Exp 18: Simulate the prediction of deadlock in operating system when all the processes announce their resource requirement in advance

Step 1: Read values for No of processes, No of Resources, Claim Matrix, Available Matrix, Resources vector
Step 2: Calculate no of resources allocated to each process
Step 3: Check whether the claim matrix is greater than the availability matrix
Step 4: if it is true NO
Allocation will be done
Else Resources will be
allocated
Step 5: Check each process which is causing deadlock
If it is found
Display deadlock causing process
Else
Requested Resources are allocated to process

Program Code:

```
#include <stdio.h>
int main()
{
   int found, flag, l, p[4][5], tp, e[4][5], i, j, k = 1, m[5], r[5], a[5], temp[5], sum = 0, tr,
c[4][5];

   printf("Enter total no of processes: ");
   scanf("%d", &tp);

   printf("Enter total no of resources: ");
   scanf("%d", &tr);

   printf("Enter claim matrix:\n");
   for(i = 0; i < tp; i++)
      for(j = 0; j < tr; j++)
      {
         scanf("%d", &c[i][j]);
      }

   printf("Enter allocation matrix:\n");
   for(i = 0; i < tp; i++)
      for(j = 0; j < tr; j++)
      {
         scanf("%d", &p[i][j]);
      }

   printf("Enter the resource vectors:\n");
   for(i = 0; i < tr; i++)
      scanf("%d", &r[i]);

   printf("Enter the availability matrix:\n");
   for(i = 0; i < tr; i++)
```

```
{
   scanf("%d", &a[i]);
   temp[i] = a[i];
}

for(i = 0; i < tp; i++)
{
   sum = 0;
   for(j = 0; j < tr; j++)
   {
      sum += p[i][j];
   }
   if(sum == 0)
   {
      m[k] = i;
      k++;
   }
}

for(i = 0; i < tp; i++)
{
   flag = 1;
   for(j = 0; j < tr; j++)
   {
      if(c[i][j] > temp[j])
      {
         flag = 0;
         break;
      }
   }
   if(flag == 1)
   {
      m[k] = i;
      k++;
      for(j = 0; j < tr; j++)
         temp[j] += p[i][j];
   }
}

printf("\nDeadlock causing processes are: ");
for(j = 0; j < tp; j++)
{
   found = 0;
   for(i = 1; i < k; i++)
   {
      if(j == m[i])
         found = 1;
   }
   if(found == 0)
      printf("%d ", j);
```

```
    }

    return 0;
}
```

Test Case 1:

```
Enter total no of processes: 3
Enter total no of resources: 3
Enter claim matrix:
7 5 3
3 2 2
9 0 2
Enter allocation matrix:
0 1 0
2 0 0
3 0 2
Enter the resource vectors:
3 3 2
Enter the availability matrix:
3 3 2

Deadlock causing processes are: 0 2
```