

Program :-

```
#include <stdio.h>
int main()
{
    int n, m, i, j, k, y, alloc[20][20], max[20][20], avail[50], ind=0;
    printf("Enter the no of Proceses:");
    scanf("%d",&n);
    printf("Enter the no of Resources:");
    scanf("%d",&m);
    printf("Enter the Allocation Matrix:");
    for (i = 0; i < n; i++)
    {
        for (j = 0; j < m; j++)
            scanf("%d",&alloc[i][j]);
    }
    printf("Enter the Max Matrix:");
    for (i = 0; i < n; i++)
    {
        for (j = 0; j < m; j++)
            scanf("%d",&max[i][j]);
    }
    printf("Enter the Available Matrix:");
    for(i=0;i<m;i++)
        scanf("%d",&avail[i]);

    int finish[n], safesequence[n], work[m], need[n][m];

    for (i = 0; i < n; i++)
    {
        for (j = 0; j < m; j++)
            need[i][j] = max[i][j] - alloc[i][j];
    }

    printf("NEED matrix is :");
    for (i = 0; i < n; i++)
    {
        printf("\n");
        for (j = 0; j < m; j++)
            printf(" %d ",need[i][j]);
    }
    for(i=0;i<m;i++)
    {
        work[i]=avail[i];
    }
    for (i = 0; i < n; i++)
    {
        finish[i] = 0;
    }

    for (k = 0; k < n; k++)
    {
        for (i = 0; i < n; i++)
        {
            if (finish[i] == 0)
            {
```

```

        int flag = 0;
        for (j = 0; j < m; j++)
        {
            if (need[i][j] > work[j])
            {
                flag = 1;
                break;
            }
        }
        if (flag == 0)
        {
            safesequence[ind++] = i;
            for (y = 0; y < m; y++)
                work[y] += alloc[i][y];
            finish[i] = 1;
        }
    }
}

printf("\nFollowing is the SAFE Sequence\n");
for (i = 0; i <= n - 1; i++)
    printf(" P%d ", safesequence[i]);
}

```

Output :-

Enter the no of Proseses:5
Enter the no of Resources:3
Enter the Allocation Matrix:0

1
0
2
0
0
3
0
2
2
1
1
0
0
2

Enter the Max Matrix:7

5
3
3
2
2
9
0
2
2
2
2
4
3
3

Enter the Available Matrix:3

3
2

NEED matrix is :

7	4	3
1	2	2
6	0	0
0	1	1
4	3	1

Following is the SAFE Sequence

P1 P3 P4 P0 P2