Program 4

Aim: Write a program to draw/create a teddy bear (elliptical face, eyes and mouth and circular tummy) using Midpoint Ellipse drawing Algorithm.

Theory:

Mid-Point Ellipse Drawing Algorithm uses symmetry of ellipse to draw an ellipse in computer graphics. This algorithm is implemented for one quadrant only. We divide the quadrant into two regions and the boundary of two regions is the point at which the curve has a slope of -1. We process by taking unit steps in the x direction to the point P (where curve has a slope of -1), then taking unit steps in the y direction and applying midpoint algorithm at every step.

Algorithm:

1. Input rx’ ry’, and ellipse center (xc’, yc’), and obtain the first point on an ellipse centered on the origin as

$$(x0, y0) = (0, ry)$$

2. Calculate the initial value of the decision parameter in region 1 as

$$p10 = r2y - r2xry + \frac{1}{4} r2x$$

3. At each xk position in region 1, starting at k = 0 perform the following test: If plk < 0, the next point along the ellipse centered on (0, 0) is (xk + 1 ,yk) and

$$P1k+1 = p1k + 2r2yxk+1 + r2y$$

Otherwise, the next point along the circle is (xk +1 , yk -1) and

$$P1k+1 = p1k + 2r2yxk+1 - 2r2xyk+1 + r2y$$

with

$$2r2yxk+1 = 2r2yxk + 2r2y \quad 2r2xyk+1 = 2r2xyk - 2r2x$$

And continue until 2r2yx >= 2r2xy.

4. Calculate the initial value of the decision parameter in region 2 using the last point (x0, y0) calculated in region 1 as

p20 = r2y(x0 + ½)2 + r2x(y0 – 1)2 – r2xr2y

5. At each yk position in region 2, starting at k = 0, perform the following test: If p2k> 0, the next point along the ellipse centered on (0, 0) is (xk , yk - 1) and

P2k+1 = p2k - 2r2xyk+1 + r2x

Otherwise, the next point along the circle is (xk + 1, yk - 1) and

P2k+1 = p2k + 2r2yxk+1 - 2r2xyk+1 + r2x

using the same incremental calculations for x and y as in region 1.

6. Determine symmetry points in the other three quadrants.

7. Move each calculated pixel position (x, y) onto the elliptical path centered on (xc , yc) and plot the coordinate values:

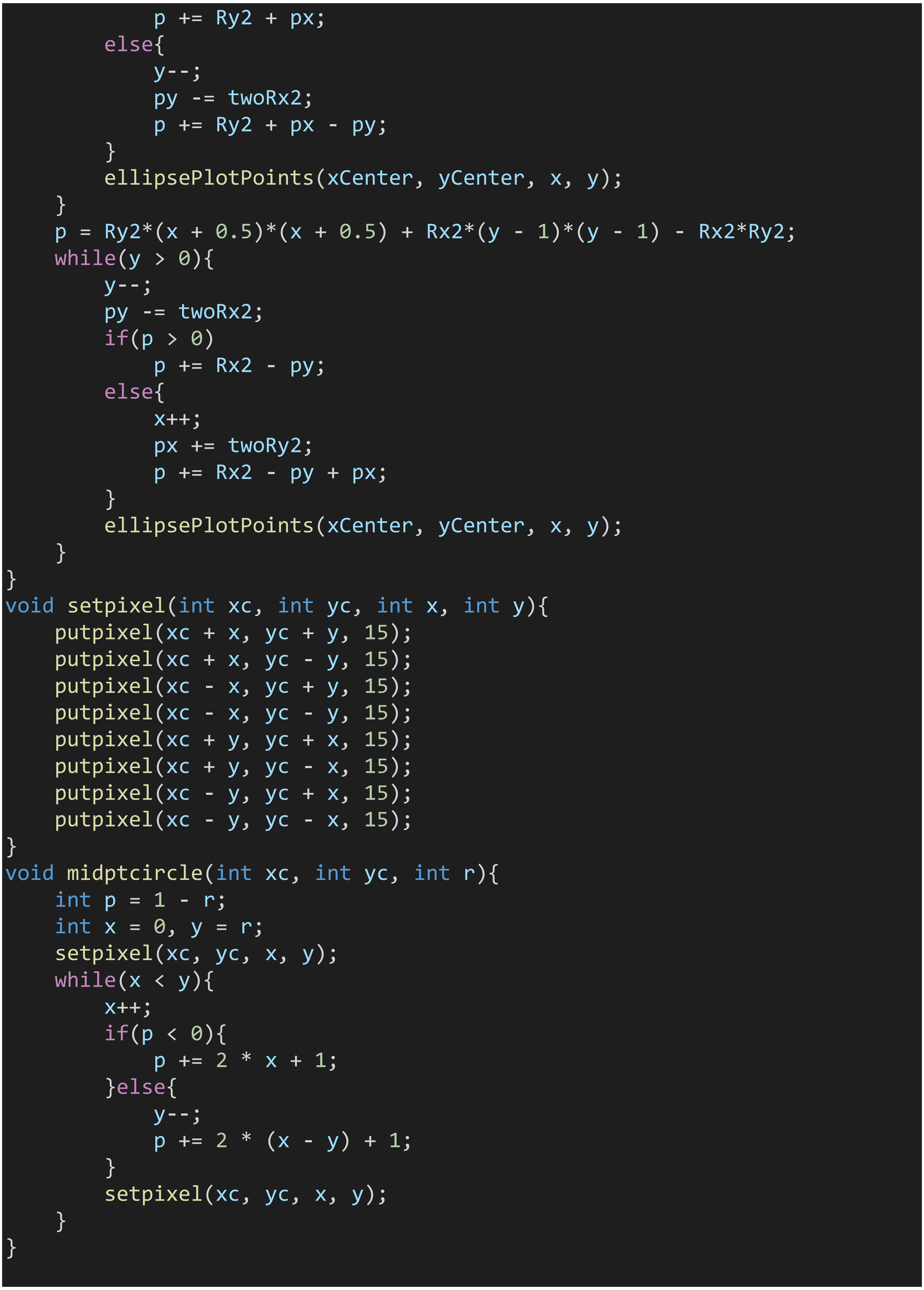
x = x + xc, y = y + yc

8. Repeat the steps for region 1 until 2r2yx >= 2r2xy.

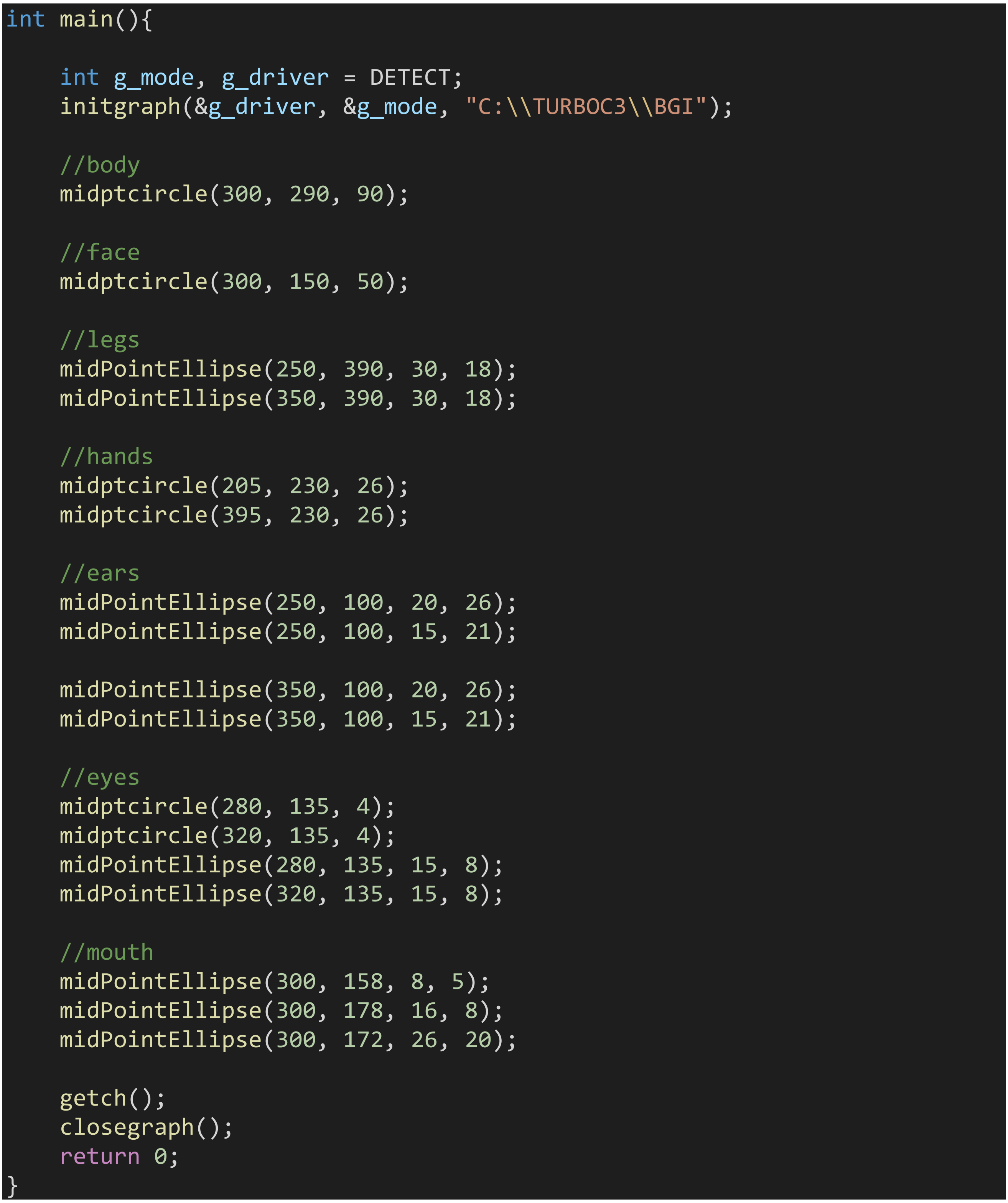
Code:

