/\*

# PROGRAM TO IMPLEMENT BANKERS ALGORITHM

Ashis Solomon

CS4B 17

MDL20CS035

\*/

**CODE:**

#include<stdio.h>

int max[10][10];

int alloc[10][10];

int need[10][10];

int avail[10];

int n,r;

void input();

void show();

void cal();

void main(){

printf("BANKERS ALGORITHM\n\n");

input();

show();

cal();

return;

}

void input(){

int i,j;

printf("\nEnter the no. of processes : ");

scanf("%d",&n);

printf("\nEnter the no. of resource instances : ");

scanf("%d",&r);

printf("\nEnter the MAX Matrix :\n");

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

for(j=0;j<r;j++){

scanf("%d",&max[i][j]);

}

}

printf("\nEnter the ALLOCATION Matrix :\n");

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

for(j=0;j<r;j++){

scanf("%d",&alloc[i][j]);

}

}

printf("\nEnter the AVAILABLE RESOURCES :\n");

for(i=0;i<r;i++){

scanf("%d",&avail[i]);

}

}

void show() {

int i, j;

printf("PROCESS\t ALLOCATION\t MAX\t AVAILABLE\t");

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

printf("\nP%d\t ", i + 1);

for (j = 0; j < r; j++) {

printf("%d ", alloc[i][j]);

}

printf("\t\t");

for (j = 0; j < r; j++) {

printf("%d ", max[i][j]);

}

printf("\t\t");

if (i == 0) {

for (j = 0; j < r; j++)

printf("%d ", avail[j]);

}

}

}

void cal() {

int finish[10], temp, need[10][10], flag = 1, k, c1 = 0;

int safe[10];

int i, j;

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

finish[i] = 0;

}

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

for (j = 0; j < r; j++) {

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

}

}

printf("\n");

while (flag) {

flag = 0;

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

int c = 0;

for (j = 0; j < r; j++) {

if ((finish[i] == 0) && (need[i][j] <= avail[j])) {

c++;

if (c == r) {

for (k = 0; k < r; k++) {

avail[k] += alloc[i][j];

finish[i] = 1;

flag = 1;

}

printf("P%d->", i);

if (finish[i] == 1) {

i = n;

}

}

}

}

}

}

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

if (finish[i] == 1) {

c1++;

} else {

printf("P%d->", i);

}

}

if (c1 == n) {

printf("\n THE SYSTEM IS IN SAFE STATE");

}

else {

printf("\n PROCESS ARE IN DEAD LOCK");

printf("\n SYSTEM IS IN UNSAFE STATE");

}

}

**OUTPUT:**

