

BLM1022 – SAYISAL ANALİZ

Dönem Ödevi Raporu

Ahmet Onur AKMAN – 16011059

A) Ödev kapsamında programlanan konular

Grafik	Basit İteras.	Regula False	M. İvers	Secant	Newton	Trapez
X	X	X	X	X	X	X
Simpson	Türev	Gauss Jordan	Jacobi	Langrange	Regresyon	
X	X	X	X	X	X	

B) Kontrol edilen program

Ödev kontrolü 22.04.2020 tarihinde, “Bir matrisin inversini bulma” programı üzerinden yapılmıştır.

Aşağıda kodu verilen “Bir matrisin inversini bulma” programı, bir matrisin boyutunu ve sırasıyla elemanlarını aldıktan sonra, bu matrisin sağ tarafına aynı boyutlu bir birim matris ekleme ve elde edilen $n \times (2 \cdot n)$ boyutlu matrisin sol $n \times n$ ’lik kısmını birim matrise çevirip, sağ tarafta orijinal matrisin inversini bırakma mantığı ile çalışır.

C) Program kodu

```
#include <stdio.h>
#include <stdlib.h>

/*
Matrisin inversi alinir.
Matris uzunlugu ve elemanlari istenir.

ornek input:
1 2 4
-1 -3 4
2 5 1

sonuc:
23 -18 -20
-9 7 8
-1 1 1
*/
void doldur(float***, int);
void invers(float***, int);

int main()
{
    int n,i,j;
    printf("\nLutfen matrisin kenar uzunlugunu veriniz...");
    scanf("%d",&n);

    float** matrix = (float**)malloc(n * sizeof(float*));
    for (i=0;i<n;i++)
    {
        matrix[i] = (float*)malloc(2*n * sizeof(float));
    }

    for (i=0;i<n;i++){
        for (j=0;j<n;j++){
            printf("\nLutfen (%d,%d) elemanini veriniz...",i,j);
            scanf("%f",&matrix[i][j]);
        }
    }
}
```

```

    }
}

doldur(&matrix,n);

printf("\n\n\nVERILEN MATRIS\n");
for (i=0;i<n;i++){
    for (j=0;j<n;j++){
        printf(" %.3f ",matrix[i][j]);
    }
    printf(" SATIR%d\n",i);
}
printf("\n\n\n");

invers(&matrix,n);

printf("\n\n\nINVERS\n");
for (i=0;i<n;i++){
    for (j=n;j<2*n;j++){
        printf(" %.3f ",matrix[i][j]);
    }
    printf(" SATIR%d\n",i);
}

return 0;
}

void doldur(float ***matrix, int n){
    int i,j;
    for (i=0;i<n;i++){
        for (j=n;j<2*n;j++){
            if ((j-n)==i){
                (*matrix)[i][j]=1;
            }
            else{
                (*matrix)[i][j]=0;
            }
        }
    }
}

void invers(float ***matrix, int n){
    int i,j,k;
    float base, a;
    for (i=0;i<n;i++){
        base=(*matrix)[i][i];
        printf("\nR%d -> R%d / %.3f",i,i,base);
        for (j=0;j<n*2;j++){
            (*matrix)[i][j]= (*matrix)[i][j]/base;
        }
        for (j=0;j<n;j++){
            if (j!=i){
                a = (*matrix)[j][i]*-1;
                printf("\nR%d -> R%d + (R%d*%.3f)",j,j,i,a);
                for(k=0;k<2*n;k++){
                    (*matrix)[j][k]=(*matrix)[j][k]+a*( (*matrix)[i][k]);
                }
            }
        }
        printf("\n");
        for (j=0;j<n;j++){
            for (k=0;k<2*n;k++){
                printf(" %.3f ",(*matrix)[j][k]);
            }
        }
        printf("\n");
    }
    printf("\n\n\n");
}
}

```

D) Alınan output

Aşağıda programın bir kere çalıştırılması sonucu alınan çıktılar verilmiştir.

Lutfen matrisin kenar uzunlugunu veriniz...3

Lutfen (0,0) elemanini veriniz...1

Lutfen (0,1) elemanini veriniz...2

Lutfen (0,2) elemanini veriniz...4

Lutfen (1,0) elemanini veriniz...-1

Lutfen (1,1) elemanini veriniz...-3

Lutfen (1,2) elemanini veriniz...4

Lutfen (2,0) elemanini veriniz...2

Lutfen (2,1) elemanini veriniz...5

Lutfen (2,2) elemanini veriniz...1

VERILEN MATRIS

1.000 2.000 4.000 SATIR0
-1.000 -3.000 4.000 SATIR1
2.000 5.000 1.000 SATIR2

R0 -> R0 / 1.000

R1 -> R1 + (R0*1.000)

R2 -> R2 + (R0*-2.000)

1.000 2.000 4.000 1.000 0.000 0.000
0.000 -1.000 8.000 1.000 1.000 0.000
0.000 1.000 -7.000 -2.000 0.000 1.000

R1 -> R1 / -1.000

R0 -> R0 + (R1*-2.000)

R2 -> R2 + (R1*-1.000)

1.000 0.000 20.000 3.000 2.000 0.000
-0.000 1.000 -8.000 -1.000 -1.000 -0.000
0.000 0.000 1.000 -1.000 1.000 1.000

R2 -> R2 / 1.000

R0 -> R0 + (R2*-20.000)

R1 -> R1 + (R2*8.000)

1.000 0.000 0.000 23.000 -18.000 -20.000
0.000 1.000 0.000 -9.000 7.000 8.000
0.000 0.000 1.000 -1.000 1.000 1.000

INVERS

23.000 -18.000 -20.000 SATIR0
-9.000 7.000 8.000 SATIR1
-1.000 1.000 1.000 SATIR2

Verilen input: ((1,2,4),(-1,-3,4),(2,5,1))

Alınan output: ((23,-18,-20),(-9,7,8),(-1,1,1))