Matris İşlemleri Yapan Program Dokümantasyonu

1) Amaç

Bu programın amacı en büyük 4x4 matrislerde toplama, çarpma, determinant bulma ve tersini alma işlemlerini yapmaktır.

2) Eklenen Kütüphaneler

<stdio.h>

<math.h>

“vmatris.h”

3) Kullanılan Fonksiyonlar

- `int control1(int x);`

```c
int control1(int x){

    while(1){
        if(x<1 || x>4){
            printf("\nGecersiz sayi\n");
            printf("Satir veya sutun sayisi 1den kucuk, 4ten buyuk olamaz!\n");
            printf("Satir sayisini girin: ");
            scanf("%d", &x);
        }
        else break;
    }

    return x;
}
```

Girilen matris boyutunun 1’den büyük 4’ten küçük olma durumunu kontrol eder.

- `int defineZero(int row, int column, int arr[row][column]);`

```c
int defineZero(int row, int column, int arr[row][column]){

    for(int i=0; i<row; i++){
        for(int j=0; j<column; j++){
            arr[i][j]=0;
        }
    }

    return arr[row][column];
}
```

Satır, sütun ve matris bilgisi verilir. Girilen matrisi 0 matrisine dönüştürür.

- `int defineArr(int row, int column, int arr[row][column]);`

```c
int defineArr(int row, int column, int arr[row][column]){

    for(int i=0; i<row; i++){
        for(int j=0; j<column; j++){
            printf("\nLutfen su elemani giriniz [%d][%d]: ", i+1, j+1);
            scanf("%d", &arr[i][j]);
        }
    }

    return arr[row][column];
}
```

Matris boyutu ve matris bilgisi verilir. Girilen matrisin elemanlarını kullanıcıdan alır.

- `float deineArrFloat(int row, float arr[row][row]);`

```c
float deineArrFloat(int row, float arr[row][row]){

    for(int i=0; i<row; i++){
        for(int j=0; j<row; j++){
            printf("\nLutfen su elemani giriniz [%d][%d]: ", i+1, j+1);
            scanf("%f", &arr[i][j]);
        }
    }

    return arr[row][row];
}
```

Matris boyutu ve matris bilgisi verilir. Girilen (float) matrisin elemanlarını kullanıcıdan alır.Matrisin tersini bulma işlemi için eklenmiştir.

- `void printArr(int row, int column, int arr[row][column]);`

```c
void printArr(int row, int column, int arr[row][column]){

    for(int i=0; i<row; i++){
        printf("\n|");
        for(int j=0; j<column; j++){
            printf("%3d ", arr[i][j]);
        }
        printf("|\n");
    }
}
```

Satır, sütun ve matris bilgisi verilir. Verilen matrisi ekrana yazdırır.

- `void printArrFloat(int row, float arr[row][row]);`

```c
void printArrFloat(int row, float arr[row][row]){

    for(int i=0; i<row; i++){
        printf("\n|");
        for(int j=0; j<row; j++){
            printf("%5.2f ", arr[i][j]);
        }
        printf("|\n");
    }
}
```

Satır, sütun ve (float) matris bilgisi verilir. Verilen (float) matrisi ekrana yazdırır.Matrisin tersini bulma işlemi için eklenmiştir.

- void sumArr(int row, int column, int arr1[row][column], int arr2[row][column]);

```c
void sumArr(int row, int column, int arr1[row][column], int arr2[row][column]){

    int sonuc[row][column];
    defineZero(row, column, sonuc);

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

            sonuc[i][j]=arr1[i][j]+arr2[i][j];
        }
    }
    printf("\n\nSonuc:\n");
    printArr(row, column, sonuc);
}
```

Satır ve sütun sayıları aynı olan iki boyutlu iki matrisin elemanlarını toplar. Ve sonuc adlı farklı bir fonksiyona değerleri atar. İçerisinde bulunan printArr fonksiyonu ile sonuc matrisini yazdırır.

