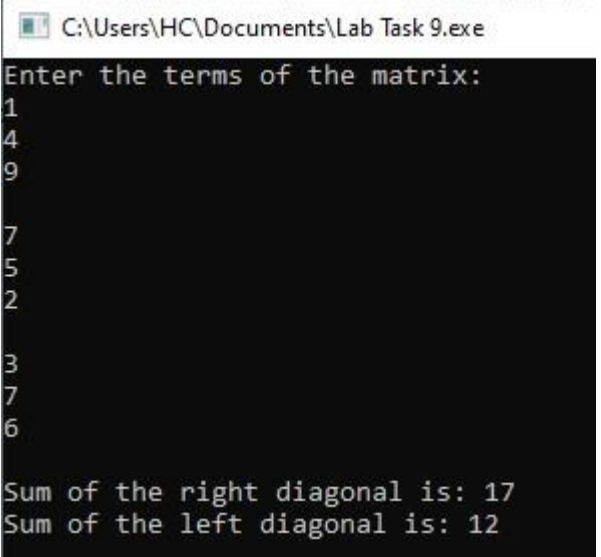


TASK 1

```
int main() {
    int Right=0,Left=0;
    int diagonal[3][3];
    cout<<"Enter the terms of the matrix:\n";
    for (int i=0;i<3;i++) {
        for (int j=0;j<3;j++) {
            cin>>diagonal[i][j];
        }
        cout<<endl;
    }
    for (int i=0;i<3;i++) {
        for (int j=0;j<3;j++) {
            if (i==j) {
                Left+=diagonal[i][j];
            }
            if (i+j==2) {
                Right+=diagonal[i][j];
            }
        }
    }
    cout<<"Sum of the right diagonal is: "<<Right<<endl;
    cout<<"Sum of the left diagonal is: "<<Left<<endl;
    return 0;
}
```



```
C:\Users\HC\Documents\Lab Task 9.exe
Enter the terms of the matrix:
1
4
9
7
5
2
3
7
6
Sum of the right diagonal is: 17
Sum of the left diagonal is: 12
```

TASK 2

```
int main () {
    int add[3][3];
    int matrix1[3][3];
    int matrix2[3][3];
    cout<<"Enter the terms for the first matrix: "<<endl;
    for (int i=0;i<3;i++) {
        for (int j=0;j<3;j++) {
            cin>>matrix1[i][j]; }
        cout<<endl; }
    cout<<"Enter the terms for the second matrix:
    "<<endl;
    for (int i=0;i<3;i++) {
        for (int j=0;j<3;j++) {
            cin>>matrix2[i][j];
        }
        cout<<endl;
    }
    for (int i=0;i<3;i++) {
        for (int j=0;j<3;j++) {
            add[i][j]=matrix1[i][j]+matrix2[i][j];
        }
    }
    cout<<"The resultant matrix is: "<<endl;
    for (int i=0;i<3;i++) {
        for (int j=0;j<3;j++) {
            cout<<add[i][j]<<" ";
        }
        cout<<endl;
    }
    return 0;
```

```
Enter the terms for the first matrix:
4
5
2
8
7
5
2
3
9
Enter the terms for the second matrix:
2
5
8
1
2
7
4
3
78
The resultant matrix is:
6 10 10
9 9 12
6 6 87
```

```
}
```

TASK 3

```
void transfunction(int matrix[3][3]) {
```

```
    int transpose[3][3];
```

```
    for (int i=0;i<3;i++) {
```

```
        for (int j=0;j<3;j++) {
```

```
            transpose[j][i]=matrix[i][j];
```

```
        }
```

```
    }
```

```
    cout<<"The original matrix is:\n";
```

```
    for (int i=0;i<3;i++){
```

```
        for (int j=0;j<3;j++){
```

```
            cout<<matrix[i][j]<<" ";
```

```
        }
```

```
    cout<<endl; }
```

```
    cout<<"The transpose of the matrix is:\n";
```

```
    for (int i=0;i<3;i++){
```

```
        for (int j=0;j<3;j++){
```

```
            cout<<transpose[i][j]<<" "; }
```

```
    cout<<endl;
```

```
}}
```

```
int main () {
```

```
    int matrix1[3][3];
```

```
    cout<<"Enter the values for the matrix: "<<endl;
```

```
    for (int i=0;i<3;i++){
```

```
        for (int j=0;j<3;j++){
```

```
            cin>>matrix1[i][j];
```

```
        }
```

```
    transfunction(matrix1);
```

