```
// A simple snake game project(program) :)

#include<iostream>
using namespace std ;
#include <stdlib.h>            // used for rand()
#include<conio.h>              // used for getch()  and kbhit()
#include<windows.h>            // used for sleep()
#include<iomanip>

bool gameover ;                // can also use flag variable
const  int height=20 ;
const int width=20 ;
int x,y,fruitx,fruity,score ;
int tailx[100],taily[100] ;
int tail,tough=0 ;

enum direction{                      // can  also use glabal const int variables
        STOP=0, LEFT,RIGHT,UP,DOWN           // of name STOP,LEFT,RIGHT,UP,DOWN having
};                              // values 0,1,2,3,4 respectivally
direction dir ;

void setup()                   // inicialise the inicial values,run only once
{
        gameover=false ;
        dir=UP ;                  // default movemebnt  of snake
        x=width/2 ;               // x and y are coordinats of default snake head
```

```
        y=height/2;                // position
        fruitx=rand()%width;       // coordinates of default randam fruit position
        fruity=rand()%height;
        score=0;
        tail=0;
}

void draw()
{
        system("cls");             //used to clear the previous screen , so that
        int i,j,k;                 //each time when draw() runs the position of
        for(i=0;i<width+1;i++)     //the box remains the same .
                cout<<"#" ;        // prints upper boundary
        cout<<endl;

        for(i=0;i<height;i++)
        {
                for(j=0;j<width;j++)
                {
                        if(j==0||j==width-1)       // prints side boundary
                                cout<<"#" ;

                        if(i==y&&j==x)             // snake head
                                cout<<"O" ;
                        else if(i==fruity&&j==fruitx)     // fruit
                                cout<<"*" ;
                        else
                        {
```

```
                                bool print=false ;          // can also use int flag variable
                                for(k=0;k<tail;k++)
                                {
                                        if(tailx[k]==j&&taily[k]==i)  // printing of tail of snake
                                        {
                                                cout<<"o" ;
                                                print=true ;
                                        }
                                }
                                if(!print)
                                        cout<<" " ;
                        }
                }
                cout<<endl ;
        }

        for(i=0;i<width+1;i++)                  // lower boundary
                cout<<"#" ;
        cout<<endl ;
        cout<<" YOUR SCORE = "<<score<<endl ;
}

void input()
{
        char ch ;
        if(!kbhit())            // keyboard hit f(), returns 1 when we hit the
        {                       // keyboard else returns 0
                switch(ch)
```

```
        {
                case 'w' : dir=UP ;
                                break ;
                case 'a' : dir=LEFT ;
                                break ;
                case 's' : dir=DOWN ;
                                break ;
                case 'd' : dir=RIGHT ;
                                break ;
                case 'x' : gameover=true ;
                                break ;
        }
}
else                            // else part  for default movement of
{                               // the snake in the same direction till
        switch(getch())                 // user don't inputs any movement
        {
                case 'w' : dir=UP ;
                                ch='w' ;
