#include <stdio.h>

#include <stdlib.h>

#define MAX 10

int allocation[MAX][MAX];

int max[MAX][MAX];

int need[MAX][MAX];

int num\_processes, num\_resources;

int i,j,k,p;

void calculateNeed();

int isSafe();

int isDeadlocked();

int main()

{

printf("Enter number of processes: ");

scanf("%d", &num\_processes);

printf("Enter number of resources: ");

scanf("%d", &num\_resources);

printf("Enter allocation matrix:\n");

for (i = 0; i < num\_processes; i++)

{

for (j = 0; j < num\_resources; j++)

{

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

}

}

printf("Enter maximum demand matrix:\n");

for (i = 0; i < num\_processes; i++)

{

for (j = 0; j < num\_resources; j++)

{

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

}

}

calculateNeed();

if (isDeadlocked())

{

printf("Deadlock detected!\n");

}

else

{

printf("No deadlock detected.\n");

}

return 0;

}

void calculateNeed()

{

for (i = 0; i < num\_processes; i++)

{

for (j = 0; j < num\_resources; j++)

{

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

}

}

}

int isSafe()

{

int work[MAX], finish[MAX] = {0};

int count = 0;

for (i = 0; i < num\_resources; i++)

{

work[i] = 0;

for (j = 0; j < num\_processes; j++)

{

work[i] += allocation[j][i];

}

}

while (count < num\_processes)

{

int found = 0;

for (p = 0; p < num\_processes; p++)

{

if (!finish[p])

{

for (j = 0; j < num\_resources; j++)

{

if (need[p][j] > work[j])

{

break;

}

}

if (j == num\_resources)

{

for (k = 0; k < num\_resources; k++)

{

work[k] += allocation[p][k];

}

finish[p] = 1;

found = 1;

count++;

}

}

}

if (!found)

{

break;

}

}

return count == num\_processes;

}

int isDeadlocked()

{

return !isSafe();

}

'''

Enter number of processes: 5

Enter number of resources: 3

Enter allocation matrix:

0 1 0

2 0 0

3 0 2

2 1 1

0 0 2

Enter maximum demand matrix:

7 5 3

3 2 2

9 0 2

2 2 2

4 3 3

Deadlock detected!

'''