j];

    }
}

for (i = 0; i < row; i++)
{
    for (j = 0; j < col; j++)
    {
        sum[i][j] = m1[i][j] + m2[i][j];
    }
}

cout << "\nThe first matrix is : \n";

for (i = 0; i < row; i++)
{
    for (j = 0; j < col; j++)
    {
        cout << m1[i][j] << "  ";
    }
    cout << endl;
}
```

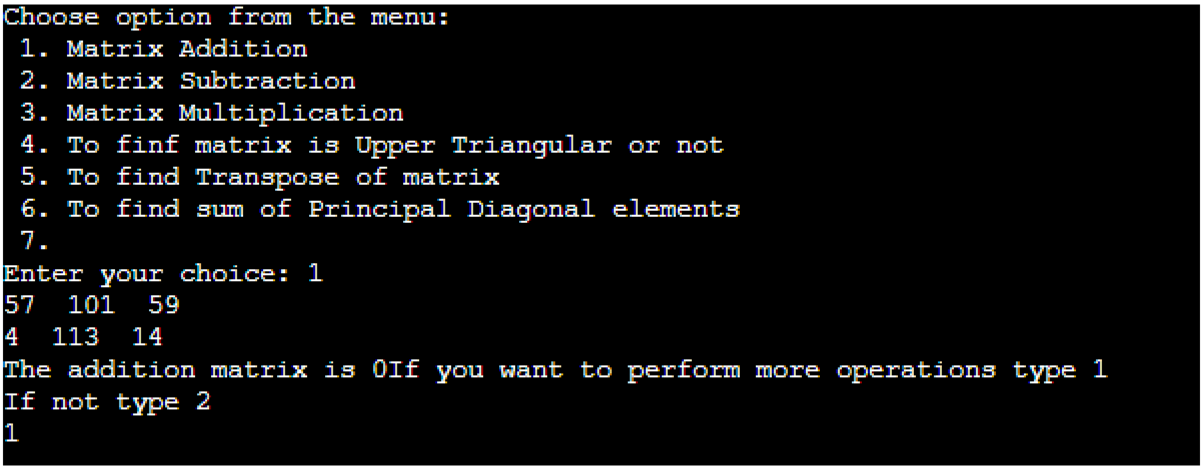
```
    cout << "\nThe second matrix is : \n";


    for (i = 0; i < row; i++)
    {
        for (j = 0; j < col; j++)
        {
            cout << m2[i][j] << "  ";
        }
        cout << endl;
    }
   gotostat:
// int recall(){
  menu();


switch(x)
{
case 1: add();
  break;
case 2: subtract();
  break;
case 3: multiply();
  break;
case 4: isUpper();
  break;
case 5: transpose();
  break;
case 6: diagonalSum();
  break;
default:
cout<<"\nInvalid choice ! ";
// }
}
cout << "If you want to perform more operations type 1\n";
cout << "If not type 2\n";
cin >> y;


if ( y == 1) {
//   recall();
  goto gotostat;
}
else {
  cout << "Thanks\n";
}
    return 0;
}
```

## Output:



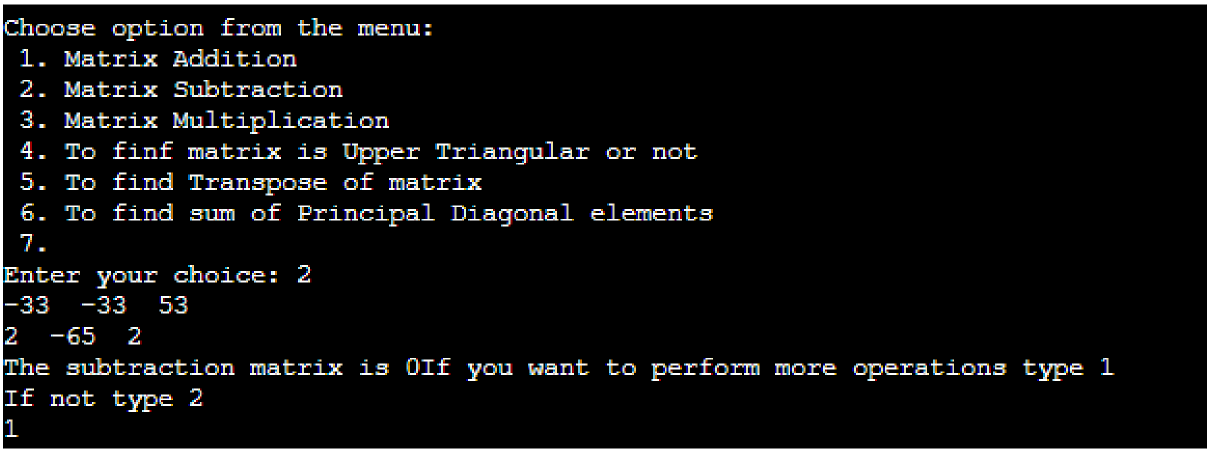
```
Enter the number of Rows and Columns of matrix : 2
3

