OS Lab Assignment no: 6

Bankers Algorithm

NITHIN JOSE
R4:46

CODE:-

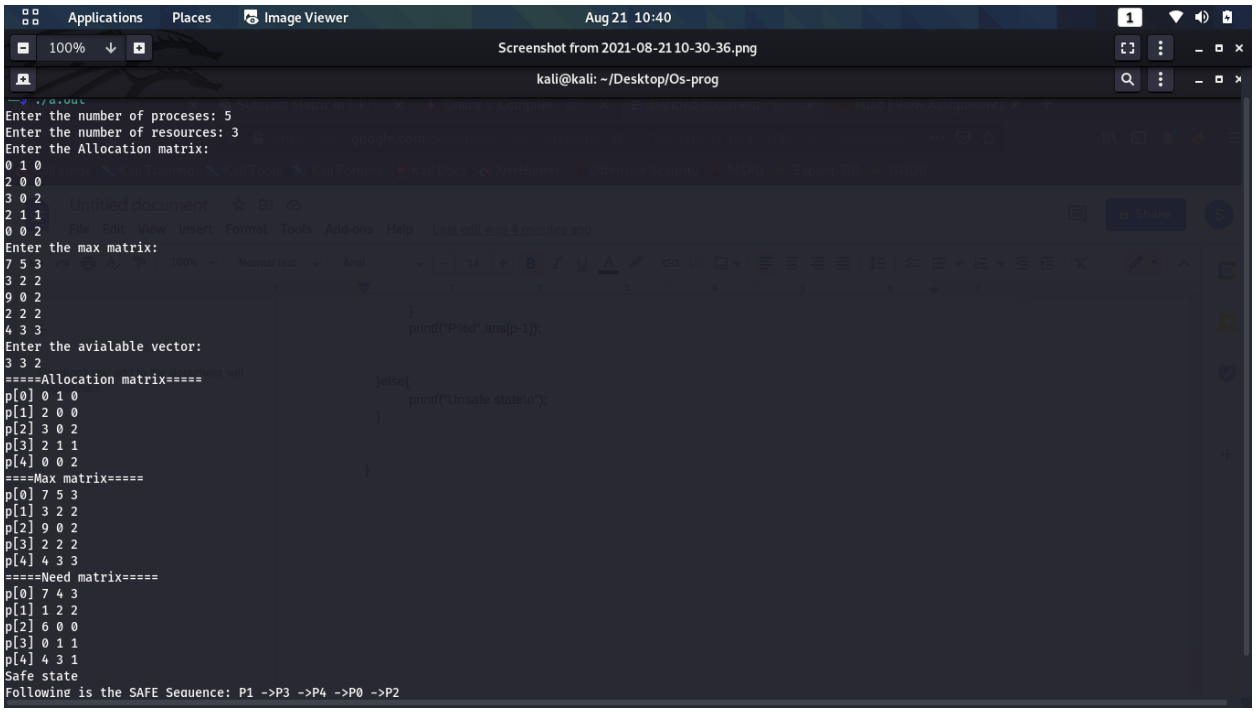
```
#include<stdio.h>
void main()
{
  int p,r,allocation[20][20],max[20][20],need[20][20],available[20],count = 0,f[20],ans[30],ind = 0;
  printf("Enter the number of proceses: ");
  scanf("%d",&p);
  printf("Enter the number of resources: ");
  scanf("%d",&r);
  printf("Enter the Allocation matrix:\n");
  for(int i=0;i<p;i++)
  {
          for(int j=0;j<r;j++)
          {
          scanf("%d",&allocation[i][j]);
          }
   }



  printf("Enter the max matrix:\n");

   for(int i=0;i<p;i++)
  {
          for(int j=0;j<r;j++)
          {
          scanf("%d",&max[i][j]);
          }
   }

   printf("Enter the avialable vector:\n");

   for(int i=0;i<r;i++)
   {
          scanf("%d",&available[i]);
   }
```

