VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

B. Tech. Semester 2020-2021

SCHOOL OF COMPUTER SCIENCE ENGINEERING

(SCOPE)

Computer Graphics and Multimedia

CSE 3021

AMIT KUMAR

19BCE1281

TABLE OF CONTENTS

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

B. Tech. Semester 2020-2021

SCHOOL OF COMPUTER SCIENCE ENGINEERING

(SCOPE)

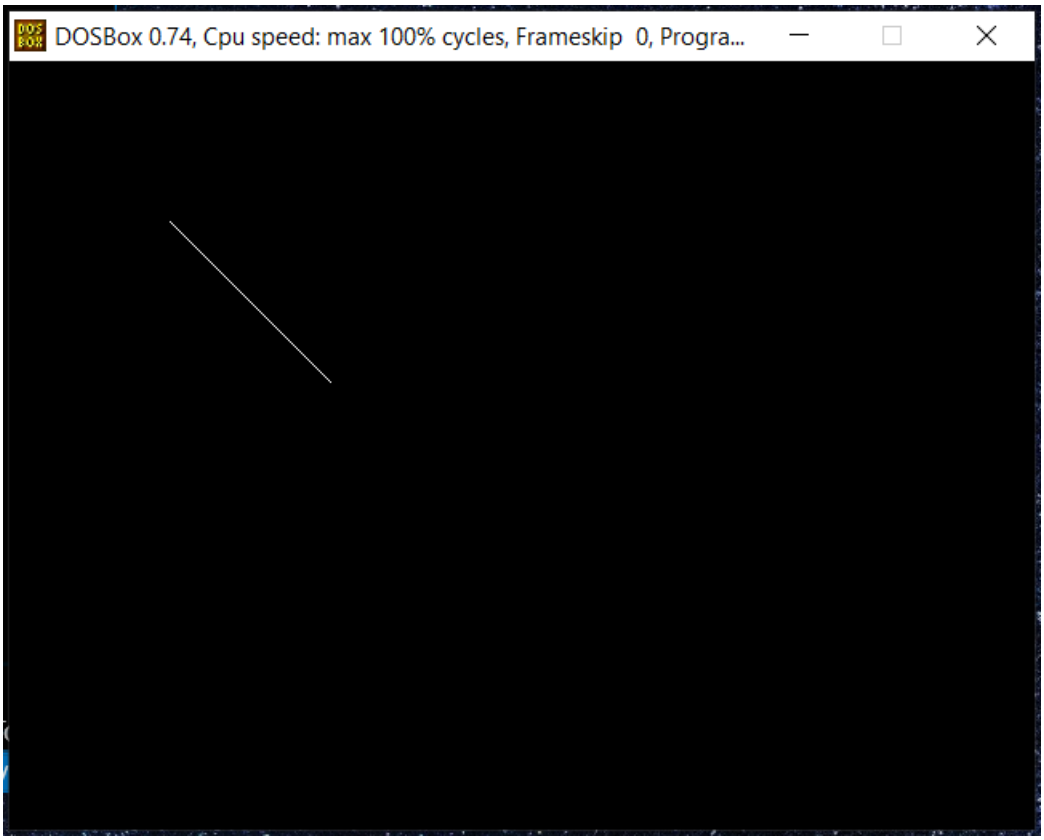
Computer Graphics and Multimedia

LAB 01

AMIT KUMAR

19BCE1281

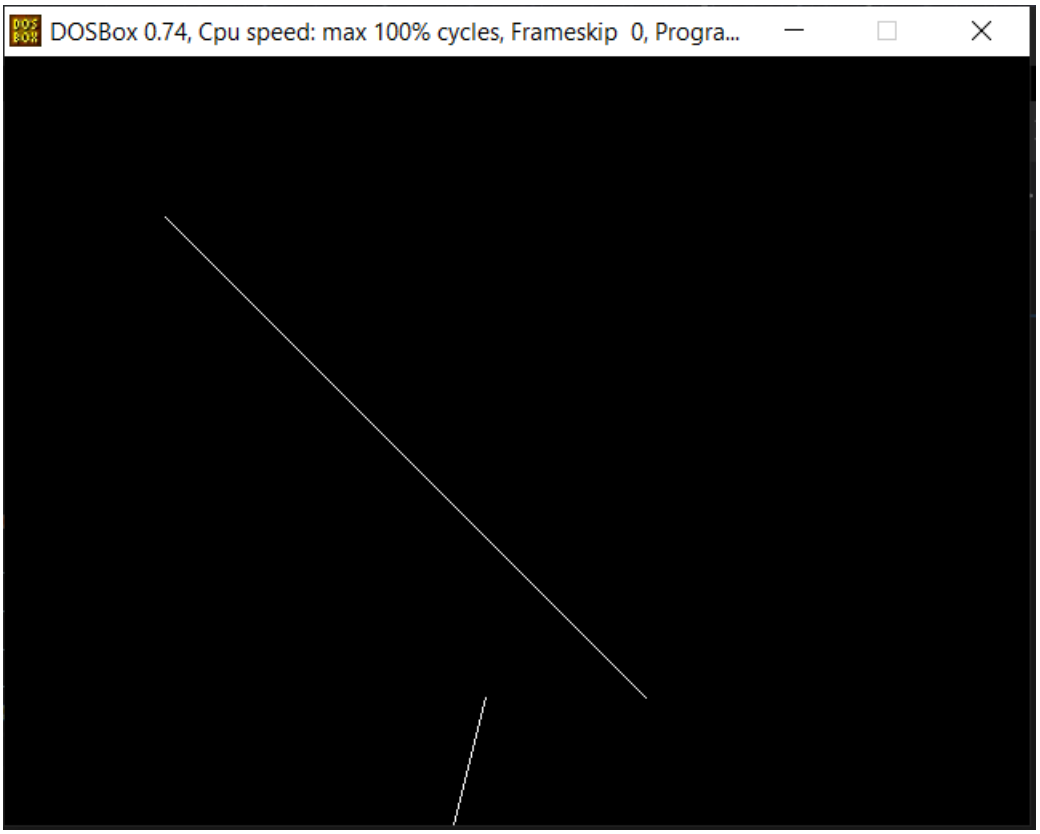
1.A line



CODE

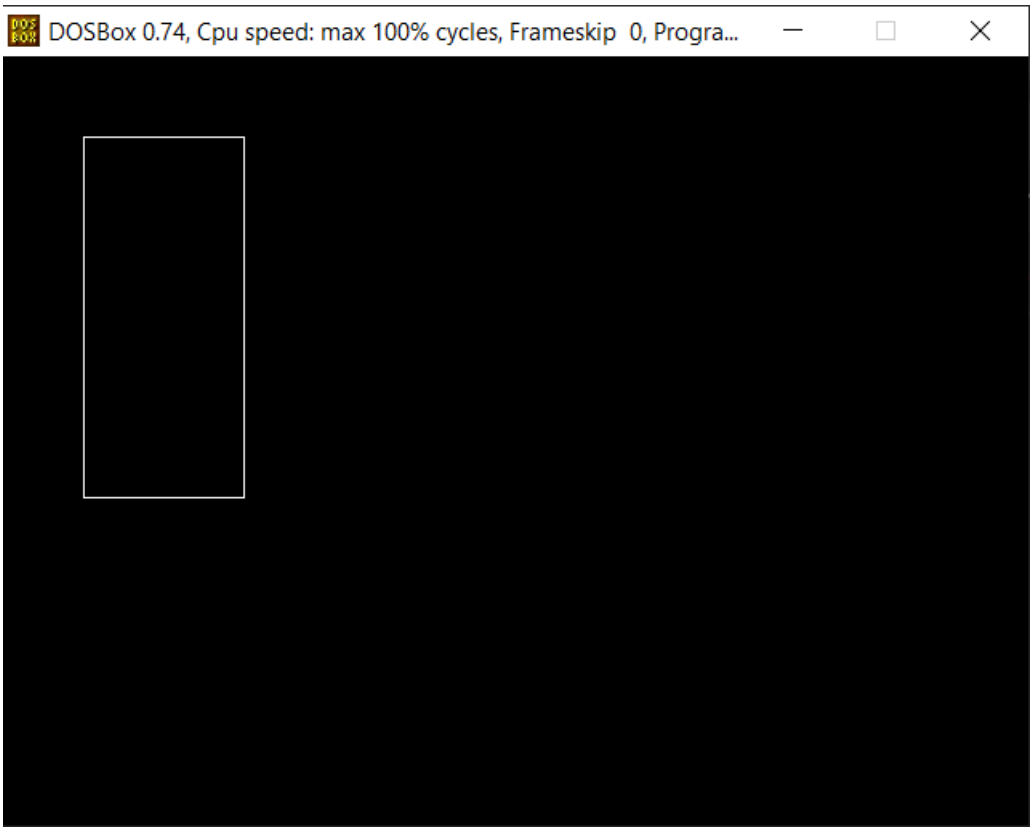
```c
#include<graphics.h>
#include<conio.h>
#include<stdio.h>

int main(){
    int gd = DETECT, gm;
    initgraph(&gd, &gm, "C:\\TURBOC3\\BGI");
    line(100,100,200,200);
    getch();
    closegraph();
    return 0;
}
```

2.Multiple lines



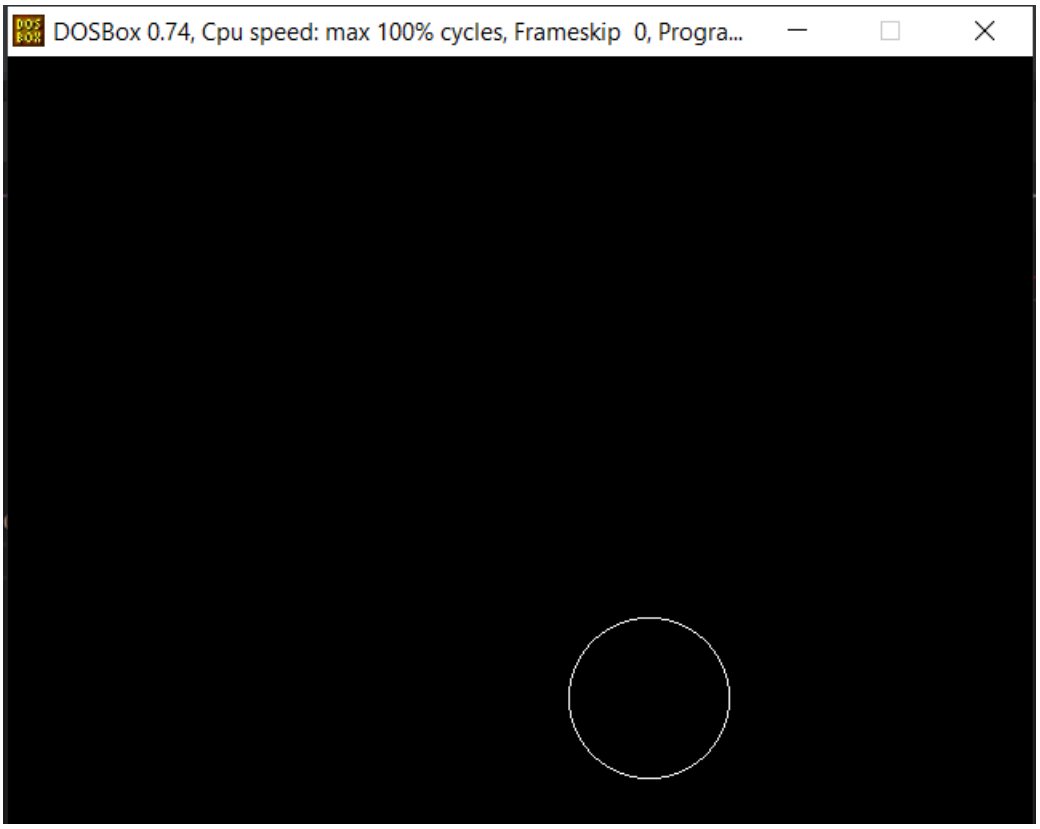

```c
#include<graphics.h>
#include<conio.h>
#include<stdio.h>

int main(){
    int gd = DETECT, gm;
    initgraph(&gd, &gm, "C:\\TURBOC3\\BGI");
    line(100,100,200,200);
    line(200,200,400,400);
    line(300,400,200,800);
    getch();
    closegraph();
    return 0;
}
```

3.Rectangle



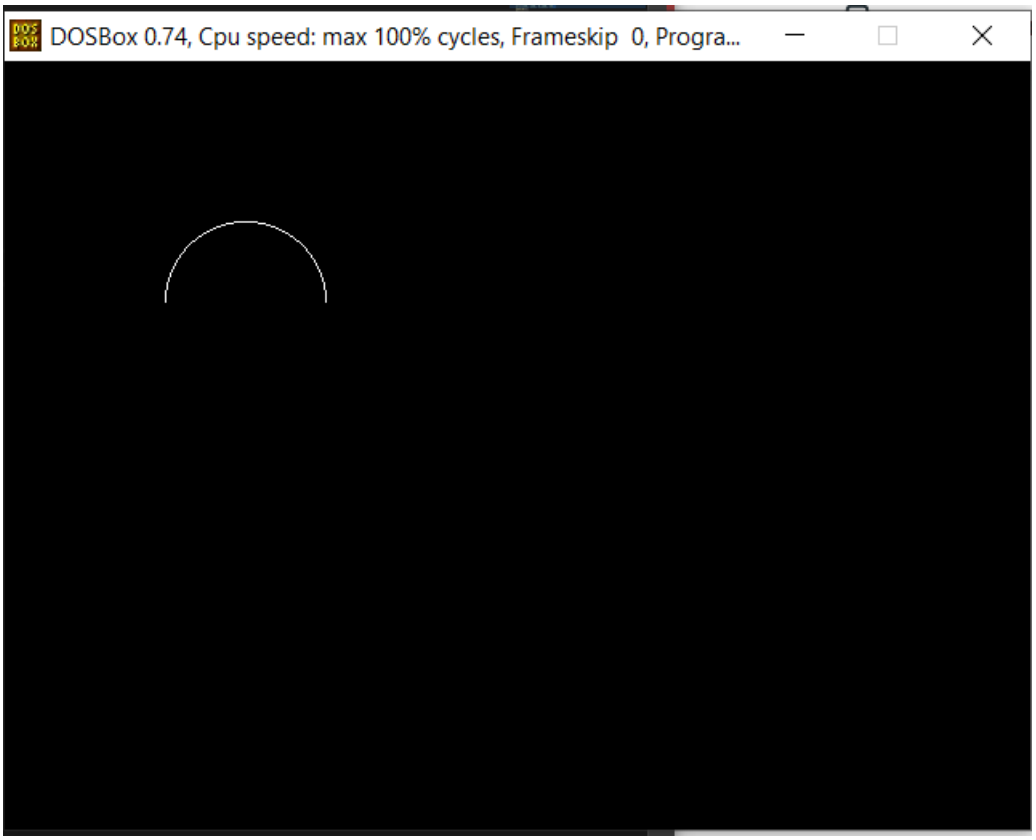

```c
#include<graphics.h>
#include<conio.h>
#include<stdio.h>


int main(){
    int gd = DETECT, gm;
    initgraph(&gd, &gm, "C:\\TURBOC3\\BGI");
    line(50,50,150,50);
    line(150,50,150,275);
    line(150,275,50,275);
    line(50,275,50,50);
    getch();
    closegraph();
    return 0;
}
```

4.circle



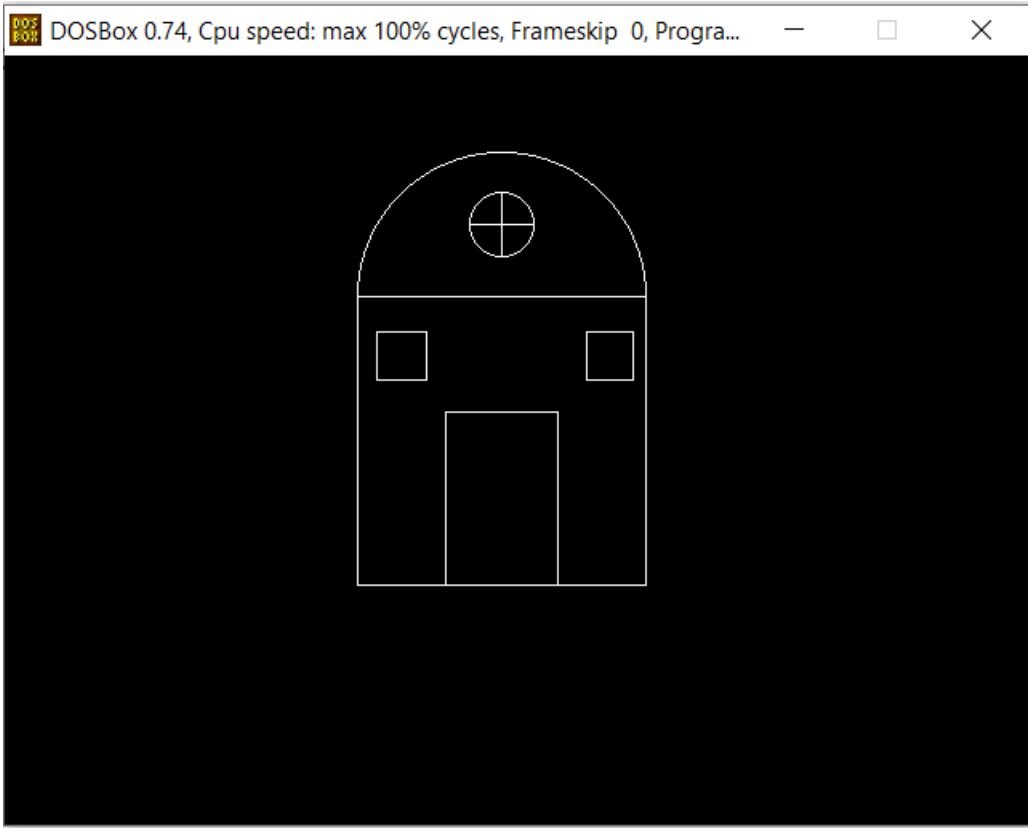

```c
#include<graphics.h>
#include<conio.h>
#include<stdio.h>

int main(){
    int gd = DETECT, gm;
    initgraph(&gd, &gm, "C:\\TURBOC3\\BGI");
    circle(400, 400, 50);
    getch();
    closegraph();
    return 0;
}
```

5. Arc




```c
#include<graphics.h>
#include<conio.h>
#include<stdio.h>

int main(){
    int gd = DETECT, gm;
    initgraph(&gd, &gm, "C:\\TURBOC3\\BGI");
    arc(150, 150, 0,180, 50)
    getch();
    closegraph();
    return 0;
}
```

6. Hut




```c
#include<graphics.h>
#include<conio.h>
#include<stdio.h>


int main(){
    int gd = DETECT, gm;
    initgraph(&gd, &gm, "C:\\TURBOC3\\BGI");
    line(220,150,400,150);
    line(400,150,400,330);
    line(400,330,220,330);
    line(220,330,220,150);
    arc(310,150,0,180,90);
    line(275,330,275,222);
    line(275,222,345,222);
    line(345,222,345,330);
    circle(310,105,20);
    line(310,125,310,85);
    line(290,105,330,105);
    line(232,202,263,202);
    line(232,172,263,172);
    line(232,172,232,202);
    line(263,172,263,202);
    line(392,202,363,202);
    line(392,202,392,172);
    line(392,172,363,172);
    line(363,202,363,172);
    getch();
    closegraph();
```

```
    return 0;
}
```

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

B. Tech. Semester 2020-2021

SCHOOL OF COMPUTER SCIENCE ENGINEERING

(SCOPE)

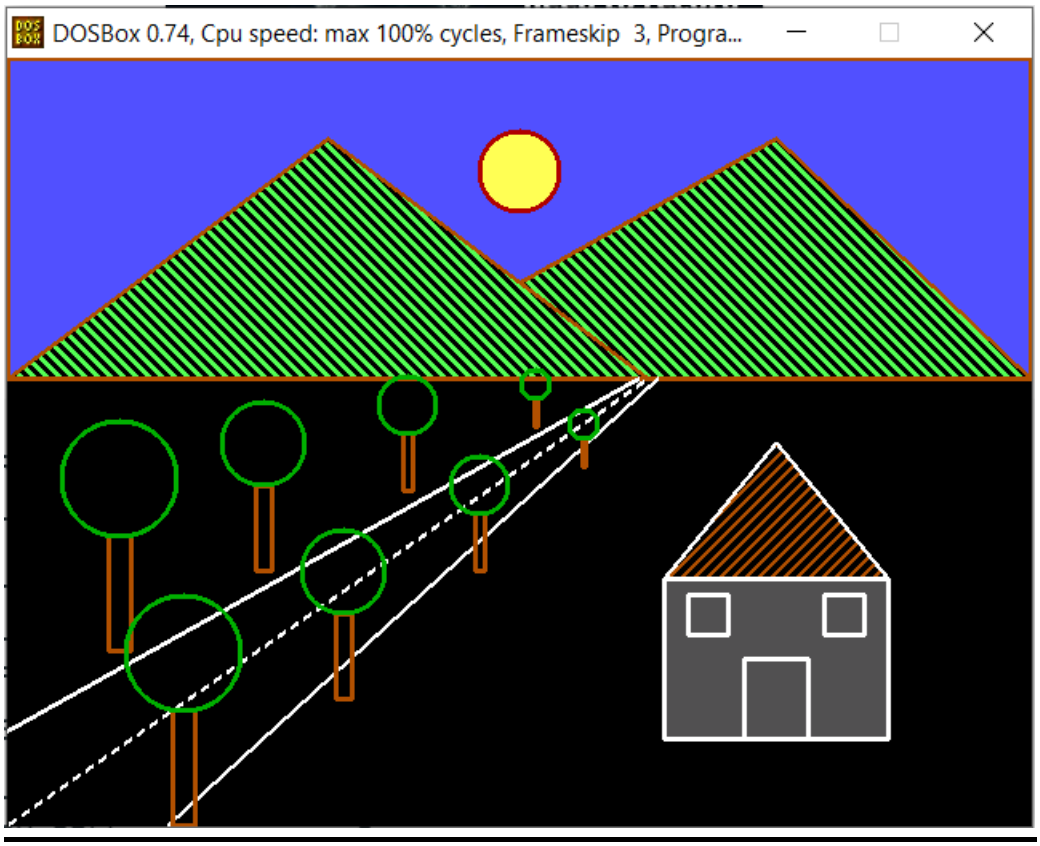
Computer Graphics and Multimedia

LAB 02

AMIT KUMAR

19BCE1281

SCENERY

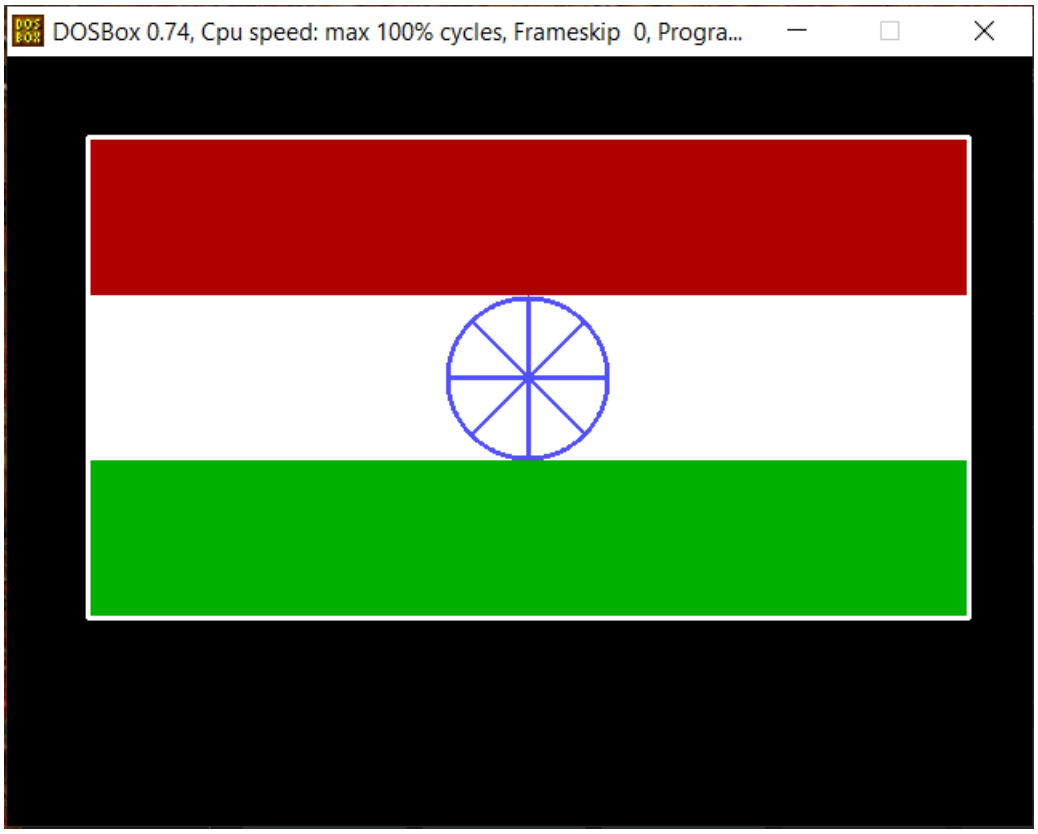



```c
#include<graphics.h>
#include<conio.h>
#include<stdio.h>

void drawtree(int x, int y, float ratio){
    int a = (int)6*ratio;
    int b = (int)60*ratio;
    int c = (int)90*ratio;
    setcolor(6);
    setlinestyle(0,1,3);
    line(x-a,y, x+a,y);
    line(x-a,y, x-a,y-b);
    line(x+a,y, x+a,y-b);
    line(x-a,y-b, x+a,y-b);
    setcolor(2);
    setfillstyle(10,14);
    circle(x,y-c,30*ratio);
    //floodfill(x,y-90,10);

}

int main(){
```

```
int arr[] = {0,200,200,50,400,200};
int arr2[] = {320, 140, 480,50,639,200,400,200,320, 140};
int arr_sky[] = {0,0,0,200,200,50,320,140,480,50,639,200,639,0,0,0};
int arr3[] = {410,425,550,425,550,325,410,325,410,425};
int arr4[] = {410, 325,480,240,550,325,410, 325};
int gd = DETECT, gm;
initgraph(&gd, &gm, "C:\\TURBOC3\\BGI");
//Start of mountain
//Mountain 1
setfillstyle(5,10);
setcolor(6);
setlinestyle(0,1,3);
line(0,200,200, 50);
line(200,50,400,200);
line(0,200,400,200);
fillpoly(3, arr);
//mountain 2
setcolor(6);
line(320,140,480,50);
line(480,50,639,200);
line(639,200,400,200);
line(400,200,320,140);
fillpoly(5, arr2);
//The sky
setfillstyle(1,9);
fillpoly(8, arr_sky);
// The sun
setcolor(4);
//arc(320,140,10,160, 70);
circle(320, 70, 25);
setfillstyle(1,14);
floodfill(320,70,RED);

//The river 479 is the max y heigth
setcolor(15);
line(394,200,0,420);
line(406,200,100,479);
setlinestyle(3,1,3);
line(400,200,0,479);
drawtree(70, 370,1.2);
drawtree(160, 320,0.9);
drawtree(250, 270,0.6);
drawtree(330, 230,0.3);
drawtree(110, 479,1.2);
drawtree(210, 400,0.9);
drawtree(295, 320,0.6);
drawtree(360, 255,0.3);
```

```
//Drawing the hut
setcolor(15);
line(410, 425,550,425);
line(410, 425,410,325);
line(410, 325,550,325);
line(550, 325,550,425);
line(410, 325,480,240);
line(480, 240,550,325);
setfillstyle(4, 6);
fillpoly(4,arr4);
setfillstyle(1, 8);
fillpoly(5,arr3);
//door {410,425,550,425,550,325,410,325,410,425}
setcolor(15);
line(460, 425, 460, 375);
line(500, 425, 500, 375);
line(460,375,500,375);
//window1
line(425,360,425,335);
line(450,360,450,335);
line(425,335,450,335);
line(425,360,450,360);
//window2
line(535,360,535,335);
line(510,360,510,335);
line(510,360,535,360);
line(510,335,535,335);
//setfillstyle(1, 8);
//fillpoly(5,arr3);
getch();
closegraph();
return 0;
}
```

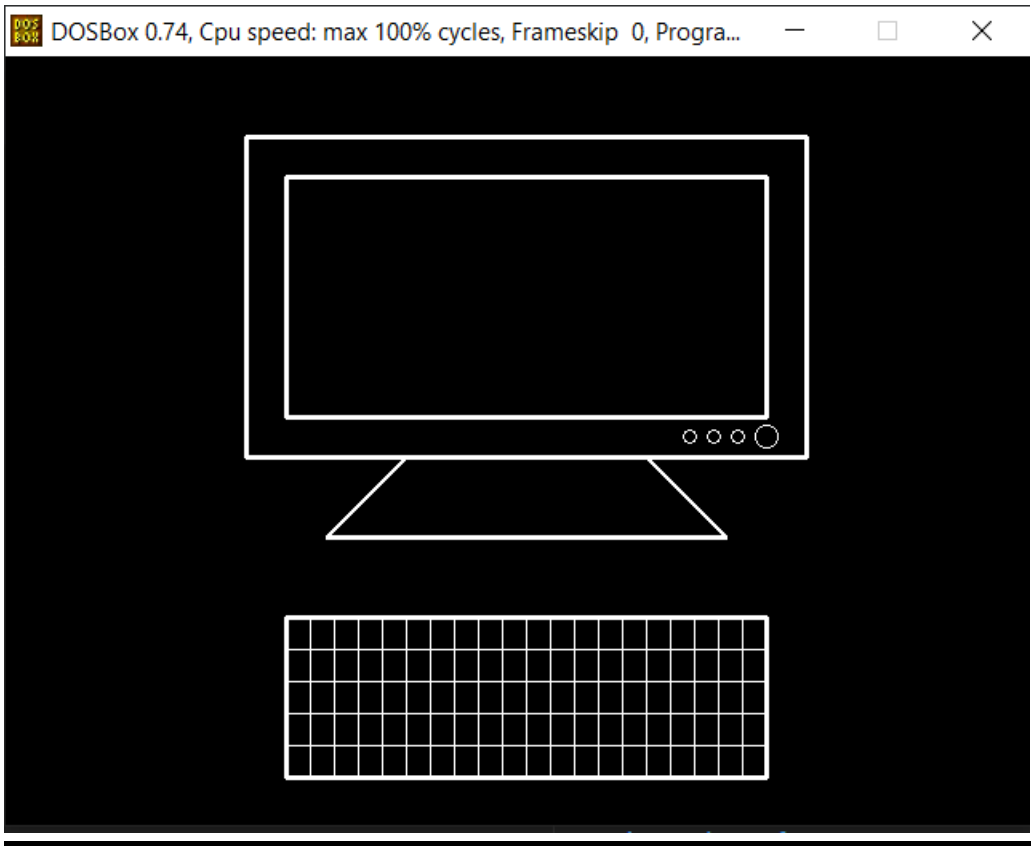
FLAG




```
#include<graphics.h>
#include<conio.h>
#include<stdio.h>

int main(){
    int s1[] = {50,50,600,50,600,150,50,150,50,50};
    int s2[] = {50,150,600,150,600,250,50,250,50,150};
    int s3[] = {50,250,600,250,600,350,50,350,50,250};
    int gd = DETECT, gm;
    initgraph(&gd, &gm, "C:\\TURBOC3\\BGI");
    setlinestyle(0,1,3);
    //Strip 1
    line(50,50,600,50);
    line(600,50,600,150);
    line(600,150,50,150);
    line(50,150,50,50);
    setfillstyle(1, 4);
    fillpoly(5,s1);

    //strip 2
```

```
    line(50,150,600,150);
    line(600,150,600,250);
    line(600,250,50,250);
    line(50,250,50,150);
    setfillstyle(1, 15);
    fillpoly(5,s2);
    //strip3
    setcolor(15);
    line(50,250,600,250);
    line(600,250,600,350);
    line(600,350,50,350);
    line(50,350,50,150);
    setfillstyle(1, 2);
    fillpoly(5,s3);
    //Wheel
    setcolor(9);
    circle(325,200,50);
    line(325,150,325,250);
    line(275,200,375,200);
    line(360,165,290,235);
    line(290,165,360,235);

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

FAVOURITE ITEM




```c
#include<graphics.h>
#include<conio.h>
#include<stdio.h>

int main(){
    int i,j;
    int gd = DETECT, gm;
    initgraph(&gd, &gm, "C:\\TURBOC3\\BGI");
    setcolor(15);
    setlinestyle(0,1,3);
    //outer box
    line(150,50,500,50);
    line(500,50,500,250);
    line(500,250,150,250);
    line(150,250,150,50);
    //stand
    line(250,250,200,300);
    line(400,250,450,300);
    line(200,300,450,300);
    //innerbox
```

```
    line(175,75,475,75);
    line(475,75,475,225);
    line(475,225,175,225);
    line(175,225,175,75);
    //buttons
    setlinestyle(0,1,1);
    circle(475,237.5,7);
    circle(457,237.5,4);
    circle(442,237.5,4);
    circle(427,237.5,4);

    //keyboard
    setlinestyle(0,1,3);
    line(175,350,175,450);
    line(175,350,475,350);
    line(475,350,475,450);
    line(475,450,175,450);
    //keys
    setlinestyle(0,1,1);
    for(i = 1; i<=20;i++){
        int x1 = 175;
        x1 +=i*15;
        line(x1,350,x1,450);
    }
    for(j = 1; j<=4;j++){
        int y1 = 350;
        y1 +=j*20;
        line(175,y1,475,y1);
    }
    getch();
    closegraph();
    return 0;
}
```

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

B. Tech. Semester 2020-2021

SCHOOL OF COMPUTER SCIENCE ENGINEERING

(SCOPE)

Graphics and Multimedia

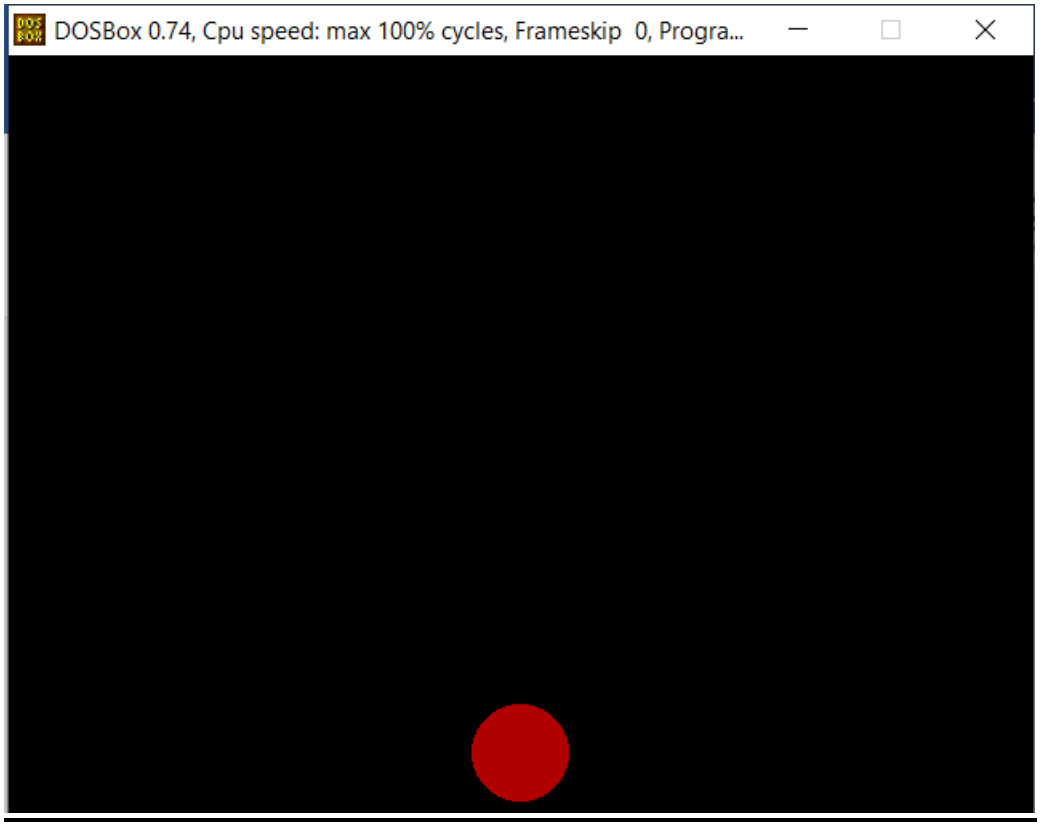
LAB 03

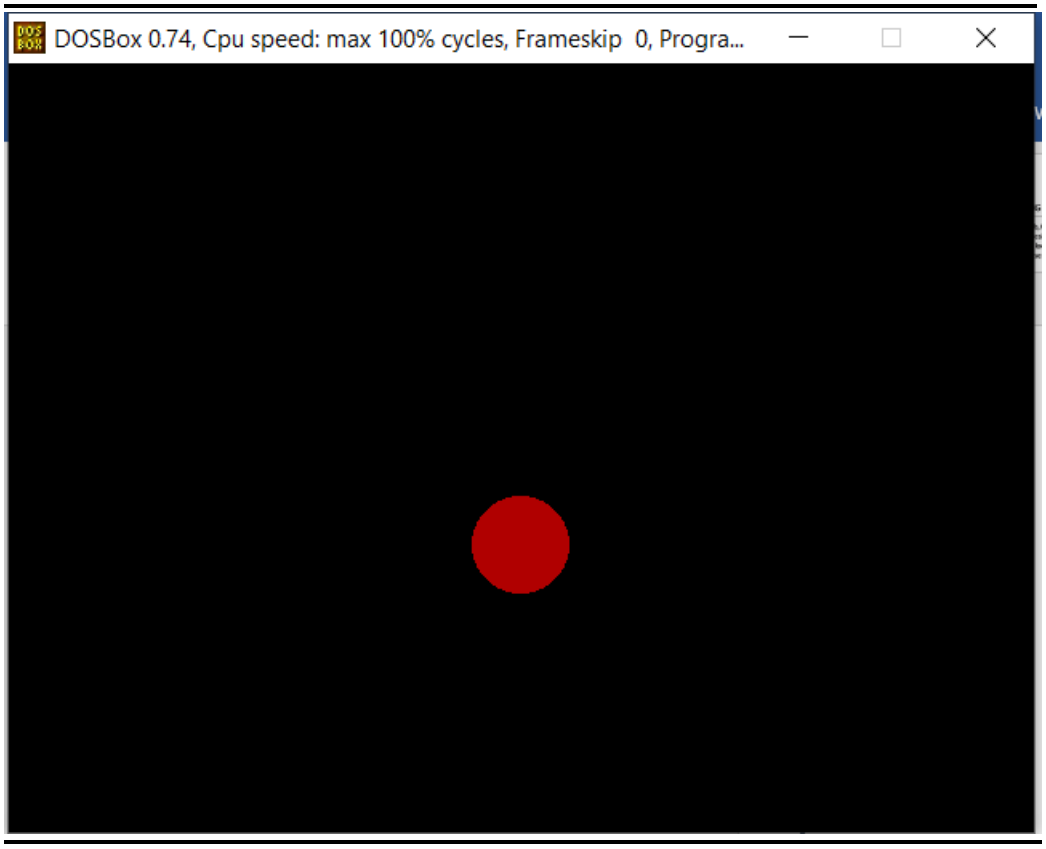
AMIT KUMAR

19BCE1281

Bouncing Ball

```
#include <stdio.h>
#include <conio.h>
#include <graphics.h>
#include <dos.h>

int main() {
 int gd = DETECT, gm;
 int i, x, y, flag=0;
 initgraph(&gd, &gm, "C:\\TURBOC3\\BGI");

 x = getmaxx()/2;
 y = 30;


 while (!kbhit()) {
  if(y >= getmaxy()-30 || y <= 30)
     flag = !flag;
     setcolor(RED);
     setfillstyle(SOLID_FILL, RED);
     circle(x, y, 30);
     floodfill(x, y, RED);
 delay(50);

 cleardevice();
 if(flag){
     y = y + 5;
 } else {
     y = y - 5;
 }
    }

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

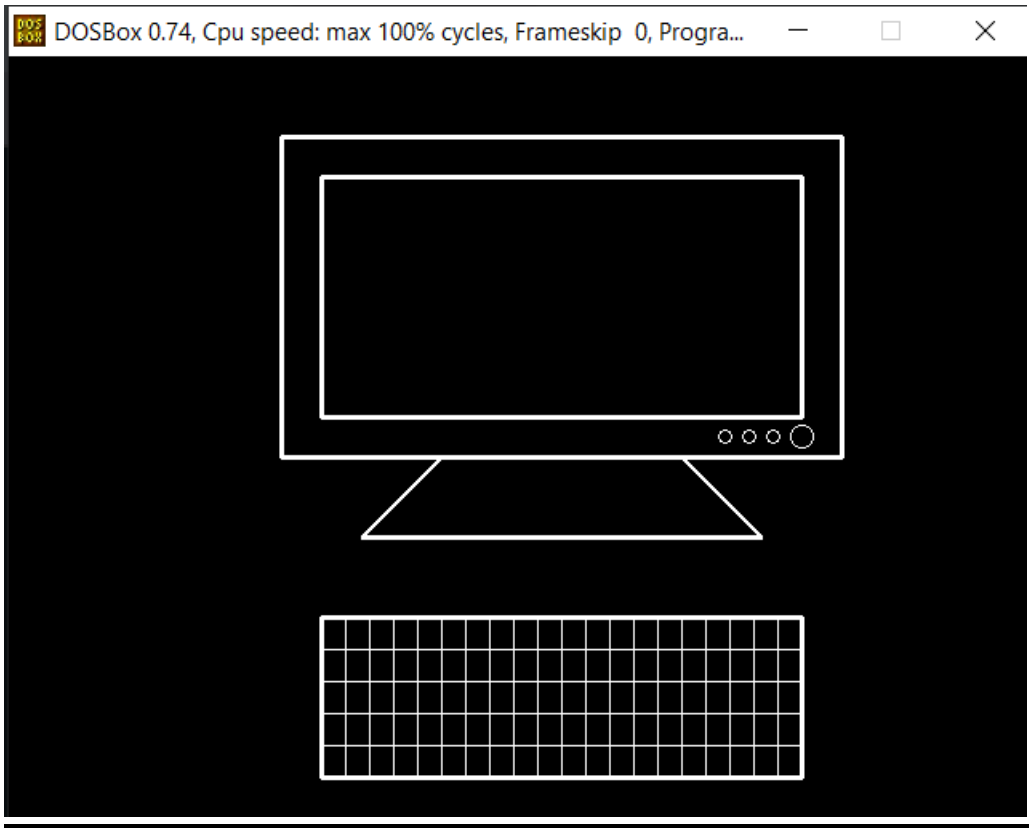

DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Progra...

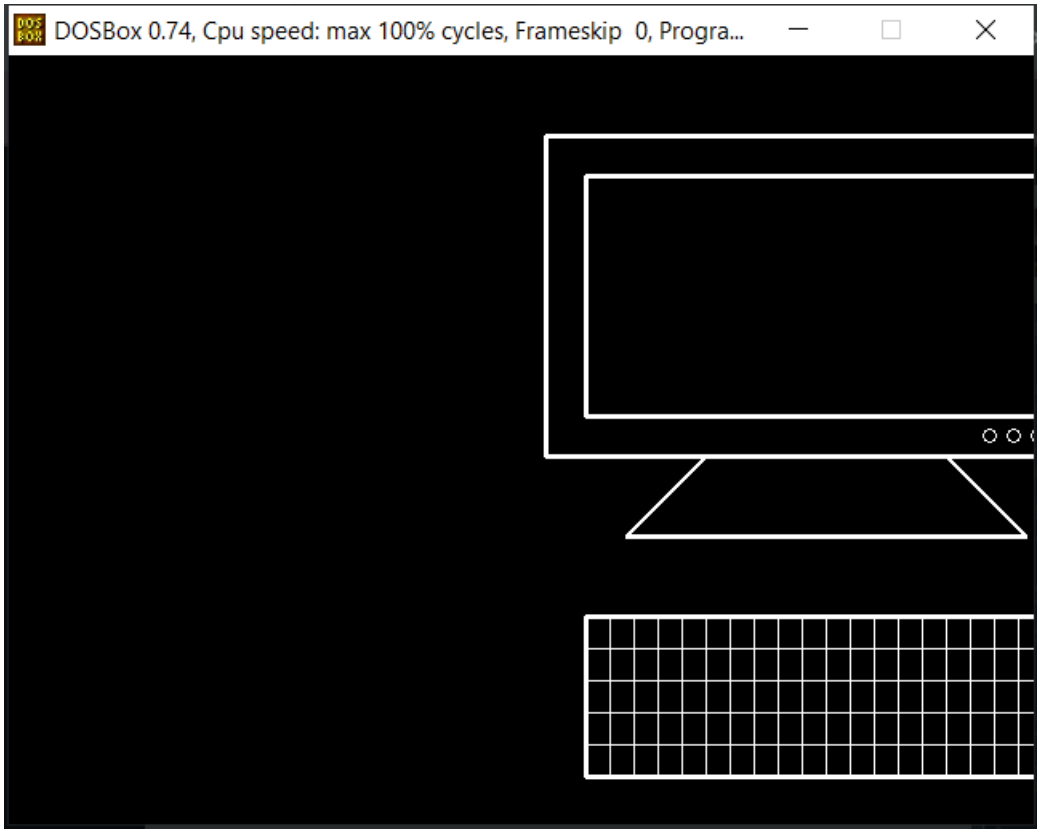

DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Progra...

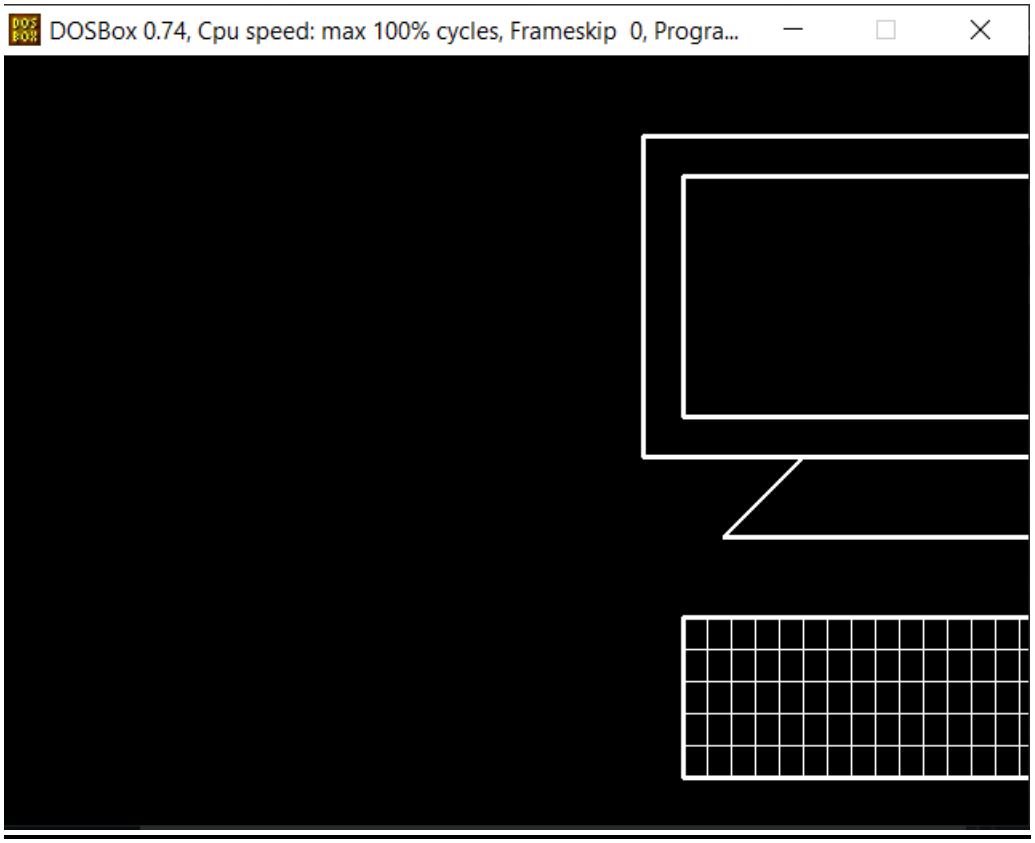
Moving Computer

```
#include<stdio.h>

#include<conio.h>

#include<graphics.h>

void main()

{
int i,j,k;
int gd = DETECT, gm;
initgraph(&gd, &gm, "C:\\TURBOC3\\BGI");


for(i=0;i<250;i++){
cleardevice();
setcolor(15);
    setlinestyle(0,1,3);
    //outer box
    line(150+i,50,500+i,50);
    line(500+i,50,500+i,250);
    line(500+i,250,150+i,250);
    line(150+i,250,150+i,50);
    //stand
    line(250+i,250,200+i,300);
    line(400+i,250,450+i,300);
    line(200+i,300,450+i,300);
    //innerbox
    line(175+i,75,475+i,75);
    line(475+i,75,475+i,225);
    line(475+i,225,175+i,225);
    line(175+i,225,175+i,75);
    //buttons
    setlinestyle(0,1,1);
    circle(475+i,237.5,7);
    circle(457+i,237.5,4);
    circle(442+i,237.5,4);
    circle(427+i,237.5,4);

    //keyboard
    setlinestyle(0,1,3);
    line(175+i,350,175+i,450);
    line(175+i,350,475+i,350);
    line(475+i,350,475+i,450);
    line(475+i,450,175+i,450);
```

```
    //keys
    setlinestyle(0,1,1);
    for(k = 1; k<=20;k++){
        int x1 = 175;
        x1 +=k*15;
        line(x1+i,350,x1+i,450);
    }
    for(j = 1; j<=4;j++){
        int y1 = 350;
        y1 +=j*20;
        line(175+i,y1,475+i,y1);
    }
delay(50);
}
getch();
closegraph();
}
```



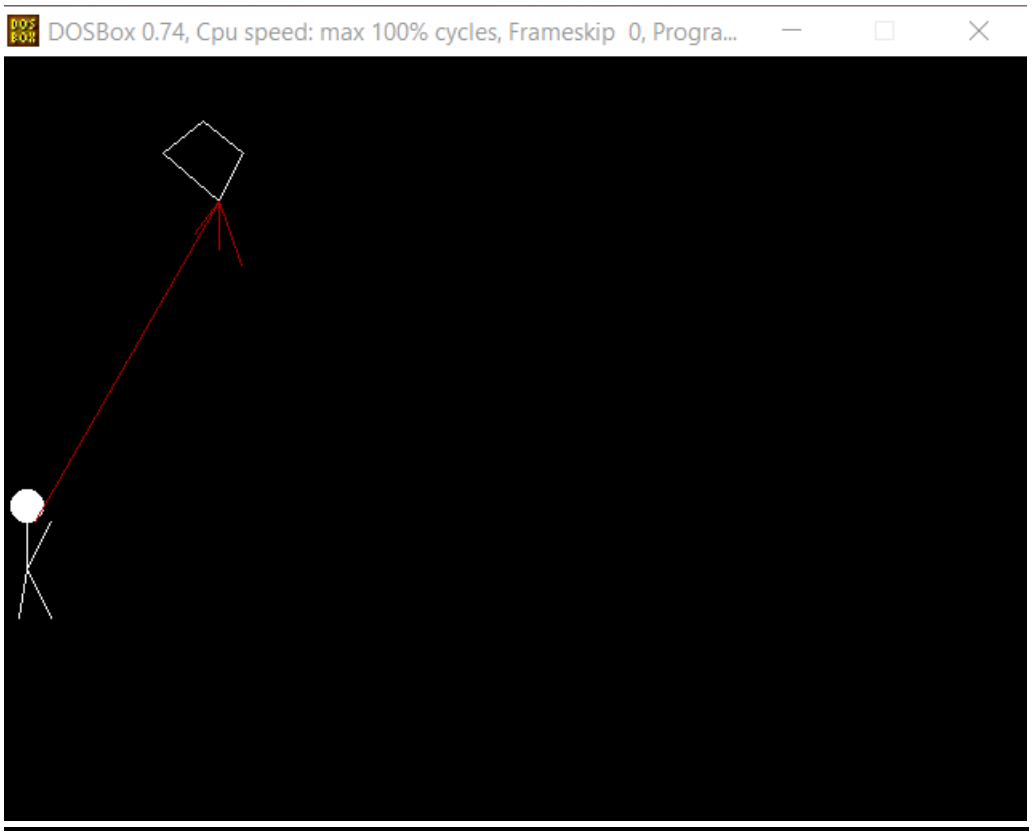

DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Progra...

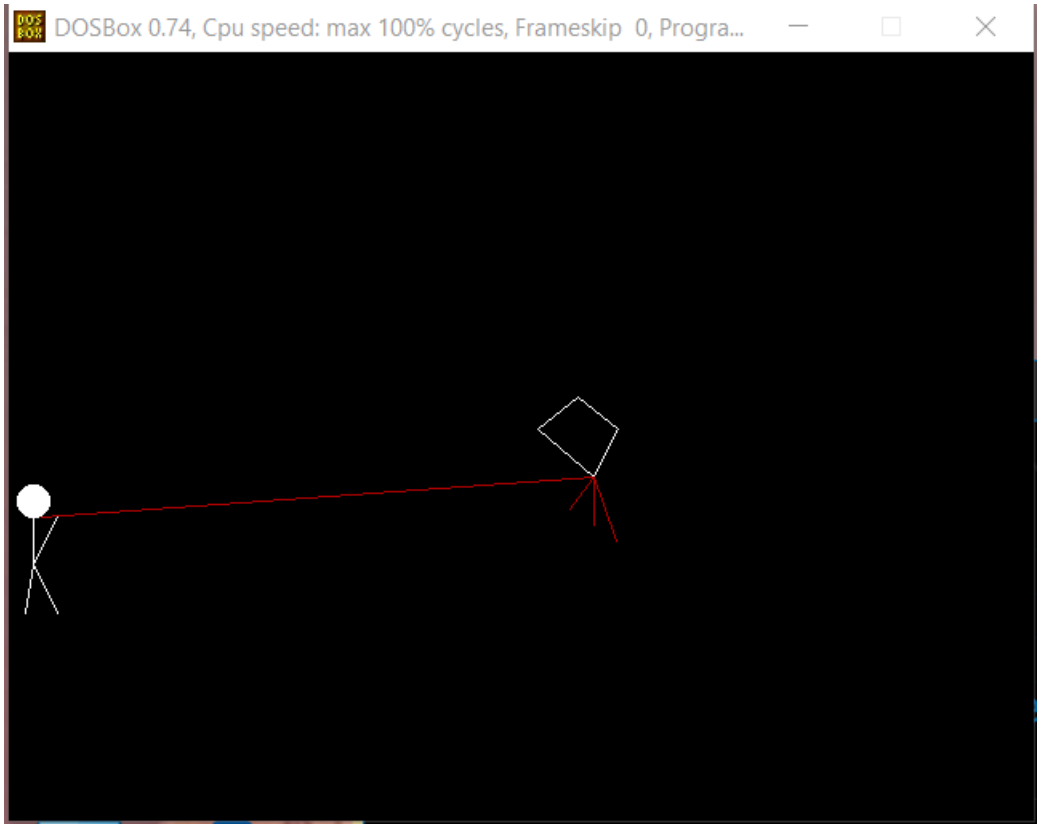

DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Progra...

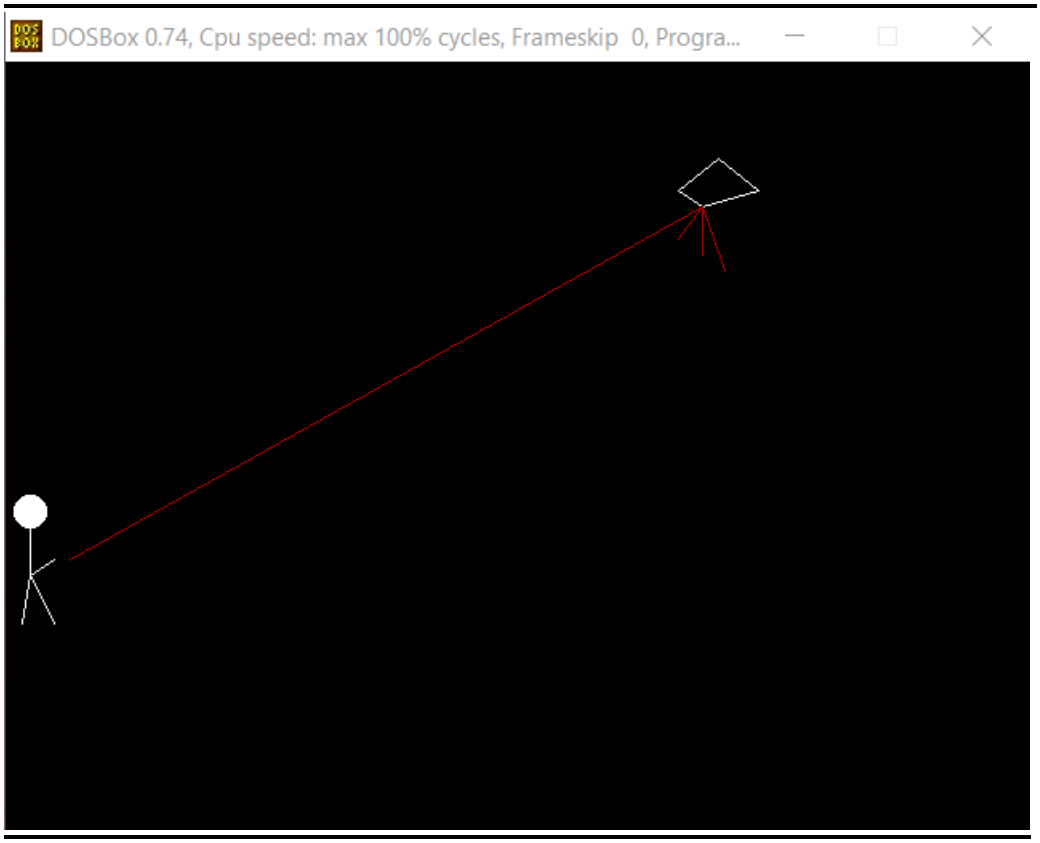
FLYING KITE

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

void main()
{
int gd=DETECT,gm;
int poly[10]; int flag=10,f=10;
int x=5,y=5,x_inc=0,y_inc=0;
initgraph(&gd, &gm, "C:\\TURBOC3\\BGI");
while(!kbhit())
{
setcolor(WHITE);
if(x<20)
    x_inc+=5;
if(x > 400)
    x_inc-=5;
if(y>300)
    y_inc-=5;
if(y<50)
    y_inc+=5;
    x=x+x_inc;
    y=y+y_inc;
 poly[0]=100+x+f;
 poly[1]=50+y+f;
 poly[2]=125+x+f;
  poly[3]=30+y+f;
 poly[4]=150+x+f;
  poly[5]=50+y+f;
 poly[6]=125+x;
 poly[7]=70+y;
 poly[8]=100+x+f;
 poly[9]=50+y+f;
drawpoly(5,poly);
fillellipse(15,280,10,10);
line(15,280,15,320);
line(15,320,10,350);
line(15,320,30,350);
line(15,320,30,300+flag); //hand
setcolor(RED);
line(30+f,300+flag,125+x,70+y);   //rope
line(125+x,70+y,110+x,90+y);
 line(125+x,70+y,125+x,100+y);
```

```
 line(125+x,70+y,139+x,110+y);
 flag=-flag;f=-f;
delay(200); // x++;y++;
cleardevice();

}
}
```




DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Progra...


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Progra...

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

B. Tech. Semester 2020-2021

SCHOOL OF COMPUTER SCIENCE ENGINEERING

(SCOPE)

Computer Graphics and Multimedia

LAB 04

AMIT KUMAR

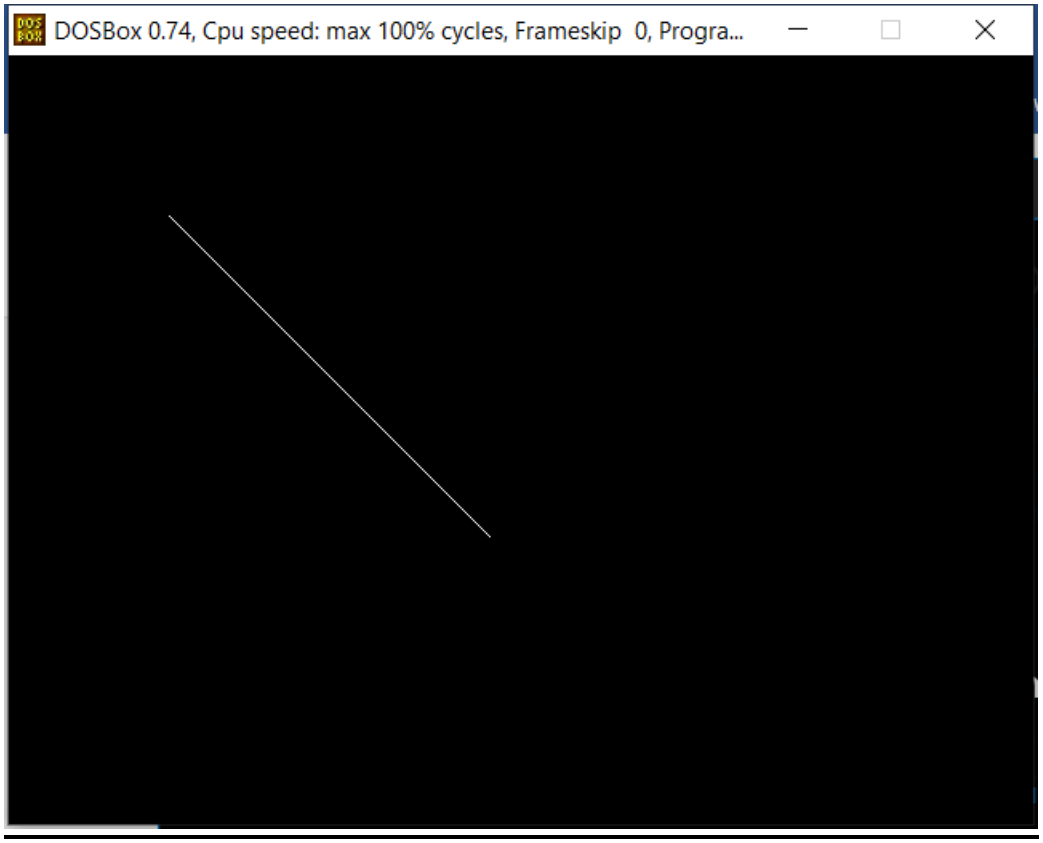
19BCE1281

Line Drawing Algorithm 1

Code

