Student Name: Aadith K Suresh
Student ID : 11714432
Email Address: aadithk200@gmail.com
Given Questions : 5

. Reena's operating system uses an algorithm for deadlock avoidance to manage the allocation of resources say three namely A, B, and C to three processes P0, P1, and P2. Consider the following scenario as reference .user must enter the current state of system as given in this example :
Suppose P0 has 0,0,1 instances , P1 is having 3,2,0 instances and P2 occupies 2,1,1 instances of A,B,C resource respectively.

Also the maximum number of instances required for P0 is 8,4,3 and for p1 is 6,2,0 and finally for P2 there are 3,3,3 instances of resources A,B,C respectively. There are 3 instances of resource A, 2 instances of resource B and 2 instances of resource C available. Write a program to check whether Reena's operating system is in a safe state or not in the following independent requests for additional resources in the

current state:

1. Request1: P0 requests 0 instances of A and 0 instances of B and 2 instances of C.
2. Request2: P1 requests for 2 instances of A, 0 instances of B and 0 instances of C.

All the request must be given by user as input.

Code:

Code for question 1:

```
#include<iostream>
#include<stdio.h>
using namespace std;
int main()
{
        int n;
        int r;
        int i,j,k;
        int nd[10][10],allocation[10][10],maximum[10][10];
        int available[10],p[10];
        printf("\nEnter the total number of process :");
        scanf("%d",&n);
        printf("\n Enter the total number ofresources available : ");
        scanf("%d",&r);
        printf("\nEnter the number instances for resources :\n");
        for(i=0;i<r;i++)
        {
          printf("R%d ",i+1);
          scanf("%d",&available[i]);
        }
         printf("\n Enter allocationation matrix  \n");
         for(i=0;i<n;i++)
         {
         printf("p%d",i+1);         p[i]=0;
         for(j=0;j<r;j++)
```

```
              {
              scanf("%d",&allocation[i][j]);
              }
     }
      printf("\n Enter the maximum matrix  \n");

      for(i=0;i<n;i++)
      {
              printf("p%d",i+1);
              for(j=0;j<r;j++)
              {
              scanf("%d",&maximum[i][j]);
              }
      }

      for(i=0;i<n;i++)
      {
              printf("\np%d\t",i+1) ;
              for(j=0;j<r;j++)
              {
        nd[i][j]=maximum[i][j]-allocation[i][j];
        printf("\t%d",nd[i][j]);
        }
      }

              printf("\n\n");
              int fl=0;
              for(i=0;i<n;i++)
              {
                      for(j=0;j<r;j++)
                      {
```

```
            if(available[j]>=nd[i][j])
            fl=1;
            else
                fl=0;
        }
    }
    if(fl==0)
    printf("the system is in Unsafe State");
    else
    printf("the system is in Safe State");
}
```