

```

/*****
**
**                                SAKARYA ÜNİVERSİTESİ
**                                BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ
**                                BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ 2021-2022
**                                PROGRAMLAMAYA GİRİŞ DERSİ YAZ OKULU ÖDEVİ
**
**                                ÖDEV NUMARASI.....:1
**                                ÖĞRENCİ ADI:Sümeyye Üstün
**                                ÖĞRENCİ NUMARASI.:B211210065
**                                DERS GRUBU:1A
*****/

```

```

#include <iostream>
#include <locale>
#include <stdlib.h>
#include <stdio.h>
#include<process.h>

```

```

using namespace std;
#define N 4
int secim;
int x[4][4];
int y[4][4];
int t[N][N];

```

```

int p, q, n;

```

```

void matrisyaz() //Kendi belirlediğim sınır olan 4 için matris
oluşturulur.

```

```

{
    srand(time(NULL));
    cout << endl;
    cout << "X Matrisi " << endl << endl;
    for (int i = 0; i < 4; i++) //Satırlar için döngü
    {
        for (int j = 0; j < 4; j++) //Sütunlar için
        {
            x[i][j] = rand() % 10; //Yine kendi
            belirlediğim sınıra kadar random sayı üretilir.

            cout << "\t" << x[i][j];

        }
        cout << endl;
    }
    cout << endl;
    cout << "Y Matrisi " << endl << endl;
    for (int i = 0; i < 4; i++) //Satırlar için döngü
    {
        for (int j = 0; j < 4; j++) //Sütunlar için
        {
            y[i][j] = rand() % 10;

```

```

        cout << "\t" << y[i][j];

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

}

void matristoplama()    // Toplam hesaplama fonksiyonu
{
    matrisyaz(); ::x[4][4], y[4][4];
    int toplam[4][4]; //Matrislerin toplamı için oluşturulan matris
    cout << endl;
    for (int i = 0; i < 4; i++)                //Satırlar için döngü
    {
        for (int j = 0; j < 4; j++)            //Sütunlar için
döngü
        {
            toplam[i][j] = x[i][j] + y[i][j];
            cout << "\t" << toplam[i][j];
        }
        cout << endl;
    }
    cout << endl;
}

void matriscarpma()    // Çarpım hesaplama fonksiyonu
{
    matrisyaz(); ::x[4][4], y[4][4];
    int carpim[4][4];                //Matrislerin çarpımı için
oluşturulan matris
    cout << endl;
    for (int i = 0; i < 4; i++)        //Satırlar için döngü
    {
        for (int j = 0; j < 4; j++)    //Sütunlar için döngü
        {
            carpim[i][j] = 0;

            for (int k = 0; k < 4; k++)    //İki matrisin
çarpılması için 1. matrisin sütun sayısı ile 2. matrisin satır sayısının
eşit olması gerektiğinden dolayı eşit olan k sayısı döndürülür.
            {
                carpim[i][j] = carpim[i][j] + (x[i][k] * y[k][j]);
            }
            cout << "\t" << carpim[i][j];
        }
        cout << endl;
    }
    cout << endl;
}

void matrisfarkalma()    // Fark hesaplama fonksiyonu
{

```

```

        matrisyaz(); ::x[4][4], y[4][4];
        int fark[4][4]; //Matrislerin farkı için oluşturulan
    matris
        cout << endl;
        for (int i = 0; i < 4; i++) //Satırlar için döngü
        {
            for (int j = 0; j < 4; j++) //Sütunlar için
                döngü
                {
                    fark[i][j] = x[i][j] - y[i][j];
                    cout << "\t" << fark[i][j];
                }
            cout << endl;
        }
        cout << endl;
    }
}

```

```

void transpoz() //Transpoz hesaplama fonksiyonu
{
    matrisyaz(); ::x[4][4], y[4][4];
    int transpoz_x[4][4]; //Matrisin transpozu için
    oluşturulan matrisler
    int transpoz_y[4][4];
    cout << endl;
    cout << "X Matrisi " << endl << endl;
    for (int i = 0; i < 4; i++) //Satırlar için döngü
    {
        for (int j = 0; j < 4; j++) //Sütunlar için
            döngü
            {
                transpoz_x[i][j] = x[j][i];
                cout << "\t" << transpoz_x[i][j];
            }
        cout << endl;
    }
    cout << endl;

    cout << "Y Matrisi " << endl << endl;
    for (int i = 0; i < 4; i++) //Satırlar için
        döngü
        {
            for (int j = 0; j < 4; j++) //Sütunlar için
                döngü
                {
                    transpoz_y[i][j] = y[j][i]; //Transpoz
                    tanımlaması yapıldı.
                    cout << "\t" << transpoz_y[i][j];
                }
            cout << endl;
        }
        cout << endl;
    }
}

```

```

void matrisinİzi() // İz hesaplama fonksiyonu
{
    matrisyaz(); ::x[4][4], y[4][4];
    int matrisiz_x = 0;
    int matrisiz_y = 0;

    cout << endl;

    matrisiz_x = x[1][1] + x[2][2] + x[3][3] + x[4][4];    //İz
    köşegenlerin toplamı olduğu için köşegenler toplanır.
    cout << "\tX Matrisinin İzi: " << matrisiz_x;

    cout << endl;
    cout << endl;

    matrisiz_y = y[1][1] + y[2][2] + y[3][3] + y[4][4];
    cout << "\tY Matrisinin İzi: " << matrisiz_y;

    cout << endl;
    cout << endl;
}

void matrisintersi() //Matrisin tersini hesaplama fonksiyonu
{
    matrisyaz(); ::x[4][4];
#define N 4

    void getCfactor(int x[4][4], int t[N][N], int p, int q, int n);
    {
        int i = 0, j = 0;
        for (int r = 0; r < n; r++) {
            for (int c = 0; c < n; c++)
                if (r != p && c != q) {
                    t[i][j++] = x[r][c];
                    if (j == n - 1) {
                        j = 0; i++;
                    }
                }
        }
    }

    int DET(int x[4][4], int n)
    {
        int D = 0;          //Determinant hesabı yapılır.
        if (n == 1)
            return x[0][0];
        int t[N][N];
        int s = 1;

        for (int f = 0; f < n; f++) {
            s = -s;

```

```

    }
    return D;
}

void ADJ(int x[4][4], int adj[N][N]) //Kofaktör hesaplama
fonksiyonu
{
    if (N == 1) { // Kofaktör kontrol ediliyor.
        adj[0][0] = 1; return;
    }
    int s = 1, t[N][N];

    for (int i = 0; i < N; i++) {
        for (int j = 0; j < N; j++) {

            getCfactor(x, t, i, j, N); //Kofaktör hesabı
yapılıyor.

            s = ((i + j) % 2 == 0) ? 1 : -1; // Satır sütun
toplamı çift sayı ise kofaktörün işareti pozitifdir
            adj[j][i] = (s) * (DET(t, N - 1)); // Satırlar
sütunlar yer değiştirir.
        }
    }
    bool INV(int x[4][4], float inv[N][N]) {
        int det = DET(x, N);
        if (det == 0) { //Determinat 0 olursa tersi
hesaplanamayacağından dolayı kontrol ediliyor.
            cout << "Tersi bulunamadı.";
            return false;
        }
        int adj[N][N]; ADJ(x, adj);
        for (int i = 0; i < N; i++) for (int j = 0; j < N; j++)
inv[i][j] = adj[i][j] / float(det); //Determinant formülünün
uygulanması.
        return true;
    }

    template<class T> void print(T A[N][N]) {
        for (int i = 0; i < N; i++) { for (int j = 0; j < N; j++) cout
<< A[i][j] << " "; cout << endl; }
    }
    int main() {
        int x[4][4];
        float inv[N][N];
        cout << "\nMatrisin Tersisi :\n"; if (INV(x, inv)) print(inv);
        return 0;
    }

}

int main()
{
    setlocale(LC_ALL, "Turkish"); //Türkçe karakterler için bir
fonksiyon

```

```

cout << "\t M E N Ü " << endl;
cout << endl;
cout << " 1- Matrisi Yazdır" << endl;
cout << endl;
cout << " 2- Matris Toplama" << endl;
cout << endl;
cout << " 3- Matris Farkı Alma" << endl;
cout << endl;
cout << " 4- Matrisi Çarpma" << endl;
cout << endl;
cout << " 5- Matrisin İzini Bulma" << endl;
cout << endl;
cout << " 6- Matris Transpozmesini Alma" << endl;
cout << endl;
cout << " 7- Matris Tersini Alma" << endl;
cout << endl;
cout << " 8- Çıkış" << endl;
cout << endl;
cout << " Seçiminizi yazdıktan sonra enter tuşuna basınız: ";
cin >> secim;

```

```

if (secim == 1)
{
    matrisyaz();
}
else if (secim == 2)
{
    matristoplama();
}
else if (secim == 3)
{
    matrisfarkalma();
}
else if (secim == 4)
{
    matriscarpma();
}
else if (secim == 5)
{
    matrisinizi();
}
else if (secim == 6)
{
    transpoz();
}

else if (secim == 7)
{
    matrisintersi();
}
else if (secim == 8)
{
}

```

```

}

```