```
#include <stdio.h>
#include <conio.h>
#include <graphics.h>
#include <dos.h>

void LineDDA(int x0, int y0, int x1, int y1){
    int dx = x1-x0, dy = y1-y0, steps,i;
    double xIncrement,yIncrement,x,y;
    if(abs(dx)>abs(dy)){
        steps = abs(dy);
    }
    else{
        steps = abs(dy);
    }

    xIncrement = (double)dx/(double)steps;
    yIncrement = (double)dy/(double)steps;

    x = x0;
    y = y0;
    putpixel((int)x, (int)y,WHITE);

    for(i = 0;i<steps;i++){
        x+=xIncrement;
        y+=yIncrement;
        putpixel((int)x, (int)y, WHITE);
    }
}

int main() {
 int gd = DETECT, gm;
 initgraph(&gd, &gm, "C:\\TURBOC3\\BGI");
 LineDDA(100,100,300,300);
 getch();
 closegraph();
 return 0;
}
```

OUTPUT



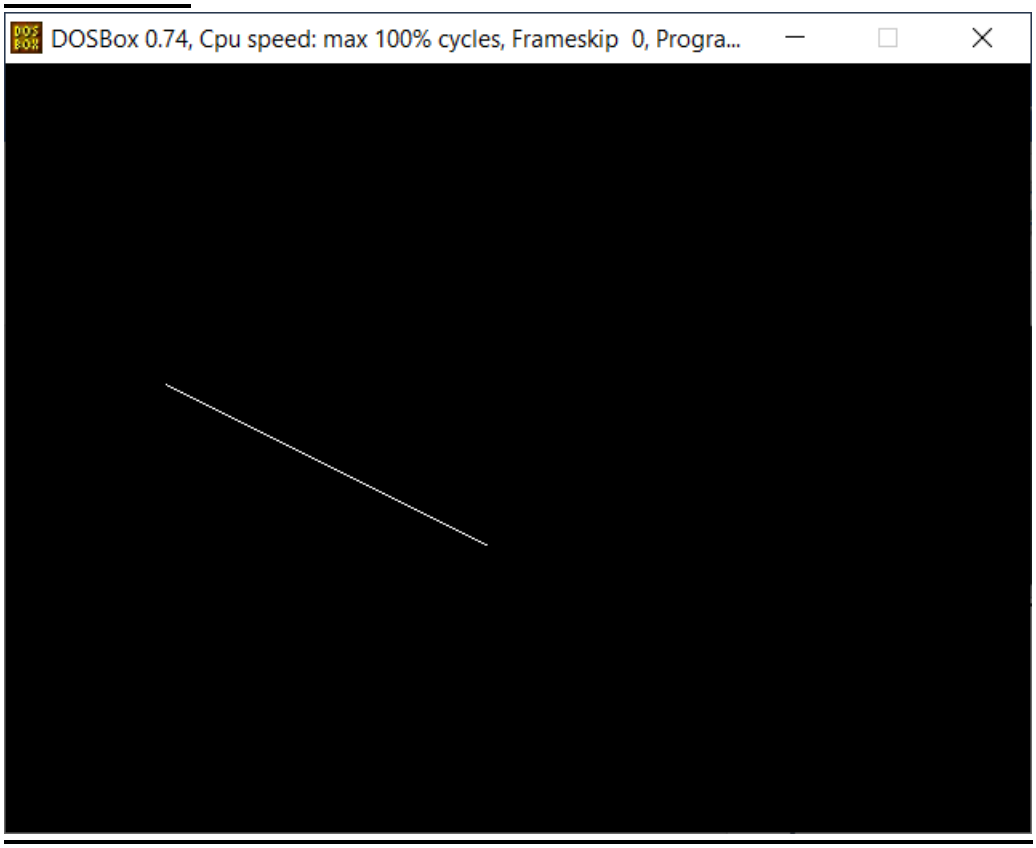
Line Drawing algorithm 2

CODE

```
#include <stdio.h>
#include <conio.h>
#include <graphics.h>
#include <dos.h>

void LineBres(int x0, int y0, int x1, int y1){
    int dx,dy,d,twoDy,twoDyMinusDx,x,y;
    dx = abs(x1-x0);
    dy = abs(y1-y0);
    d = 2*dy-dx;
    twoDy = 2*dy;
    twoDyMinusDx = 2*(dy-dx);
    if(x0>x1){
        x = x1;
        y = y1;
        x1 = x0;
    }
    else{
        x = x0;
        y = y0;
    }
    putpixel(x,y,WHITE);
    while(x<x1){
        x++;
        if(d<0){
            d+=twoDy;
        }
        else{
            y++;
            d+=twoDyMinusDx;
        }
        putpixel(x,y,WHITE);
    }
}

int main() {
 int gd = DETECT, gm;
 initgraph(&gd, &gm, "C:\\TURBOC3\\BGI");
 LineBres(100,100,300,300);
 getch();
 closegraph();
 return 0;
```

```
}
```

OUTPUT



VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

B. Tech. Semester 2020-2021

SCHOOL OF COMPUTER SCIENCE ENGINEERING

(SCOPE)

Graphics and Multimedia

LAB 05

AMIT KUMAR

19BCE1281

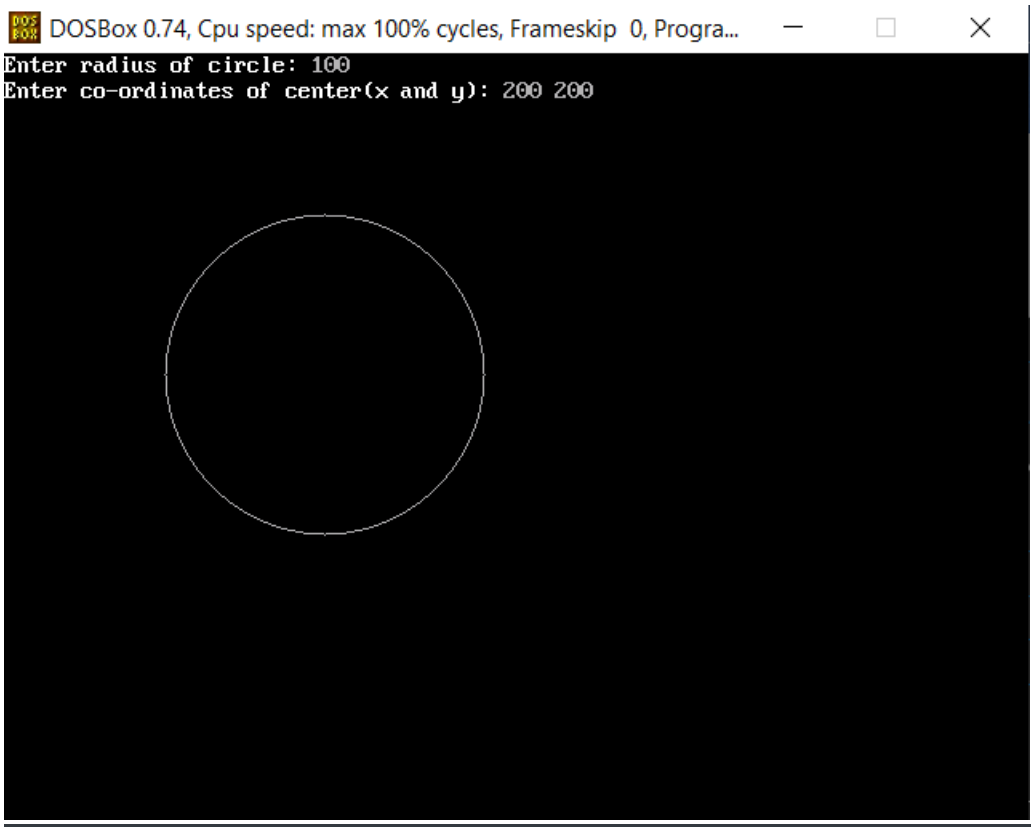
Midpoint circle drawing

Code

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

void drawcircle(int x0, int y0, int radius)
{
    int x = radius;
    int y = 0;
    int err = 0;

    while (x >= y)
    {
    putpixel(x0 + x, y0 + y, 7);
    putpixel(x0 + y, y0 + x, 7);
    putpixel(x0 - y, y0 + x, 7);
    putpixel(x0 - x, y0 + y, 7);
    putpixel(x0 - x, y0 - y, 7);
    putpixel(x0 - y, y0 - x, 7);
    putpixel(x0 + y, y0 - x, 7);
    putpixel(x0 + x, y0 - y, 7);

    if (err <= 0)
    {
        y += 1;
        err += 2*y + 1;
    }

    if (err > 0)
    {
        x -= 1;
        err -= 2*x + 1;
    }
    }
}

int main()
{
    int gdriver=DETECT, gmode, error, x, y, r;
    initgraph(&gdriver, &gmode, "c:\\turboc3\\BGI");

    printf("Enter radius of circle: ");
    scanf("%d", &r);

    printf("Enter co-ordinates of center(x and y): ");
```

```
    scanf("%d%d", &x, &y);
    drawcircle(x, y, r);
    getch();
    closegraph();

    return 0;
}
```

Results



Liang- Barsky Line clipping

Code

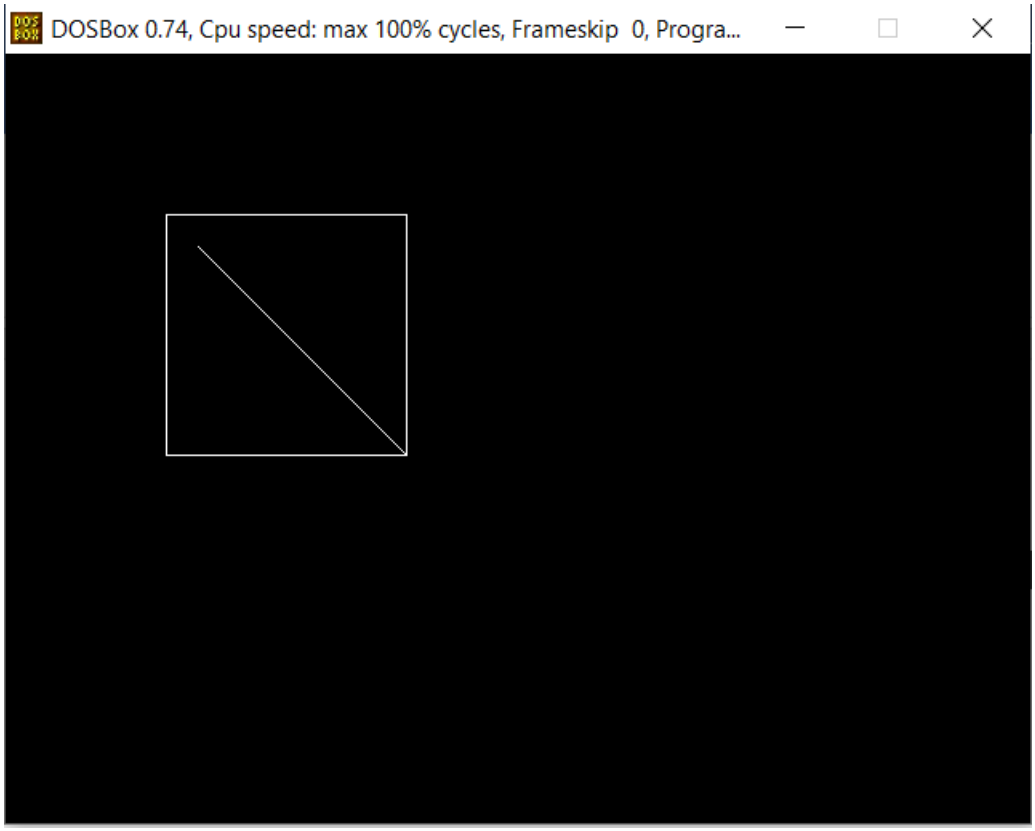
```
#include<stdio.h>
#include<graphics.h>
#include<math.h>
#include<dos.h>

void main()
{
    int i,gd=DETECT,gm;
    int x1,y1,x2,y2,xmin,xmax,ymin,ymax,xx1,xx2,yy1,yy2,dx,dy;
    float t1,t2,p[4],q[4],temp;
```

```
x1=120;
y1=120;
x2=300;
y2=300;

xmin=100;
ymin=100;
xmax=250;
ymax=250;

initgraph(&gd,&gm,"c:\\turboc3\\BGI");
rectangle(xmin,ymin,xmax,ymax);
dx=x2-x1;
dy=y2-y1;

p[0]=-dx;
p[1]=dx;
p[2]=-dy;
p[3]=dy;

q[0]=x1-xmin;
q[1]=xmax-x1;
q[2]=y1-ymin;
q[3]=ymax-y1;

for(i=0;i<4;i++)
{
    if(p[i]==0)
    {
        printf("line is parallel to one of the clipping boundary");
        if(q[i]>=0)
        {
            if(i<2)
            {
                if(y1<ymin)
                {
                    y1=ymin;
                }

                if(y2>ymax)
                {
                    y2=ymax;
                }

                line(x1,y1,x2,y2);
            }
```

```
            if(i>1)
            {
                if(x1<xmin)
                {
                    x1=xmin;
                }

                if(x2>xmax)
                {
                    x2=xmax;
                }

                line(x1,y1,x2,y2);
            }
        }
    }
}

t1=0;
t2=1;

for(i=0;i<4;i++)
{
    temp=q[i]/p[i];

    if(p[i]<0)
    {
        if(t1<=temp)
            t1=temp;
    }
    else
    {
        if(t2>temp)
            t2=temp;
    }
}

if(t1<t2)
{
    xx1 = x1 + t1 * p[1];
    xx2 = x1 + t2 * p[1];
    yy1 = y1 + t1 * p[3];
    yy2 = y1 + t2 * p[3];
    line(xx1,yy1,xx2,yy2);
}

delay(5000);
closegraph();
```

```
}
```

Result



Weiler Atherton Polygon clipping

Code

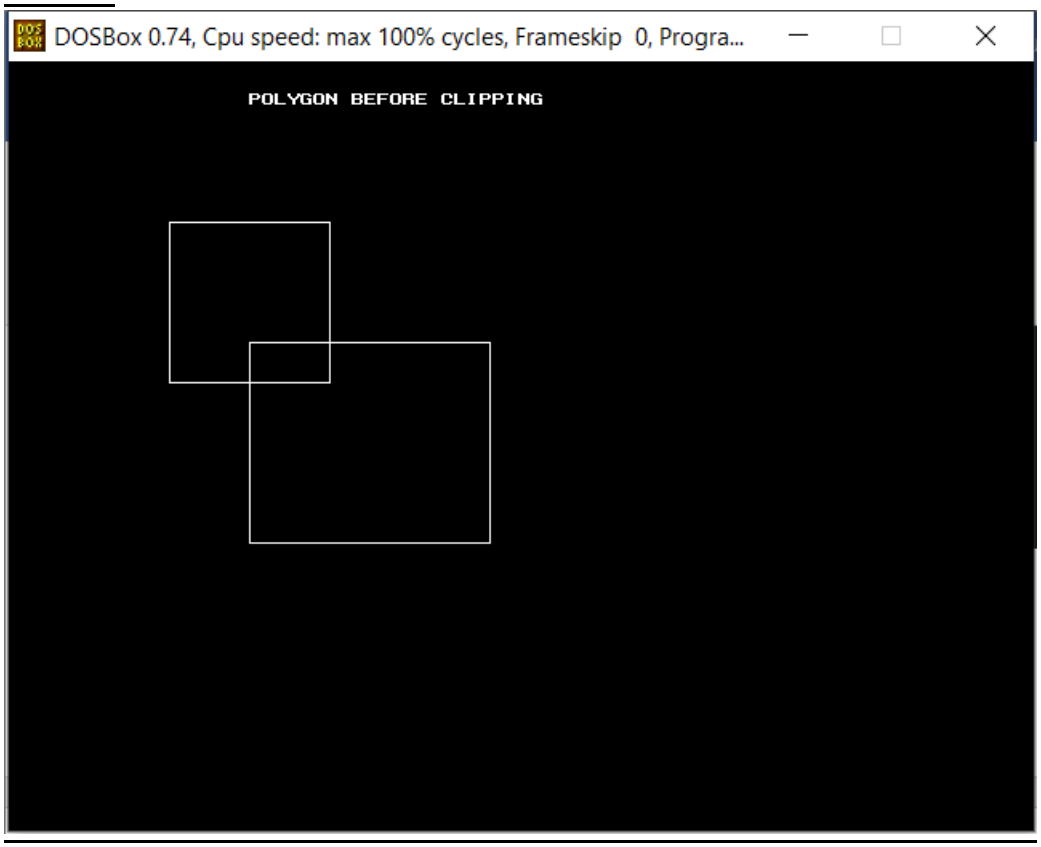
```
#include <stdio.h>
#include <graphics.h>
#include <conio.h>
#include <math.h>
#include <process.h>
#define TRUE 1
#define FALSE 0
typedef unsigned int outcode;
outcode CompOutCode(float x,float y);
enum  {  TOP = 0x1,
BOTTOM = 0x2,
RIGHT = 0x4,
LEFT = 0x8
};
float xmin,xmax,ymin,ymax;
void clip(float x0,float y0,float x1,float y1)
```

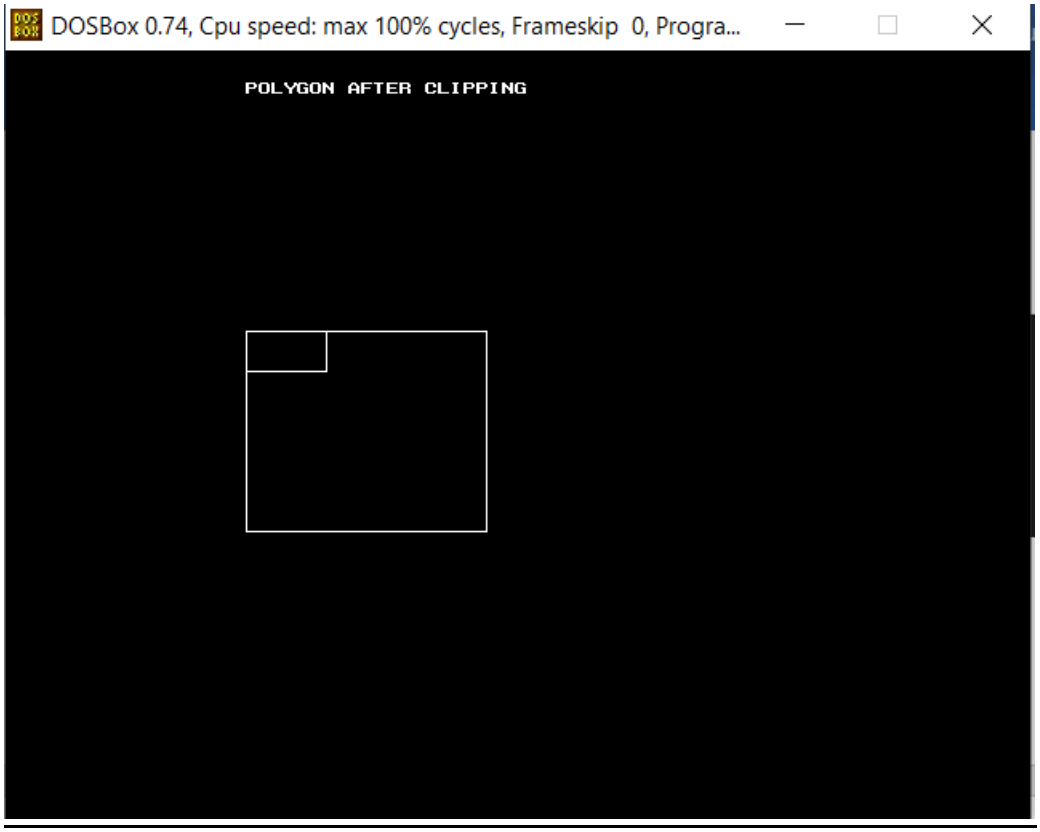
```
{
outcode outcode0,outcode1,outcodeOut;
int accept = FALSE,done = FALSE;
outcode0 = CompOutCode(x0,y0);
outcode1 = CompOutCode(x1,y1);
do
{
    if(!(outcode0|outcode1))
    {
        accept = TRUE;
        done = TRUE;
    }
    else
    if(outcode0 & outcode1)
        done = TRUE;
    else
    {
        float x,y;
        outcodeOut = outcode0?outcode0:outcode1;
        if(outcodeOut & TOP)
        {
            x = x0+(x1-x0)*(ymax-y0)/(y1-y0);
            y = ymax;
        }
        else if(outcodeOut & BOTTOM)
        {
            x = x0+(x1-x0)*(ymin-y0)/(y1-y0);
            y = ymin;
        }
        else if(outcodeOut & RIGHT)
        {
            y = y0+(y1-y0)*(xmax-x0)/(x1-x0);
            x = xmax;
        }
        else
        {
            y = y0+(y1-y0)*(xmin-x0)/(x1-x0);
            x = xmin;
        }
        if(outcodeOut==outcode0)
        {
            x0 = x;
            y0 = y;
            outcode0 = CompOutCode(x0,y0);
        }
        else
        {
            x1 = x;
```

```
            y1 = y;
            outcode1 = CompOutCode(x1,y1);
          }
    }
}while(done==FALSE);
if(accept)
    line(x0,y0,x1,y1);
outtextxy(150,20,"POLYGON AFTER CLIPPING");
rectangle(xmin,ymin,xmax,ymax);
}
outcode CompOutCode(float x,float y)
{
    outcode code = 0;
    if(y>ymax)
        code|=TOP;
    else if(y<ymin)
            code|=BOTTOM;
    if(x>xmax)
        code|=RIGHT;
    else if(x<xmin)
        code|=LEFT;
    return code;
}
void main( )
{
float x1,y1,x2,y2;
int gdriver = DETECT, gmode, n,poly[14],i;
clrscr( );
printf("Enter the no of sides of polygon:");
scanf("%d",&n);
printf("\nEnter the coordinates of polygon\n");
for(i=0;i<2*n;i++)
{
    scanf("%d",&poly[i]);
}
poly[2*n]=poly[0];
poly[2*n+1]=poly[1];
printf("Enter the rectangular coordinates of clipping window\n");
scanf("%f%f%f%f",&xmin,&ymin,&xmax,&ymax);
initgraph(&gdriver, &gmode, "c:\\turboc3\\BGI");

outtextxy(150,20,"POLYGON BEFORE CLIPPING");
drawpoly(n+1,poly);
rectangle(xmin,ymin,xmax,ymax);
getch( );
cleardevice( );
for(i=0;i<n;i++)
clip(poly[2*i],poly[(2*i)+1],poly[(2*i)+2],poly[(2*i)+3]);
```

```
getch( );
restorecrtmode( );
}
```

Result




DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Progra...
POLYGON AFTER CLIPPING

