Index

S. No.	Name of the Experiment	Page No.	Date of Experiment	Date of Submission	Remarks
1.	Pixel highlighting.	1	28/9	12/10	
2.	Drawing line.	2	28/9	12/10	Attained 12/10/22
3.	Drawing circle.	3	28/9	12/10	
4.	Drawing rectangle.	4	28/9	12/10	
5.	Drawing triangle	5	28/9.	12/10	
6.	Drawing ellipse	6	28/9	12/10	
7.	Drawing name	7	12/10	9/10	Attained 9/11/22
8.	Drawing line (DDA algorithm)	8	12/10	9/10	
9	Drawing line (Bresenham's algorithm	9	2/11	9/10	
10	Drawing Circle using Midpoint algorithm	11	9/11	23/11	Attained 23/11/22
11.	Drawing Ellipse using Midpoint algorithm	13	9/11	23/11	
12	Drawing circles filled with style.	15	23/11	7/12	Attained 7/12/22
13	Creating screensaver using circles	16	23/11	7/12	
14	Line clipping (Cohen - Sutherland)	17			
15	2D transformations	23			

Expt. No. _____ Date _____

Page No. 1

Program-1 → Program to highlight a pixel.

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

int main(){
    int gd=DETECT, gm;
    initgraph(&gd,&gm,"C:\\TC\\BGI");
    putpixel(25,25,Red);
    getch();
    closegraph();
    return 0;
}
```

Teacher's Signature _____

Program-2 → Program to draw a line.

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

int main() {
    int gd = DETECT, gm;
    initgraph(&gd, &gm, "C:\\TC\\BGI");

    line(100, 100, 200, 100);
    getch();
    closegraph();
    return 0;
}
```

Expt. No. _____ Date _____

Page No. 3

Program-3 → Program to draw a circle.

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

int main(){
    int gd= DETECT, gm;
    initgraph(&gd, &gm, "C:\\TC\\BGI");

    circle(150,150,30);
    getch();
    closegraph();
    return 0;
}
```

Teacher's Signature _____

Expt. No. _____ Date _____

Page No. 4

Program-4 → Program to draw a rectangle.

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

int main(){
    int gd=DETECT, gm;
    initgraph(&gd, &gm, "C:\\TC\\BGI");

    rectangle(100,100,200,200);
    getch();
    closegraph();
    return 0;
}
```

Teacher's Signature _____

Program-5 → Program to draw a trigangle.

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

int main(){
    int gd= DETECT, gm;
    initgraph(&gd, &gm, "C:\\TC\\BGI");

    line(300, 100, 200, 200);
    line(300, 100, 400, 200);
    line(200, 200, 400, 200);
    getch();
    closegraph();
    return 0;
}
```

Program - 6 → Program to draw an ellipse

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

int main() {
    int gd = DETECT, gm;
    initgraph(&gd, &gm, "C:\\TC\\BGI");

    ellipse(250, 200, 0, 360, 100, 50);
    getch();
    closegraph();
    return 0;
}
```

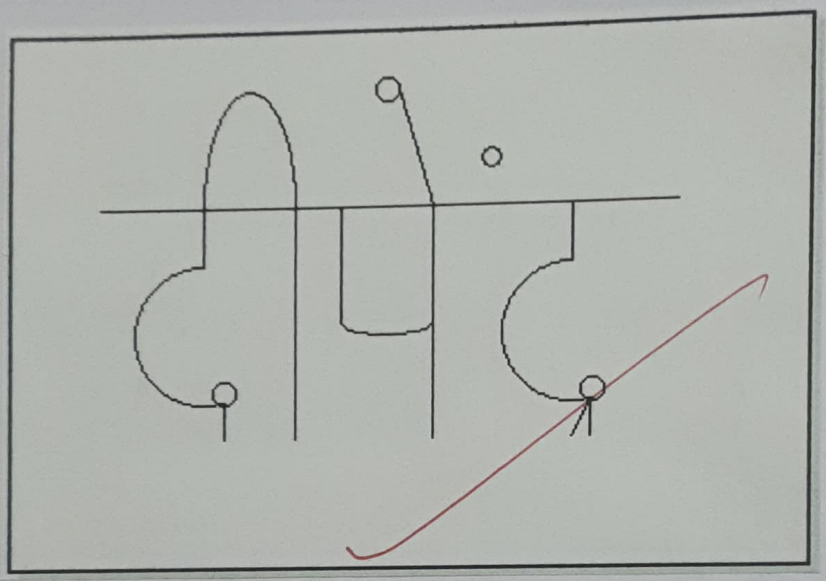
Handling
12/10/22

Teacher's Signature _____

Program - 7 → Program to print my name using graphics.

