

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

```
int max[100][100];
int alloc[100][100];
int need[100][100];
int avail[100];
int n, r;
```

```
void INPUT();
void SHOW();
void CAL();
```

```
int main()
{
    INPUT();
    SHOW();
    CAL();
    return 0;
}
```

```
void INPUT()
{
    int i, j;
    printf("Enter the no of Processes: ");
    scanf("%d", &n);
    printf("Enter the no of resource instances: ");
    scanf("%d", &r);
    printf("Enter the Max Matrix:\n");
    for (i = 0; i < n; i++)
        for (j = 0; j < r; j++)
            scanf("%d", &max[i][j]);
    printf("\nEnter the Allocation Matrix:\n");
    for (i = 0; i < n; i++)
        for (j = 0; j < r; j++)
            scanf("%d", &alloc[i][j]);
    printf("\nEnter the available Resources:\n");
    for (j = 0; j < r; j++)
        scanf("%d", &avail[j]);
}
```

```
void SHOW()
{
    int i, j;
    printf("\nProcess\t Allocation\t Max\t\t Available");
    for (i = 0; i < n; i++)
    {
        printf("\nP%d\t ", i + 1);
        for (j = 0; j < r; j++)
            printf("%d ", alloc[i][j]);
        printf("\t\t");
        for (j = 0; j < r; j++)
            printf("%d ", max[i][j]);
        printf("\t\t");
        if (i == 0)
            for (j = 0; j < r; j++)
                printf("%d ", avail[j]);
    }
}
```

```

void CAL()
{
    int i,j, k;
    for (i = 0; i < n; i++)
    {
        for (j = 0; j < r; j++)
        {
            need[i][j] = max[i][j] - alloc[i][j];
        }
    }

    int finish[100] = {0};
    int flag = 1;

    while (flag)
    {
        flag = 0;
        for (i = 0; i < n; i++)
        {
            if (finish[i] == 0)
            {
                int c = 0;
                for (j = 0; j < r; j++)
                {
                    if (need[i][j] <= avail[j])
                    {
                        c++;
                        if (c == r) {
                            for (k = 0; k < r; k++)
                            {
                                avail[k] += alloc[i][k];
                            }
                            finish[i] = 1;
                            flag = 1;
                        }
                    }
                }
            }
        }
    }

    int deadlock = 0;
    for (i = 0; i < n; i++)
    {
        if (finish[i] == 0)
        {
            deadlock = 1;
            printf("\n\nSystem is in Deadlock and the Deadlocked process is P%d\n", i + 1);
        }
    }

    if (!deadlock)
    {
        printf("\nNo Deadlock Occurs\n");
    }
}

```