```
#include <stdlib.h>

#include <stdio.h>


int main()

{

int Allocated[10][10], Max[10][10], Need[10][10], Available[10], Completed[10], Safe_Sequence[10];

int p, r, i, j, process, count;

count = 0;


printf("Enter the number of processes : ");

scanf("%d", &p);

/*proses sayısı ve  resource sayısın ı kullanıcıdan aldım örnekdeki sayıları girdiğimizde kod
çalışmaktadır resimler eklenmiştir.*/


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

        {

                Completed[i] = 0;

        }

printf("\n\nEnter the number of different type resources( if same type enter one ) : ");

scanf("%d", &r);

/* iç içe for döngüyle matris yoluyl allocation ve max örnekde verilmişti onları doldurduk*/

printf("\n\nEnter the allocation for each process : ");

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

{

        printf("\nFor process %d : ",i + 1);

        for(j = 0; j < r; j++)

                {

                        scanf("%d", &Allocated[i][j]);
```

```
                }
}

printf("\n\nEnter the Max Matrix for each process : ");
for(i = 0; i < p; i++)
{
        printf("\nFor process %d : ", i + 1);
        for(j = 0; j < r; j++)
                {
                        scanf("%d", &Max[i][j]);
                }
}

printf("\n\nEnter the Available Resources : ");
for(i = 0; i < r; i++)
                {
                scanf("%d", &Available[i]);
                }
/*bu döngüde need=max- allocation yoluyla needmatrisi dolduruldu ve aşağıdaki fonksiyonda bastırıldı*/
for(i = 0; i < p; i++)
        {
                for(j = 0; j < r; j++)
                        {
                                Need[i][j] = Max[i][j] - Allocated[i][j];
                        }
        }
system("CLS");
```

```
do
{
        printf("\n Max Matrix:\tAllocation Matrix:\tNeed Matrix:\n");
        for(i = 0; i < p; i++)
        {
                for( j = 0; j < r; j++)
                        printf("%d  ", Max[i][j]);
                printf("\t\t");
                for( j = 0; j < r; j++)
                        printf("%d  ", Allocated[i][j]);
                printf("\t\t\t");
                for(j=0; j < r; j++)
                  printf("%d  ",  Need[i][j]);
                printf("\n");
        }

        process = -1;
```

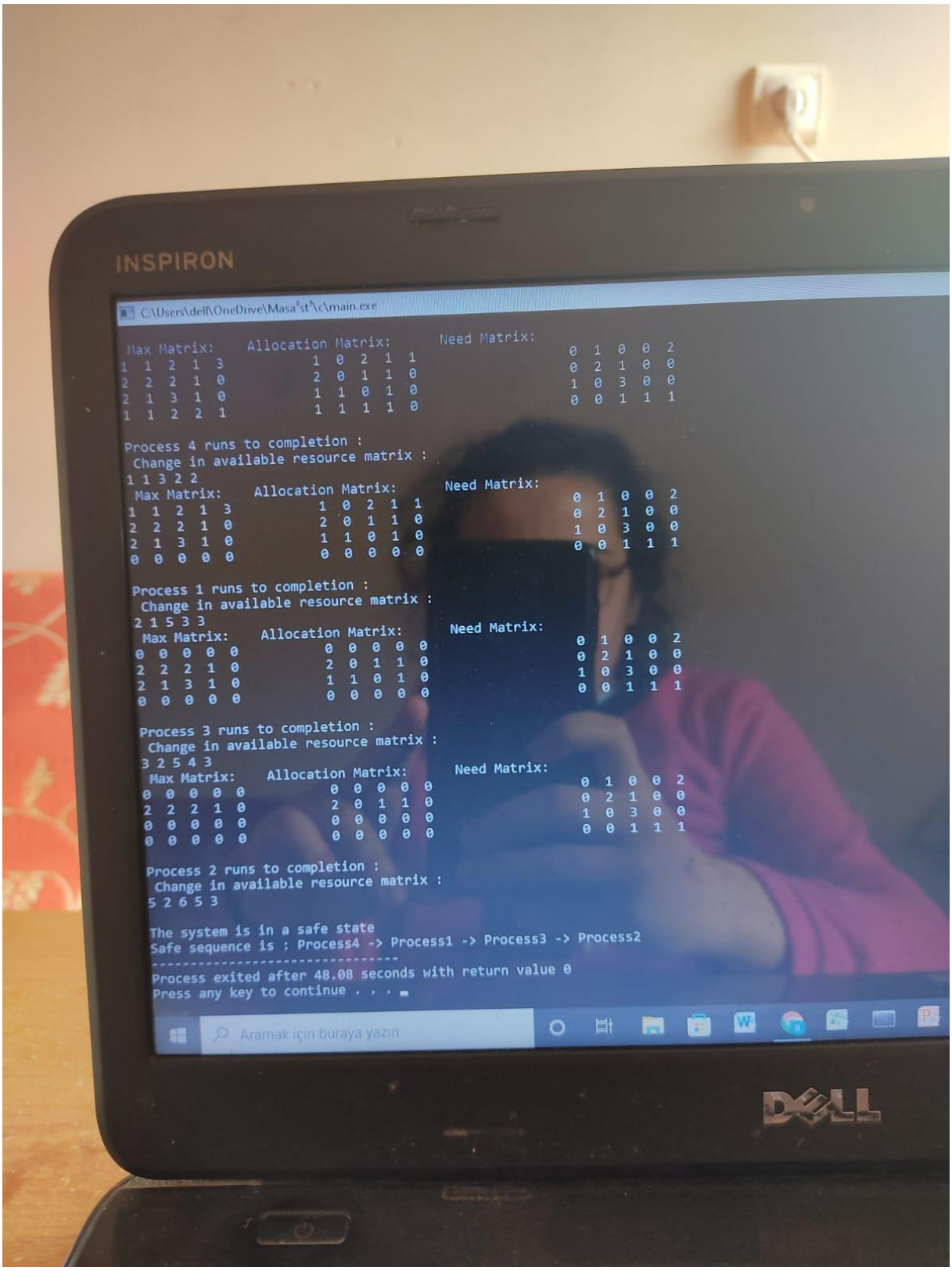
/* elimde bulunanlar available karışılıyormu kontrolü yapıldı karşılamıyorsa process -1 diyerek break komutuyla döngüden çıktık.karşılıyorsa available ile allocation toplayarak yeni available yaratıldı */

```
        for(i = 0; i < p; i++)
        {
                if(Completed[i] == 0)
                {
                        process = i ;
                        for(j = 0; j < r; j++)
                        {
                                if(Available[j] < Need[i][j])
```

```
                        {
                                process = -1;
                                break;
                        }
                }
        }
        if(process != -1)
                break;
}

if(process != -1)
{
        printf("\nProcess %d runs to completion : ", process + 1);
        Safe_Sequence[count] = process + 1;
        count++;
        printf("\n Change in available resource matrix :\n");
        for(j = 0; j < r; j++)
        {
                Available[j] += Allocated[process][j];
                Allocated[process][j] = 0;
                Max[process][j] = 0;
                Completed[process] = 1;


                printf("%d ", Available[j]);
        }
```

```
        }
}while(count != p && process != -1);


if(count == p)
{
        printf("\n\nThe system is in a safe state\n");
        printf("Safe sequence is : ");
        for( i = 0; i < p; i++)
                {       printf("Process%d ", Safe_Sequence[i]);
                        if(i<p-1)
                        {
                                printf("-> ");
                        }
                }
}
else
        {
                printf("\nThe system is in an unsafe state ");
        }
}
```

Sonuç olarak örnekde verilen çıktıya benzer şeyler yapmaya çalışıldı.sistemin güvenli durumda çalışabilmesi için boşta olan kaynak verilen sayılar girildiğinde ilk olarak available 4. Proses karşıladı available=available+allocation 11322 oldu ; sırasıyla 1. Proses karşılandı available 21533 ;proses 3 – 32543 ; Proses 2 -52653 kaynaklar proseslerin istedikleri sırayla çalışmaı için yeterlioldu sistem güvenli durumdadır ölümcül kilitlenme oluşmaz


INSPIRON
C:\Users\dell\OneDrive\Masa³st³\c\main.exe
Max Matrix:
1 1 2 1 3
2 2 2 1 0
2 1 3 1 0
1 1 2 2 1
Allocation Matrix:
1 0 2 1 1
2 0 1 1 0
1 1 0 1 0
1 1 1 1 0
Need Matrix:
0 1 0 0 2
0 2 1 0 0
1 0 3 0 0
0 0 1 1 1
Process 4 runs to completion :
Change in available resource matrix :
1 1 3 2 2
Max Matrix:
1 1 2 1 3
2 2 2 1 0
2 1 3 1 0
0 0 0 0 0
Allocation Matrix:
1 0 2 1 1
2 0 1 1 0
1 1 0 1 0
0 0 0 0 0
Need Matrix:
0 1 0 0 2
0 2 1 0 0
1 0 3 0 0
0 0 1 1 1
Process 1 runs to completion :
Change in available resource matrix :
2 1 5 3 3
Max Matrix:
0 0 0 0 0
2 2 2 1 0
2 1 3 1 0
0 0 0 0 0
Allocation Matrix:
0 0 0 0 0
2 0 1 1 0
1 1 0 1 0
0 0 0 0 0
Need Matrix:
0 1 0 0 2
0 2 1 0 0
1 0 3 0 0
0 0 1 1 1
Process 3 runs to completion :
Change in available resource matrix :
3 2 5 4 3
Max Matrix:
0 0 0 0 0
2 2 2 1 0
0 0 0 0 0
0 0 0 0 0
Allocation Matrix:
0 0 0 0 0
2 0 1 1 0
0 0 0 0 0
0 0 0 0 0
Need Matrix:
0 1 0 0 2
0 2 1 0 0
1 0 3 0 0
0 0 1 1 1
Process 2 runs to completion :
Change in available resource matrix :
5 2 6 5 3
The system is in a safe state
Safe sequence is : Process4 -> Process1 -> Process3 -> Process2
Process exited after 48.08 seconds with return value 0
Press any key to continue . . .
Aramak için buraya yazın
DELL