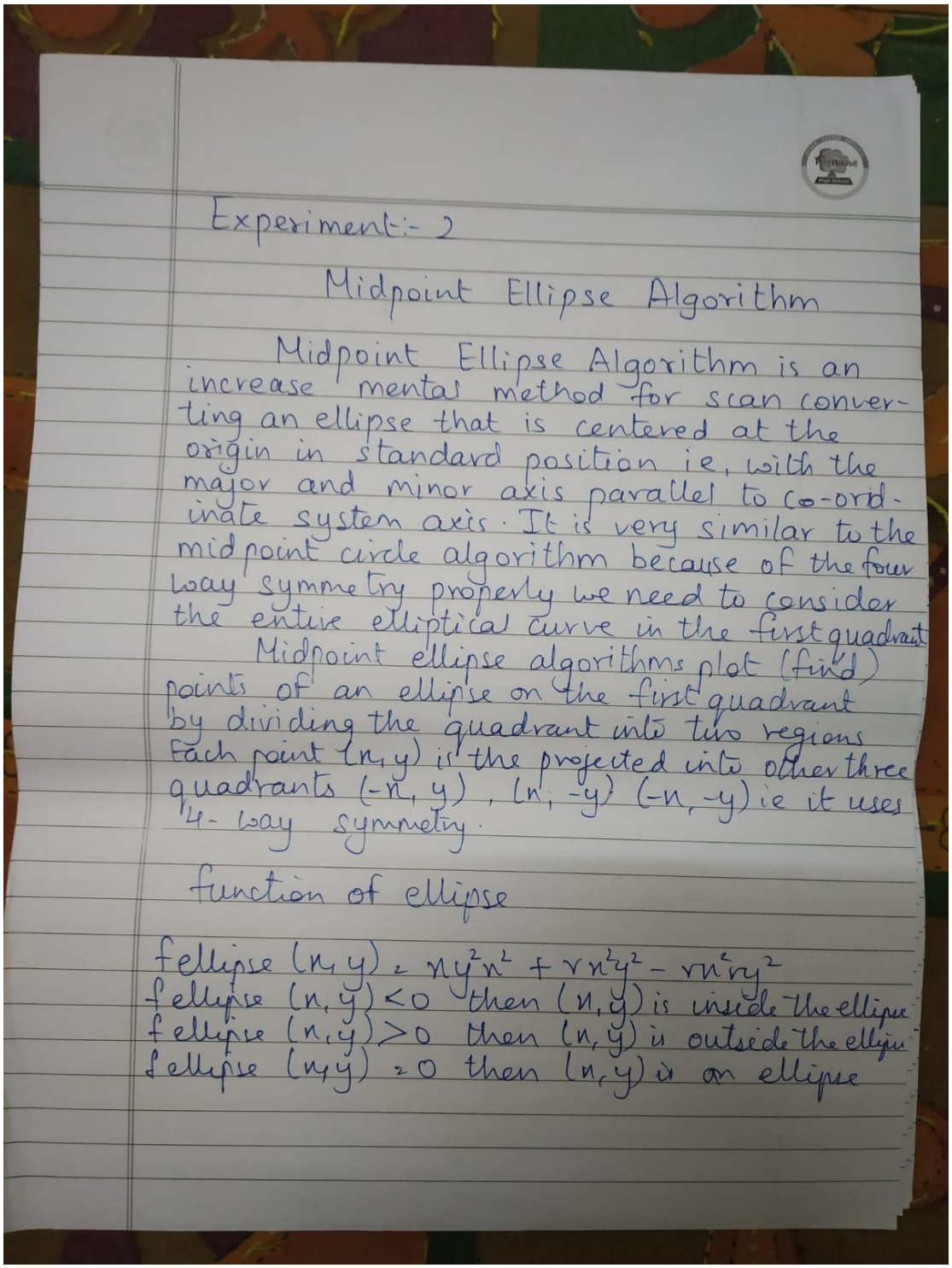
Assignment performed by Gaurav Amarnani, D7A Roll no. 67.
CG LAB 2 - Midpoint Ellipse Algorithm.

Aim: Implement midpoint Ellipse algorithm.