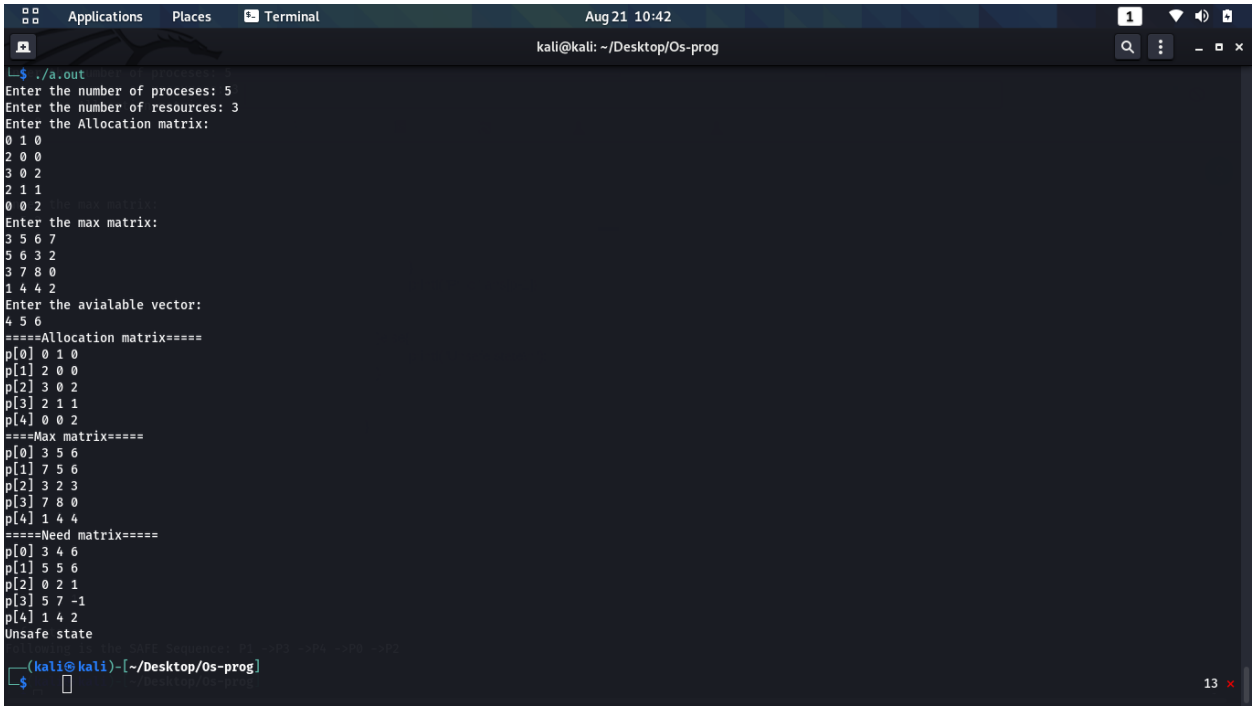
```
        for(int i=0;i<p;i++)
{
        for(int j=0;j<r;j++)
        {
        need[i][j]=max[i][j]-allocation[i][j];
        }
 }

        for (int k=0;k<p; k++)
 {
        f[k] = 0;
 }

        printf("=====Allocation matrix=====\n");
for(int i=0;i<p;i++)
{
        printf("p[%d] ",i);
        for(int j=0;j<r;j++)
        {
        printf("%d ",allocation[i][j]);
        }
        printf("\n");
 }

printf("====Max matrix=====\n");

 for(int i=0;i<p;i++)
{
        printf("p[%d] ",i);
        for(int j=0;j<r;j++)
        {
        printf("%d ",max[i][j]);
        }
        printf("\n");
 }

        printf("=====Need matrix=====\n");

        for(int i=0;i<p;i++)
{
        printf("p[%d] ",i);
        for(int j=0;j<r;j++)
        {
        printf("%d ",need[i][j]);
```

```
        }
        printf("\n");
 }



 for (int k = 0; k < 5; k++) {
        for (int i = 0; i < p; i++) {
        if (f[i] == 0) {

        int flag = 0;
        for (int j = 0; j < r; j++) {
        if (need[i][j] > available[j]){
        flag = 1;
        break;
        }
        }

        if (flag == 0) {
        ans[ind++] = i;
        for (int y = 0; y < r; y++)
        available[y] += allocation[i][y];
        f[i] = 1;
        }
        }
        }

}

        for (int k=0;k<p; k++)
 {
        if(f[k]==0)
        {
        count++;
        }

 }

 if(count == 0)
 {
        printf("Safe state\n");
        printf("Following is the SAFE Sequence");
for (int i = 0; i < p-1; i++)
        {
```

```
        printf("P%d ->",ans[i]);

        }
        printf("P%d",ans[p-1]);


  }else{
        printf("Unsafe state\n");
  }


}
```

OUTPUT:




Applications
Places
Terminal
Aug 21 10:42
kali@kali: ~/Desktop/Os-prog
└─$ ./a.out
Enter the number of proceses: 5
Enter the number of resources: 3
Enter the Allocation matrix:
0 1 0
2 0 0
3 0 2
2 1 1
0 0 2
Enter the max matrix:
3 5 6 7
5 6 3 2
3 7 8 0
1 4 4 2
Enter the avialable vector:
4 5 6
=====Allocation matrix=====
p[0] 0 1 0
p[1] 2 0 0
p[2] 3 0 2
p[3] 2 1 1
p[4] 0 0 2
====Max matrix=====
p[0] 3 5 6
p[1] 7 5 6
p[2] 3 2 3
p[3] 7 8 0
p[4] 1 4 4
=====Need matrix=====
p[0] 3 4 6
p[1] 5 5 6
p[2] 0 2 1
p[3] 5 7 -1
p[4] 1 4 2
Unsafe state
┌──(kali㉿kali)-[~/Desktop/Os-prog]
└─$
13