Aim- Implementation of Banker’s algorithm.

Code-

**#include<stdio.h>**

**#include<stdlib.h>**

**int Allocation[5][3] = {{0,1,0},{2,0,0},{0,0,2},{2,1,1},{0,0,2}};**

**int request[5][3] = {{0,0,0},{2,0,2},{12,0,1},{1,0,0},{0,0,2}};**

**int available[3] = {3,3,2};**

**int isSafeState(int p){**

**int work[3];**

**for(int i=0;i<3;i++){**

**work[i]=available[i];**

**}**

**int count = 0;**

**int finished[5] = {0,0,0,0,0};**

**if(p!=-1){**

**finished[p] = 1;**

**for(int j=0;j<3;j++){**

**work[j]+=Allocation[p][j];**

**//printf("work = %d \n",work[j]);**

**}**

**count = 1;**

**}**

**while(1){**

**int curcount = count;**

**for(int i=0;(i<5) ;i++){**

**if((finished[i]!=1)){**

**int canBeAllocated = 1;**

**for(int j=0;j<3;j++){**

**if(work[j]<request[i][j]){**

**//printf("work = %d \n",work[j]);**

**canBeAllocated = 0;**

**break;**

**}**

**}**

**//printf("canbeallcoated = %d\n",canBeAllocated);**

**if(canBeAllocated==1){**

**for(int j=0;j<3;j++){**

**work[j]+=Allocation[i][j];**

**//printf("work = %d \n",work[j]);**

**}**

**finished[i]=1;**

**curcount++;**

**}**

**}**

**}**

**//printf("curcount = %d count = %d\n",curcount,count);**

**if(curcount==5){**

**return 1;**

**}**

**if(curcount==count){**

**return 0;**

**}**

**count = curcount;**

**}**

**return 0;**

**}**

**int main(){**

**if(isSafeState(2)==1){**

**printf("The system is not in deadlock\n");**

**}**

**else{**

**printf("The system is in deadlock state\n");**

**}**

**return 0;**

**}**

Output-

