```
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, array[10], z=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) {
                 array[z++]=i;
              printf("Resouces can be allocated to Process:%d and available resources are: ", (i
+ 1));
              for (j = 0; j < m; j++) {
                 printf("%d ", ava[j]);
              }
              printf("\n");
              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];
     }
  }
  printf("\nNeed Matrix:\n");
  for (i = 0; i < n; i++) //printing need matrix
  {
     for (j = 0; j < m; j++)
        printf("%d ",need[i][j]);
     printf("\n");
  }
```

```
if (count == n) {
    printf("\nSystem is in safe mode \n<");
    for(i=0;i<n;i++)
    printf("P%d ",(array[i]+1));
    printf(">\n");
} else {
    printf("\nSystem is not in safe mode deadlock occurred \n");
}
return 0;
}
```

Output:

```
PS D:\US Code\Os\" ; if ($i) { gcc bankers\Os\ 2.c -o bankers\Os\ 2. } ; if ($i) { gcc bankers\Os\ 2.c -o bankers\Os\ 2. } ; if ($i) { sinkers\Os\ 2.c -o bankers\Os\ 2.c -o bankers\Os\
```

Observation:

```
to simulate Bankers algorithm for deadlock any
261+123.
 a) write a e program
 Hinclude (stolio-h)
           maines of maines of reactions of avaliand, need (10)(10), 1, 1, 1, 1, 1, 10, 10), promotion of the notion of the n
  int mainer of
          c, count=0, drivay(10), 2=0;
          printfluenter number of processes & number of resources required in).
          scant (" God God", Gn, Gm);
             printferenter total number of required and resources for each in a
             forci=0; icn; ite)
                       Br (j=0; jem; 3++)
                                 Scant ( " oped", Greg [i] (j]);
              Print+("Enter number of allocated resources and the each prousing, 1)
             for (1:0) icn : 1++)
                         Por (j=0; Jem; j++)
                              sconf (" olod", ball[i][j]);
             printf("Enter number of available resources in);
              for (1=0; 1 cm; 177)
                         Stan + (" old", dava (i)());
              for li=o; icn; ita)
                          for (j=0; j cm; j++)
                                      need(itis) = reg(itis) = all (itis);
             for (1=0; icu (1++)
                           #log[i]=1
            K= 1;
            while (K)
                          KaO;
                     Postiso; icn; i++) &
                              if (flag(i)) d
                                       (=b',
                                    for (5=0) j(m) j++)
                                          i + (need[i](i] (= ava[i])
                                                         cart ;
                                 : $ cc = = m ) 4
                                               array (2++)=1;
                                     print ("Resources on be alterated to proused gava ore 1/H)
                                     Por (100) 2 km; 3 ++ 70
                                                                                   (CD mus,
                               mint ( " m " );

for (5 = 0) 5 < m; 5 + 7) {

and (1) 1 + = all (1) (1) );

4 all (1) (1) = 0;
                      Alog(i)=0;
            y count ++;
```

```
Auci=0; (2n; 1+4) {
           it ( tray [i]! = prev(i]) 1
    for (1=0; icn; it+)
        prev (1) = / (by (1);
Alut 41" IN Need Matrix : In" );
 for (1=0); (n); ++1)
      for (5=0; 3 LM ) ++)
        orint(" vod", med (ilis);
'A (count = = n ) 1
      printin in System is in stage mode in " (");
      forti=0; icn; in+)
      minter pyod " (arroy (i)+1));
       minter yours.
    printe ("in system is not in safe mode deadlock occurredin");
Jeturn 0;
output ".
 Ender number of process & no. of resources 5 3
tutes to roll number of required resources
 753
 322
 901
 212
Enter allocated resources
 0 10
 200
 302
 211
 002
Enter number of available resources.
                                         32.
                                        & available me: 332
Resources can be allocated to Process: 2
                                        & overlook me: 5 32
             be all to from 4
                                            available are: 2 43
            be allocated to
                             gras: 5
                                        Ce
       Con
                                            available mo: 745
                allowed to press:1
                                        en available ani,755
            be
                allocated to proces: 3
Resones on be
Need matrin:
 7 43
  0 0
 4
```