- void multiArr(int row, int column, int row2, int column2, int arr[row][column], int arr2[row2][column2]);

İki boyutlu arr, arr2 dizileri ve bunların boyutları (row ve column arr1'e ait row2 ve column2 arr2 dizisine ait olmak üzere) verilir.

```c
void multiArr(int row, int column, int row2, int column2, int arr[row][column], int arr2[row2][column2]){

    int sonuc[row][column2];
    defineZero(row, column2, sonuc);

    for(int i=0; i<row; i++){
        for(int j=0; j<column2; j++){
            for(int k=0; k<row2; k++){
                sonuc[i][j] += arr[i][k] * arr2[k][j];
            }
        }
    }

    printf("\n\nSonuc:\n");
    printArr(row, column2, sonuc);
}
```

1. For döngüsünde 1. dizinin satır sayısı, 2.for döngüsünde 2. dizinin sütun sayısı tutulur. En içteki for dögüsü ise 1. dizinin sütun veya 2. dizinin satır sayısı kadar döndürülür ve bu iki dizi en üçteki döngü

kadar çarpılır ve çarpımlar toplanarak iki boyutlu sonuc dizisine akrtarılır. Daha sonra ise sonuc dizisi printArr fonksiyonu ile ekrana bastırılır.

- int detOne(int row, int arr[row][row])

```
int detOne(int row, int arr[row][row]){

    int det=0;
    det=arr[0][0];

    return det;
}
```

2 boyutlu dizi ve boyutu verilen 1x1 matrisin determinantını hesaplar.

- int detTwo(int row, int arr[row][row])

```
int detTwo(int row, int arr[row][row]){

    int det=0;
    det=((arr[0][0]*arr[1][1])-(arr[0][1]*arr[1][0]));

    return det;
}
```

2 boyutlu dizi ve boyutu verilen 2x2 matrisin determinantını hesaplar.

- int detThree(int row, int arr[row][row])

```
int detThree(int row, int arr[row][row]){

    int det=0;
    det=(arr[0][0]*arr[1][1]*arr[2][2])+(arr[0][1]*arr[1][2]*arr[2][0])+(arr[0][2]*arr[1][0]*arr[2][1])-(arr[0][2]*arr[1][1]*arr[2]
        [0])-(arr[0][1]*arr[1][0]*arr[2][2])-(arr[0][0]*arr[1][2]*arr[2][1]);

    return det;
}
```

2 boyutlu dizi ve boyutu verilen 3x3 matrisin determinantını hesaplar. Sarrus yöntemi kullanılır.

- int detFour(int row, int arr[row][row])

2 boyutlu dizi ve boyutu verilen 4x4 matrisin determinantını hesaplar.

```
int detFour(int row, int arr[row][row]){

    int a, b;
    int det=0;
    int arr2[row][row];

    for(int i=0; i<row; i++){
        a=0;
        b=0;

        for(int j=1; j<row; j++){
            for(int k=0; k<row; k++){

                if(k==i){
                    continue;
                }

                arr2[a][b]=arr[j][k];
                b++;
                if(b==a-1){
                    a++;
                    b=0;
                }
            }
        }
        det += arr[0][i]*pow(-1, i)*detThree(3, arr2);
    }

    return det;
}
```

Tanımlanan a ve b değişkenleri, geçici dizinin (arr2) satır ve sütun sayısını belirtir. Her k == i durumunda o satır ve sütun görmezden gelinerek yeni iki boyutlu bir dizi oluşturulur (3x3 boyutlu arr2 matrisi). Her satır sonuna geldiğinde (b==a-1 durumu) alt satıra (a++) ve satırın başına (b==0) geçilir. Dizi boyutu kadar 3 boyutlu determinant hesaplama fonkiyonu çağırılarak kofaktör yöntemiyle 4x4 matrisin determinantı hesaplanır.

- **float** identityMatrix(**int** row, **float** arr[row][row])

```
179 float identityMatrix(int row, float arr[row][row]){
180
181     for(int i=0; i<row; i++){
182         for(int j=0; j<row; j++){
183             if(i==j){
184                 arr[i][j]=1;
185             }
186             else{
187             arr[i][j]=0;
188             }
189         }
190     }
191     return arr[row][row];
192 }
193
```

Boyutu ve verilen 2 boyutlu (float) diziyi birim matrise dönüştürmek için kullanılır.