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

B. Tech. Semester 2020-2021

SCHOOL OF COMPUTER SCIENCE ENGINEERING

(SCOPE)

Graphics and Multimedia

LAB 06

AMIT KUMAR

19BCE1281

2-D Transformation

Code

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
#include<graphics.h>
int ch,x,y,az,i,w,ch1,ch2,xa,ya,ra,a[10],b[10],da,db;
float x1,y1,az1,w1,dx,dy,theta,x1s,y1s,sx,sy,a1[10],b1[10];
void main()
{
    int gm ,gr;
    clrscr();
    detectgraph(&gm,&gr);
    initgraph(&gm,&gr,"c:\\turboc3\\BGI");
    printf("Enter the upper left corner of the rectangle:\n");
    scanf("%d%d",&x,&y);
    printf("Enter the lower right corner of the rectangle:\n");
    scanf("%d%d",&az,&w);
    rectangle(x,y,az,w);
    da=az-x;
    db=w-y;
    a[0]=x;
    b[0]=y;
    a[1]=x+da;
    b[1]=y;
    a[2]=x+da;
    b[2]=y+db;
    a[3]=x;b[3]=y+db;
    while(1)
    {
        printf("******2D Transformations*******\n");
        printf("1.Translation\n2.Rotation\n3.Scaling\n4.Reflection\n5.Shearing
\n6.Exit\nEnter your choice:\n");
        scanf("%d",&ch);
        switch(ch)
    {
    case 1:
        detectgraph(&gm,&gr);
        initgraph(&gm,&gr,"c:\\turboc3\\BGI");
        rectangle(x,y,az,w);
        printf("*******Translation*******\n\n");
        printf("Enter the value of shift vector:\n");
        scanf("%f%f",&dx,&dy);
        x1=x+dx;
        y1=y+dy;
        az1=az+dx;
        w1=w+dy;
        rectangle(x1,y1,az1,w1);
```

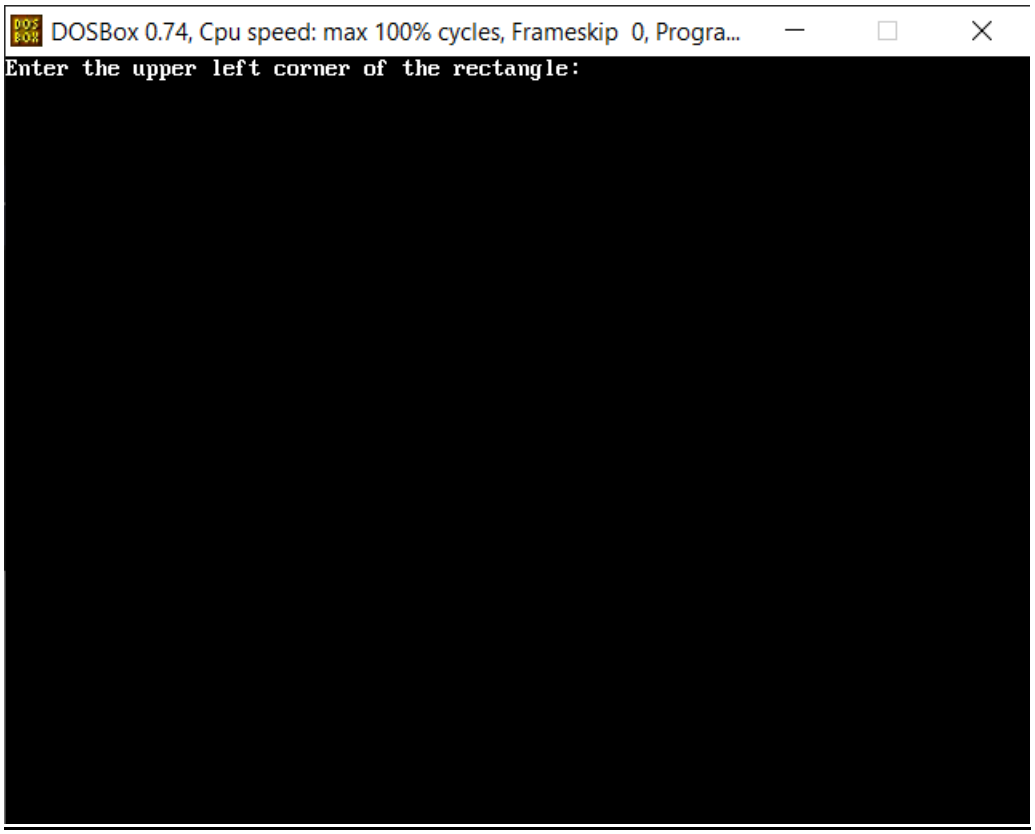
```
        break;
    case 2:
        detectgraph(&gm,&gr);
        initgraph(&gm,&gr,"c:\\turboc3\\BGI");
        rectangle(x,y,az,w);
        printf("*******Rotation*******\n\n");
        printf("Enter the value of fixed point and angle of rotation:Enter the
 value of fixed point and angle of rotation:\n");
        scanf("%d%d%d",&xa,&ya,&ra);
        theta=(float)(ra*(3.14/180));
        for(i=0;i<4;i++)
        {
            a1[i]=(xa+((a[i]-xa)*cos(theta)-(b[i]-ya)*sin(theta)));
            b1[i]=(ya+((a[i]-xa)*sin(theta)+(b[i]-ya)*cos(theta)));
        }
        for(i=0;i<4;i++)
        {
            if(i!=3)
                line(a1[i],b1[i],a1[i+1],b1[i+1]);
            else
                line(a1[i],b1[i],a1[0],b1[0]);
        }
        break;
    case 3:
        detectgraph(&gm,&gr);
        initgraph(&gm,&gr,"c:\\turboc3\\BGI");
        rectangle(x,y,az,w);
        printf("********Scaling*******\n\n");
        printf("Enter the value of scaling factor:\n");
        scanf("%f%f",&sx,&sy);
        x1=x*sx;
        y1=y*sy;
        az1=az*sx;
        w1=w*sy;
        rectangle(x1,y1,az1,w1);
        break;
    case 4:
        detectgraph(&gm,&gr);
        initgraph(&gm,&gr,"c:\\turboc3\\BGI");
        rectangle(x,y,az,w);
        printf("*******Reflection*********\n");
        printf("1.About x-axis\n2.About y-
axis\n3.About both axis\nEnter your choice:\n");
        scanf("%d",&ch1);
        switch(ch1)
        {
            case 1:
            printf("Enter the fixed point\n");
```

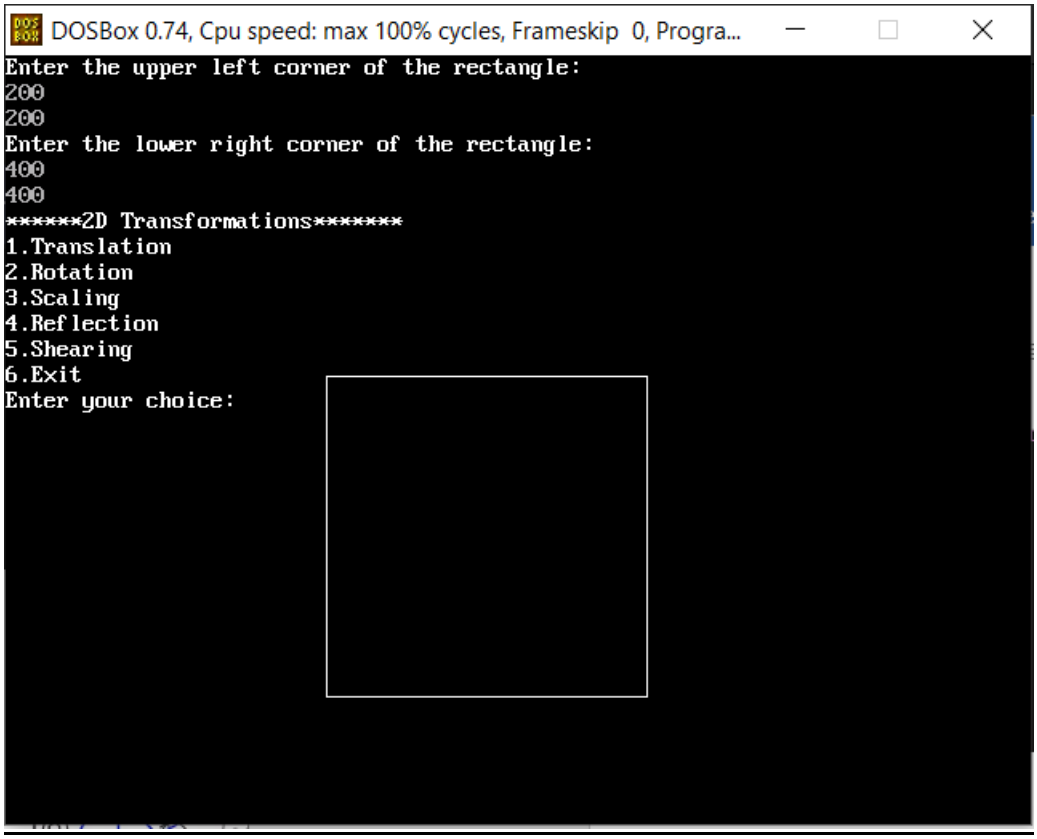
```
            scanf("%d%d",&xa,&ya);
            theta=(float)(90*(3.14/180));
            for(i=0;i<4;i++)
            {
                a1[i]=(xa+((a[i]-xa)*cos(theta)-(-b[i]-ya)*sin(theta)));
                b1[i]=(ya+((a[i]-xa)*sin(theta)+(-b[i]-ya)*cos(theta)));
            }
            for(i=0;i<4;i++)
            {
                if(i!=3)
                    line(a1[i],b1[i],a1[i+1],b1[i+1]);
                else
                    line(a1[i],b1[i],a1[0],b1[0]);
            }
            break;
            case 2:
                printf("Enter the fixed point\n");
                scanf("%d%d",&xa,&ya);
                theta=(float)(270*(3.14/180));
                for(i=0;i<4;i++)
                {
                    a1[i]=(xa+((-a[i]-xa)*cos(theta)-(b[i]-ya)*sin(theta)));
                    b1[i]=(ya+((-a[i]-xa)*sin(theta)+(b[i]-ya)*cos(theta)));
                }
                for(i=0;i<4;i++)
                {
                    if(i!=3)
                        line(a1[i],b1[i],a1[i+1],b1[i+1]);
                    else
                        line(a1[i],b1[i],a1[0],b1[0]);
                }
                break;
            case 3:
                printf("Enter the fixed point\n");
                scanf("%d%d",&xa,&ya);
                theta=(float)(180*(3.14/180));
                for(i=0;i<4;i++)
                {
                    a1[i]=(xa+((-a[i]-xa)*cos(theta)-(-b[i]-ya)*sin(theta)));
                    b1[i]=(ya+((-a[i]-xa)*sin(theta)+(-b[i]-ya)*cos(theta)));
                }
                for(i=0;i<4;i++)
                {
                    if(i!=3)
                        line(a1[i],b1[i],a1[i+1],b1[i+1]);
                    else
                        line(a1[i],b1[i],a1[0],b1[0]);
                }
```

