MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

DATA STRUCTURE LAB

MINI SKILL BASED PROJECT

SUBMITTED BY:

ASMITA JAIN

0901EO201017

SUBMITTED TO:

PROF. NAMRATA AGRAWAL

QUESTION

DAP A PROGRAM TO SHOW POSSIBLE MOVEMENTS OF QUEEN ON A CHESS BOARD FROM ITS CURRENT POSITION. ALSO, CHECK IF POSITION ENTERED IS VALID OR NOT.

(NOTE: X-AXIS IS MARKED FROM '1' TO '8', Y-AXIS IS MARKED FROM 'a' TO 'h'.)

ANSWER

```
#include <iostream>
using namespace std;

bool check(int,char);
void horizontal(int x,char y);
void vertical(int x,char y);
void right_diagonal(int x,char y);
void left_diagonal(int x,char y);

int c=0;
char a[9]={'\0','a','b','c','d','e','f','g','h'};

int main()
{

   int x,choice=1;
   char y;
```

```
while(choice)
{
    cout<<"ENTER COORDINATES OF QUEEN (1-8<space>a-h) : "<<endl;
    cin>>x;
    cin>>y;
    if(check(x,y))
       cout<<"INVALID POSITION"<<endl;
    else
    {
    cout<<"The possible positions can be: "<<endl;
    horizontal(x,y);

    cout<<endl;
    vertical(x,y);

    cout<<endl;
    left_diagonal(x,y);

    cout<<endl;
    right_diagonal(x,y);

    cout<<endl<<endl;
    cout<<"TOTAL NO. OF POSSIBLE POSITIONS: "<<c<<endl;
    }
```

```
        cout<<endl<<"DO YOU WANT TO CONTINUE?(0/1): ";
        cin>>choice;
        cout<<endl<<endl;
    }
    return 0;
}
bool check(int x,char y)
{
    if((x>8 || x<1) || (y<'a' || y>'h'))
       return true;
    return false;
}

void horizontal(int x,char y)
{
     cout<<"Horizontal movement: ";
     for(int i=7;i>=0;i--)
     {
        if((8-i)==x)
         continue;
        cout<<"("<<8-i<<","<<y<<")"<<"  ";
        c++;
     }
```

```
}

void vertical(int x,char y)
{
   int i,j;
   cout<<"Vertical movement: ";
     for(j=7;j>=0;j--)
     {
        i=8-j;
        if((a[i]==y))
         continue;
        cout<<"("<<x<<","<<a[i]<<")"<<" ";
        c++;
     }
}

void right_diagonal(int x,char y)
{
   int i,j;
   j=(int)(y-'a') ;
     i=x-1;
     cout<<"Right diagonal movement: ";
     while(((j)>=1) && ((i)>=1))
     {
```

```
        cout<<"("<<i<<","<<a[j]<<")"<<" ";
        c++;
        i--;
        j--;
    }
    j=(int)(y-'a')+2 ;
    i=x+1;
    while(((j)<=8) && ((i)<=8))
    {
        cout<<"("<<i<<","<<a[j]<<")"<<" ";
        c++;
        i++;
        j++;
    }
}
void left_diagonal(int x,char y)
{
   int i,j;
   cout<<"Left diagonal movement: ";
    j=(int)(y-'a')+2 ;
    i=x-1;
    while(((j)<=8) && ((i)>=1))
    {
        cout<<"("<<i<<","<<a[j]<<")"<<" ";
```

```
        c++;
        i--;
        j++;
    }
    j=(int)(y-'a') ;
    i=x+1;
    while(((j)>=1) && ((i)<=8))
    {
        cout<<"("<<i<<","<<a[j]<<")"<<" ";
        c++;
        i++;
        j--;
    }
}
```

OUTPUT

```
ENTER COORDINATES OF QUEEN (1-8<space>a-h) :
3 e
The possible positions can be:
Horizontal movement: (1,e)  (2,e)  (4,e)  (5,e)  (6,e)  (7,e)  (8,e)
Vertical movement: (3,a)  (3,b)  (3,c)  (3,d)  (3,f)  (3,g)  (3,h)
Left diagonal movement: (2,f) (1,g) (4,d) (5,c) (6,b) (7,a)
Right diagonal movement: (2,d) (1,c) (4,f) (5,g) (6,h)

TOTAL NO. OF POSSIBLE POSITIONS: 25

DO YOU WANT TO CONTINUE?(0/1): 1


ENTER COORDINATES OF QUEEN (1-8<space>a-h) :
5 g
The possible positions can be:
Horizontal movement: (1,g)  (2,g)  (3,g)  (4,g)  (6,g)  (7,g)  (8,g)
Vertical movement: (5,a)  (5,b)  (5,c)  (5,d)  (5,e)  (5,f)  (5,h)
Left diagonal movement: (4,h) (6,f) (7,e) (8,d)
Right diagonal movement: (4,f) (3,e) (2,d) (1,c) (6,h)

TOTAL NO. OF POSSIBLE POSITIONS: 48

DO YOU WANT TO CONTINUE?(0/1): 1


ENTER COORDINATES OF QUEEN (1-8<space>a-h) :
9 a
INVALID POSITION

DO YOU WANT TO CONTINUE?(0/1): 1


ENTER COORDINATES OF QUEEN (1-8<space>a-h) :
3 i
INVALID POSITION

DO YOU WANT TO CONTINUE?(0/1): 0
```