**EXPERIMENT: 17** To Illustrate the deadlock avoidance concept by simulating Banker’s algorithm with C

**PROGRAM:**

#include <stdio.h>

int main() {

int n = 5, m = 3;

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

int max[5][3] = {{7,5,3},{3,2,2},{9,0,2},{2,2,2},{4,3,3}};

int avail[3] = {3,3,2};

int need[5][3], f[5]={0}, ans[5], ind=0;

for (int i=0;i<n;i++)

for (int j=0;j<m;j++)

need[i][j]=max[i][j]-alloc[i][j];

for (int k=0;k<n;k++) {

for (int i=0;i<n;i++) {

if (!f[i]) {

int flag=0;

for (int j=0;j<m;j++)

if (need[i][j]>avail[j]) { flag=1; break; }

if (!flag) {

ans[ind++]=i;

for (int y=0;y<m;y++) avail[y]+=alloc[i][y];

f[i]=1;

}

}

}

}

int safe=1;

for (int i=0;i<n;i++) if (!f[i]) safe=0;

if (safe) {

printf("Safe sequence: ");

for (int i=0;i<n;i++) printf("P%d ",ans[i]);

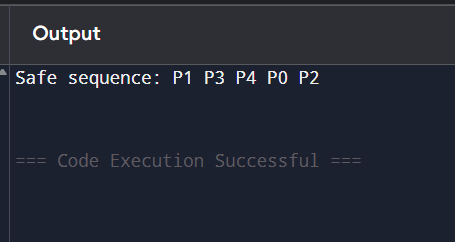
printf("\n");

} else printf("System is in DEADLOCK.\n");

return 0;

}

**OUTPUT:**

****