Theory:



Experiment:- 2

Midpoint Ellipse Algorithm

Midpoint Ellipse Algorithm is an increase mental method for scan converting an ellipse that is centered at the origin in standard position ie, with the major and minor axis parallel to co-ordinate system axis. It is very similar to the midpoint circle algorithm because of the four way symmetry property we need to consider the entire elliptical curve in the first quadrant

Midpoint ellipse algorithms plot (find) points of an ellipse on the first quadrant by dividing the quadrant into two regions Each point (x, y) is the projected into other three quadrants $(-x, y)$, $(x, -y)$ $(-x, -y)$ ie it uses 4- way symmetry.

function of ellipse

$f_{ellipse}(x, y) = ry^2x^2 + rx^2y^2 - rx^2ry^2$

$f_{ellipse}(x, y) < 0$ then (x, y) is inside the ellipse

$f_{ellipse}(x, y) > 0$ then (x, y) is outside the ellipse

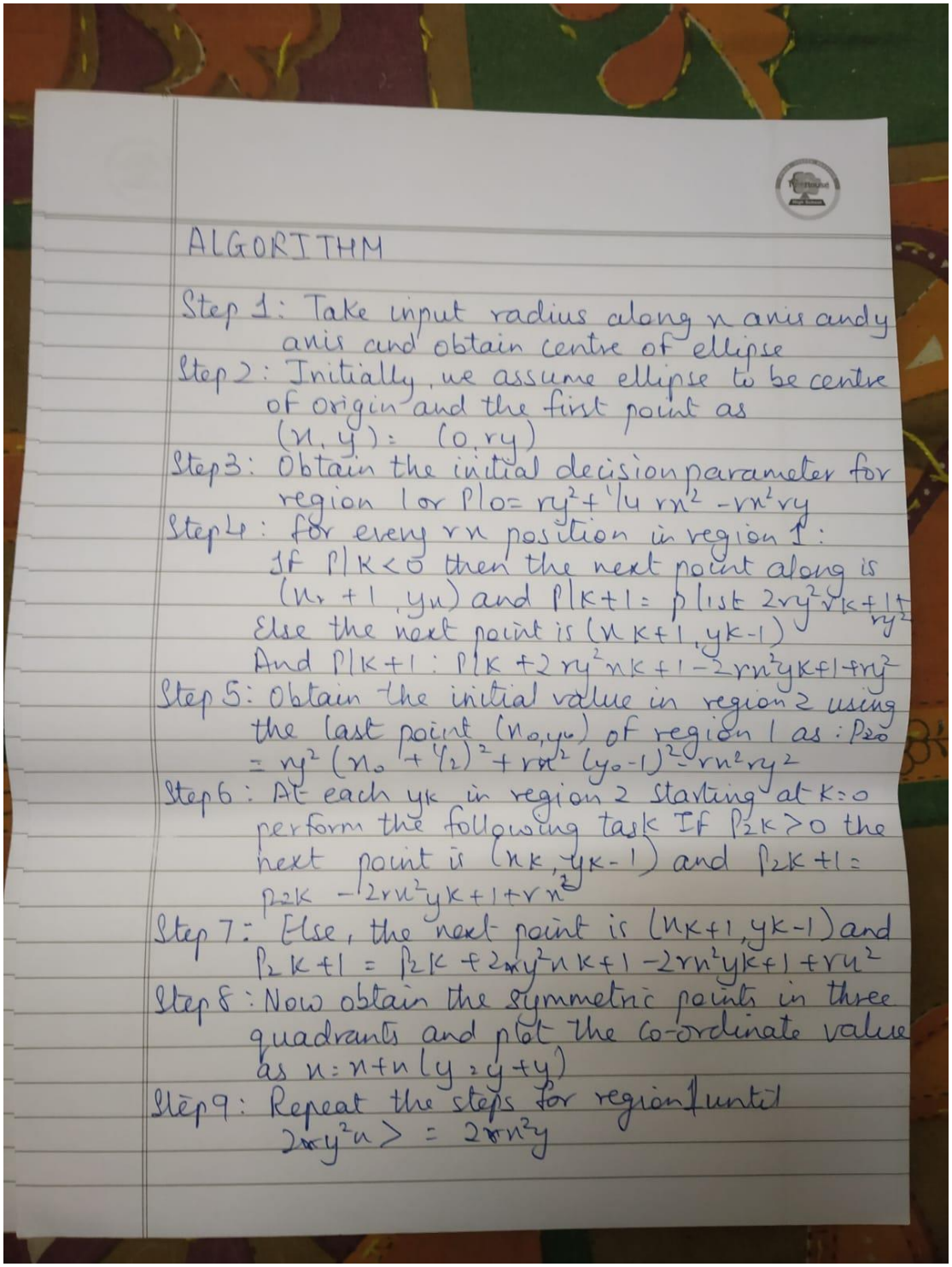
$f_{ellipse}(x, y) = 0$ then (x, y) is on ellipse

Decision Parameter :-

Initially we have two decision parameter p_{10} in region 1 and p_{20} in region 2

$$p_{10} = ry^2 + \frac{1}{4}rn^2 - rn^2ry$$

Algorithm:



ALGORITHM

Step 1: Take input radius along x axis and y axis and obtain centre of ellipse

Step 2: Initially, we assume ellipse to be centre of origin and the first point as $(x, y) = (0, ry)$

Step 3: Obtain the initial decision parameter for region 1 or $P1_0 = ry^2 + 1/4\, rx^2 - rx^2 ry$

Step 4: For every x position in region 1:
If $P1_K < 0$ then the next point along is $(x_K + 1, y_K)$ and $P1_{K+1} = P1_K + 2ry^2 x_{K+1} + ry^2$
Else the next point is $(x_K + 1, y_K - 1)$
And $P1_{K+1} = P1_K + 2ry^2 x_{K+1} - 2rx^2 y_{K+1} + ry^2$

Step 5: Obtain the initial value in region 2 using the last point (x_0, y_0) of region 1 as: $P2_0 = ry^2 (x_0 + 1/2)^2 + rx^2 (y_0 - 1)^2 - rx^2 ry^2$

Step 6: At each y_K in region 2 starting at $K = 0$ perform the following task If $P2_K > 0$ the next point is $(x_K, y_K - 1)$ and $P2_{K+1} = P2_K - 2rx^2 y_{K+1} + rx^2$

Step 7: Else, the next point is $(x_{K+1}, y_K - 1)$ and $P2_{K+1} = P2_K + 2ry^2 x_{K+1} - 2rx^2 y_{K+1} + rx^2$

Step 8: Now obtain the symmetric points in three quadrants and plot the co-ordinate value as $x = x + x_c$, $y = y + y_c$