```
                break;
        }
        break;
    case 5:
        detectgraph(&gm,&gr);
        initgraph(&gm,&gr,"c:\\turboc3\\BGI");
        rectangle(x,y,az,w);
        printf("*******Shearing******\n\n");
        printf("1.x-direction shear\n2.y-
direction shear\nEnter your choice:\n");
        scanf("%d",&ch2);
        switch(ch2)
        {
            case 1:
                printf("Enter the value of shear:\n");
                scanf("%f",&x1s);
                x1=x+(y*x1s);
                y1=y;
                az1=az+(w*x1s);
                w1=w;
                rectangle(x1,y1,az1,w1);
                break;
            case 2:
                printf("Enter the value of shear:\n");
                scanf("%f",&y1s);
                x1=x;
                y1=y+(x*y1s);
                az1=az;
                w1=w+(az*y1s);
                rectangle(x1,y1,az1,w1);
                break;
        }
        break;
    case 6:
        exit(0);
    }
    }
    getch();
}
```

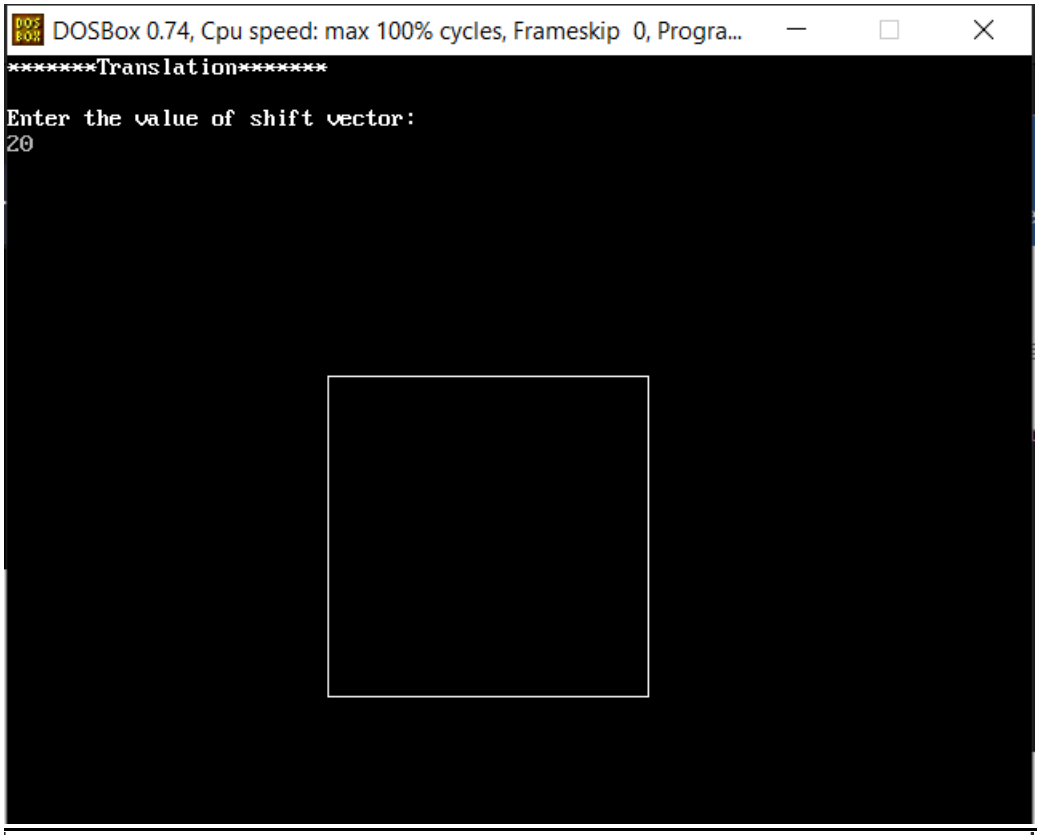
Results

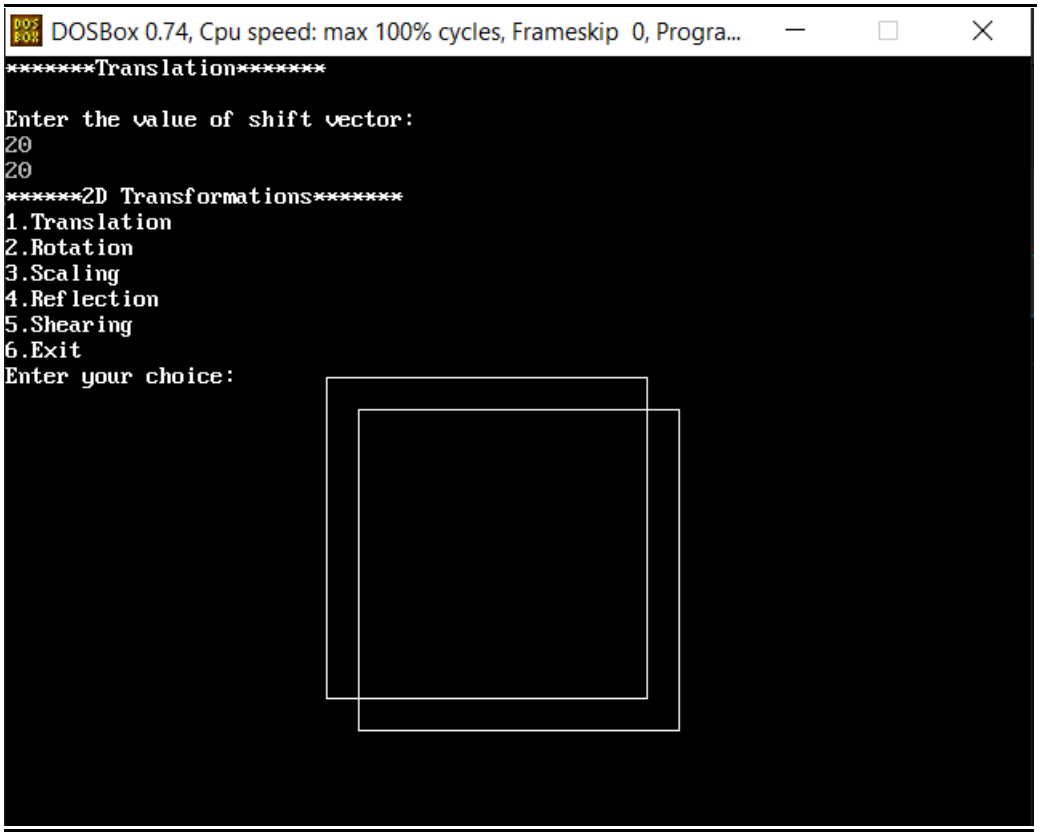
Initialisation



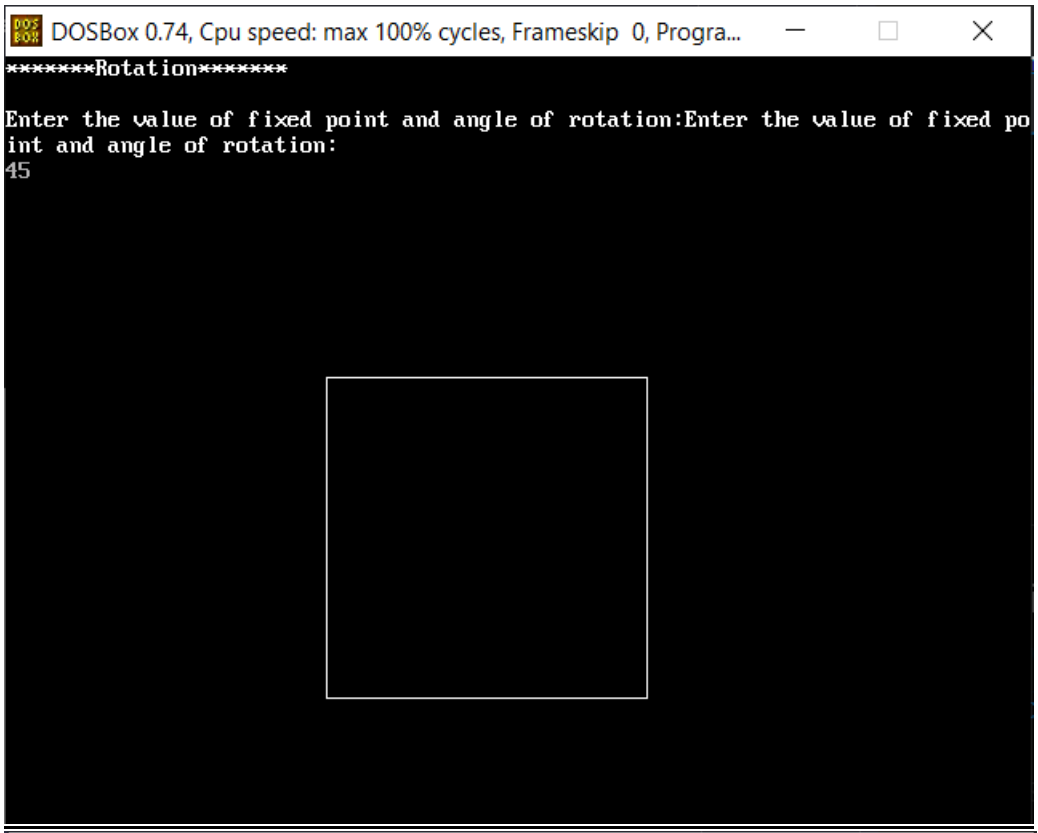


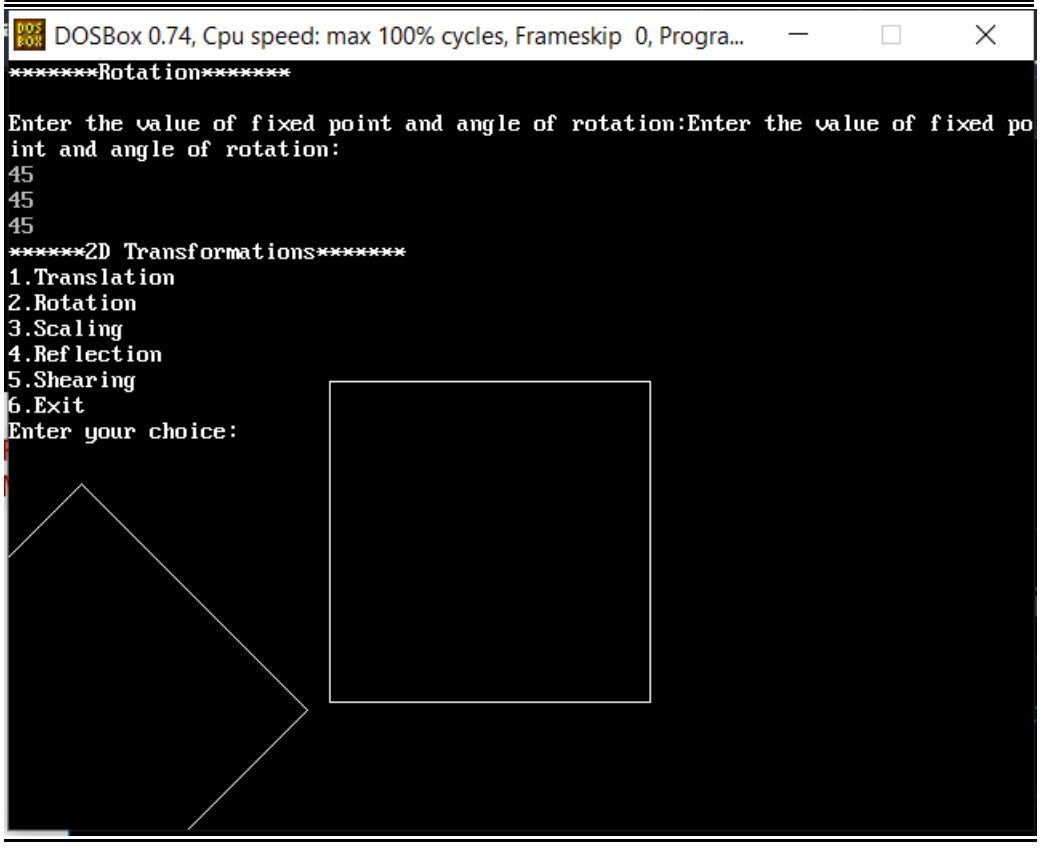
Translation


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Progra...
*******Translation*******
Enter the value of shift vector:
20

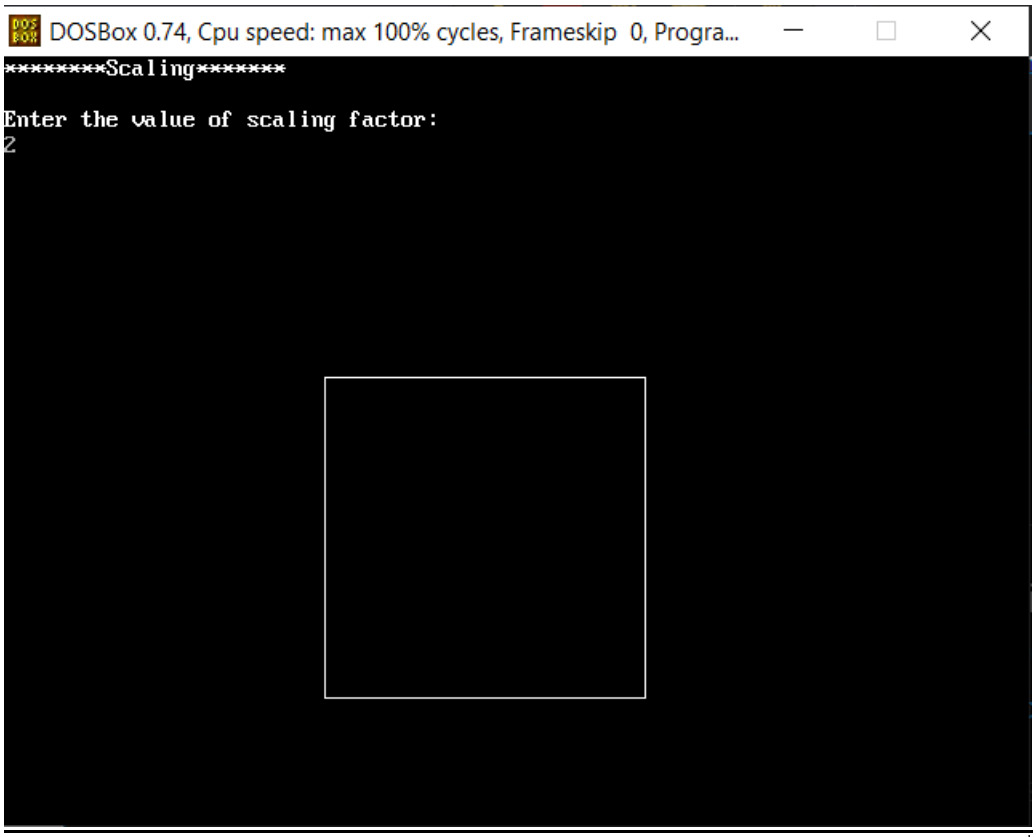

DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Progra...
*******Translation*******
Enter the value of shift vector:
20
20
******2D Transformations*******
1.Translation
2.Rotation
3.Scaling
4.Reflection
5.Shearing
6.Exit
Enter your choice:

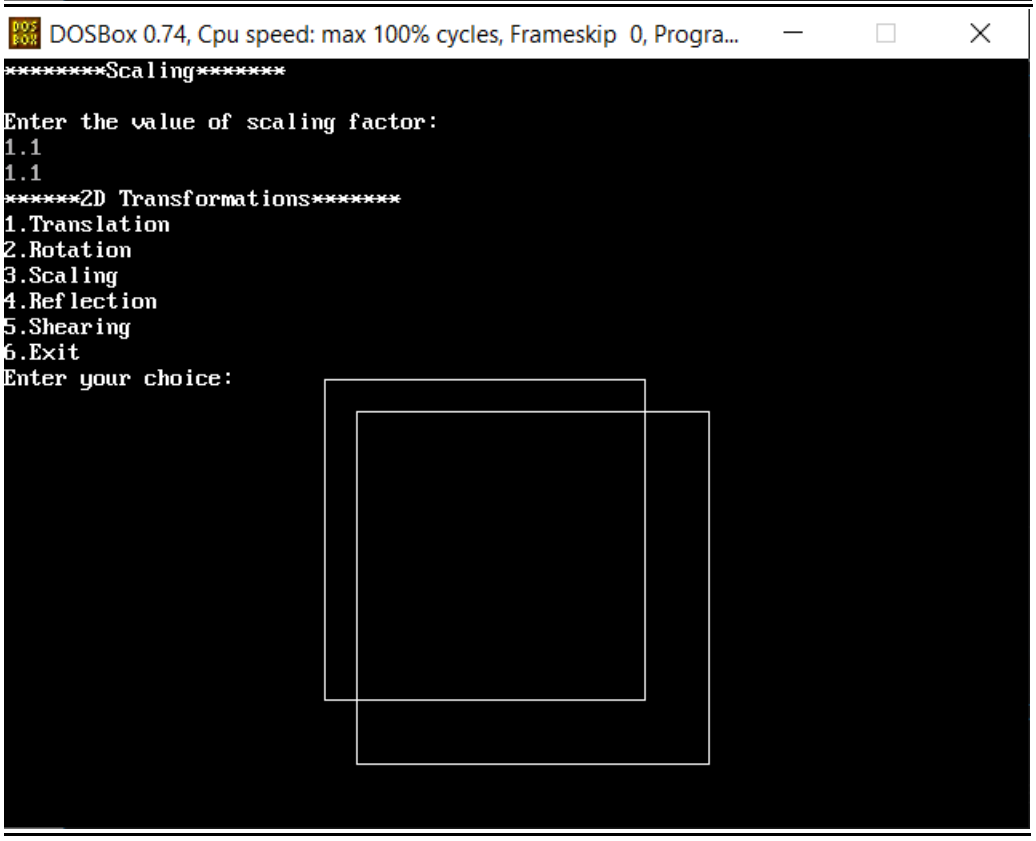
Rotation


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Progra...
*******Rotation*******
Enter the value of fixed point and angle of rotation:Enter the value of fixed po
int and angle of rotation:
45

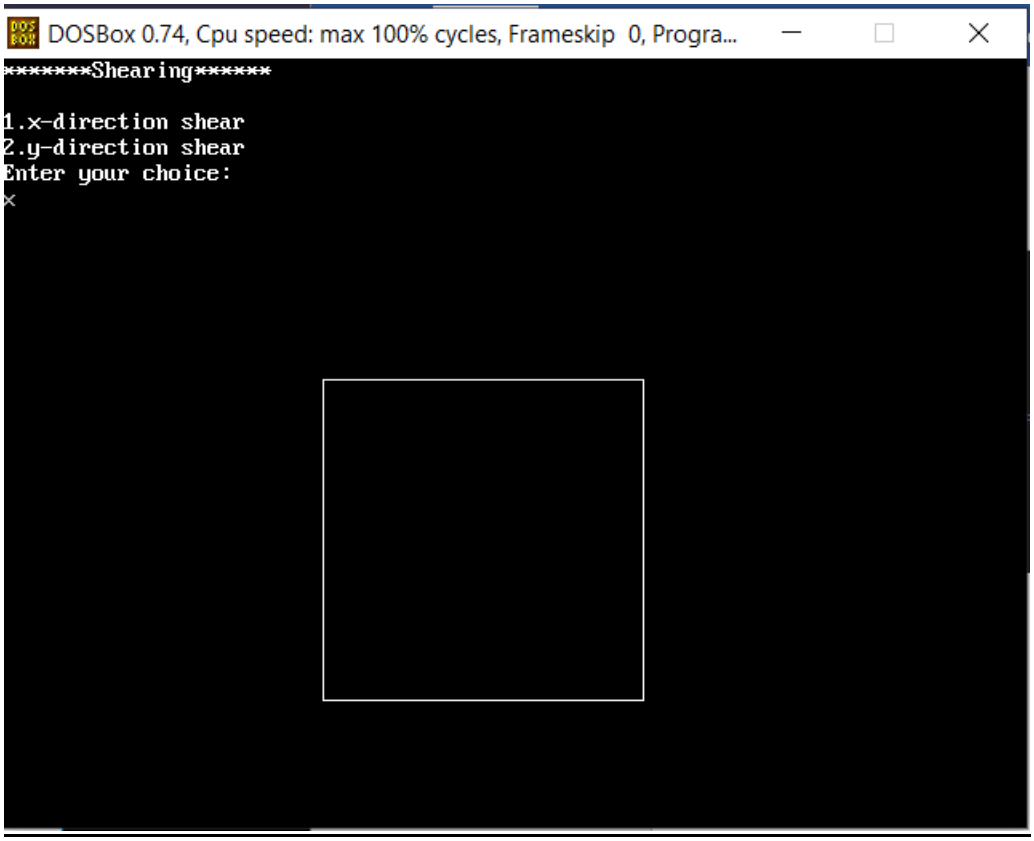

DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Progra...
*******Rotation*******
Enter the value of fixed point and angle of rotation:Enter the value of fixed po
int and angle of rotation:
45
45
45
******2D Transformations*******
1.Translation
2.Rotation
3.Scaling
4.Reflection
5.Shearing
6.Exit
Enter your choice:

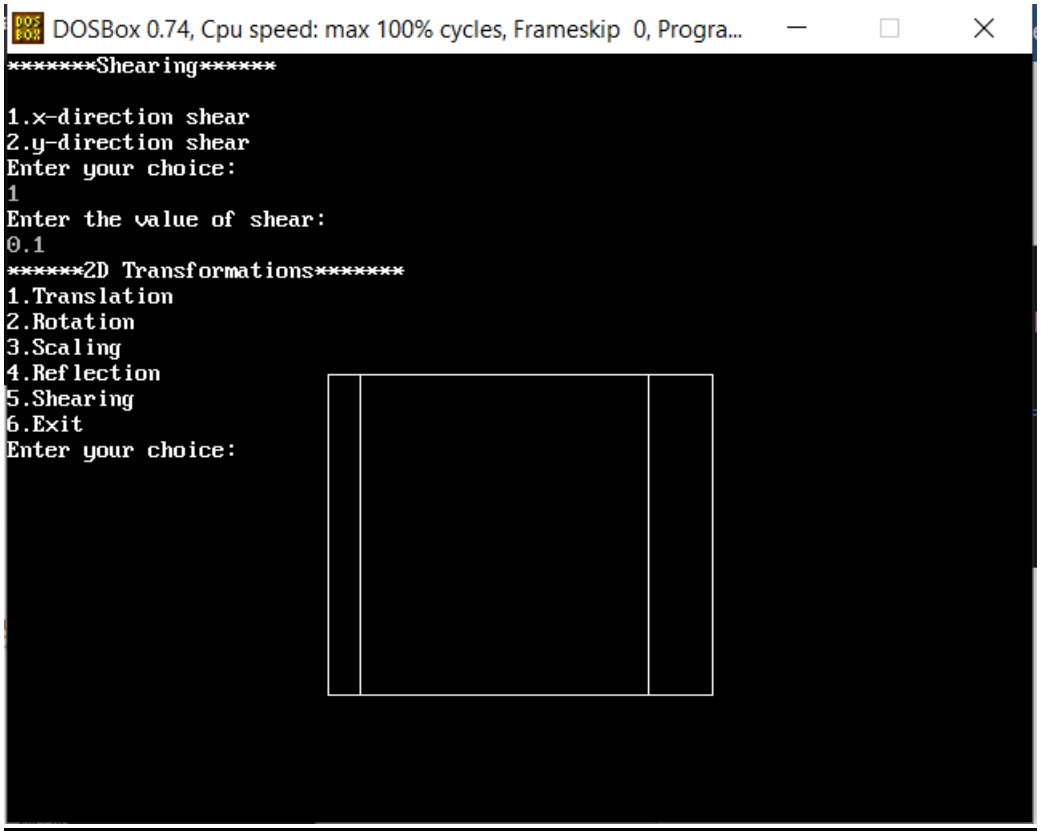
Scaling


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Progra...
********Scaling*******
Enter the value of scaling factor:
2

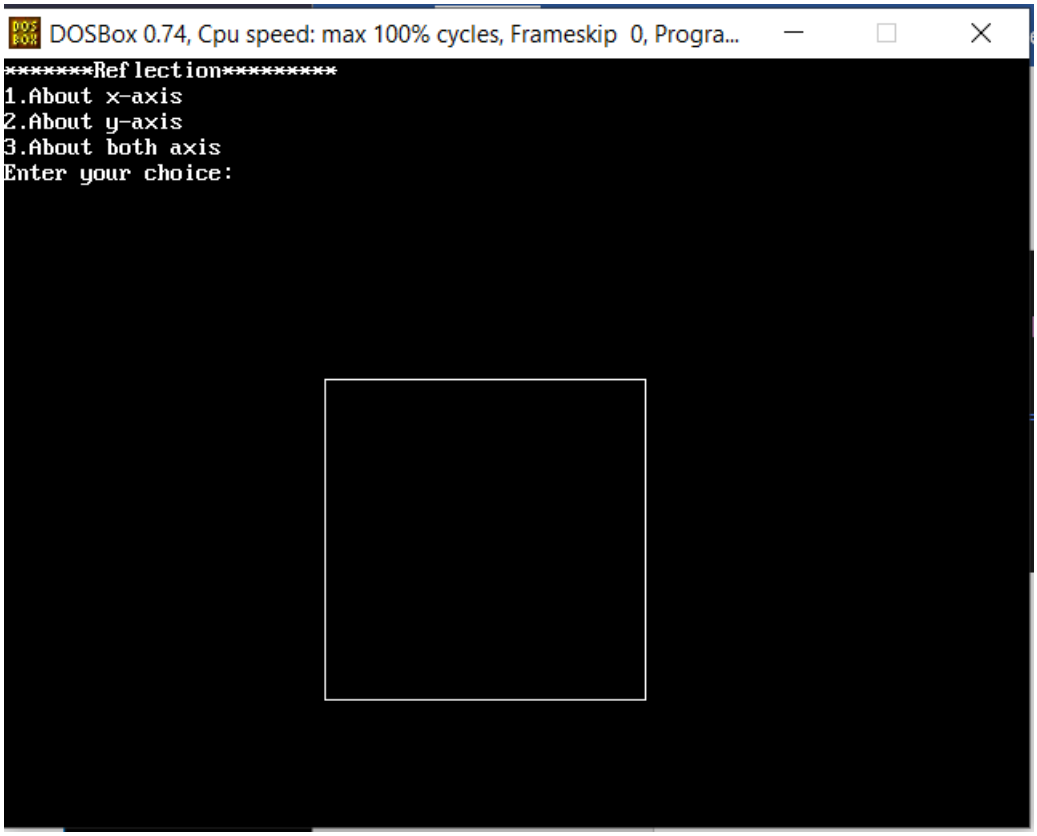

DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Progra...
********Scaling*******
Enter the value of scaling factor:
1.1
1.1
******2D Transformations*******
1.Translation
2.Rotation
3.Scaling
4.Reflection
5.Shearing
6.Exit
Enter your choice:

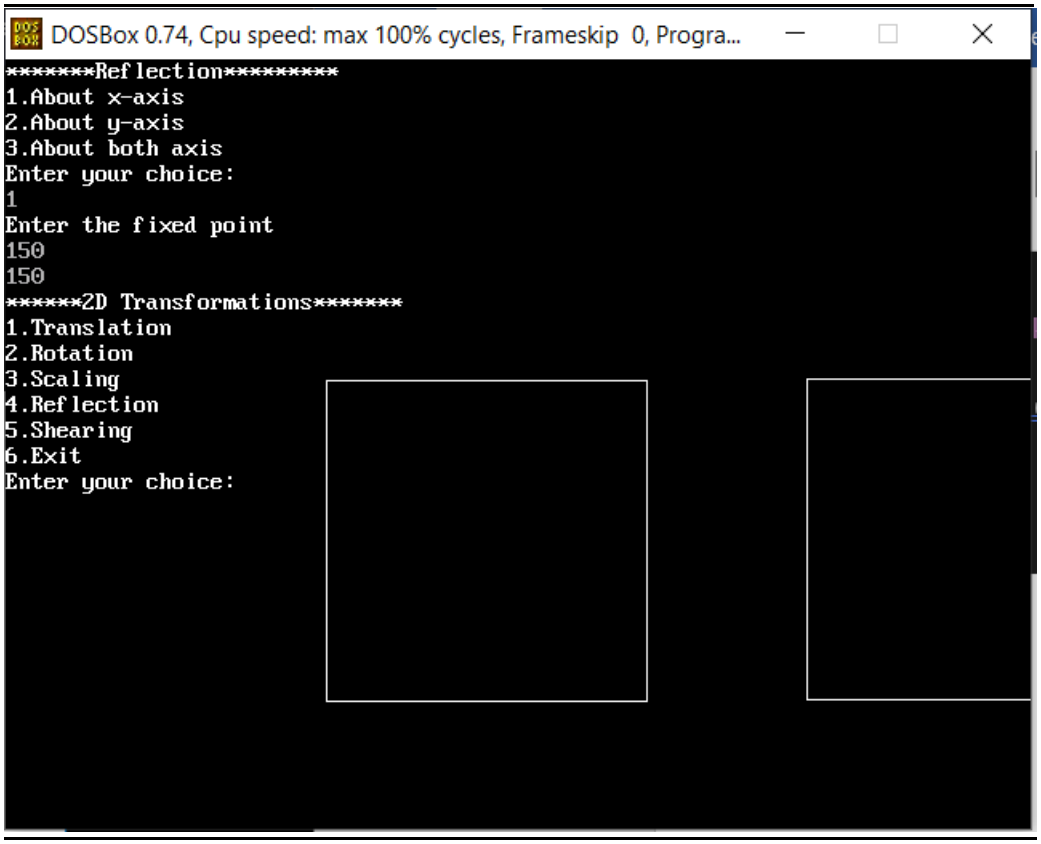
Shearing


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Progra...
*******Shearing******
1.x-direction shear
2.y-direction shear
Enter your choice:


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Progra...
*******Shearing******
1.x-direction shear
2.y-direction shear
Enter your choice:
1
Enter the value of shear:
0.1
******2D Transformations*******
1.Translation
2.Rotation
3.Scaling
4.Reflection
5.Shearing
6.Exit
Enter your choice:

Reflection


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Progra...
*******Reflection*********
1.About x-axis
2.About y-axis
3.About both axis
Enter your choice:


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Progra...
*******Reflection*********
1.About x-axis
2.About y-axis
3.About both axis
Enter your choice:
1
Enter the fixed point
150
150
******2D Transformations*******
1.Translation
2.Rotation
3.Scaling
4.Reflection
5.Shearing
6.Exit
Enter your choice:

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

B. Tech. Semester 2020-2021

SCHOOL OF COMPUTER SCIENCE ENGINEERING

(SCOPE)

Graphics and Multimedia

LAB 07

AMIT KUMAR

19BCE1281

3-D Transformation

Code

```
#include<stdio.h>

#include<conio.h>

#include<graphics.h>

#include<math.h>

void trans();

//void axis();

void scale();

void rotate();

int maxx,maxy,midx,midy;

void main()

{

            int ch;

            int gd=DETECT,gm;

            detectgraph(&gd,&gm);

            initgraph(&gd,&gm,"c:\\turboc3\\BGI");

            printf("\n 1.Translation \n 2.Scaling\n 3.Rotation \n 4.exit");

            printf("\nenter your choice\n");

            scanf("%d",&ch);

            do

            {

                        switch(ch)

                        {

                                    case 1 :            trans();
```

```
                                                    getch();
                                                //    closegraph();
                                                    break;

                                    case 2 :            scale();
                                                    getch();
                                                //    closegraph();
                                                    break;

                                    case 3 :            rotate();
                                                    getch();
                                                //    closegraph();
                                                    break;

                                    case 4 :        break;
                        }
                        printf("\nenter your choice\n");
                        scanf("%d",&ch);
            } while(ch<4);
}
void trans()
{
            int x,y,z,o,x1,x2,y1,y2;
            maxx=getmaxx();
```

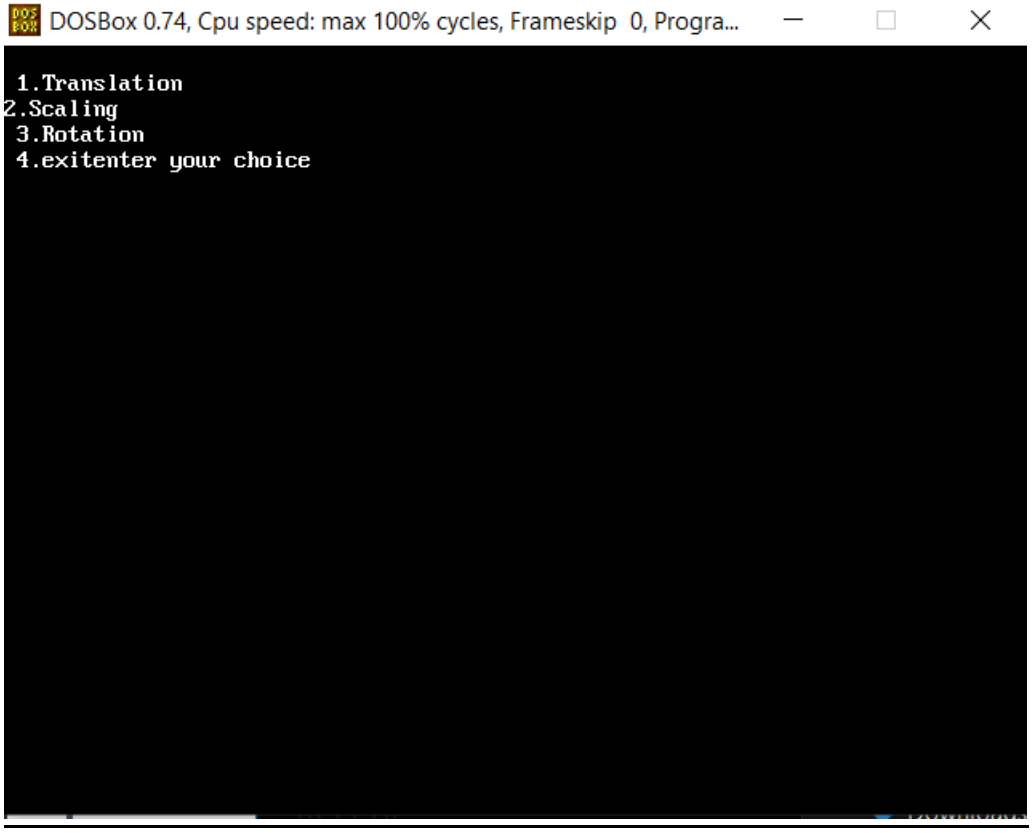
```
            maxy=getmaxy();
            midx=maxx/2;
            midy=maxy/2;
            //axis();
            bar3d(midx+50,midy-100,midx+60,midy-90,10,1);
            printf("\nEnter translation factor\n");
            scanf("%d%d",&x,&y);
            printf("\nAfter translation:\n");
            bar3d(midx+x+50,midy-(y+100),midx+x+60,midy-(y+90),10,1);
}
void scale()
{
            int x,y,z,o,x1,x2,y1,y2;
            maxx=getmaxx();
            maxy=getmaxy();
            midx=maxx/2;
            midy=maxy/2;
            //axis();
            bar3d(midx+50,midy-100,midx+60,midy-90,5,1);
            printf("\nbefore translation\n");
            printf("\nEnter scaling factors\n");
            scanf("%d %d %d", &x,&y,&z);
            printf("\nAfter scaling\n");
            bar3d(midx+(x*50),midy-(y*100),midx+(x*60),midy-(y*90),5*z,1);
```

```
}
void rotate()
{
            int x,y,z,o,x1,x2,y1,y2;
            maxx=getmaxx();
            maxy=getmaxy();
            midx=maxx/2;
            midy=maxy/2;
            //axis();
            bar3d(midx+50,midy-100,midx+60,midy-90,5,1);
            printf("\nEnter rotating angle\n");
            scanf("%d",&o);
            x1=50*cos(o*3.14/180)-100*sin(o*3.14/180);
            y1=50*sin(o*3.14/180)+100*cos(o*3.14/180);
            x2=60*cos(o*3.14/180)-90*sin(o*3.14/180);
            y2=60*sin(o*3.14/180)+90*cos(o*3.14/180);
   //       axis();
    //       printf("After rotation about z axis");
    //       bar3d(midx+x1,midy-y1,midx+x2,midy-y2,5,1);
            //axis();
            printf("\nAfter rotation  about x axis\n");
            bar3d(midx+50,midy-x1,midx+60,midy-x2,5,1);
            //axis();
            printf("\nAfter rotation about yaxis\n");
```

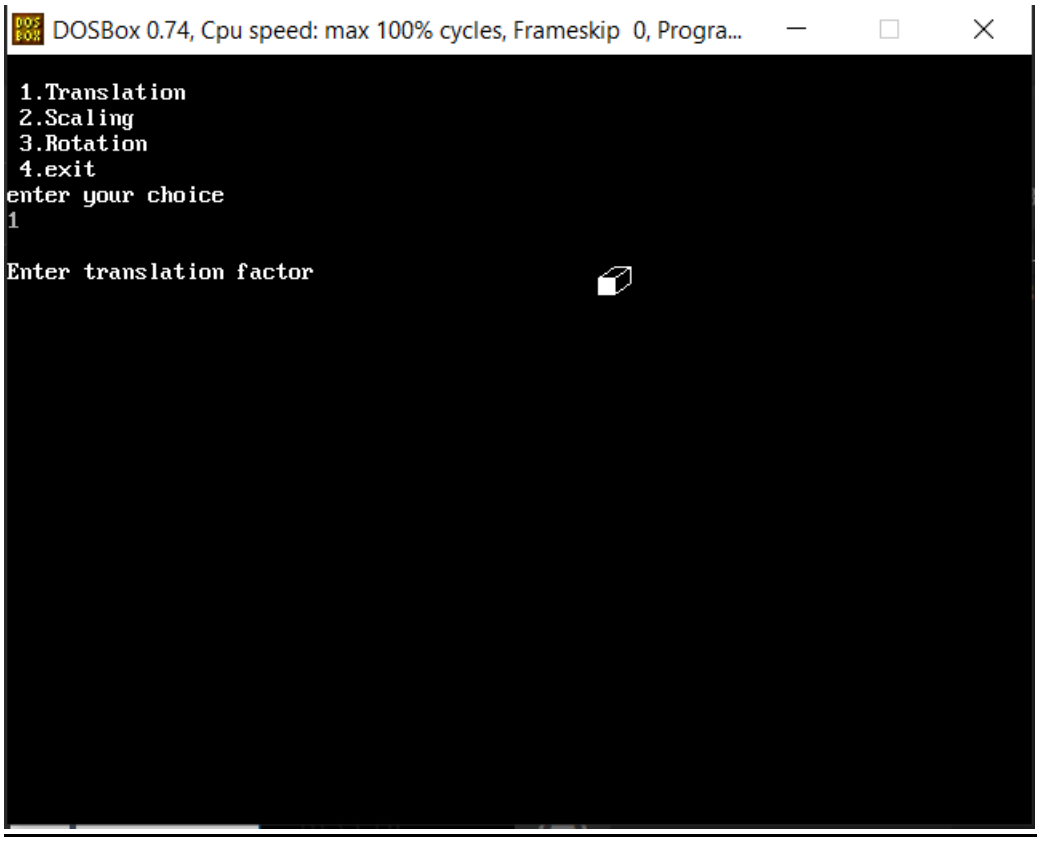
```
            bar3d(midx+x1,midy-100,midx+x2,midy-90,5,1);

}
```

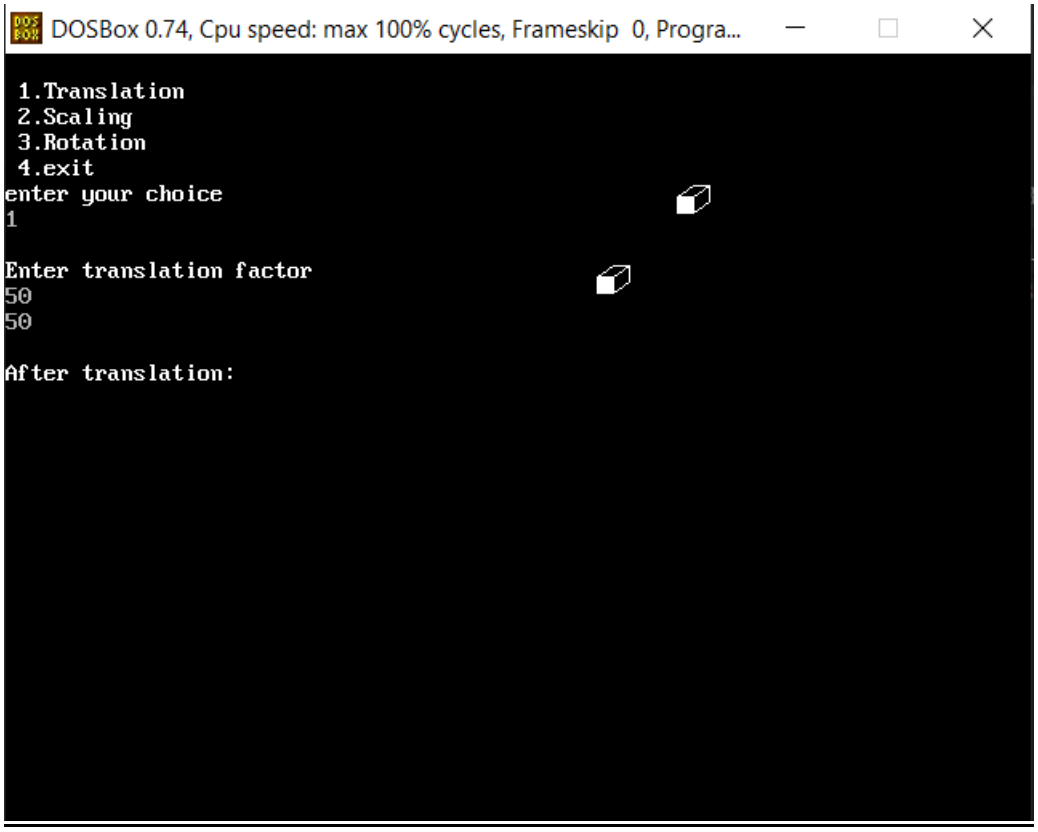
Result

Initialization

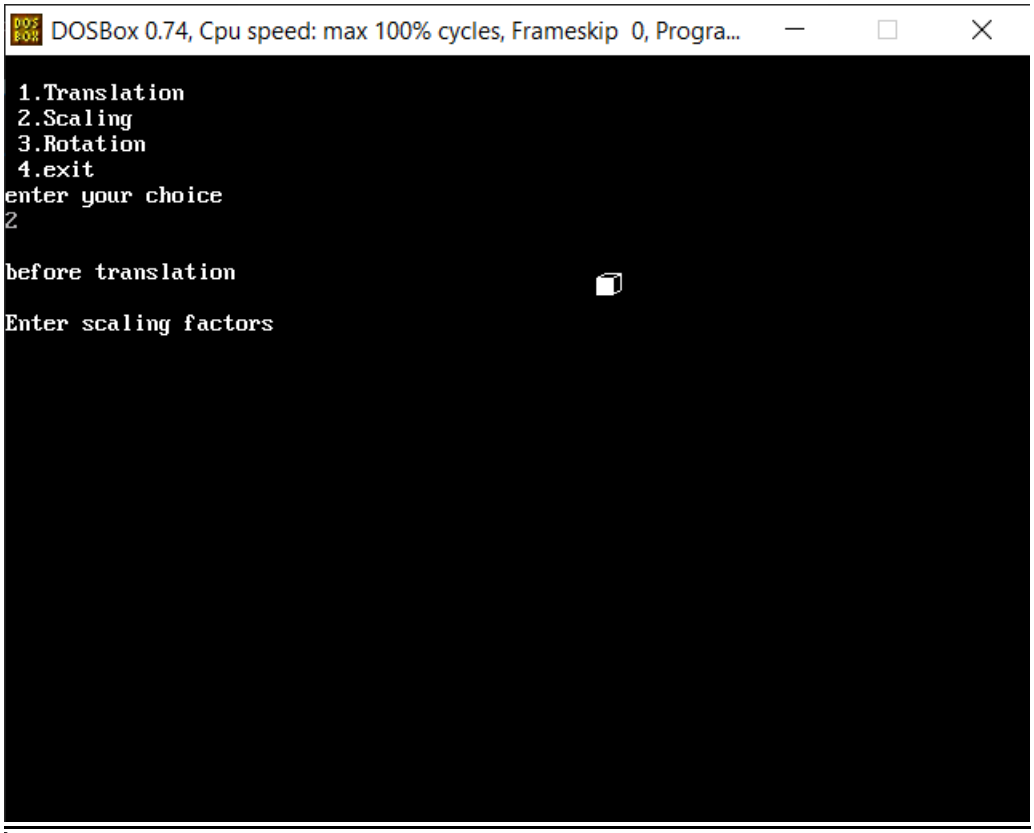


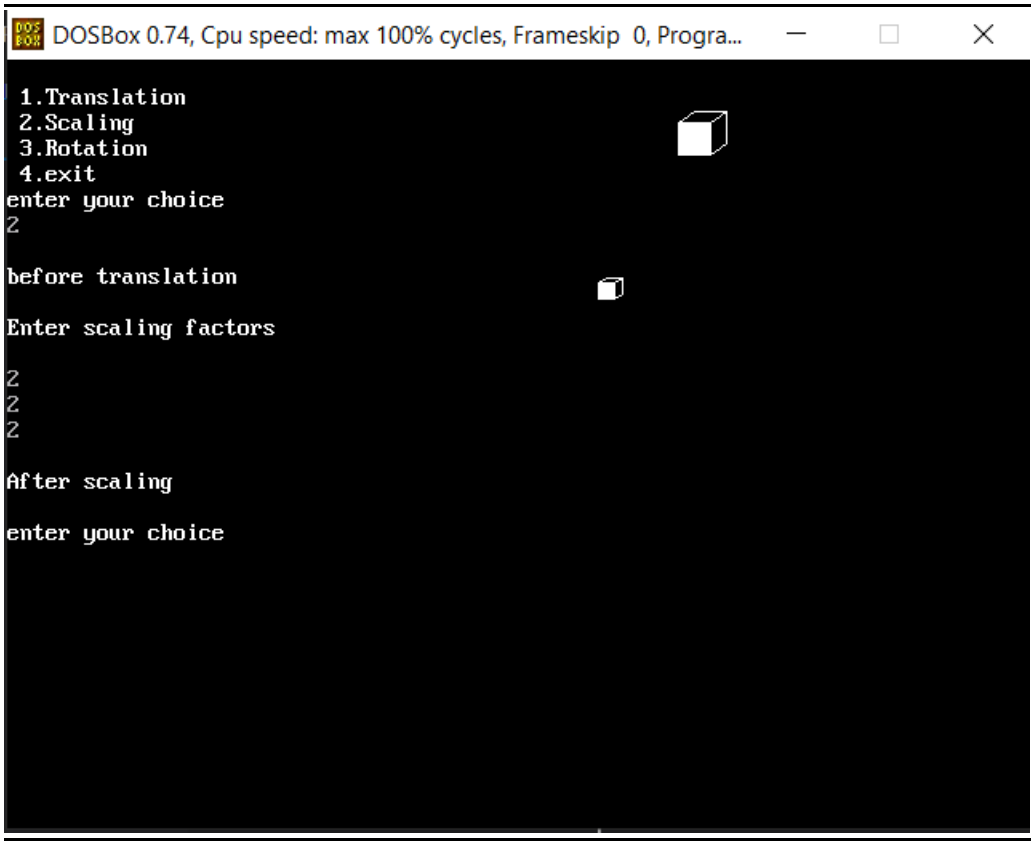
Translation



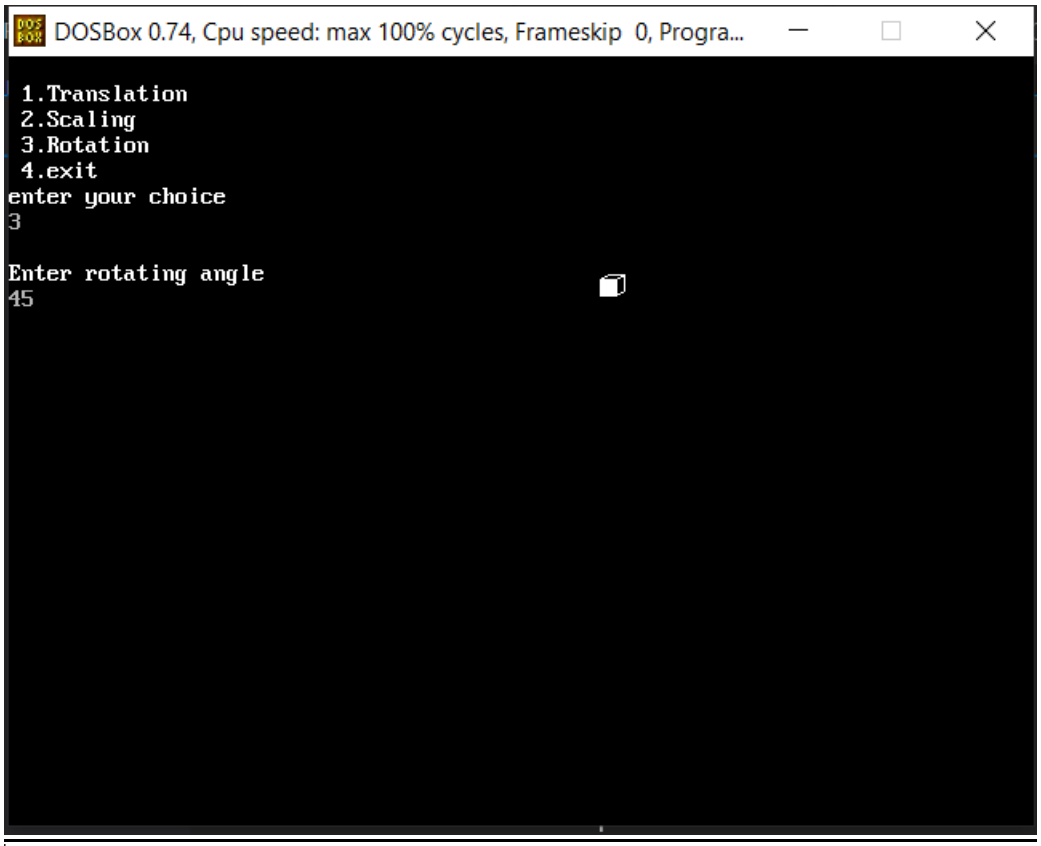

DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Progra...
1.Translation
2.Scaling
3.Rotation
4.exit
enter your choice
1
Enter translation factor
50
50
After translation:

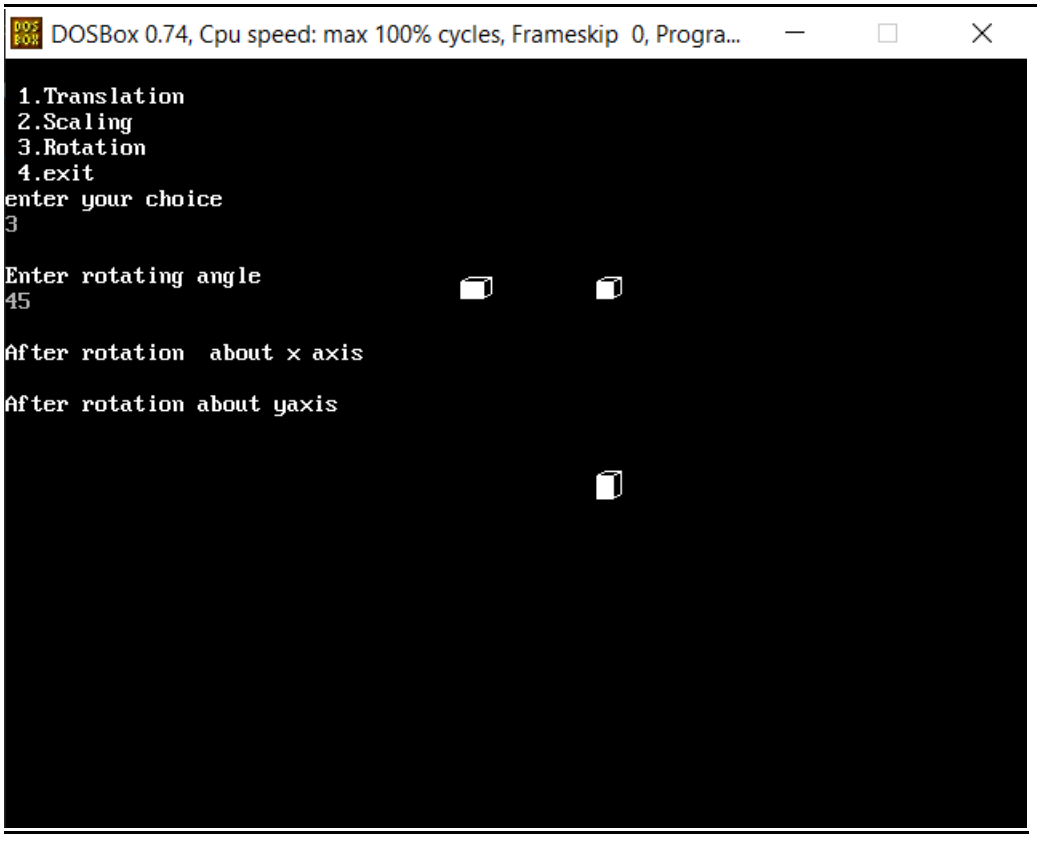
Scaling


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Progra...
1.Translation
2.Scaling
3.Rotation
4.exit
enter your choice
2
before translation
Enter scaling factors


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Progra...
1.Translation
2.Scaling
3.Rotation
4.exit
enter your choice
2
before translation
Enter scaling factors
2
2
2
After scaling
enter your choice

Rotation


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Progra...
1.Translation
2.Scaling
3.Rotation
4.exit
enter your choice
3
Enter rotating angle
45


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Progra...
1.Translation
2.Scaling
3.Rotation
4.exit
enter your choice
3
Enter rotating angle
45
After rotation about x axis
After rotation about yaxis

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

B. Tech. Semester 2020-2021

SCHOOL OF COMPUTER SCIENCE ENGINEERING

(SCOPE)

Graphics and Multimedia

LAB 08

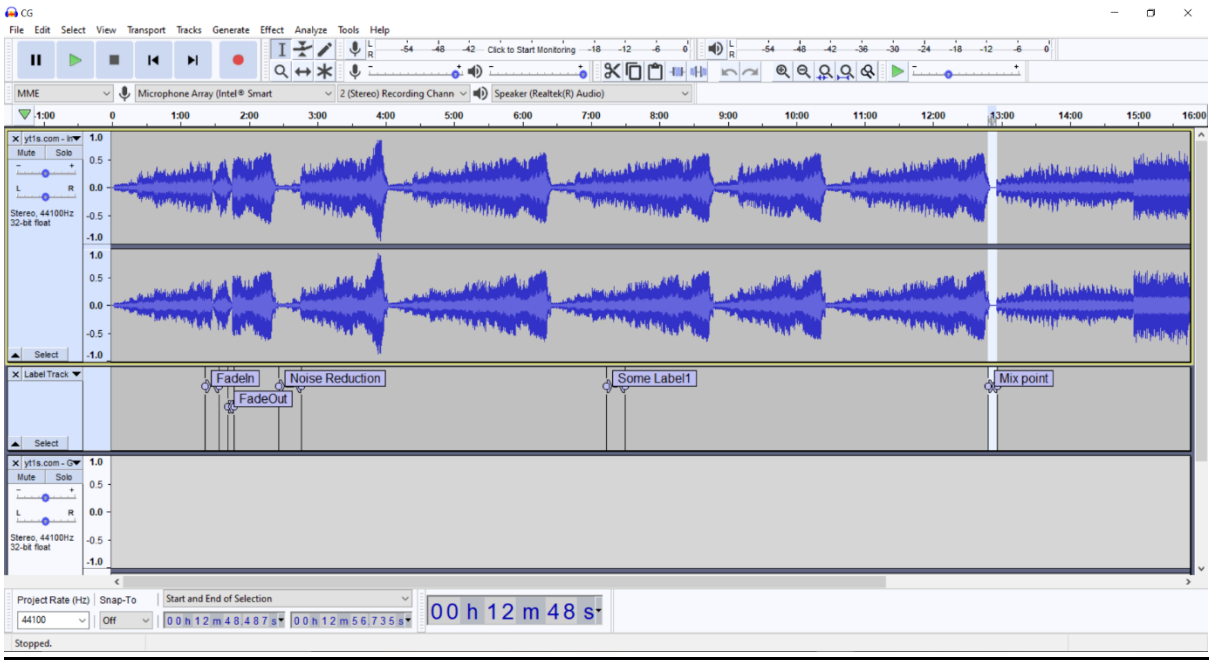
AMIT KUMAR

19BCE1281

Link to Drive

https://drive.google.com/drive/folders/1lMcHjYLo6PMQQLTqdjlJAXIjq3sxeyn5?usp=sharing

Result SS



VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

B. Tech. Semester 2020-2021

SCHOOL OF COMPUTER SCIENCE ENGINEERING

(SCOPE)

Graphics and Multimedia

LAB 09

AMIT KUMAR

19BCE1281

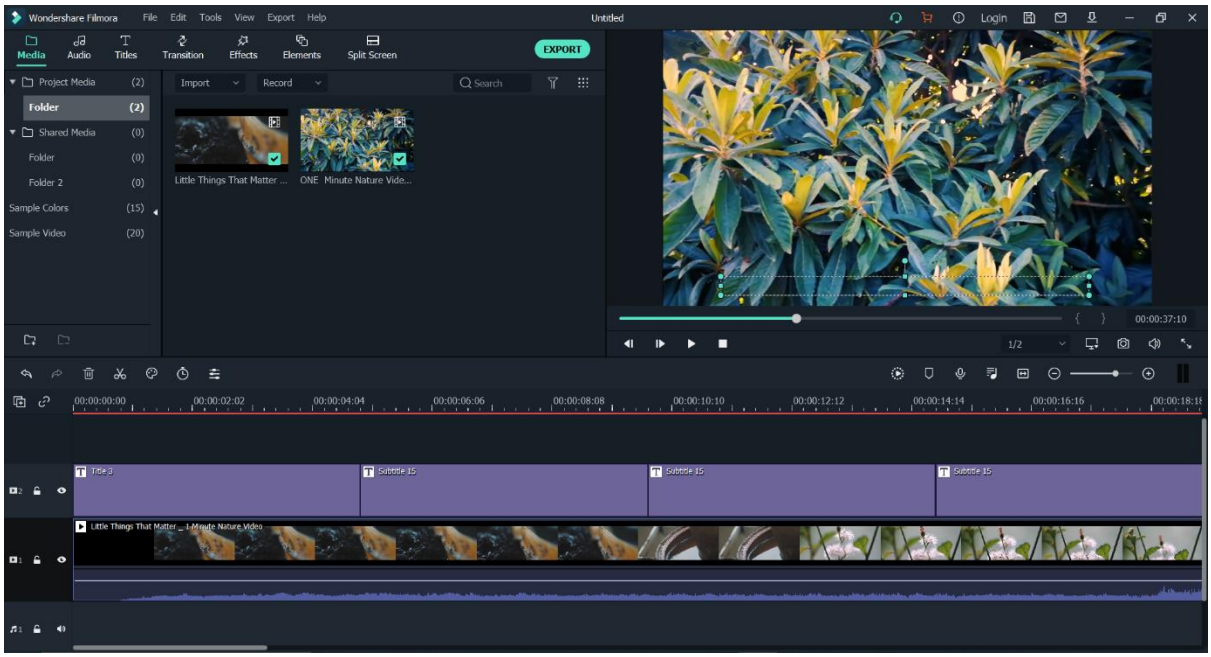
Tool – Filmora X

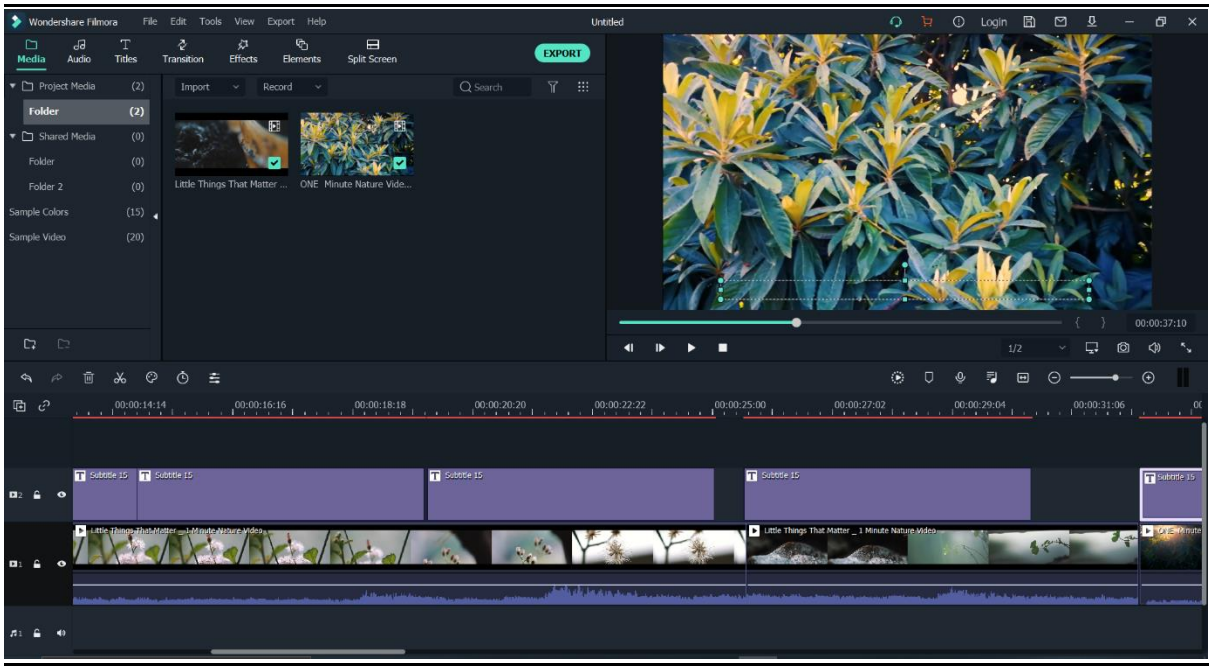
Link to Drive

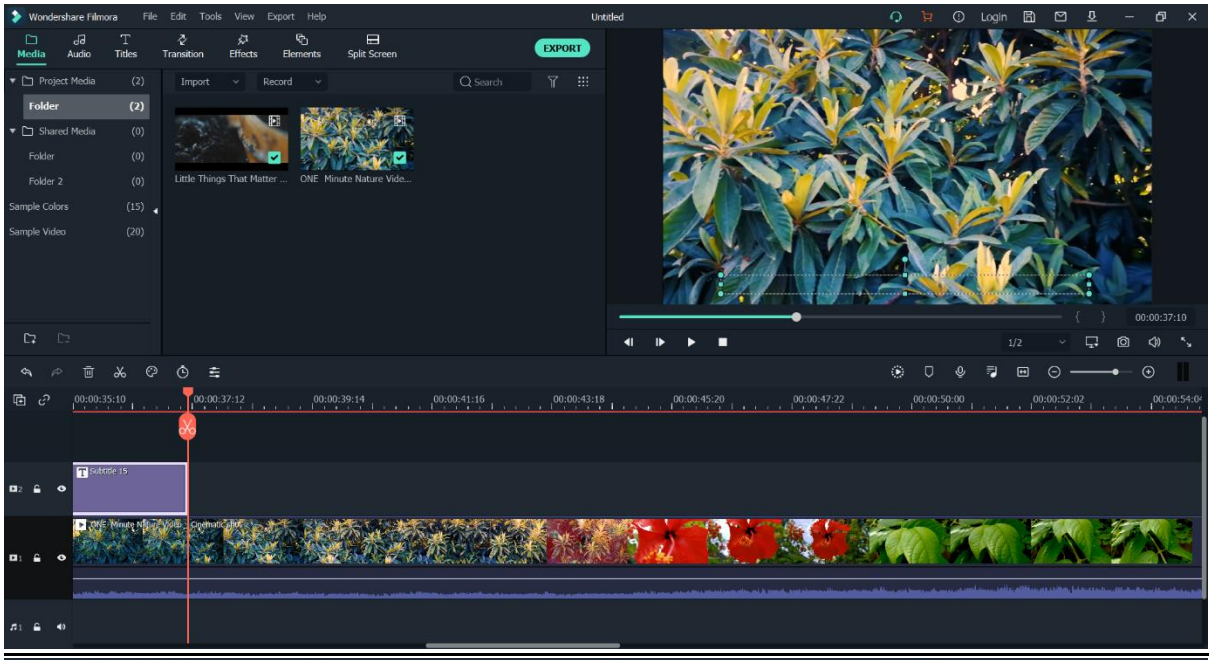
https://drive.google.com/drive/folders/1LUmj7MFWPhme0brX-z7ertxYODbqAzqb?usp=sharing

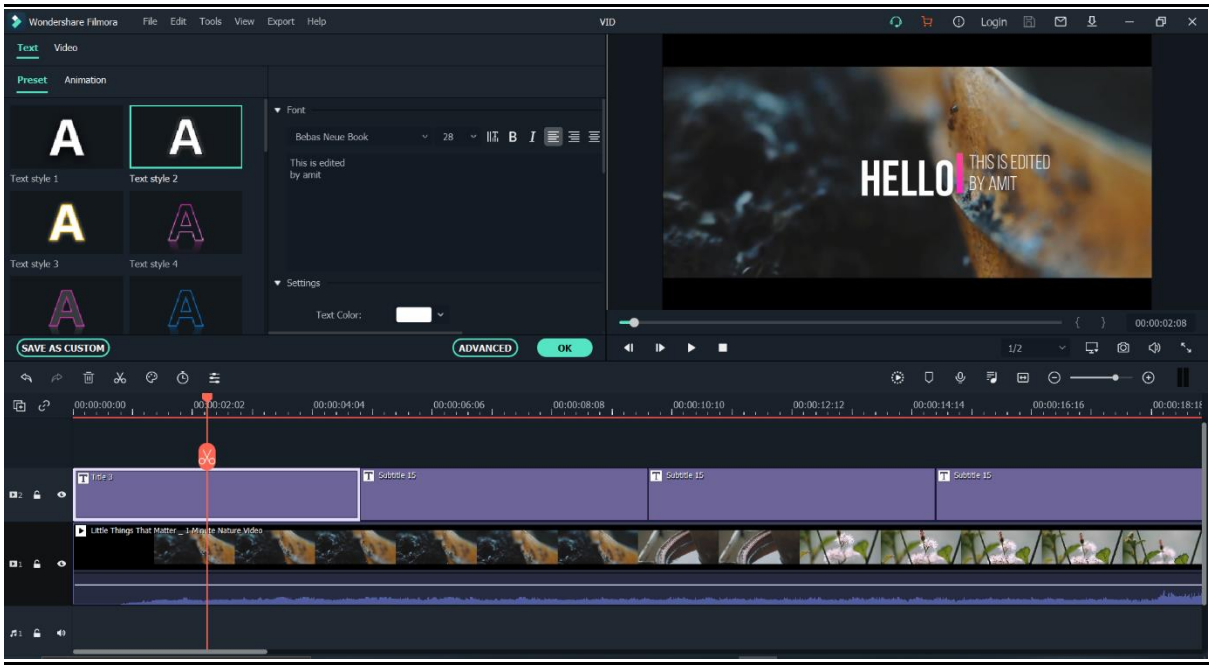
Result SS

(Trim, crop, rotate, join video, subtitles, edit video dimension)

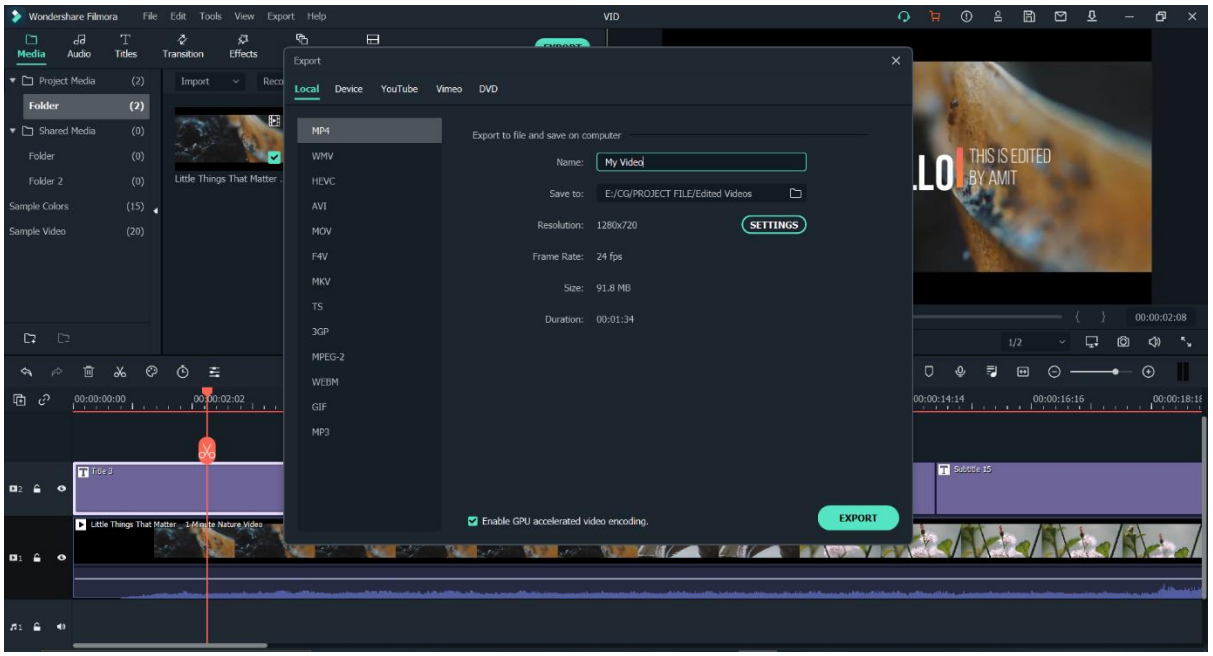


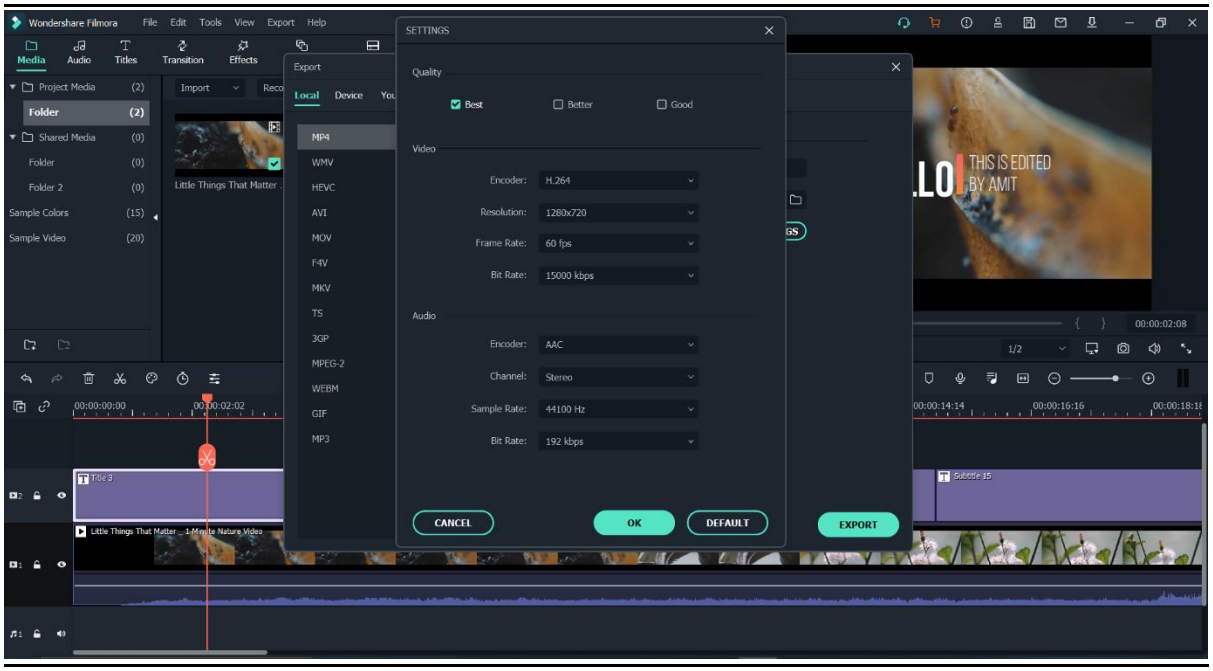



Wondershare Filmora
File Edit Tools View Export Help
Untitled
Login
Media Audio Titles Transition Effects Elements Split Screen
EXPORT
Project Media (2)
Folder (2)
Shared Media (0)
Folder (0)
Folder 2 (0)
Sample Colors (15)
Sample Video (20)
Import Record Search
Little Things That Matter ...
ONE Minute Nature Vide...
00:00:37:10
1/2
00:00:35:10 00:00:37:12 00:00:39:14 00:00:41:16 00:00:43:18 00:00:45:20 00:00:47:22 00:00:50:00 00:00:52:02
Subtitle 15


Wondershare Filmora
File Edit Tools View Export Help
VID
Login
Text Video
Preset Animation
Text style 1
Text style 2
Text style 3
Text style 4
Font
Bebas Neue Book
28
This is edited by amit
Settings
Text Color:
SAVE AS CUSTOM
ADVANCED
OK
HELLO
THIS IS EDITED BY AMIT
00:00:02:08
1/2
00:00:00:00 00:00:02:02 00:00:04:04 00:00:06:06 00:00:08:08 00:00:10:10 00:00:12:12 00:00:14:14 00:00:16:16
Title 3
Subtitle 15
Subtitle 15
Subtitle 15
Little Things That Matter _ 1 Minute Nature Video

Change BIT Rate, Change FPS





VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

B. Tech. Semester 2020-2021

SCHOOL OF COMPUTER SCIENCE ENGINEERING

(SCOPE)

Graphics and Multimedia

LAB 10

AMIT KUMAR

19BCE1281

Tool – Alice 3

Link to Drive

https://drive.google.com/drive/folders/1sPejJTbRwDc_SmrSwgDKNQ-g60QmCSwX?usp=sharing

Result SS