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

int main(){
    int gd = DETECT, gm;
    initgraph(&gd, &gm, "C:\\TC\\BGI");
    line(50,100,300,100);
    ellipse(115,100,0,180,20,50);
    line(135,100,135,200);
    line(95,100,95,125);
    ellipse(95,155,90,290,30,30);
    circle(104,180,5);
    line(104,185,104,200);
    line(155,100,155,150);
    line(195,100,195,200);
    ellipse(175,150,180,360,20,5);
    line(195,100,180,50);
    circle(175,50,5);
    line(255,100,255,125);
    ellipse(255,155,90,290,30,30);
    line(263,185,263,200);
    circle(264,180,5);
    line(263,185,255,200);
    circle(220,80,4);
    getch();
    closegraph();
}
```

Output ->

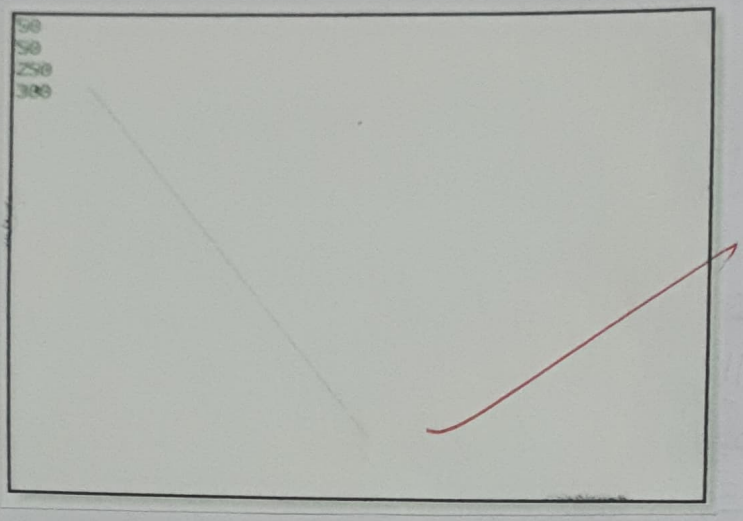


Program-8 → Program to draw line using DDA algorithm.

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

int main(){
    int gd = DETECT, gm;
    initgraph(&gd, &gm, "C:\\TC\\BGI");
    float x1, x2, y1, y2, dx, dy, x, y, xinc, yinc, step;
    scanf("%f %f %f %f", &x1, &y1, &x2, &y2);
    dx = abs(x2-x1);
    dy = abs(y2-y1);
    if(dx>dy) step = dx;
    else step = dy;
    xinc = dx/step;
    yinc = dy/step;
    x=x1; y=y1;
    putpixel(x,y, CYAN);
    while(x!=x2 && y!=y2){
        x = x+xinc;
        y = y+yinc;
        putpixel(x,y, CYAN);
    }
    getch();
    closegraph();
    return 0;
}
```

Teacher's Signature _____

Output →

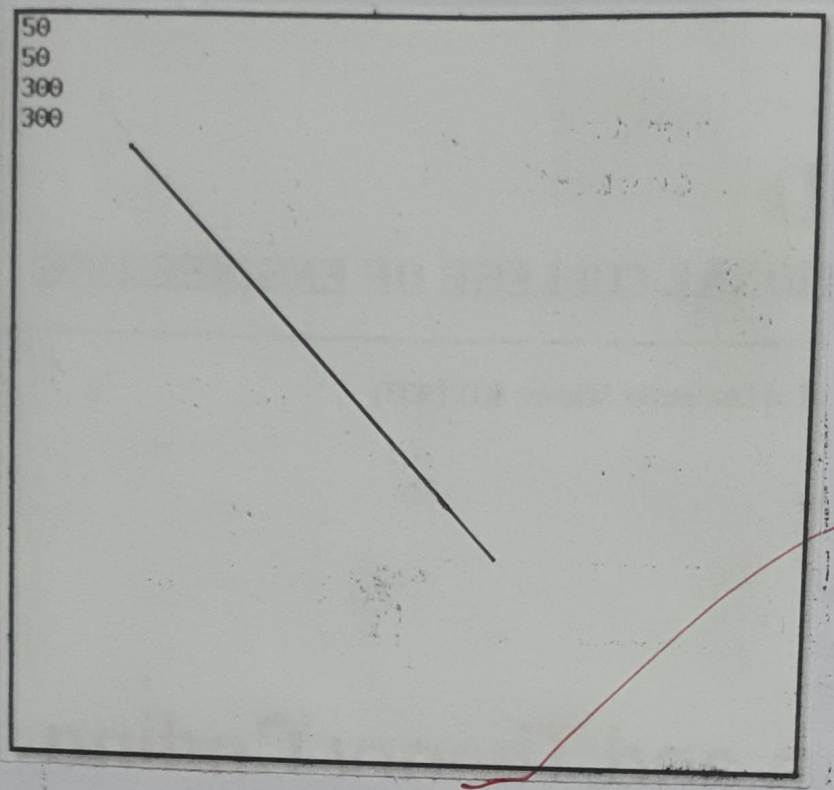


Expt. No. _____ Page No. 9

Program - 9 → Program to draw line using Bresenham's Algorithm

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

int main(){
    int gd=DETECT, gm;
    initgraph(&gd, &gm, "C:\\TC\\BGI");
    float x1, x2, y1, y2, x, y, p;
    scanf("%f %f %f %f", &x1, &y1, &x2, &y2);
    x = x1; y = y1;
    putpixel(x1, y1, CYAN);
    dx = abs(x2 - x1);
    dy = abs(y2 - y1);
    p = (2*dy - dx);
    while(x <= x2){
        if(p >= 0){
            putpixel(x, y, CYAN);
            y = y+1;
            p = p + (2*dy - 2*dx);
        }else{
            putpixel(x, y, CYAN);
            p = p + 2*dy;
        }
        x++;
    }
```

Teacher's Signature _____


50
50
300
300

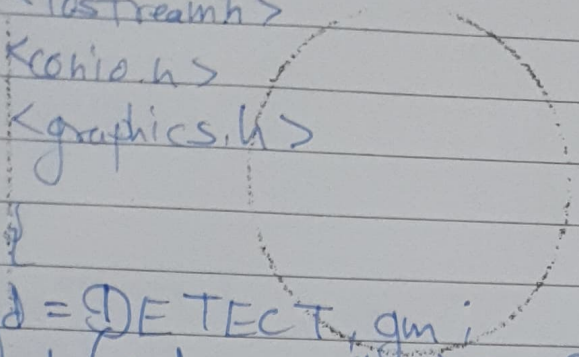
```
getch();
closegraph();
return 0;
}
```

Attendance
9/11/22

Expt. No. _____ Date _____

Page No. 11

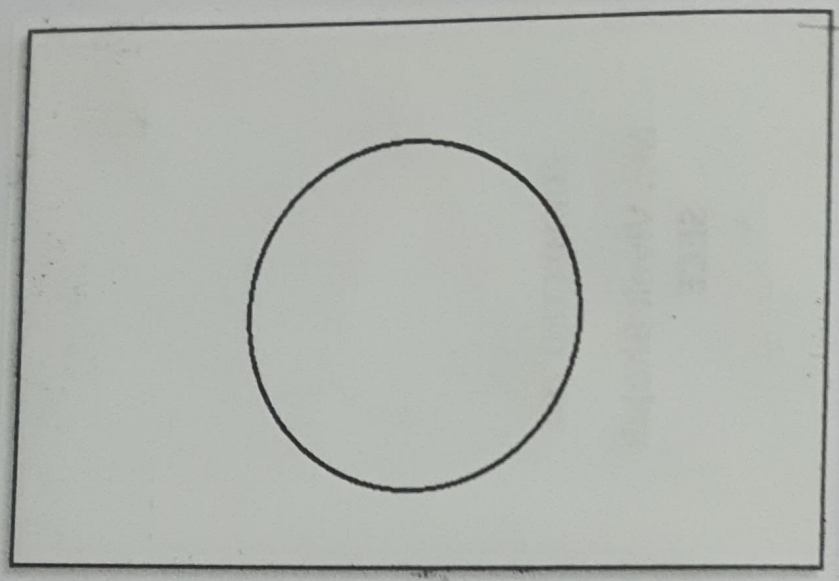
Program 10 → Program to draw a circle using Midpoint circle drawing algorithm.



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

int main(){
    int gd=DETECT, gm;
    initgraph(&gd, &gm, "C:\TC\BGI");
    int x, y, xc, yc, r;
    float p;
    cout<<"Enter centre's coordinates and radius: ";
    cin>>xc>>yc>>r;
    p=5/4-r;
    x=0;
    y=r;

    while(x<=y){
        if(P<0) p=p+2*x+3;
        else{
            p=p+2*(x-y)+5;
            y--;
        }
        putpixel(xc+x, yc+y, RED);
        putpixel(xc+y, yc+x, RED);
        putpixel(xc+y, yc-x, RED);
        putpixel(xc+x, yc-y, RED);
```

Teacher's Signature _____

```
        putpixel(xc-x, yc-y, RED);
        putpixel(xc-y, yc-x, RED);
        putpixel(xc-y, yc+x, RED);
        putpixel(xc-x, yc+y, RED);
    }
    x++;
}
getch();
closegraph();
return 0;
}
```



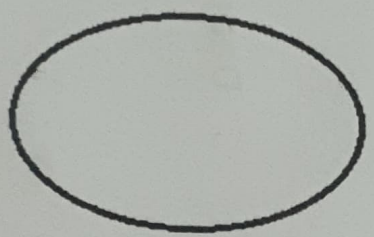
Program-11 → Drawing ellipse using midpoint algorithm.

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

int main(){
    int gd= DETECT, gm;
    initgraph(&gd, &gm, "C:\\TC\\BGI");
    float rx, ry, xc, yc;
    printf("Enter the center's coordinates and x and y radius: ");
    scanf("%f %f %f %f", &xc, &yc, &rx, &ry);
    putpixel(xc, yc-ry, RED);
    float p1 = ((ry*ry)-(rx*rx*ry)+((rx*rx)/4));
    float x=0, y=ry;
    while((2*ry*ry*x)<(2*rx*rx*y)){
        putpixel(xc+x, yc+y, RED);
        putpixel(xc-x, yc+y, RED);
        putpixel(xc+x, yc-y, RED);
        putpixel(xc-x, yc-y, RED);
        if(p1<0){
            p1 = p1+(2*ry*ry*(x+1))-(2*(rx*rx)*(y+1))
                 +(ry*ry);
            x++; y--;
        }
    }
    float p2 = ((ry*ry)*(x+0.5)*(x+0.5))+(rx*rx)*((y-1)*(y-1))
               -((rx*rx)*(ry*ry));
```

```
while(y>0){
    putpixel(xc+x, yc+y, RED);
    putpixel(xc+x, yc-y, RED);
    putpixel(xc-x, yc+y, RED);
    putpixel(xc-x, yc-y, RED);
    if(p2<0){
        p2=p2+(2*ry*ry*(x+1))-(2*rx*rx*(y+1))+(rx*rx);
        x++; y--;
    }
    else{
        p2=p2-(2*(rx*rx)*(y+1))+(rx*rx);
        y--;
    }
}
getch();
closegraph();
return 0;
}
```

Answered
23/11/22

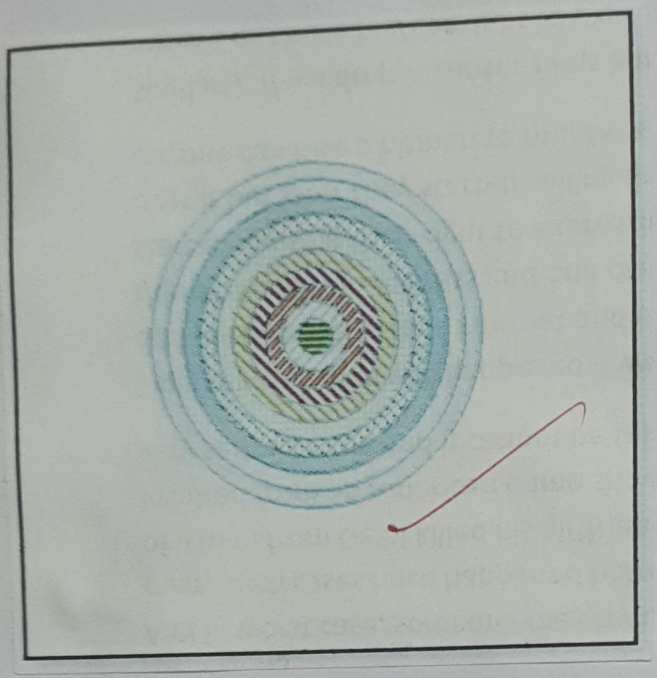
Enter the X radius and Y radius and centre point of the ellipse: 80
50
200
200



Program-12 → Write a program to multiple eccentric circles filled with different styles using floodfill and setfillstyle.

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

int main() {
        int gd = DETECT, gm;
        initgraph(&gd, &gm, "C://TC//BGI");
        int x, y;
        x = getmaxx()/2;
        y = getmaxy()/2;
        setbkcolor(WHITE);
        setcolor(CYAN);
        int style = 2;
        for(int i=0; i<100; i=i+20) {
                circle(x, y, i);
                setfillstyle(3, style%15);
                floodfill(x-2+i, y, style%15);
                style++; delay(10);
        }
        getch();
        closegraph();
        return 0;
}
```

Output →



Program-13 → Write a program to create a screensaver which display multiple circles of random sizes and style.

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

int main(){
    int gd=DETECT, gm;
    initgraph(&gd, &gm, "C:\\TC\\BGI");
    int x, y, r; style=2;
    for(int i=0; i<=30; i++){
        delay(100);
        x=rand()%500;
        y=rand()%500;
        r=rand()%100;
        circle(x,y,r);
        setfillstyle(style%15, style%15);
        floodfill(x-2+i, y, style%15);
        style++;
    }
    getch();
    closegraph();
    return 0;
}
```

Anubhav
7/12/22

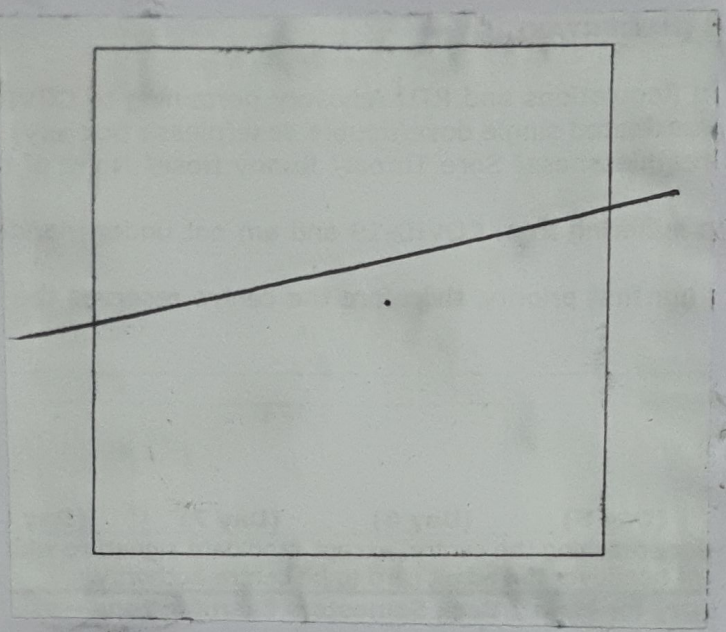
Output →

Program-14 → Write a program to clip a line using Cohen-Sutherland line clipping algorithm.

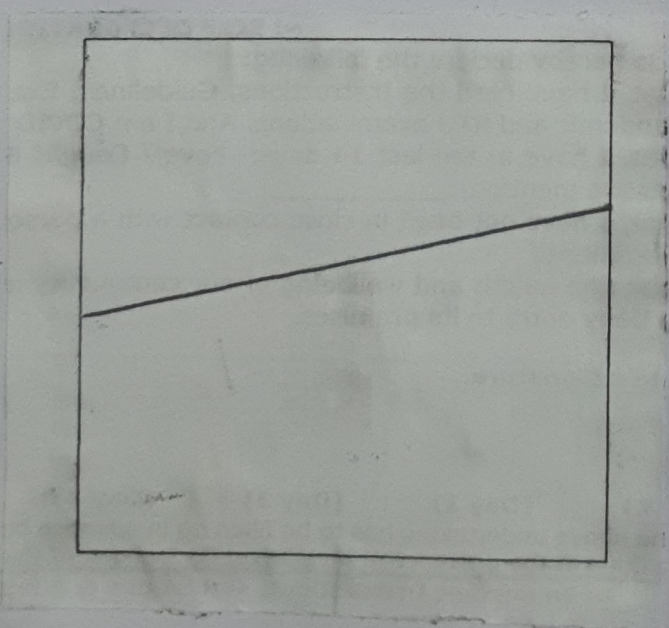
```
#include<iostream.h>
#include<graphics.h>
#include<math.h>
#include<conio.h>
using namespace std.

void main() {
        int rcode_begin[4]={0,0,0,0}, rcode_end[4]={0,0,0,0}
            region_code[4];
        int w_xmax, w_xmin, w_ymax, w_ymin, flag=0;
        float slope;
        int x1, x2, y1, y2, i, xc, yc;
        int gd= DETECT, gm;
        initgraph (&gd, &gm, "c:// TC//BGI");
        cout<<"Enter Xmin & Ymin";
        cin>> w_xmin>> w_ymin
        cout<<"Enter Xmax & Ymax";
        cin>> w_xmax>> w_ymax;
        cout<<"Enter the initial points of the line: ";
        cin>> x1 >> y1
        cout<<"Enter the final points of the line:";
        cin>> x2>> y2;
        clrscr();
        rectangle (w_xmin, w_ymin, w_xmax, w_ymax);
        line(x1, y1, x2, y2);
```

Teacher's Signature _____

Output →



Before line clipping



After line clipping.

```
if (y1 < w_ymin) {
        rcode_begin[1] = 1;
        flag = 1;
}
if (x1 > w_xmax) {
        rcode_begin[2] = 1;
        flag = 1;
}
if (x1 < w_xmin) {
        rcode_begin[3] = 1;
        flag = 1;
}
if (y1 > w_ymax) {
        rcode_begin[0] = 1
        flag = 1;
}
if (y2 > w_ymax) {
        rcode_end[0] = 1
        flag = 1;
}
if (y2 < w_ymin) {
        rcode_end[1] = 1;
        flag = 1;
}
if (x2 < w_xmin) {
        rcode_begin rcode_end[3] = 1;
        flag = 1;
}
```

```
if (x2 > w_xmax) {
        rcode_end[2] = 1;
        flag = 1;
    }
if (flag == 0) {
    cout << "No line clipping required as it is already in
            the window." << endl;
  }
  flag = 1;
  for (i=0; i<4; i++) {
      region_code[i] = rcode_begin[i] && rcode_end[i];
      if (region_code[i] == 1)
              flag = 0;
    }
  if (flag == 0) {
        cout << "Line is completely outside \n";
    }
  else {
     slope = (float)(y2-y1)/(x2-x1);
     if (rcode_begin[2] == 0 && rcode_begin[3] == 1) {
          y1 = y1 + (float)(w_xmin - x1) * slope;
          x1 = w_xmin;
       }
     if (rcode_begin[2] == 1 && rcode_begin[3] == 0) {
          y1 = y1 + (float)(w_xmax - x) * slope;
          x = w_xmax;
     }
```

```
if(rcode_begin[0]==1 && rcode[i]==0){
        x1 = x1+(float)(w_ymax-y)/slope;
        y1 = w_ymax;
}
if(rcode_begin[0]==0 && rcode_begin[1]==1){
        x1 = x1+(float)(w_ymin-y)*slope;
        y1 = w_ymin;
}
if(rcode_end[2]==0 && rcode_end[3]==1){
        y2 = y2+(float)(w_xmin-x2)*slope;
        x2 = w_xmin;
}
if(rcode_end[2]==1 && rcode_end[3]==0){
        y2 = y2+(float)(w_xmax-x2)*slope;
        x2 = w_xmax;
}
if(rcode_end[0]==1 && rcode_end[1]==0){
        x2 = x2+(float)(w_ymax-y2)/slope;
        y2 = w_ymax;
}
if(rcode[0]==0 && rcode[1]==1){
        x2 = x2+(float)(w_ymin-y2)/slope;
        y2 = w_ymin;
}
}
delay(100);
clearviewport();
rectangle(w_xmin, w_ymin, w_xmax, w_ymax);
```

```
setcolor(RED);
line(x1, y1, x2, y2);
getch();
closegraph();
return 0;
}
```

Program-15 → 2D Transformation program

```
#include<iostream.h>
#include<graphics.h>
#include<maths
using namespace std;

int main() {
    int gd = DETET, gm, s;
    initgraph (&gd, &gm, "c://TC//BGI");
    cout << "1. Translation \n 2. Rotation \n 3. Scaling \n 4.
      Reflection \n 5. Shearing" << endl;
    cout << "Selection: ";
    cin >> s;
  switch(s){
   case 1:{
     int x1=200, y1=150, x2=300, y2=250;
     int tx =50, ty=50;
     cout "Rectangle before translation. \n";
     rectangle (x1, y1, x2, y2);
      cout << "Rectangle after translation \n";
     rectangle (x1+tx, y1+ty, x2+tx, y2+ty);
      getch();
     break;
   }

  case 2: {
     long x1 = 200, y1 = 200, x2 = 300, y2 = 300;
     double a;
     cout << "Rectangle before rotation \n";
```

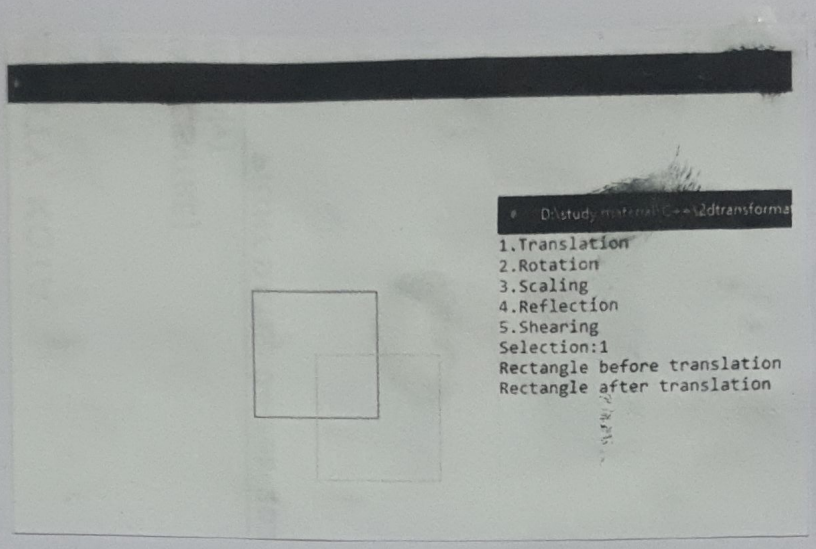
```
rectangle(x1, y1, x2, y2);
cout << "Enter angle of rotation:";
cin >> a;
a = (a * 3.14)/180;
long xr = x1 + ((x2 - x1) * cos(a) - (y2 - y1) * sin(a));
long yr = y1 + ((x2 - x1) * sin(a) + (y2 - y1) * cos(a));
cout << "Rectangle after rotation\n";
rectangle(x1, y1, xr, yr);
getch();
break;
}
Case 3: {
int x1 = 30, y1 = 30, x2 = 70, y2 = 70, x = 2, y = 2;
cout << "Rectangle before scaling\n";
rectangle(x1, y1, x2, y2);
cout << "Rectangle after scaling\n";
rectangle(x1 * x, y1 * y, x2 * x, y2 * y);
getch();
break;
}
Case 4: {
int x1 = 200, y1 = 300, x2 = 500, y2 = 300, x3 = 350, y3 = 400;
cout << "Triangle before reflection\n";
line(x1, y1, x2, y2);
line(x1, y1, x3, y3);
line(x2, y2, x3, y3);
cout << "Triangle after reflection\n";
line(x1, -y1 + 500, x2, -y2 + 500);
line(x1, -y1 + 500, x3, -y3 + 500);
```

Teacher's Signature

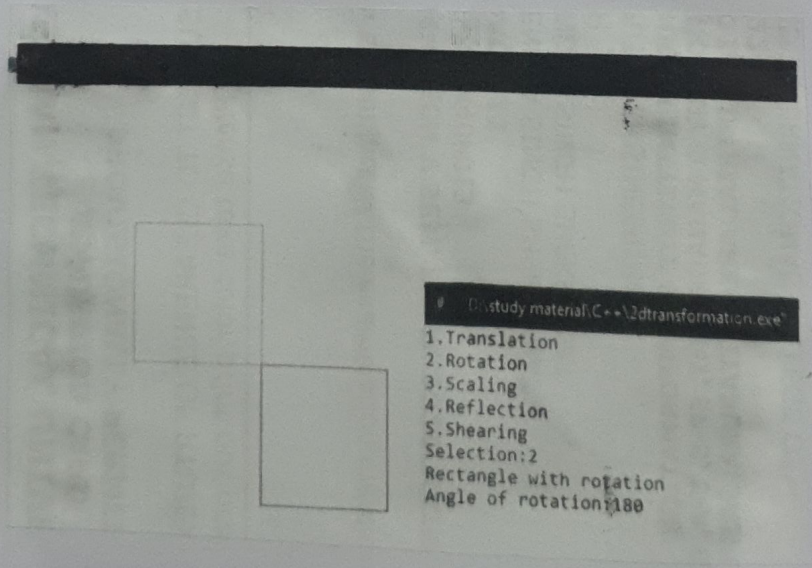
```
        line(x2, -y2+500, x3, -y3+500);
        getch();
        break;
    }
    case 5:{
        int x1=400, y1=100, x2=600, y2=100, x3=400, y3=200,
            x4=600, y4=200, shx=2;
        cout<<"Rectangle before shearing\n";
        line(x1,y1,x2,y2);
        line(x2,y2,x3,y3);
        line(x3,y3,x4,y4);
        line(x1,y1,x4,y4);
        cout<<"Rectangle after shearing\n";
        x1=x1+shx*y1;
        x2=x2+shx*y2;
        x3=x3+shx*y3;
        x4=x4+shx*y4;
        line(x1,y1,x2,y2);
        line(x2,y2,x3,y3);
        line(x3,y3,x4,y4);
        line(x1,y1,x4,y4);
        getch();
        break()
    }
    default:{
        cout<<"Invalid selection"<<endl;
        break;
    }
}
```

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

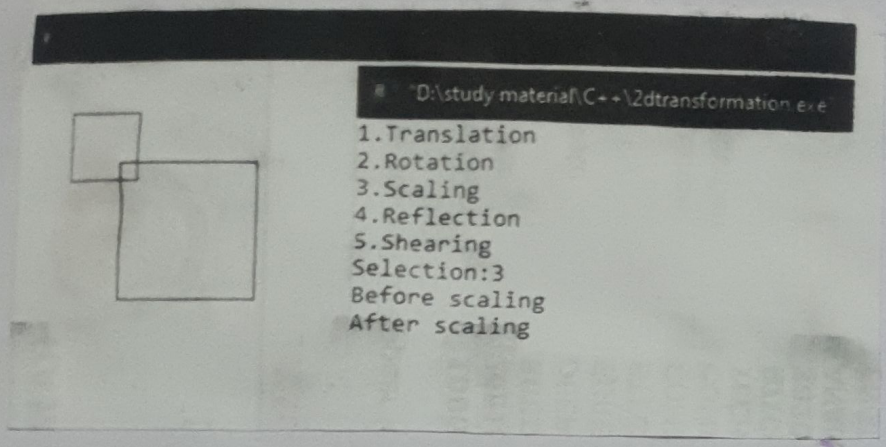
Output →



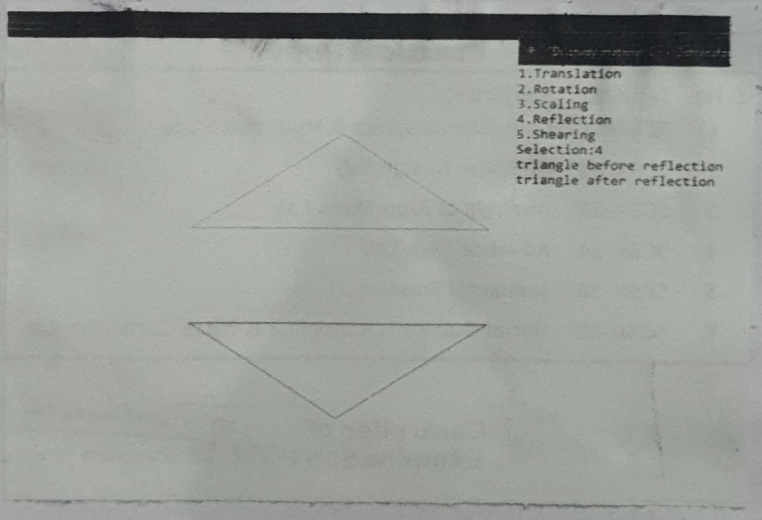
Translation



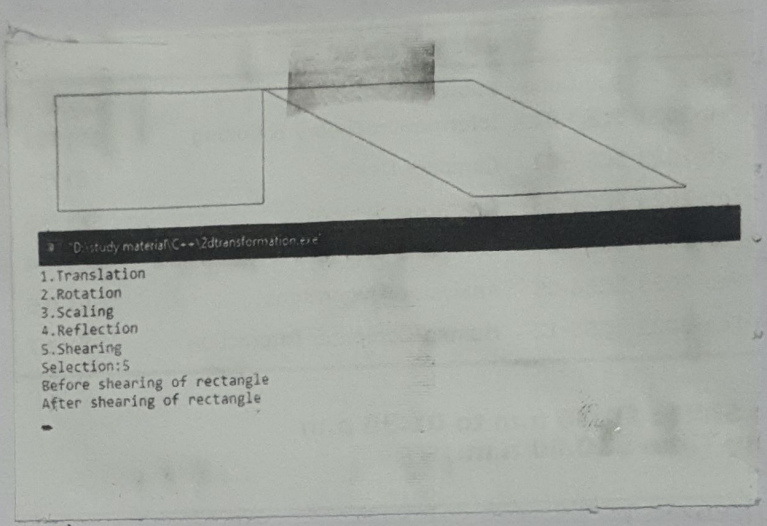
Rotation.



Scaling



Reflection.


D:\study material\C++\2dtransformation.exe
1.Translation
2.Rotation
3.Scaling
4.Reflection
5.Shearing
Selection:5
Before shearing of rectangle
After shearing of rectangle

Shearing.