Step 9: Repeat the steps for region 1 until $2ry^2 x >= 2rx^2 y$

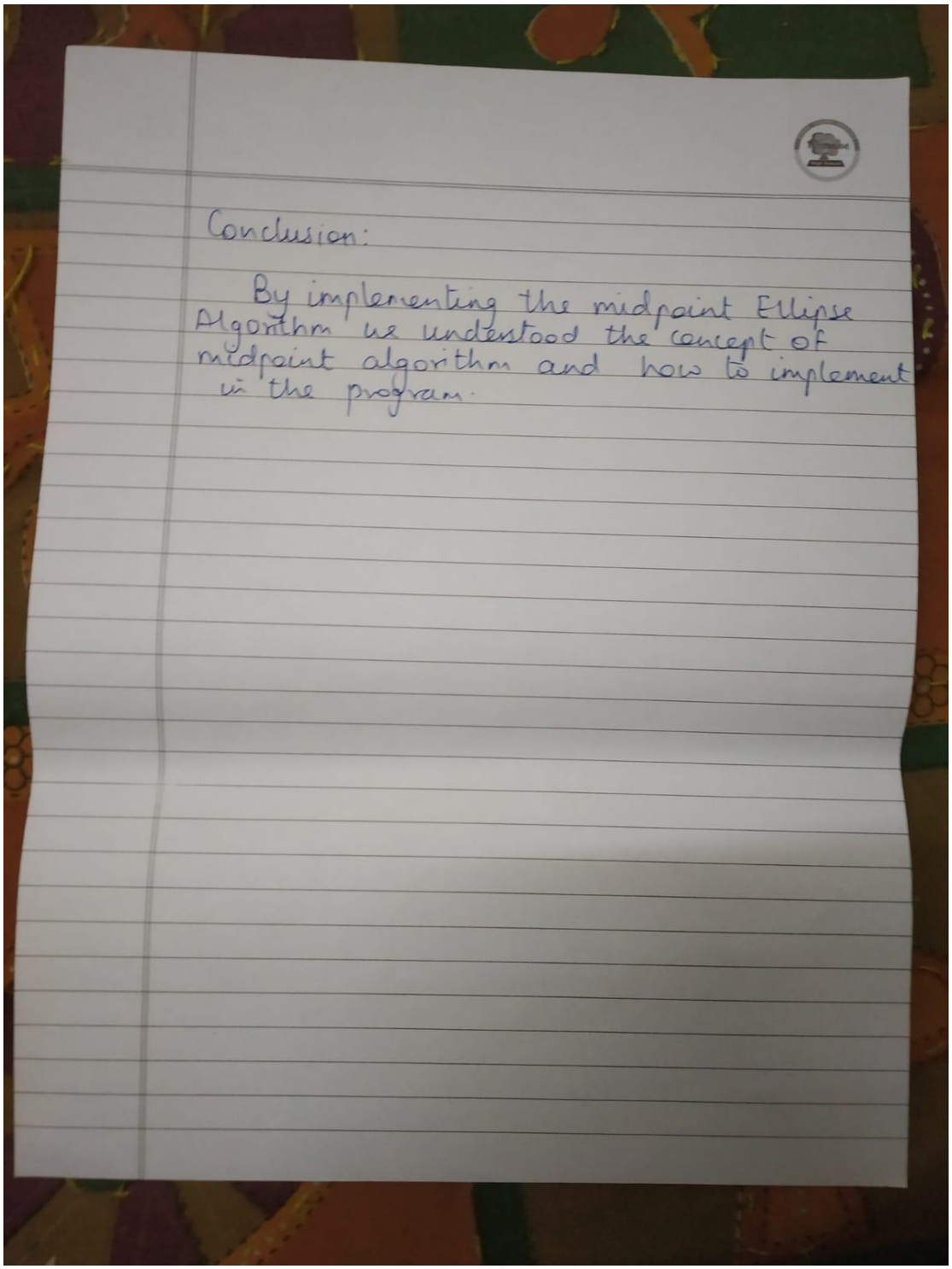
Program:

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

//Performed by Gaurav Amarnani DSE CMPN.

void midptellipse(int rx, int ry, int xc, int yc) {

float dx,dy,d1,d2,x,y;
x = 0;
y = ry;
d1 = (ry * ry) - (rx * rx * ry) + (0.25 * rx * rx);
dx = 2 * ry * ry * x;
dy = 2 * rx * rx * y;
while (dx < dy) {
putpixel( x + xc, y + yc, WHITE);
putpixel( -x + xc, y + yc, WHITE);
putpixel( x + xc, -y + yc, WHITE);
putpixel( -x + xc, -y + yc, WHITE);

if (d1 < 0) {
x++;
dx = dx + (2*ry*ry);
d1 = d1 + dx+(ry*ry);
}
else {
y--;
dx = dx + (2 * ry * ry);
dy = dy - (2 * rx * rx);
d1 = d1 + dx - dy + (ry * ry);
}
}


d2 = ((ry * ry) * ((x + 0.5) * (x + 0.5)))+ ((rx * rx) * ((y - 1) * (y - 1))) - (rx * rx * ry *
ry);
while (y >= 0) {
putpixel( x + xc, y + yc, WHITE);
putpixel( -x + xc, y + yc, WHITE);
putpixel( x + xc, -y + yc, WHITE);
putpixel( -x + xc, -y + yc, WHITE);

if (d2 > 0) {
```

```
y--;
dy = dy - (2 * rx * rx);
d2 = d2 + (rx * rx) - dy;
}
else {
y--;
x++;
dx = dx + (2 * ry * ry);
dy = dy - (2 * rx * rx);
d2 = d2 + dx - dy + (rx * rx);
}
}
}

void main() {
int gd = DETECT, gm;
initgraph(&gd, &gm,"c:\\turboc3\\bgi");
midptellipse(10, 15, 50, 50);
getch();
}
```

Output:

Conclusion:


Conclusion:
By implementing the midpoint Ellipse Algorithm we understood the concept of midpoint algorithm and how to implement in the program.