- **void** inverse(**int** row, **float** arr[row][row], **float** iArr[row][row])

Boyutu, 2 boyutlu (float) dizi(arr[][]) ve aynı boyutta olan birim matris verilir (iArr).printArrFloat fonksiyonu yardımıyla girilen 2 boyutlu dizinin(matrisin) tersi bulma işlemi yapılır.

```
194
195 void inverse(int row, float arr[row][row], float iArr[row][row]){
196
197     float a,b;
198     for(int i=0; i<row; i++){
199         a=arr[i][i];
200             for(int j=0; j<row; j++){
201                 arr[i][j]=(arr[i][j])/a;
202                 iArr[i][j]=(iArr[i][j])/a;
203             }
204             for(int k=0; k<row; k++){
205                 if(k!=i){
206                     b=arr[k][i];
207                     for(int j=0; j<row; j++){
208                         arr[k][j]=arr[k][j]-(arr[i][j]*b);
209                         iArr[k][j]=arr[k][j]-(iArr[i][j]*b);
210                     }
211                 }
212             }
213         }
214     printf("\nGirilen matrisin tersi:\n");
215     printArrFloat(row, iArr);
216
217 }
218
```

Bu fonksiyon Gauss Jordan yötemiyle çalışır.İlk for dögüsünde satır, ikinci for dögüsünde ise sütun tutulur ve hem birim matris hem de birim matrisin [i][j] de bulunan elemanı matrisi satır eşelon formuna getirmek için önce kendinden bir önceki satırdaki elemana bölünür.(198-203)

Daha sonra matrisin kösegeninde olmayan elemanlar (k!=i) kontrolüyle sağlanır.Buradaki eleman orijinal matriste bulunan elemanla çarpılıp kendisinde çıkarılır.Orijinal matrisi satır eşelon formuna getirmek için yapılan tüm elementer işlemler, birim matrise de uygulanırsa oluşan yeni matris orijinal matrisin tersi olur.Orijinal matrisi satır eşelon forma getirmek için uygulanan işlemler birim matrise de uygulandığından birim matris orijinal matrisin tersi haline gelir.Yeni oluşan iArr matrisi printArrFloat fonksiyonu ile ekrana bastırılır.

4) Int main Yapısı

```
char exit = 'Y';

while(exit== 'y' || exit=='Y')
```

```
printf("\n\nMenuye donmek ister misiniz (Y/N): ");
getchar();
scanf("%s", &exit);
```

Bu while döngüsü ile programın sürekli döngü içerisinde kalması sağlanır.

```
printf("-----------------------------\n");
printf("|                           |\n");
printf("| 1 -> Toplama              |\n");
printf("| 2 -> Determinant Hesaplama |\n");
printf("| 3 -> Tersini Alma         |\n");
printf("| 4 -> Noktasal Carpim      |\n");
printf("|                           |\n");
printf("-----------------------------\n");
printf("\n\n");
printf("Matris islemleri yapma uygulamasina hosgeldiniz!\n");
printf("Lutfen yapmak istediginiz islemi secin: ");
scanf("%d", &gate);
while(1){
    if (gate<1 || gate>4){
        printf("Gecersiz secim!\nLutfen tekrar deneyin: ");
        scanf("%d", &gate);
    }
else break;
}
```

Menü ve kontorlü yukarıda gösterildiği üzere while(1) dögüsü ile sağlanmıştır.Matrisişlemleri 1, 2, 3, 4 ile girilebilen switch-case yapısına sokulmuştur.Bu switch case yapılarında farklı matris işlemleri yukarıda belirtilen fonksiyonlar yardımıyla yapılabilmektedir.