                                break ;
                case 'a' : dir=LEFT ;
                                ch='a' ;
                                break ;
                case 's' : dir=DOWN ;
                                ch='s' ;
                                break ;
                case 'd' : dir=RIGHT ;
                                ch='d' ;
```

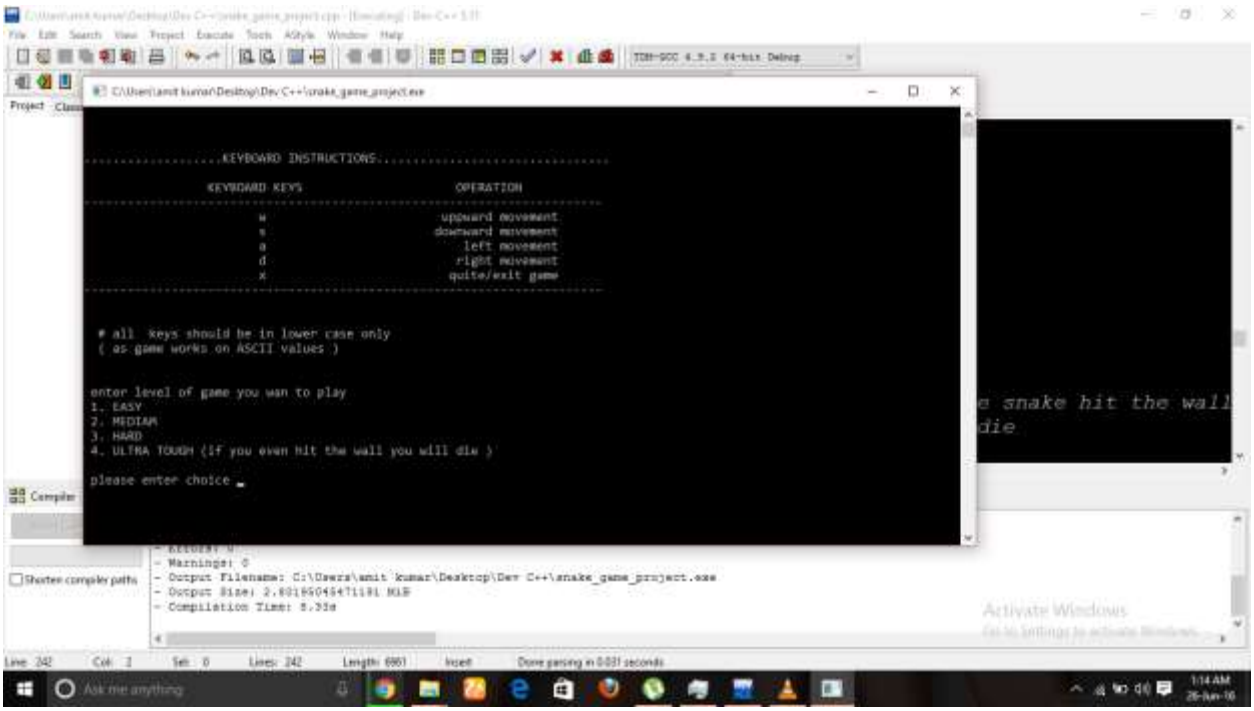
```
                                        break;
                        case 'x' : gameover=true;
                                        break;
                }
        }
}

void logic()
{
        int prevx,prevy,prev2x,prev2y,i;       //line 120-136 for  the tail
        prevx=tailx[0];
        prevy=taily[0];
        tailx[0]=x;
        taily[0]=y;

        for(i=1;i<tail;i++)                    // SWAPPING of values of array
        {
                prev2x=tailx[i];
                prev2y=taily[i];

                tailx[i]=prevx;
                taily[i]=prevy;

                prevx=prev2x;
                prevy=prev2y;
        }

        switch(dir)
```

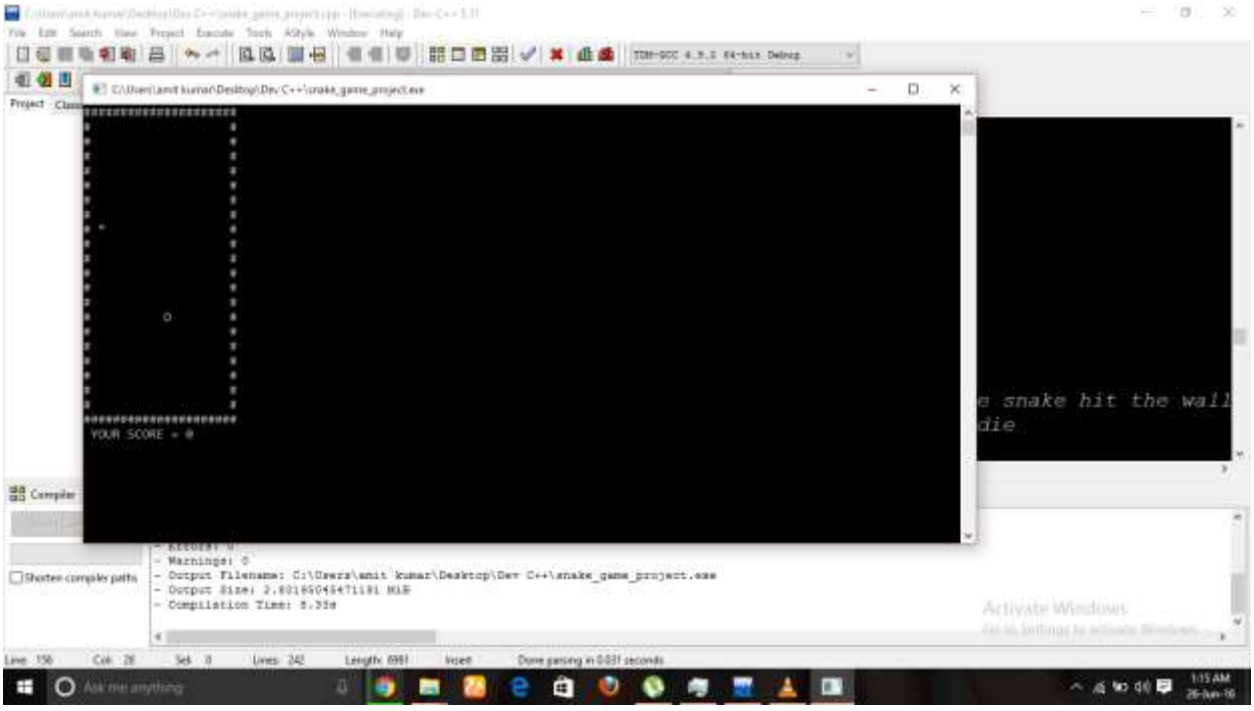
```
{
        case UP :   y-- ;
                        break ;
        case DOWN : y++ ;
                        break ;
        case LEFT : x-- ;
                        break ;
        case RIGHT : x++ ;
                        break ;
        default : break ;
}

for(i=0;i<tail;i++)                 // if snake hits its own tail it will
{                               // die
        if(tailx[i]==x&&taily[i]==y)
        {
                gameover=true ;
                system("cls");
                cout<<"\n GAME OVER : you hit/bite your tail" ;
        }
}

if(tough==1)
{
        if(x<0||x>width||y<0||y>height)         // if the snake hit the wall it
        {                               // will die
                gameover=true ;
                system("cls");
```

