NAME: MUHAMMAD SAIM NOMANI
ROLL NO: DT-22030
SUBJECT: OPERATING SYSTEM
CODE: CT-353
DATA SCIENCE
THIRD YEAR

OS LAB: 8

CODE:

```
#include <stdio.h>
#include <conio.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() {
   printf("********** Deadlock Detection Algo ************\n");
   input();
   show();
   cal();
   getch();
   return 0;
}

void input() {
   int i, j;
   printf("Enter the number of Processes: ");
   scanf("%d", &n);

   printf("Enter the number 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("Enter the Allocation Matrix:\n");
    for (i = 0; i < n; i++) {
        for (j = 0; j < r; j++) {
            scanf("%d", &alloc[i][j]);
        }
    }

    printf("Enter the Available Resources:\n");
    for (j = 0; j < r; j++) {
        scanf("%d", &avail[j]);
    }
}

void show() {
    int i, j;
    printf("\nProcess\tAllocation\tMax\t\tAvailable\n");

    for (i = 0; i < n; i++) {
        printf("P%d\t", i + 1);
        for (j = 0; j < r; j++) {
            printf("%d ", alloc[i][j]);
        }

        printf("\t");

        for (j = 0; j < r; j++) {
            printf("%d ", max[i][j]);
        }

        printf("\t");

        if (i == 0) {
            for (j = 0; j < r; j++) {
                printf("%d ", avail[j]);
            }
        }

        printf("\n");
    }
}
```

```
void cal() {
    int finish[100], dead[100], safe[100];
    int flag = 1, i, j, k, c1 = 0;

    for (i = 0; i < n; i++) {
        finish[i] = 0;
    }

    // Calculate need matrix
    for (i = 0; i < n; i++) {
        for (j = 0; j < r; j++) {
            need[i][j] = max[i][j] - alloc[i][j];
        }
    }

    while (flag) {
        flag = 0;
        for (i = 0; i < n; i++) {
            int c = 0;
            for (j = 0; j < r; j++) {
                if (finish[i] == 0 && 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;
    printf("\n");

    for (i = 0; i < n; i++) {
        if (finish[i] == 0) {
            dead[deadlock++] = i;
        }
    }

    if (deadlock > 0) {
        printf("System is in Deadlock. The deadlocked processes are:\n");
        for (i = 0; i < deadlock; i++) {
```

```
            printf("P%d\t", dead[i]);
        }
        printf("\n");
    } else {
        printf("No Deadlock detected. System is in a safe state.\n");
    }
}
```

OUTPUT: