PAGE NO:
DATE:

8) Write C code to implement deadlock detection program.

```
#include<stdio.h>
#include<stdbool.h>
#define NUM_PROCESSES 5
#define NUM_RESOURCES 3

int available[NUM_Resources];
int allocation[NUM_Resources][Num_Resources];
int request[Num_PROCESSES][NUM_RESOURCES];

bool deadlockDetection(int *safeSequence){
    int work[NUM_RESOURCES];
    bool finish[NUM_PROCESSES];
    for(int i=0; i<NUM_RESOURCES; i++){
        work[i] = available[i];
    }
    int count=0;
    while(count<NUM_processES){
        bool found=false;
        for(int i=0; i<NUM_processES; i++){
            if(!finish[i]){
                bool canProceed = true;
                for(int j=0;j<NUM_RESOURCES;
                    j++){
                    if(request[i][j] > work[j]){
                        canProceed = false;
                        break;
                    }
                }
                if(canProceed){
                    for(int j=0; j<NUM_RESOURCES;j++){
                        work[j] += allocation[i][i];
                    }
```

```
            safeSequence[count++] = i;
            finish[i] = true;
            found = false;
        }
    }
}
if (!found) {
    break;
}
}

for (int i=0; i<NUM_PROCESSES; i++) {
    if (!finish[i]) {
        printf("Deadlock detected. Process
            P%d is in deadlock.\n", i);
        return false;
    }
}

printf("No deadlock detected. The system
        is in a safe state.\n");
printf("\n");
return true;
}

int main() {
    int i, j;
    printf("Enter the available Resources
    for (i=0; i<NUM_RESOURCES; i++) {
        scanf("%d", &available[i]);
    }

    printf("Enter Allocation Matrix:\n");
    for (i=0; i<NUM_Processes; i++) {
        for (j=0; j<NUM_RESOURCES; j++) {
            scanf("%d", &allocation[i][j]
        }
    }
```

```
printf("Enter the Request matrix:\n");
for(i=0; i<NUM_Processes; i++){
    for(j=0; j<NUM_RESOURCES; j++){
        scanf("%d", &request[i][j]);
    }
}

int safeSequence[NUM_processes];
deadlockDetection(safeSequence);
return 0;
}
```

Out put

```
Enter the available Resource vector:
3 3 2
Enter the allocation matrix:
0 1 0
2 0 0
3 0 2
2 1 1
0 0 2
Enter the Request matrix:
7 4 3
1 2 2
6 0 0
0 1 1
0 0 2
No Deadlock detected: The system is in a
safe State
Safe Sequence: P1 P3 P4 P0 P2
```