**Program 6(a): Implementation of Point Translational Transformations.**

#include<iostream>

#include<graphics.h>

using namespace std;

voidtranslatePoint( int P[], int T[])

{

cout<<"Original Coordinates :"<<P[0]<<","<<P[1];

putpixel (P[0], P[1], WHITE);

P[0] = P[0] + T[0];

P[1] = P[1] + T[1];

cout<<"\nTranslated Coordinates :"<< P[0]<<","<< P[1];

putpixel (P[0], P[1], WHITE);

}

int main()

{

initwindow(500,500);

int P[2] = {300, 200};

int T[] = {100, 100};

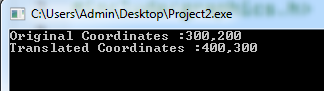
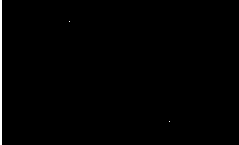
translatePoint (P, T);

getch();

return 0;

}

**OUTPUT:**

**Program 6(b): Implementation of Line Translational Transformations.**

#include<iostream>

#include<graphics.h>

using namespace std;

voidtranslateLine ( int P[][2], int T[])

{ setcolor (2);

line(P[0][0], P[0][1], P[1][0], P[1][1]);

P[0][0] = P[0][0] + T[0];

P[0][1] = P[0][1] + T[1];

P[1][0] = P[1][0] + T[0];

P[1][1] = P[1][1] + T[1];

setcolor(3);

line(P[0][0], P[0][1], P[1][0], P[1][1]); }

int main()

{ initwindow(200,200);

int P[2][2] = {100, 70, 80, 98};

int T[] = {70, 100};

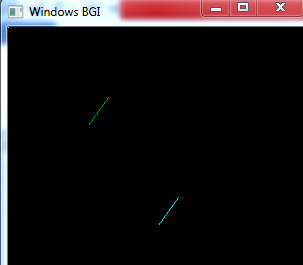
translateLine (P, T);

getch();

return 0;

}

**OUTPUT:**



**Program 6(c): Implementation of rectangle translational transformation.**

#include<iostream>

#include<graphics.h>

using namespace std;

voidtranslateRectangle ( int P[][2], int T[])

{ setcolor (2);

rectangle (P[0][0], P[0][1], P[1][0], P[1][1]);

P[0][0] = P[0][0] + T[0];

P[0][1] = P[0][1] + T[1];

P[1][0] = P[1][0] + T[0];

P[1][1] = P[1][1] + T[1];

rectangle (P[0][0], P[0][1], P[1][0], P[1][1]);

}

int main()

{

initwindow(200,200);

int P[2][2] = {50, 80, 100, 160};

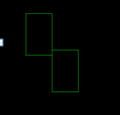
int T[] = {50,70 }; // translation factor

translateRectangle (P, T);

getch();

return 0; }

**OUTPUT:**



**Program 8: Implementation of rotational transformation.**

#include<iostream>

#include<graphics.h>

#include<math.h>

using namespace std;

int main()

{

int x1,y1,x2,y2;

double s,c, angle;

initwindow(500,500);

setcolor(RED);

cout<<"Enter coordinates of line: ";

cin>>x1>>y1>>x2>>y2;

setbkcolor(WHITE);

line(x1,y1,x2,y2);

setbkcolor(WHITE);

cout<<"Enter rotation angle: ";

cin>>angle;

setbkcolor(WHITE);

c = cos(angle \*3.14/180);

s = sin(angle \*3.14/180);

x1 = floor(x1 \* c + y1 \* s);

y1 = floor(-x1 \* s + y1 \* c);

x2 = floor(x2 \* c + y2 \* s);

y2 = floor(-x2 \* s + y2 \* c);

line(x1, y1 ,x2, y2);

getch();

return 0;

}

**OUTPUT:**

