**PROGRAM 14**

**Write a program to implement 2-D translation.**

#include<stdio.h>

#include<graphics.h>

#define ROUND(a) ((int)(a+0.5))

void ddaline(int x1, int y1, int x2, int y2)

{

float xsteps, ysteps, x=x1, y=y1;

int dx = x2-x1;

int dy = y2-y1;

int steps,k=1;

if(abs(dx)>=abs(dy))

steps=abs(dx);

else steps=abs(dy);

xsteps= dx/(float)steps;

ysteps= dy/(float)steps;

putpixel(ROUND(x),ROUND(y),15);

while(k<=steps)

{

x+=xsteps;

y+=ysteps;

putpixel(ROUND(x), ROUND(y),15);

k++;

}

}

void translate(int x1, int y1, int x2, int y2, int tx, int ty)

{

ddaline(x1,y1,x2,y2);

ddaline(x1+tx,y1+ty,x2+tx,y2+ty);

}

int main()

{

int x1, x2, y1, y2, tx, ty;

int gdriver = DETECT, gmode, errorcode;

initgraph(&gdriver, &gmode, "..\\");

errorcode = graphresult();

if (errorcode != grOk)

{

printf("Graphics error: %s\n", grapherrormsg(errorcode));

printf("Press any key to halt:");

getch();

exit(1);

}

printf("Enter start point\n");

scanf("%d %d", &x1, &y1);

printf("Enter end point\n");

scanf("%d %d", &x2, &y2);

printf("Enter value for \'tx\' and \'ty\' \n");

scanf("%d %d", &tx, &ty);

translate(x1, y1, x2, y2, tx, ty);

getch();

closegraph();

return 0;

}

**OUTPUT 14**