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

void ellipsePlotPoints(int xCenter, int yCenter, int x, int y){
    putpixel(xCenter + x, yCenter + y, WHITE);
    putpixel(xCenter - x, yCenter + y, WHITE);
    putpixel(xCenter + x, yCenter - y, WHITE);
    putpixel(xCenter - x, yCenter - y, WHITE);
}
void midPointEllipse(int xCenter, int yCenter, int Rx, int Ry){
    int Rx2 = Rx * Rx;
    int Ry2 = Ry * Ry;
    int twoRx2 = 2 * Rx2;
    int twoRy2 = 2 * Ry2;
    int p, x = 0, y = Ry, px = 0;
    int py = twoRx2 * y;
    ellipsePlotPoints(xCenter, yCenter, x, y);
    p = Ry2 - (Rx2 * Ry) + (0.25 * Rx2);
    while(px < py){
        x++;
        px += twoRy2;
        if (p < 0)
```



```
            p += Ry2 + px;
        else{
            y--;
            py -= twoRx2;
            p += Ry2 + px - py;
        }
        ellipsePlotPoints(xCenter, yCenter, x, y);
    }
    p = Ry2*(x + 0.5)*(x + 0.5) + Rx2*(y - 1)*(y - 1) - Rx2*Ry2;
    while(y > 0){
        y--;
        py -= twoRx2;
        if(p > 0)
            p += Rx2 - py;
        else{
            x++;
            px += twoRy2;
            p += Rx2 - py + px;
        }
        ellipsePlotPoints(xCenter, yCenter, x, y);
    }
}
void setpixel(int xc, int yc, int x, int y){
    putpixel(xc + x, yc + y, 15);
    putpixel(xc + x, yc - y, 15);
    putpixel(xc - x, yc + y, 15);
    putpixel(xc - x, yc - y, 15);
    putpixel(xc + y, yc + x, 15);
    putpixel(xc + y, yc - x, 15);
    putpixel(xc - y, yc + x, 15);
    putpixel(xc - y, yc - x, 15);
}
void midptcircle(int xc, int yc, int r){
    int p = 1 - r;
    int x = 0, y = r;
    setpixel(xc, yc, x, y);
    while(x < y){
        x++;
        if(p < 0){
            p += 2 * x + 1;
        }else{
            y--;
            p += 2 * (x - y) + 1;
        }
        setpixel(xc, yc, x, y);
    }
}
```



```c
int main(){

    int g_mode, g_driver = DETECT;
    initgraph(&g_driver, &g_mode, "C:\\TURBOC3\\BGI");

    //body
    midptcircle(300, 290, 90);

    //face
    midptcircle(300, 150, 50);

    //legs
    midPointEllipse(250, 390, 30, 18);
    midPointEllipse(350, 390, 30, 18);

    //hands
    midptcircle(205, 230, 26);
    midptcircle(395, 230, 26);

    //ears
    midPointEllipse(250, 100, 20, 26);
    midPointEllipse(250, 100, 15, 21);

    midPointEllipse(350, 100, 20, 26);
    midPointEllipse(350, 100, 15, 21);

    //eyes
    midptcircle(280, 135, 4);
    midptcircle(320, 135, 4);
    midPointEllipse(280, 135, 15, 8);
    midPointEllipse(320, 135, 15, 8);

    //mouth
    midPointEllipse(300, 158, 8, 5);
    midPointEllipse(300, 178, 16, 8);
    midPointEllipse(300, 172, 26, 20);

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

Output: