NAME : Pratiksha Thorat

ROLL NO :TCOB26

Deadlock Avoidance

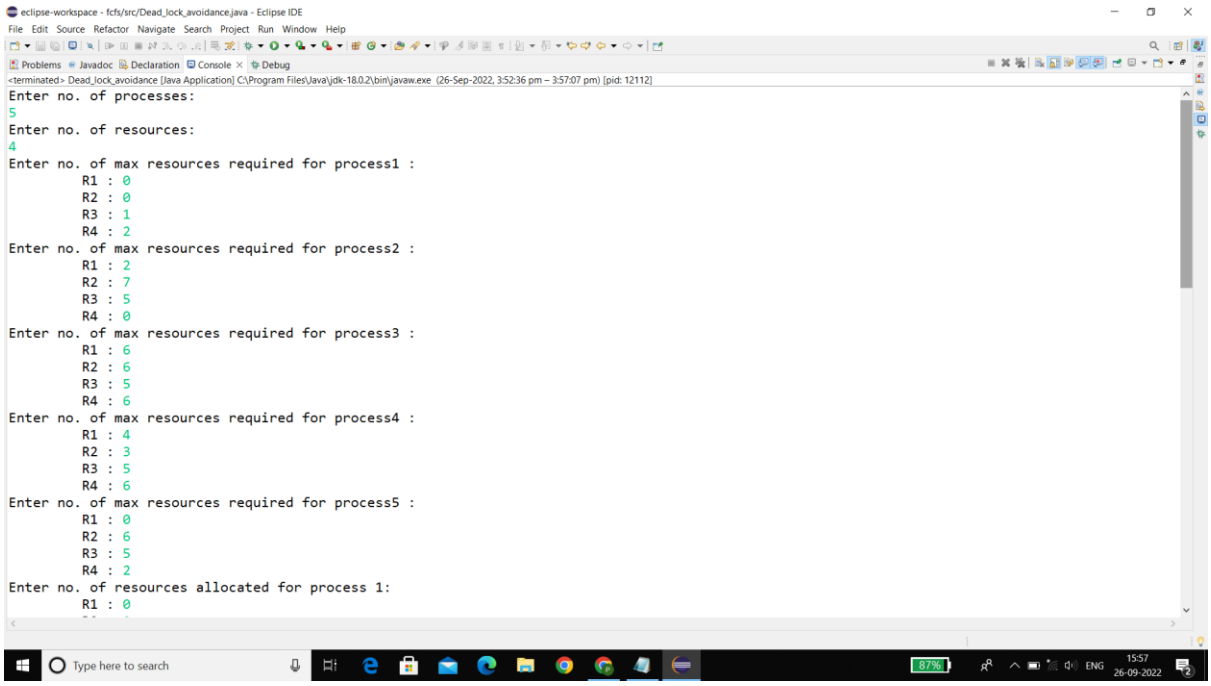
```
import java.util.*;
public class dead_lock_avoidance
{
	public static void main(String args[])
	{
		Scanner sc=new Scanner(System.in);
		System.out.println("Enter no. of processes: ");
		int m=sc.nextInt();
		System.out.println("Enter no. of resources: ");
		int n=sc.nextInt();
		int [][]max=new int[m][n];
		int [][]allocation=new int[m][n];
		int [][]need=new int[m][n];
		int [][]new_available=new int[1][n];
		int f[]=new int[m];
		Queue<String>  safe = new LinkedList<>();

		for(int i=0;i<m;i++)
		{
			f[i]=0;
		}
		for(int i=0;i<m;i++)
		{
			System.out.println("Enter no. of max resources required for process"+(i+1)+" : ");
			for(int j=0;j<n;j++)
			{
```

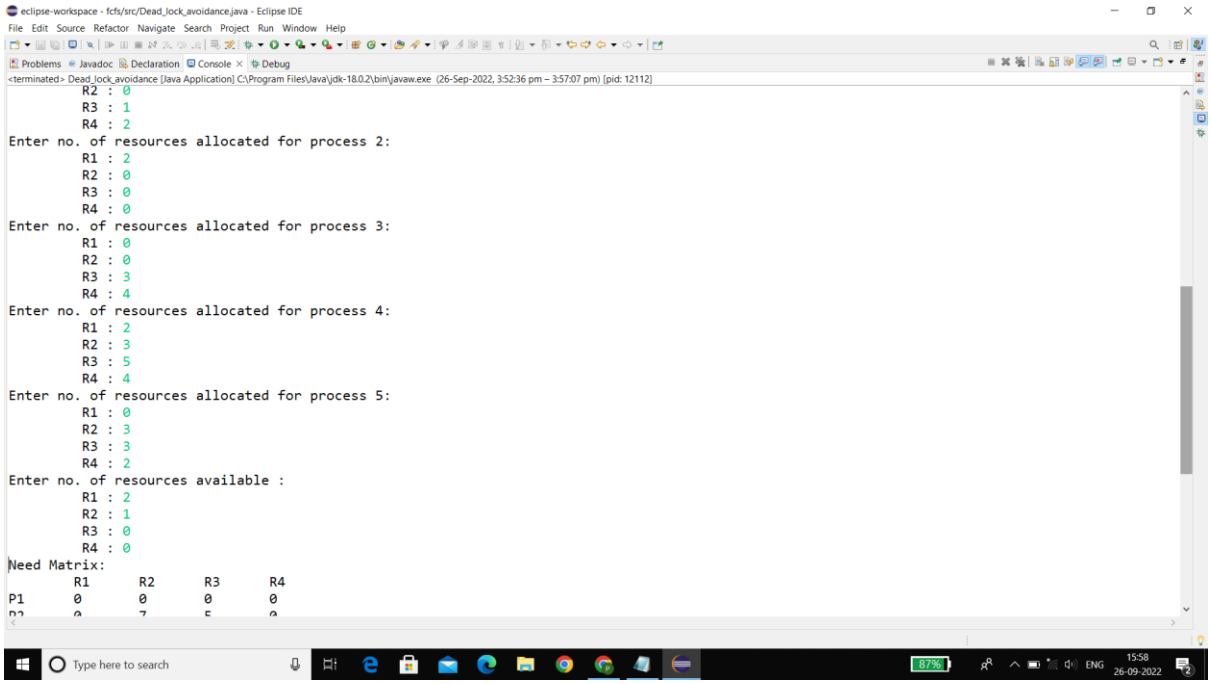
```
                    System.out.print("\t");
                    System.out.print(" R"+(j+1)+" : ");
                    max[i][j]=sc.nextInt();
                }
            }

            for(int i=0;i<m;i++)
            {
                System.out.println("Enter no. of resources allocated for process
"+(i+1)+":");
                for(int j=0;j<n;j++)
                {
                    System.out.print("\t");
                    System.out.print(" R"+(j+1)+" : ");
                    allocation[i][j]=sc.nextInt();
                }
            }

            for(int i=0;i<1;i++)
            {
                System.out.println("Enter no. of resources available :");
                for(int j=0;j<n;j++)
                {
                    System.out.print("\t");
                    System.out.print(" R"+(j+1)+" : ");
                    new_available[i][j]=sc.nextInt();
                }
            }

            for(int i=0;i<m;i++)
            {
```

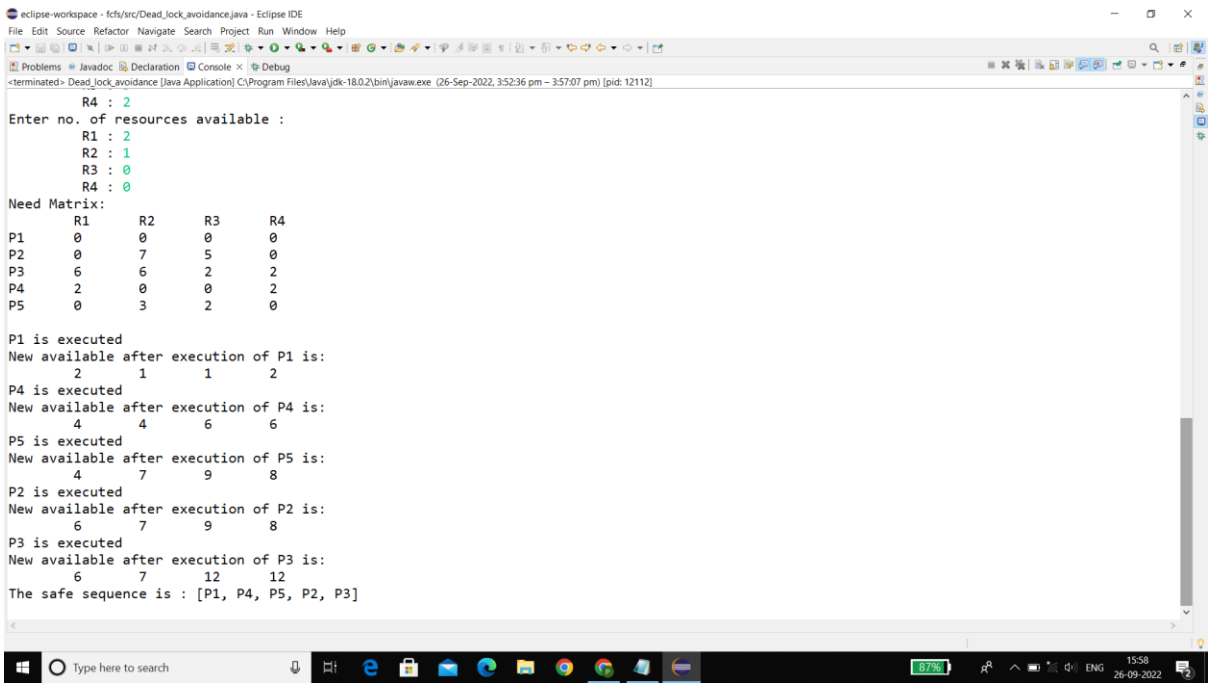
```
        for(int j=0;j<n;j++)
        {
                need[i][j]=max[i][j]-allocation[i][j];
        }
}

System.out.println("Need Matrix:");
for(int i=0;i<n;i++)
{
        System.out.print("\tR"+(i+1));
}
System.out.println();
for(int i=0;i<m;i++)
{
        System.out.print("P"+(i+1));
        for(int j=0;j<n;j++)
        {
                System.out.print("\t"+need[i][j]);
        }
        System.out.println();
}
System.out.println();

while(true)
{
        for(int i=0;i<m;i++)
        {
                int flag=0;
                for(int j=0;j<n;j++)
                {
```

```
                    flag=0;
                    if(f[i]==0 && need[i][j]<=new_available[0][j])
                    {
                        flag=1;
                    }
                    else
                    {
                        flag=0;
                        break;
                    }
                }

                if(flag==1)
                {
                    safe.add("P"+(i+1));
                    System.out.println("P"+(i+1)+" is executed");
                    System.out.println("New available after execution of
P"+(i+1)+" is:");
                    for(int j=0;j<n;j++)
                    {
                        {

    new_available[0][j]=new_available[0][j]+allocation[i][j];
                                f[i]=1;

    System.out.print("\t"+new_available[0][j]);
                        }
                    }
                    System.out.println();
                }
            }
```

```
                int flag_f=0;
                for(int k=0;k<n;k++)
                {
                        flag_f=0;
                        if(f[k]==1)
                        {
                                flag_f=1;
                        }
                        else
                                break;
                }
        if(flag_f==1)
                        break;
        }
        System.out.println("The safe sequence is : "+safe);
        sc.close();
    }
}
```




eclipse-workspace - fcfs/src/Dead_lock_avoidance.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Problems Javadoc Declaration Console Debug
<terminated> Dead_lock_avoidance [Java Application] C:\Program Files\Java\jdk-18.0.2\bin\javaw.exe (26-Sep-2022, 3:52:36 pm – 3:57:07 pm) [pid: 12112]
R2 : 0
R3 : 1
R4 : 2
Enter no. of resources allocated for process 2:
R1 : 2
R2 : 0
R3 : 0
R4 : 0
Enter no. of resources allocated for process 3:
R1 : 0
R2 : 0
R3 : 3
R4 : 4
Enter no. of resources allocated for process 4:
R1 : 2
R2 : 3
R3 : 5
R4 : 4
Enter no. of resources allocated for process 5:
R1 : 0
R2 : 3
R3 : 3
R4 : 2
Enter no. of resources available :
R1 : 2
R2 : 1
R3 : 0
R4 : 0
Need Matrix:
R1 R2 R3 R4
P1 0 0 0 0
Type here to search
87%
ENG
15:58
26-09-2022


eclipse-workspace - fcfs/src/Dead_lock_avoidance.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Problems Javadoc Declaration Console Debug
<terminated> Dead_lock_avoidance [Java Application] C:\Program Files\Java\jdk-18.0.2\bin\javaw.exe (26-Sep-2022, 3:52:36 pm – 3:57:07 pm) [pid: 12112]
R4 : 2
Enter no. of resources available :
R1 : 2
R2 : 1
R3 : 0
R4 : 0
Need Matrix:
R1 R2 R3 R4
P1 0 0 0 0
P2 0 7 5 0
P3 6 6 2 2
P4 2 0 0 2
P5 0 3 2 0
P1 is executed
New available after execution of P1 is:
2 1 1 2
P4 is executed
New available after execution of P4 is:
4 4 6 6
P5 is executed
New available after execution of P5 is:
4 7 9 8
P2 is executed
New available after execution of P2 is:
6 7 9 8
P3 is executed
New available after execution of P3 is:
6 7 12 12
The safe sequence is : [P1, P4, P5, P2, P3]
Type here to search
87%
ENG
15:58
26-09-2022