6. DEAD LOCK

#include<stdio.h>

int main()

{

int max[3][3] = {{3, 6, 8}, {4, 3, 3}, {3, 4, 4}};

int alloc[3][3] = {{3, 3, 3}, {2, 0, 3}, {1, 2, 4}};

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

int need[3][3], finish[3] = {0}, work[3], i, j, k, count = 0;

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

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

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

}

}

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

work[i] = avail[i];

}

while(count < 3) {

int found = 0;

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

if(finish[i] == 0) {

int safe = 1;

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

if(need[i][j] > work[j]) {

safe = 0;

break;

}

}

if(safe) {

for(k = 0; k < 3; k++) {

work[k] += alloc[i][k];

}

finish[i] = 1;

found = 1;

count++;

}

}

}

if(!found) {

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

printf("The deadlocked processes are: ");

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

if(finish[i] == 0) {

printf("%d ", i);

}

}

printf("\n");

return 0;

}

}

printf("The system is not in a deadlock state.\n");

return 0;

}

OUT PUT