C:\Users\HC\Documents\Lab Task 9.exe

Enter the values for the matrix:

2

4

1

5

8

6

4

5

9

The original matrix is:

2 4 1

5 8 6

4 5 9

The transpose of the matrix is:

2 5 4

4 8 5

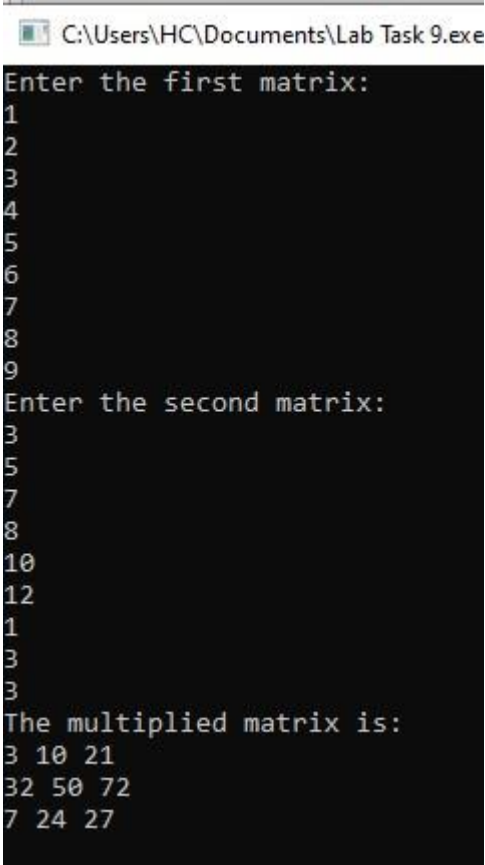
1 6 9

```
return 0;
}
```

TASK 4

```
void multi(int mat1[3][3], int mat2[3][3]) {
    int mat3[3][3];
    for (int i=0;i<3;i++) {
        for (int j=0;j<3;j++) {
            mat1[i][j]*mat2[i][j];
            mat3[i][j]=mat1[i][j]*mat2[i][j];} }
    cout<<"The multiplied matrix is:"<<endl;
    for (int i=0;i<3;i++) {
        for (int j=0;j<3;j++) {
            cout<<mat3[i][j]<<" ";
        }
        cout<<endl;
    }
}

int main () {
    int mat1[3][3],mat2[3][3];
    cout<<"Enter the first matrix: "<<endl;
    for (int i=0;i<3;i++) {
        for (int j=0;j<3;j++) {
            cin>>mat1[i][j];
        } }
    cout<<"Enter the second matrix: "<<endl;
    for (int i=0;i<3;i++) {
        for (int j=0;j<3;j++) {
            cin>>mat2[i][j];
        } }
    multi(mat1,mat2);
    return 0;
```




```
C:\Users\HC\Documents\Lab Task 9.exe
Enter the first matrix:
1
2
3
4
5
6
7
8
9
Enter the second matrix:
3
5
7
8
10
12
1
3
3
The multiplied matrix is:
3 10 21
32 50 72
7 24 27
```

```
}
```

TASK 5

```
void table(int n, int i=1) {  
    if (i<=10) {  
        cout<<n<<" x "<<i<<" = "<<n*i<<endl;  
        table(n,i+1);  
    }  
}  
  
int main () {  
    int num;  
    cout<<"Enter the number for the table: "<<endl; cin>>num;  
    table(num);  
    return 0;  
}
```

 C:\Users\HC\Documents\Lab Task 9.exe

```
Enter the number for the table:  
14  
14 x 1 = 14  
14 x 2 = 28  
14 x 3 = 42  
14 x 4 = 56  
14 x 5 = 70  
14 x 6 = 84  
14 x 7 = 98  
14 x 8 = 112  
14 x 9 = 126  
14 x 10 = 140
```

HOMETASK 1

```
#include <iostream>
```

```
double determinant2x2(double a, double b, double c, double d) {  
    return a * d - b * c;  
}
```

```
double determinant3x3(double matrix[3][3]) {  
    double a = matrix[0][0], b = matrix[0][1], c = matrix[0][2];  
    double d = matrix[1][0], e = matrix[1][1], f = matrix[1][2];  
    double g = matrix[2][0], h = matrix[2][1], i = matrix[2][2];  
  
    return a * determinant2x2(e, f, h, i) - b * determinant2x2(d, f, g, i) + c * determinant2x2(d,  
e, g, h);
```

```
}
```

```
void inverse3x3(double matrix[3][3], double inverse[3][3]) {
```

```
    double det = determinant3x3(matrix);
```

```
    if (det == 0) {
```

```
        std::cout << "Inverse does not exist as the determinant is zero.\n";
```

```
        return;
```

```
    }
```

```
    double invDet = 1.0 / det;
```

```
    for (int i = 0; i < 3; ++i) {
```

```
        for (int j = 0; j < 3; ++j) {
```

```
            int sign = ((i + j) % 2 == 0) ? 1 : -1;
```

```
            inverse[j][i] = sign * determinant2x2(matrix[(i + 1) % 3][(j + 1) % 3],
```

```
                matrix[(i + 1) % 3][(j + 2) % 3],
```

```
                matrix[(i + 2) % 3][(j + 1) % 3],
```

```
                matrix[(i + 2) % 3][(j + 2) % 3]) * invDet;
```

```
        }
```

```
    }
```

```
}
```

```
void displayMatrix(double matrix[3][3]) {
```

```
    for (int i = 0; i < 3; ++i) {
```

```
        for (int j = 0; j < 3; ++j) {
```

```
            std::cout << matrix[i][j] << " ";
```

```
        }
```

```

        std::cout << "\n";
    }
}

int main() {
    double matrix[3][3] = {{1, 2, 3},
                           {4, 5, 6},
                           {7, 8, 9}};

    double inverse[3][3];

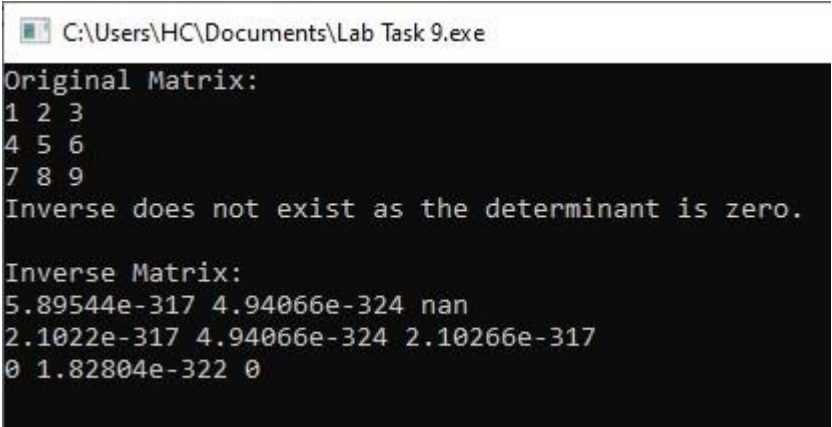
    std::cout << "Original Matrix:\n";
    displayMatrix(matrix);

    inverse3x3(matrix, inverse);

    std::cout << "\nInverse Matrix:\n";
    displayMatrix(inverse);

    return 0;
}

```



```

C:\Users\HC\Documents\Lab Task 9.exe
Original Matrix:
1 2 3
4 5 6
7 8 9
Inverse does not exist as the determinant is zero.

Inverse Matrix:
5.89544e-317 4.94066e-324 nan
2.1022e-317 4.94066e-324 2.10266e-317
0 1.82804e-322 0

```