**Bankers algorithm**

#include <stdio.h>

int n, m, i, j, k,alloc[10][10],max[10][10],avail[10],ch,t,add[10];

void main()

{

printf("Enter the number of process:");

scanf("%d",&n);

printf("\nEnter the number of resources:");

scanf("%d",&m);

printf("\nEnter the allocation array");

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

{

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

{

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

}

}

printf("\nEnter the maximum available array");

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

{

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

{

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

}

}

printf("\nEnter the total available number of resources:");

for(i=0;i<m;i++)

{

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

}

printf("Is there any request from the process, if yes (1),no (0)");

scanf("%d",&ch);

if(ch==1)

{

printf("Enter the process number for which there is an additional request");

scanf("%d",&t);

printf("Enter the number of instances required for each resource");

for(i=0;i<m;i++)

{

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

}

for(i=0;i<m;i++)

{

alloc[t][i]+=add[i];

}

if(max[t][0]<alloc[t][0]||max[t][1]<alloc[t][1]||max[t][2]<alloc[t][2])

printf("It is not a valid request");

else

{

for(i=0;i<m;i++)

{

avail[i]-=add[i];

}

bankers();

}

}

else

bankers();

}

void bankers()

{

int f[n], ans[n], ind = 0;

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

{

f[k] = 0;

}

int need[n][m];

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

{

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

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

}

int y = 0;

for (k = 0; k < 5; k++)

{

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

{

if (f[i] == 0)

{

int flag = 0;

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

{

if (need[i][j] > avail[j])

{

flag = 1;

break;

}

}

if (flag == 0)

{

ans[ind++] = i;

for (y = 0; y < m; y++)

avail[y] += alloc[i][y];

f[i] = 1;

break;

}

}

}

}

int flag = 1;

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

{

if (f[i] == 0)

{

flag = 0;

printf("The following system is not safe");

break;

}

}

if (flag == 1)

{

printf("Following is the SAFE Sequence\n");

for (i = 0; i < n - 1; i++)

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

printf(" P%d", ans[n - 1]);

}

}

OUTPUT

