Write a C program to simulate Banker's algorithm for the purpose of deadlock avoidance.

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
   int N, M = 3, ind = 0;
   printf("\nEnter the number of processess: ");
   scanf("%d", &N);

   int alloc[N][M], max[N][M], need[N][M], finished[N], ans[N],avail[M];

   printf("\nEnter allocated resources\n");
   for (int i = 0; i < N; i++)
   {
      printf("For Process %d: ", i);
      for (int j = 0; j < M; j++)
      {
         scanf("%d", &alloc[i][j]);
      }
   }

   printf("\nEnter Maximum resources\n");
   for (int i = 0; i < N; i++)
   {
      printf("For Process %d: ", i);
      for (int j = 0; j < M; j++)
      {
         scanf("%d", &max[i][j]);
```

```
    }
}

printf("\nEnter available resources\n");
for (int i = 0; i < M; i++)
{
    scanf("%d",&avail[i]);
}

for (int i = 0; i < N; i++)
{
    finished[i] = 0;
}

for (int i = 0; i < N; i++)
{
    for (int j = 0; j < M; j++)
    {
        need[i][j] = max[i][j] - alloc[i][j];
    }
}

for (int k = 0; k < N; k++)
{
    for (int i = 0; i < N; i++)
    {
        if (finished[i] == 0)
        {
            int flag = 0;
            for (int j = 0; j < M; j++)
```

```
            {
                if (need[i][j] > avail[j])
                {
                    flag = 1;
                    break;
                }
            }

            if (flag == 0)
            {
                ans[ind++] = i;
                for (int p = 0; p < M; p++)
                {
                    avail[p] += alloc[i][p];
                }
                finished[i] = 1;
            }
        }
    }
}

int flag = 1;
for (int i = 0; i < N; i++)
{
    if (finished[i] == 0)
    {
        flag = 0;
        printf("The System is NOT safe\n");
        break;
    }
```

```
    }

    if (flag == 1)
    {
        printf("\nSafe Sequence:\n");
        for (int i = 0; i < N - 1; i++)
        {
            printf("P%d --> ", ans[i]);
        }
        printf("P%d\n", ans[N - 1]);
    }
}
```

OUTPUT:

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL
PS C:\Users\Admin\Desktop\Vignesh> gcc Bankers_Algorithm.c
PS C:\Users\Admin\Desktop\Vignesh> .\a.exe

Enter the number of processess: 5

Enter allocated resources
For Process 0: 0 1 0
For Process 1: 2 0 0
For Process 2: 3 0 2
For Process 3: 2 1 1
For Process 4: 0 0 2

Enter Maximum resources
For Process 0: 7 5 3
For Process 1: 3 2 2
For Process 2: 9 0 2
For Process 3: 2 2 2
For Process 4: 4 3 3

Enter available resources
3 3 2

Safe Sequence:
P1 --> P3 --> P4 --> P0 --> P2
PS C:\Users\Admin\Desktop\Vignesh>
```

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

PS C:\Users\Admin\Desktop\Vignesh> gcc Bankers_Algorithm.c
PS C:\Users\Admin\Desktop\Vignesh> .\a.exe

Enter the number of processess: 5

Enter allocated resources
For Process 0: 0 1 0
For Process 1: 3 0 2
For Process 2: 3 0 2
For Process 3: 2 1 1
For Process 4: 0 0 2

Enter Maximum resources
For Process 0: 7 5 3
For Process 1: 3 2 2
For Process 2: 9 0 2
For Process 3: 2 2 2
For Process 4: 4 3 3

Enter available resources
3 3 2

Safe Sequence:
P1 --> P2 --> P3 --> P4 --> P0
PS C:\Users\Admin\Desktop\Vignesh>
```