PRACTICAL 7:

AIM: Write a Program to perform 2D translation transformation.

Source Code:

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

void translatePoint ( int P[], int T[])
{
   /* init graph and putpixel are used for
      representing coordinates through graphical
      functions
   */
   int gd = DETECT, gm, errorcode;
   initgraph(&gd,&gm,"C:\\TURBOC3\\BGI");

   printf("Original Coordinates : %d,%d",P[0],P[1]);

   putpixel (P[0], P[1], 4);

   // calculating translated coordinates
   P[0] = P[0] + T[0];
   P[1] = P[1] + T[1];

   printf("\nTranslated Coordinates : %d,%d",P[0],P[1]);

   // Draw new coordinatses
   putpixel (P[0], P[1],3);
   getch();
   closegraph();
}

void translateLine ( int P[][2], int T[])
{
   /* init graph and line() are used for
      representing line through graphical
      functions
   */
   int gd = DETECT, gm, errorcode;
   initgraph(&gd,&gm,"C:\\TURBOC3\\BGI");
```

```
    // drawing original line using graphics functions
    setcolor (5);
    line(P[0][0], P[0][1], P[1][0], P[1][1]);

    // calculating translated coordinates
    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];

    // drawing translated line using graphics functions
    setcolor(1);
    line(P[0][0], P[0][1], P[1][0], P[1][1]);
    getch();
    closegraph();
}

void translateRectangle ( int P[][2], int T[])
{
    /* init graph and rectangle() are used for
    representing rectangle through graphical functions */
    int gd = DETECT, gm, errorcode;
    initgraph(&gd,&gm,"C:\\TURBOC3\\BGI");
    setcolor (2);
    // rectangle (Xmin, Ymin, Xmax, Ymax)
    // original rectangle
    rectangle (P[0][0], P[0][1], P[1][0], P[1][1]);

    // calculating translated coordinates
    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];

    // translated rectangle (Xmin, Ymin, Xmax, Ymax)
    setcolor(14);
    rectangle (P[0][0], P[0][1], P[1][0], P[1][1]);
    // closegraph();
        getch();
}
```

```
int main()
{
   //for point transformation
   int P[2] = {150,180}; // coordinates of point
   int Q[] = {120,110}; // translation factor
   //for line transformation
   int R[2][2] = {50, 80, 180, 280}; // coordinates of point
   int S[] = {20, 10}; // translation factor

   // Xmin, Ymin, Xmax, Ymax as rectangle
   // coordinates of top left and bottom right points
   int T[2][2] = {50,180,150,280};
   int U[] = {40, 50}; // translation factor

   translatePoint (P, Q);
   translateLine (R, S);
   translateRectangle (T, U);
   return 0;
}
```

Output: