```
/*Resource Allocation Graph*/

#include<stdio.h>

int main()

{

int Resources[6][6],Process[5];

int resource=0,i,j,n,row,column,count=-1;

printf("You can Implement for 4 processes and 3 columns of resources\n");

printf("Process Sequence\n");

printf("Enter the number of processes\n");

scanf("%d",&n);

printf("Enter Process Numbers (0-4)\n");

for(i=0;i<n;i++)

{

scanf("%d",&Process[i]);//Process count is entered here.Preferably 4-5
processes can be considered

}

printf("Enter number of rows for resource array\n");

scanf("%d",&row); //Program is designed to handle 4 rows

printf("Enter number of columns for resource array\n");

scanf("%d",&column); //Program is designed to handle 3 columns

printf("Enter resources");

for(i=0;i<row;i++)

{

for(j=0;j<column;j++)

{

scanf("%d",&Resources[i][j]);            //Resources are allocated in a i x j
matrix

}

}

printf("Process Show\n");

for(i=0;i<n;i++)

{

printf("%d",Process[i]);       //To display processes
```

```
}
printf("\nResource Matrix  Process Number\n");
printf("A \t B \t  C \n");
for(i=0;i<1;i++)
{
for(j=0;j<1;j++)
{
printf("%d \t %d \t %d \t <--
%d\n",Resources[0][0],Resources[0][1],Resources[0][2],Process[0]);
printf("%d \t %d \t %d \t <--
%d\n",Resources[1][0],Resources[1][1],Resources[1][2],Process[1]);
printf("%d \t %d \t %d \t <--
%d\n",Resources[2][0],Resources[2][1],Resources[2][2],Process[2]);
printf("%d \t %d \t %d \t <--
%d",Resources[3][0],Resources[3][1],Resources[3][2],Process[3]);
}
printf("Check for Resource Competition/Possibility of Deadlock\n");
printf("Please check resource matrix\n");
printf("\nEnter a resource to check if collision happens");
scanf("%d",&resource);
for(i=0;i<row;i++)
{
for(j=0;j<column;j++)
{
if(resource==Resources[i][j])
{
printf("Matches[row][column]   %d \t %d\n",i,j);
count=count+1;
}
}
}
}
printf("Counts of deadlock = %d \n ",count);
return 0;
```

}

/* OUTPUT

```
Enter the number of processes
4
Enter Process Numbers (0-4)
0
1
2
3
Enter number of rows for resource array
4
Enter number of columns for resource array
3
Enter resources
1
2
3
4
5
4
3
2
1
0
2
1
Process Show
P0
P1
P2
P3
Resource Matrix  Process Number
A  B   C
1  2  3       <--   0
4  5  4       <--   1
3  2  1       <--   2
0  2  1       <--   3
```

*/