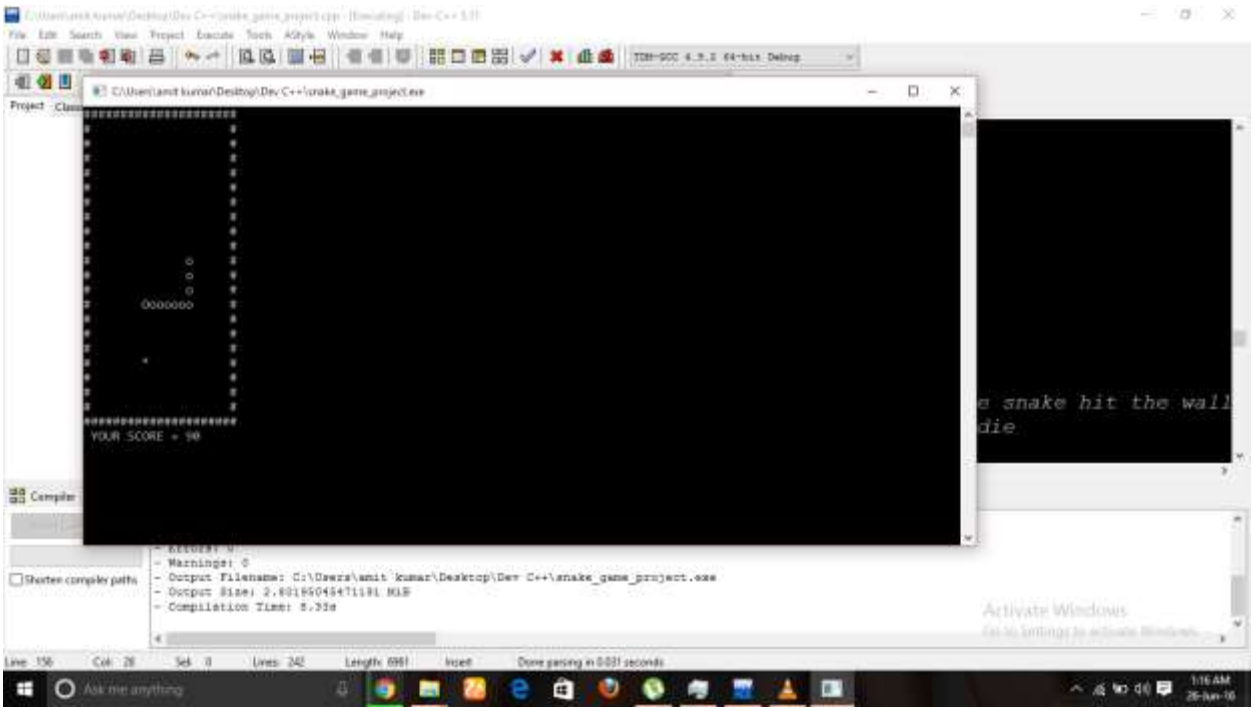
```
                cout<<"\n GAME OVER : you hit the wall " ;
            }
        }
        else
        {
            if(x<0)                 // if snake hit the wall it will continue
                x=width-1 ;             // from same position on the opposite
            else if(x>width-2)          // wall
                x=0 ;

            if(y<0)
                y=height-1 ;
            else if(y>height)
                y=0 ;
        }

        if(x==fruitx&&y==fruity)            // if snake ate the fruit score
        {                           // should increase and new fruit
            score+=10 ;                 // position should be atomatically
            fruitx=rand()%width ;           // genereted on the screen
            fruity=rand()%height ;
            tail++ ;                    // length of tail increases
        }
}

int main()
{
        int n,choice ;
```

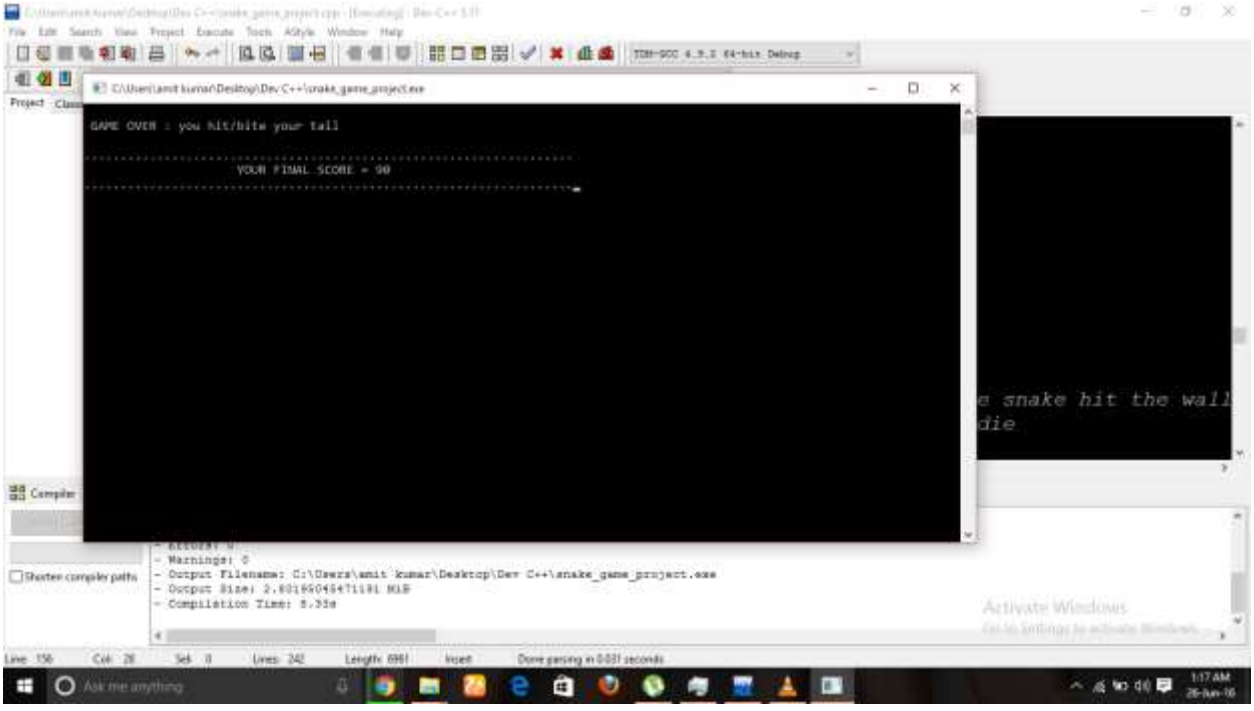
```
cout<<"\n\n\n.....KEYBOARD INSTRUCTIONS.....";
cout<<endl<<endl<<setw(30)<<"KEYBOARD KEYS"<<setw(30)<<"OPERATION";
cout<<"\n-----------------------------------------------------------";
cout<<endl<<setw(25)<<"w"<<setw(40)<<"uppward movement";
cout<<endl<<setw(25)<<"s"<<setw(40)<<"downward movement";
cout<<endl<<setw(25)<<"a"<<setw(40)<<"left movement";
cout<<endl<<setw(25)<<"d"<<setw(40)<<"right movement";
cout<<endl<<setw(25)<<"x"<<setw(40)<<"quite/exit game";
cout<<"\n-----------------------------------------------------------";
cout<<"\n\n\n # all  keys should be in lower case only\n  ( as game works on ASCII values )";

cout<<"\n\n\n enter level of game you wan to play " ;
cout<<"\n 1. EASY \n 2. MEDIAM \n 3. HARD  " ;
cout<<"\n 4. ULTRA TOUGH (if you even hit the wall you will die )";
cout<<"\n\n please enter choice " ;
cin>>choice ;

if(choice==1)
        n=100 ;
else if(choice==2)
        n=50 ;
