8. Write a C program to simulate deadlock detection

```
Code:
#include <stdio.h>
int main() {
  int n, m, all[10][10], req[10][10], ava[10], need[10][10];
  int i, j, k, flag[10], prev[10], c, count = 0;
  printf("Enter number of processes and number of resources required \n");
  scanf("%d %d", &n, &m);
  printf("Enter total number of required resources %d for each process\n", n);
  for (i = 0; i < n; i++)
     for (j = 0; j < m; j++)
        scanf("%d", &req[i][j]);
  printf("Enter number of allocated resources %d for each process\n", n);
  for (i = 0; i < n; i++)
     for (j = 0; j < m; j++)
        scanf("%d", &all[i][j]);
  printf("Enter number of available resources \n");
  for (i = 0; i < m; i++)
     scanf("%d", &ava[i]);
  for (i = 0; i < n; i++)
     for (j = 0; j < m; j++)
        need[i][j] = req[i][j] - all[i][j];
  for (i = 0; i < n; i++)
     flag[i] = 1;
  k = 1;
  while (k) {
     k = 0; // Reset the value of k for each iteration of the loop
     for (i = 0; i < n; i++) {
        if (flag[i]) {
          c = 0;
          for (j = 0; j < m; j++) {
             if (need[i][j] <= ava[j]) {
                C++;
             }
          }
```

```
if (c == m) {
              for (j = 0; j < m; j++) {
              for (j = 0; j < m; j++) {
                 ava[j] += all[i][j];
                 all[i][j] = 0;
              }
              flag[i] = 0;
              count++;
          }
        }
     }
     // Check if the current state is different from the previous state
     for (i = 0; i < n; i++) {
        if (flag[i] != prev[i]) {
           k = 1;
           break;
        }
     for (i = 0; i < n; i++) {
        prev[i] = flag[i];
     }
   }
   if (count == n) {
     printf("\nNo deadlock");
  } else {
     printf("\nDeadlock occurred \n");
  }
   return 0;
}
```

Output:

```
PS D:\VS Code\OS> cd "d:\VS Code\OS\" ; if ($?) { gcc bankersV2.c -o bankersV2 } ; if ($?) { .\bankersV2 }

Enter number of processes and number of resources required

3 3

Enter total number of required resources 3 for each process

7 5 3

3 2 2

9 0 2

Enter number of allocated resources 3 for each process

0 1 0

2 0 0

3 0 2

Enter number of available resources

1 2 3
```

Observation:

```
as write a congram to simulally deadlock defection.
#include & stolio. h.
        int of maining, regliothol, and (id), need fiotio), is, it, stapping, processing, man, authorized, regliothol, processing, and live of the processing of the
  from morring of
    printing Enter number of processes & number of resources in all
     sean + (autod doda, wo, bom);
      printer Enter total may pusourus (n");
        Our (1=0) 120; (+1)
               Par (3-03:20 ) 1++ )
       privale" Enter allocarbed m");
            for (1=03/cm3/++)
               Rox (j=0; j tar; j+r)
     Scanfillero d", Coallissis),
Minter Enter available resources in");
           for (1=0); 2m jitt dava(1);
                 for (j-0; j(m))++)
      for ( i = 0 ; ich ; i ++ )
         K-1, need (136) = reg (136) - all (136)
    white (K) of
                  Kaoj
Are (10) jenj (+1) (
                                             (20)
                                   Per (5-0) i (m) (it+) (1) 12 (need (1) 1) x.
                                                          C+a;
                                  Countary,
              for ( =0; | en; | 177) d.
                   i + ( + eag ( is ! = prev ( is ) ) d
   it ccount = = n)1
                             private "m No deadlock");
                          private ("InDecolork occurred");
output "
   the number of processes
   Enter stotal resources.
   $ 32 2 L
Entre accorded resources
           00
    302.
     mer available resources
 No Deadlock.
```