```
Script started on 2019-03-06 09:11:49+0530
praveen@praveen$ cat Deadlock.c
#include<stdio.h>
int M = 3, N = 2;
int RAG[10][10];
int WFG[10][10];
int stack[10], v[10], top = -1, flag = 0, l = 0, cycle = 0;
void print(int a[][10], int n, int m)
{
    for(int i = 0; i < n; i++)
    {
        for(int j = 0; j < m; j++)
        {
            printf("%d ", a[i][j]);
        }
        printf("\n");
    }
}
void request()
{
    printf("PID and RID: ");
    int p, r, flag = 0;
    scanf("%d %d", &p, &r);
    p--; r--; r+=M;
    RAG[p][r] = 1;
    for(int i = 0; i < M+N; i++)
    {
        if(RAG[r][i] == 1) {
            flag = 1;
        }
    }
    if(flag == 0) {
        RAG[p][r] = 0;
        RAG[r][p] = 1;
    }
}
void release()
{
    printf("PID and RID: ");
    int p, r;
    scanf("%d %d", &p, &r);
    p--; r--; r+=M;
    if(RAG[r][p] == 0) return;
    RAG[r][p] = 0;
    for(int i = 0; i < M; i++)
    {
      if(RAG[i][r] == 1) {
          RAG[r][i] = 1;
          RAG[i][r] = 0;
          break;
      }
    }
}
void checkcycle(int a, int o)
{
    v[a] = 1;
    top++;
    stack[top] = a;
    for(int i = 0; i < M+N && cycle == 0; i++)
    {
        if(RAG[a][i] == 1 && i == o && l >= 3) {
            cycle = 1;
            break;
        }
```

```
        if(v[i]==0 && RAG[a][i]==1) {
            l++;
            checkcycle(i, o);
        }
    }
    if(cycle == 1) {
      if(stack[top] >= M) printf("R%d ", stack[top] - M + 1);
      else printf("P%d ", stack[top] + 1);
      top--;
    }
}
void detect()
{
    for(int a = 0; a < M+N; a++)
    {
        cycle = 0;
        checkcycle(a, a);
      for(int i = 0; i < M+N; i++) v[i] = 0;
        if(cycle == 1) {
            printf("Deadlock!\n");
            break;
        }
    }
    if(cycle == 0) printf("No deadlock!\n");
}
void waitforgraph()
{
    for(int i = 0; i < M; i++)
      for(int j = 0; j < M; j++)
          WFG[i][j] = 0;
    for(int i = 0; i < M; i++)
    {
      for(int j = M; j < M+N; j++)
      {
          if(RAG[i][j] == 1)
          {
            int r = j;
            for(int k = 0; k < M; k++)
            {
                if(RAG[r][k] == 1) {
                  r = k; break;
                }
            }
            WFG[i][r] = 1;
          }
      }
    }
}
int main()
{
    int ch = 0;
    printf("Enter no. of processes and resources: ");
    scanf("%d %d", &M, &N);
    do {
      printf("1. Request 2. Release 3. Detect Deadlock 4. Print RAG 5. Print WFG
6. Exit\nChoice: ");
      scanf("%d", &ch);
      switch(ch)
      {
          case 1: request(); break;
          case 2: release(); break;
          case 3: detect(); break;
          case 4: print(RAG, M+N, M+N); break;
          case 5: waitforgraph(); print(WFG, M, M); break;
```

```
      }
      printf("\n");
    }while(ch != 6);
    return 0;
}
praveen@praveen$ gcc Deadlock.c
praveen@praveen$ ./a.out
Enter no. of processes and resources: 5 4
1. Request 2. Release 3. Detect Deadlock 4. Print RAG 5. Print WFG 6. Exit
Choice: 1
PID and RID: 1 3

1. Request 2. Release 3. Detect Deadlock 4. Print RAG 5. Print WFG 6. Exit
Choice: 1
PID and RID: 2 2

1. Request 2. Release 3. Detect Deadlock 4. Print RAG 5. Print WFG 6. Exit
Choice: 3# #1
PID and RID: 3 0

1. Request 2. Release 3. Detect Deadlock 4. Print RAG 5. Print WFG 6. Exit
Choice: 4# #1
PID and RID: 4 1

1. Request 2. Release 3. Detect Deadlock 4. Print RAG 5. Print WFG 6. Exit
Choice: 1
PID and RID: 1 1

1. Request 2. Release 3. Detect Deadlock 4. Print RAG 5. Print WFG 6. Exit
Choice: 2 3
PID and RID: 0

1. Request 2. Release 3. Detect Deadlock 4. Print RAG 5. Print WFG 6. Exit
Choice: 1
PID and RID: 0 0

1. Request 2. Release 3. Detect Deadlock 4. Print RAG 5. Print WFG 6. Exit
Choice: 1
PID and RID: 3 0

1. Request 2. Release 3. Detect Deadlock 4. Print RAG 5. Print WFG 6. Exit
Choice: 3
No deadlock!

1. Request 2. Release 3. Detect Deadlock 4. Print RAG 5. Print WFG 6. Exit
Choice: 1
PID and RID: 1# #4 3

1. Request 2. Release 3. Detect Deadlock 4. Print RAG 5. Print WFG 6. Exit
Choice: 3
R3 P4 R1 P1 Deadlock!

1. Request 2. Release 3. Detect Deadlock 4. Print RAG 5. Print WFG 6. Exit
Choice: 4
0 0 0 0 0 1 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 1 0
0 0 1 0 0 0 0 0 0
0 0 0 1 0 0 0 0 0
0 1 0 0 0 0 0 0 0
1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
```

```
1. Request 2. Release 3. Detect Deadlock 4. Print RAG 5. Print WFG 6. Exit
Choice: 5
0 0 0 1 0
0 0 0 0 0
0 0 0 0 0
1 0 0 0 0
0 0 0 0 0

1. Request 2. Release 3. Detect Deadlock 4. Print RAG 5. Print WFG 6. Exit
Choice: 2
PID and RID: 4 3

1. Request 2. Release 3. Detect Deadlock 4. Print RAG 5. Print WFG 6. Exit
Choice: 3
R3 P4 R1 P1 Deadlock!

1. Request 2. Release 3. Detect Deadlock 4. Print RAG 5. Print WFG 6. Exit
Choice: 2
PID and RID: 4 1

1. Request 2. Release 3. Detect Deadlock 4. Print RAG 5. Print WFG 6. Exit
Choice: 3
No deadlock!

1. Request 2. Release 3. Detect Deadlock 4. Print RAG 5. Print WFG 6. Exit
Choice: 6

praveen@praveen$ exit
exit

Script done on 2019-03-06 09:23:34+0530
```