Enter the elements of first matrix :
12
34
56
3
24
8

Enter the elements of second matrix :
45
67
3
1
89
6

The first matrix is :
12   34   56
3   24   8

The second matrix is :
45   67   3
1   89   6
```

**Addition:**



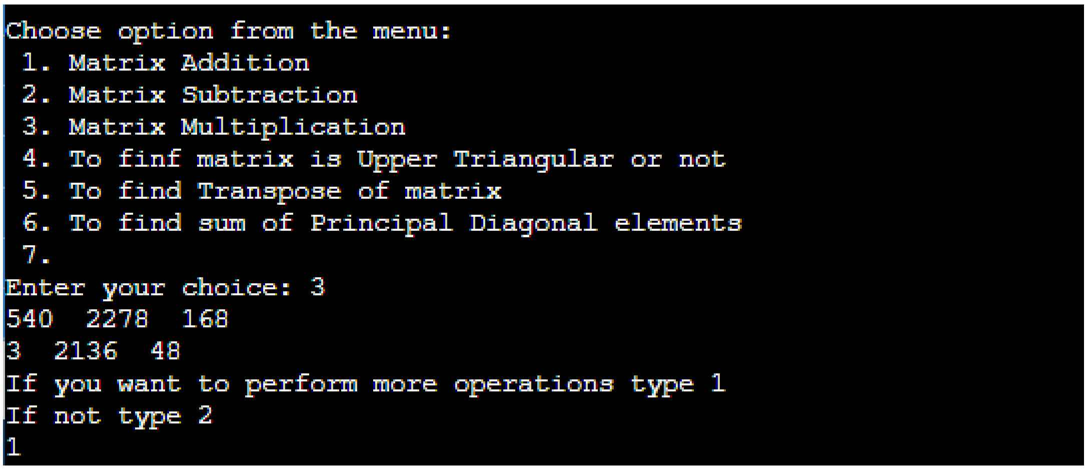
```
Choose option from the menu:
 1. Matrix Addition
 2. Matrix Subtraction
 3. Matrix Multiplication
 4. To finf matrix is Upper Triangular or not
 5. To find Transpose of matrix
 6. To find sum of Principal Diagonal elements
 7.
Enter your choice: 1
57  101  59
4  113  14
The addition matrix is 0If you want to perform more operations type 1
If not type 2
1
```

**Substraction:**



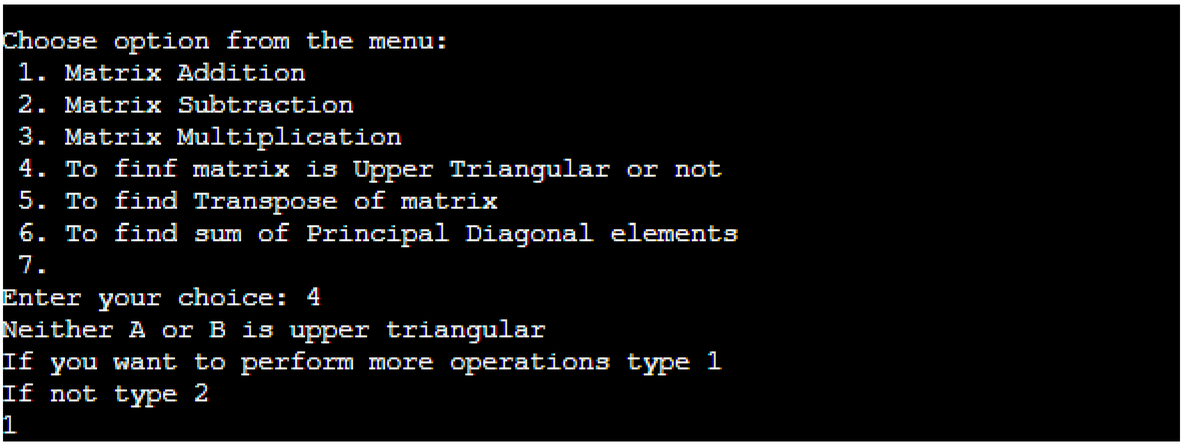
```
Choose option from the menu:
 1. Matrix Addition
 2. Matrix Subtraction
 3. Matrix Multiplication
 4. To finf matrix is Upper Triangular or not
 5. To find Transpose of matrix
 6. To find sum of Principal Diagonal elements
 7.
Enter your choice: 2
-33  -33  53
2  -65  2
The subtraction matrix is 0If you want to perform more operations type 1
If not type 2
1
```

**Multiplication:**



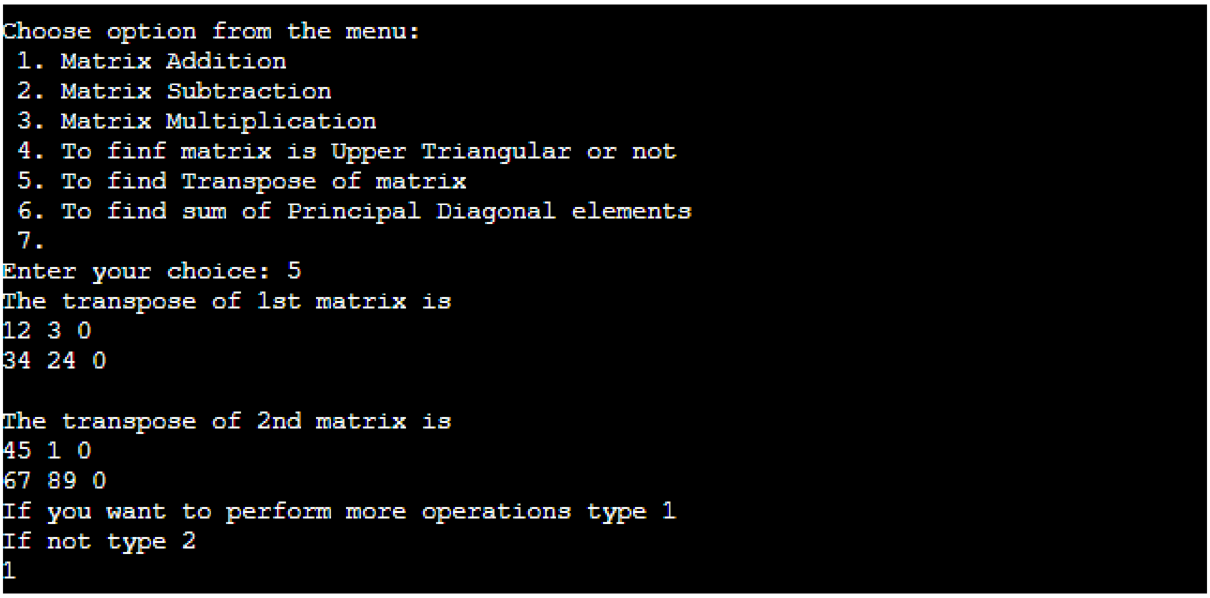
```
Choose option from the menu:
 1. Matrix Addition
 2. Matrix Subtraction
 3. Matrix Multiplication
 4. To finf matrix is Upper Triangular or not
 5. To find Transpose of matrix
 6. To find sum of Principal Diagonal elements
 7.
Enter your choice: 3
540  2278  168
3  2136  48
If you want to perform more operations type 1
If not type 2
1
```

**Upper Triangular or not:**



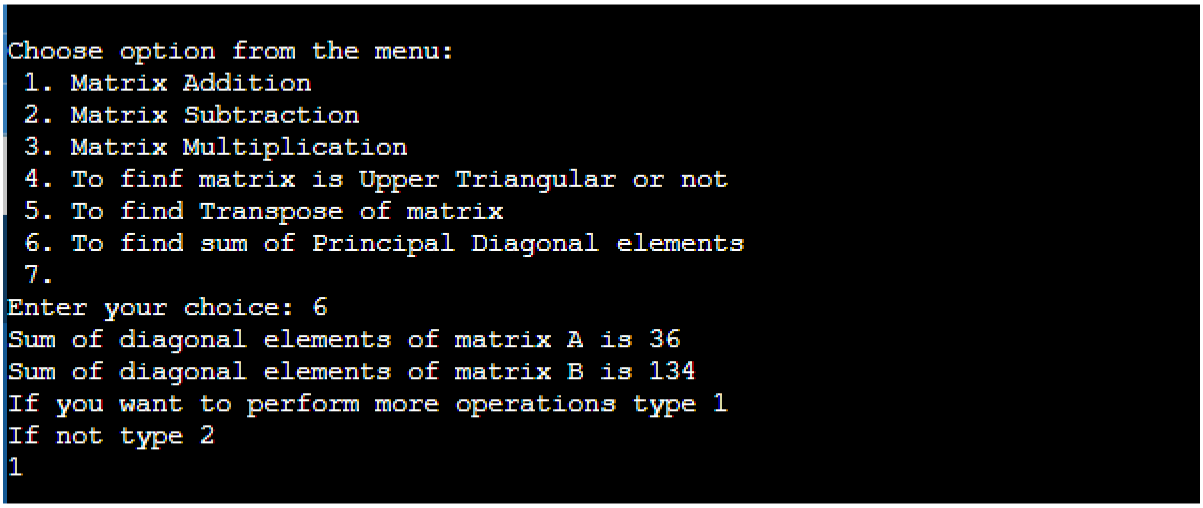
```
Choose option from the menu:
 1. Matrix Addition
 2. Matrix Subtraction
 3. Matrix Multiplication
 4. To finf matrix is Upper Triangular or not
 5. To find Transpose of matrix
 6. To find sum of Principal Diagonal elements
 7.
Enter your choice: 4
Neither A or B is upper triangular
If you want to perform more operations type 1
If not type 2
1
```

**Transpose of matrix:**

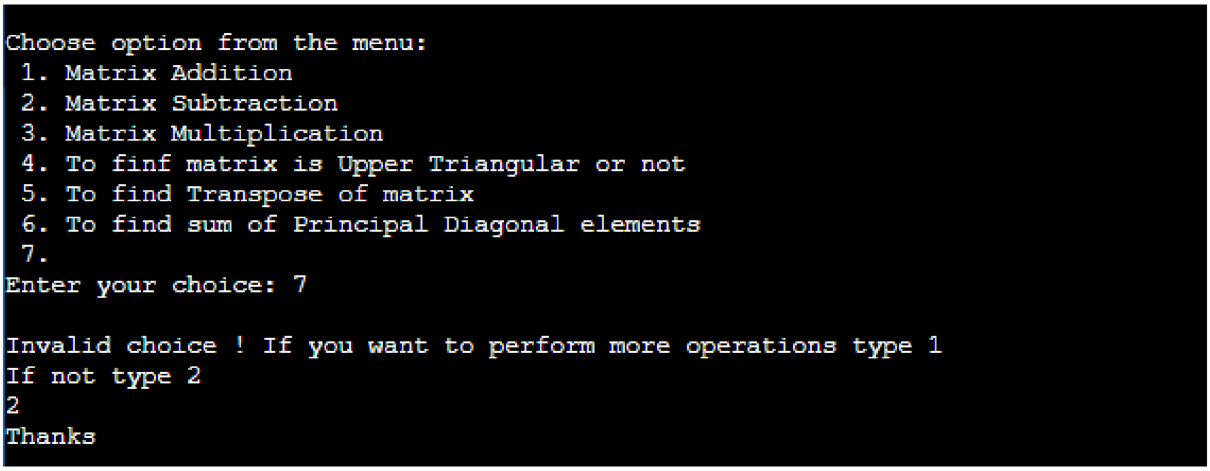


```
Choose option from the menu:
 1. Matrix Addition
 2. Matrix Subtraction
 3. Matrix Multiplication
 4. To finf matrix is Upper Triangular or not
 5. To find Transpose of matrix
 6. To find sum of Principal Diagonal elements
 7.
Enter your choice: 5
The transpose of 1st matrix is
12 3 0
34 24 0

The transpose of 2nd matrix is
45 1 0
67 89 0
If you want to perform more operations type 1
If not type 2
1
```

**Sum of principal diagonal elements:**



```
Choose option from the menu:
 1. Matrix Addition
 2. Matrix Subtraction
 3. Matrix Multiplication
 4. To finf matrix is Upper Triangular or not
 5. To find Transpose of matrix
 6. To find sum of Principal Diagonal elements
 7.
Enter your choice: 6
Sum of diagonal elements of matrix A is 36
Sum of diagonal elements of matrix B is 134
If you want to perform more operations type 1
If not type 2
1
```



```
Choose option from the menu:
 1. Matrix Addition
 2. Matrix Subtraction
 3. Matrix Multiplication
 4. To finf matrix is Upper Triangular or not
 5. To find Transpose of matrix
 6. To find sum of Principal Diagonal elements
 7.
Enter your choice: 7

Invalid choice ! If you want to perform more operations type 1
If not type 2
2
Thanks
```