Avoidance::

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

#define MAX_PROCESSES  100
#define MAX_RESOURCES  100

int available[MAX_RESOURCES];
int maximum[MAX_PROCESSES][MAX_RESOURCES];
int allocation[MAX_PROCESSES][MAX_RESOURCES];
int need[MAX_PROCESSES][MAX_RESOURCES];
int num_processes;
int num_resources;
int safeSeq[MAX_PROCESSES];
int ss=0;
void calculate_need() {
   int i, j;
   for (i = 0; i < num_processes; i++) {
      for (j = 0; j < num_resources; j++) {
         need[i][j] = maximum[i][j] - allocation[i][j];
      }
   }
}

int is_safe() {
   int i, j;
   int work[MAX_RESOURCES];
   int finish[MAX_PROCESSES];


   for (i = 0; i < num_resources; i++) {
      work[i] = available[i];
   }
   for (i = 0; i < num_processes; i++) {
      finish[i] = 0;
   }


   int found = 1;
   while (found) {
      found = 0;
      for (i = 0; i < num_processes; i++) {
         if (!finish[i]) {
            int can_finish = 1;
            for (j = 0; j < num_resources; j++) {
               if (need[i][j] > work[j]) {
                  can_finish = 0;
                  break;
               }
            }
            if (can_finish) {
               found = 1;
               finish[i] = 1;
               safeSeq[ss]=i;
               ss++;
```

```
                for (j = 0; j < num_resources; j++) {
                    work[j] += allocation[i][j];
                }
            }
        }
    }
}

for (i = 0; i < num_processes; i++) {
    if (!finish[i]) {
        return 0;
    }
}
return 1;
}

void detect_deadlock() {
int i, j;


printf("Enter the number of processes: ");
scanf("%d", &num_processes);
printf("Enter the number of resources: ");
scanf("%d", &num_resources);


printf("Enter the available array:\n");
for (i = 0; i < num_resources; i++) {
    scanf("%d", &available[i]);
}

// Input maximum and allocation arrays
printf("Enter the maximum matrix:\n");
for (i = 0; i < num_processes; i++) {
    for (j = 0; j < num_resources; j++) {
        scanf("%d", &maximum[i][j]);
    }
}
printf("Enter the allocation matrix:\n");
for (i = 0; i < num_processes; i++) {
    for (j = 0; j < num_resources; j++) {
        scanf("%d", &allocation[i][j]);
    }
}

calculate_need();
if (is_safe()) {
    printf("No deadlock detected.\n");
     printf("Safe sequence is as follows:\n");
    for(i = 0;i<ss;i++){
        printf("%d ",safeSeq[i]);
    }
    printf("\n");
} else {
    printf("Deadlock detected.\n");
}
}
```

```
int main() {
   detect_deadlock();
   return 0;
}
```

Detection-->

```
#include<stdio.h>
#include<stdlib.h>
#define MAX_PROCESSES  100
#define MAX_RESOURCES  100

int available[MAX_RESOURCES];
int allocation[MAX_PROCESSES][MAX_RESOURCES];
int request[MAX_PROCESSES][MAX_RESOURCES];
int num_processes;
int num_resources;

int is_safe() {
   int i, j;
   int avai_resources[MAX_RESOURCES];
   int state[MAX_PROCESSES];
   for (i = 0; i < num_resources; i++) {
      avai_resources[i] = available[i];
   }
   for (i = 0; i < num_processes; i++) {
      state[i] = 0;
   }
   int found = 1;
   while (found) {
      found = 0;
      for (i = 0; i < num_processes; i++) {
         if (!state[i]) {
            int can_finish = 1;
            for (j = 0; j < num_resources; j++) {
               if (request[i][j] > avai_resources[j]) {
                  can_finish = 0;
                  break;
               }
            }
            if (can_finish) {
               found = 1;
               state[i] = 1;
               for (j = 0; j < num_resources; j++) {
                  avai_resources[j] += allocation[i][j];
               }
            }
         }
      }
   }

   for (i = 0; i < num_processes; i++) {
      if (!state[i]) {
         return 0;
      }
   }
```

```
    return 1;
}

void detect_deadlock() {
    if (is_safe()) {
        printf("No deadlock detected.\n");
    } else {
        printf("Deadlock detected.\n");
    }
}

int main() {
    int i, j;
    printf("Enter the number of processes: ");
    scanf("%d", &num_processes);
    printf("Enter the number of resources: ");
    scanf("%d", &num_resources);
    printf("Enter the available resources: ");
    for (i = 0; i < num_resources; i++) {
        scanf("%d", &available[i]);
    }
    printf("Enter the allocation matrix:\n");
    for (i = 0; i < num_processes; i++) {
        printf("Process %d: ", 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++) {
        printf("Process %d: ", i);
        for (j = 0; j < num_resources; j++) {
            scanf("%d", &request[i][j]);
        }
    }
    detect_deadlock();

    return 0;
}
```