else if(choice==3)
        n=10 ;
else
{
        tough=1 ;
        n=70 ;
```

```
        }

        system("cls");
        setup();
        while(!gameover)            // can also use while(flag) we only need to run the
        {                   // loop infinitr no of times
            draw();
            input();
            logic();
            Sleep(n);               // f() to delay the output screen each time it
        }                   // get executed for n milli-secondes

        cout<<"\n\n.....";
        cout<<"\n              YOUR FINAL SCORE = "<<score ;
        cout<<"\n.....";

        getch();
        getch();
        return 0 ;
}
```


C:\Users\amit kumar\Desktop\Dev C++\snake_game_project.exe
.....KEYBOARD INSTRUCTIONS.....
KEYBOARD KEYS OPERATION
w upward movement
s downward movement
a left movement
d right movement
x quite/exit game
all keys should be in lower case only
(as game works on ASCII values)
enter level of game you wan to play
1. EASY
2. MEDIAM
3. HARD
4. ULTRA TOUGH (if you even hit the wall you will die)
please enter choice
- Warnings: 0
- Output Filename: C:\Users\amit kumar\Desktop\Dev C++\snake_game_project.exe
- Output Size: 2.8018604547119141 MiB
- Compilation Time: 0.33s


C:\Users\amit kumar\Desktop\Dev C++\snake_game_project.exe
YOUR SCORE = 0
- Warnings: 0
- Output Filename: C:\Users\amit kumar\Desktop\Dev C++\snake_game_project.exe
- Output Size: 2.8018604547119141 MiB
- Compilation Time: 0.33s


YOUR SCORE = 90
- Warnings: 0
- Output Filename: C:\Users\amit kumar\Desktop\Dev C++\snake_game_project.exe
- Output Size: 2.8018504547119141 MiB
- Compilation Time: 0.33s
Activate Windows


GAME OVER : you hit/bite your tail
YOUR FINAL SCORE = 90
- Warnings: 0
- Output Filename: C:\Users\amit kumar\Desktop\Dev C++\snake_game_project.exe
- Output Size: 2.8018504547119141 MiB
- Compilation Time: 0.33s
Activate Windows