Case 1: Toplama işlemi

```
46
47          case 1://Toplama işlemi
48          printf("\nToplama islemi!\n\n");
49          printf("Lutfen satir sayisini girin: ");
50          scanf("%d", &row);
51          row=control1(row);
52          printf("Lutfen sutun sayisini girin: ");
53          scanf("%d", &column);
54          column=control1(column);
55
56
57          printf("\n1. matris icin:\n");
58          defineArr(row,column,arr1);
59          printArr(row,column,arr1);
60
61          printf("\n\n2. matris icin:\n");
62          defineArr(row,column,arr2);
63          printArr(row,column,arr2);
64
65          sumArr(row, column,arr1,arr2);
66
67          printf("\n\nMenuye donmek ister misiniz (Y/N): ");
68          getchar();
69          scanf("%s", &exit);
70          break;
71
72
```

Case 2: Determinant işlemi

```c
case 2://Determinant işlemi
printf("\nDeterminant Hesaplama!\n\n");
printf("Determinant sadece kare matrislerde bulunabilir\n");
printf("Lutfen satir ve sutun sayisini girin: ");
scanf("%d", &row);
row=control1(row);
defineArr(row, row, arr1);
printf("Girilen matris:\n");
printArr(row, row, arr1);
printf("\n\nMatrisin determinanti: ");

if(row==1){
    printf("%d", detOne(row, arr1));
}
if(row==2){
    printf("%d", detTwo(row, arr1));
}
if(row==3){
    printf("%d", detThree(row, arr1));
}
if(row==4){
    printf("%d", detFour(row, arr1));
}


printf("\n\nMenuye donmek ister misiniz (Y/N): ");
getchar();
scanf("%s", &exit);
break;
```

Case 3: Tersini Alma İşlemi

```c
case 3://Tersini alma işlemi
printf("\nTersini Alma!\n\n");
printf("Kare matrislerin determinanti alinabilir\n");
printf("Lutfen satir ve sutun sayisini girin: ");
scanf("%d", &row);
row=control1(row);
deineArrFloat(row,arrF);
printf("Girilen matris:\n");
printArrFloat(row,arrF);
printf("\n");
identityMatrix(row,ideArr);
inverse(row, arrF, ideArr);



printf("\n\nMenuye donmek ister misiniz (Y/N): ");
getchar();
scanf("%s", &exit);
break;
```

Case 4: Çarpma İşlemi

```c
case 4://Noktasal çarpım işlemi
printf("\nNoktasal Carpim!\n\n");
printf("1. Matrisin sutun sayisi ile 2. matrisin satir sayisi esit olmali\n\n");
printf("Lutfen 1. matris icin satir sayisini girin: ");
scanf("%d", &row);
printf("Lutfen 1. matris icin sutun sayisini girin: ");
scanf("%d", &column);

printf("Lutfen 2. matris icin satir sayisini girin: ");
scanf("%d", &row2);
printf("Lutfen 2. matris icin sutun sayisini girin: ");
scanf("%d", &column2);

while(1){
    if( column!=row2 || row<1 || row>4 || column<1 || column>4 || row2<1 || row2>4 || column2<1 || column2>4){
        printf("1. Matrisin sutun sayisi ile 2. matrisin satir sayisi esit olmali\n");
        printf("\nSatir ve sutun sayilari 1den kucuk 4ten buyuk olamaz!\n\n");
        printf("Lutfen 1. matris icin satir sayisini girin: ");
        scanf("%d", &row);
        printf("Lutfen 1. matris icin sutun sayisini girin: ");
        scanf("%d", &column);

        printf("Lutfen 2. matris icin satir sayisini girin: ");
        scanf("%d", &row2);
        printf("Lutfen 2. matris icin sutun sayisini girin: ");
        scanf("%d", &column2);
    }
    else break;
}

printf("\n1. matris icin:\n");
defineArr(row,column,arr1);
printArr(row,column,arr1);

printf("\n\n2. matris icin:\n");
defineArr(row2,column2,arr2);
printArr(row2,column2,arr2);

multiArr(row,column,row2,column2,arr1,arr2);

printf("\n\nMenuye donmek ister misiniz (Y/N): ");
getchar();
scanf("%s", &exit);
break;
```

5) Referanslar

https://bilgisayarkavramlari.com/2008/11/19/matrisin-tersinin-alinmasi-mantrix-inverse/

Hazırlayan

Hilmi Ege Kara

220229046

Kocaeli Üniversitesi Yazılım Mühendisliği Bölümü