**NATIONAL INSTITUTE OF TECHNOLOGY , RAIPUR**

Diagram

Description automatically generated

**COMPUTER GRAPHICS LAB REPORT**

**NAME:** ANIMESH AGRAWAL

**ROLL NO**.: 20118010

**SUBJECT:** COMPUTER GRAPHICS LAB

**COURSE:** B. TECH

**BRANCH:** INFORMATION TECHNOLOGY (IT)

**SEMESTER:** 6th

**Submitted To:** DR. SANJAY KUMAR

**Que 1). Write a program to draw the pixel (x,y) and display the color of the pixel.**

**Code :**

#include <stdio.h>

#include <conio.h>

#include <graphics.h>

void main()

{

    int gd, gm = DETECT;

    initgraph(&gm, &gd, "C:\\TURBOC3\\BGI");

    putpixel(500, 400, RED);

    putpixel(100, 500, WHITE);

    putpixel(250, 200, BLUE);

    putpixel(200, 100, YELLOW);

    putpixel(240, 150, GREEN);

    putpixel(270, 300, RED);

    getch();

}

**Result :**



**Que 2). Write a program to draw hut and smiley using various graphics function.**

**Draw a hut :**

#include <conio.h>

#include <graphics.h>

int main(){

    int gdr = DETECT, gm;

    initgraph(&gdr, &gm, "C:\\TURBOC3\\BGI");

    line(100, 100, 150, 50);

    line(150, 50, 200, 100);

    line(150, 50, 350, 50);

    line(350, 50, 400, 100);

    rectangle(100, 100, 200, 200);

    rectangle(200, 100, 400, 200);

    rectangle(130, 130, 170, 200);

    rectangle(250, 120, 350, 180);

    setfillstyle(2, 3);

    floodfill(131, 131, WHITE);

    floodfill(201, 101, WHITE);

    setfillstyle(SOLID\_FILL, 7);

    floodfill(101, 101, WHITE);

    floodfill(150, 52, WHITE);

    floodfill(163, 55, WHITE);

    floodfill(251, 121, WHITE);

    getch();

    closegraph();

   return 0;

}

**Result :**

**Icon

Description automatically generated with low confidence**

**Draw a smiley :**

#include <graphics.h>

#include <conio.h>

int main()

{

    int graphicdriver = DETECT, graphicmode;

    initgraph(&graphicdriver, &graphicmode, "C:\\TURBOC3\\BGI");

    outtextxy(10, 10 + 10, "Program to draw a smiley face in C graphics");

    setcolor(YELLOW);

    circle(300, 100, 40);

    setfillstyle(SOLID\_FILL, YELLOW);

    floodfill(300, 100, YELLOW);

    setcolor(BLACK);

    setfillstyle(SOLID\_FILL, BLACK);

    fillellipse(310, 85, 2, 6);

    fillellipse(290, 85, 2, 6);

    ellipse(300, 100, 205, 335, 20, 9);

    ellipse(300, 100, 205, 335, 20, 10);

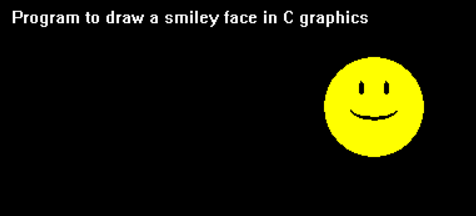
    ellipse(300, 100, 205, 335, 20, 11);

    getch();

    return 0;

}

**Result :**



**Que 3). Write a program to implement DDA line drawing algorithm.**

**Code :**

#include <graphics.h>

#include <math.h>

#include <stdio.h>

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

int abs(int n) { return ((n > 0) ? n : (n \* (-1))); }

void DDA(int X0, int Y0, int X1, int Y1){

    int dx = X1 - X0;

    int dy = Y1 - Y0;

    int steps = abs(dx) > abs(dy) ? abs(dx) : abs(dy);

    float Xinc = dx / (float)steps;

    float Yinc = dy / (float)steps;

    float X = X0;

    float Y = Y0;

    for (int i = 0; i <= steps; i++){

        putpixel(round(X), round(Y),

                 RED);

        X += Xinc;

        Y += Yinc;

        delay(10);

    }

}

int main(){

    int gd = DETECT, gm;

    initgraph(&gd, &gm, "C:\\TURBOC3\\BGI");

    int X0 = 2, Y0 = 2, X1 = 14, Y1 = 16;

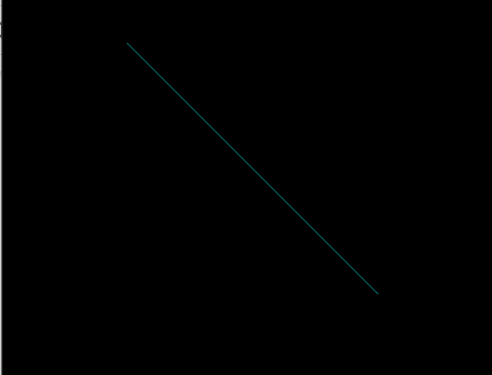
    DDA(2, 2, 14, 16);

    return 0;

}

**Result :**

Text

Description automatically generated 

**Que 4). Write a program to implement Bresenham’s line drawing algorithm.**

**Code :**

#include <stdio.h>

#include <conio.h>

#include <graphics.h>

#include <math.h>

void swap(int &x, int &y){

    int k = x;

    x = y;

    y = k;

}

int main(){

    int gd = DETECT, gm = DETECT, x1, x2, y1, y2, dx, dy, p, k;

    float m = 0;

    printf("Enter the starting point x1 & y1\n");

    scanf("%d%d", &x1, &y1);

    printf("Enter the end point x2 & y2\n");

    scanf("%d%d", &x2, &y2);

    dx = abs(x2 - x1);

    dy = abs(y2 - y1);

    m = (float)(y2 - y1) / (x2 - x1);

    initgraph(&gd, &gm, "C:\\Turboc3\\BGI");

    cleardevice();

    if (fabs(m) > 1){

        swap(x1, y1);

        swap(x2, y2);

        swap(dx, dy);

    }

    if ((x1 > x2)){

        x1 = x2;

        y1 = y2;

    }

    p = 2 \* dy - dx;

    for (k = 0; k < abs(dx); k++){

        if (p < 0){

            p = p + 2 \* dy;

        }

        else{

            if (m < 0)

                y1--;

            else

                y1++;

            p = p + (2 \* dy) - (2 \* dx);

        }

        if (fabs(m) <= 1)

            putpixel(x1++, y1, 15);

        else

            putpixel(y1, x1++, 15);

    }

    getch();

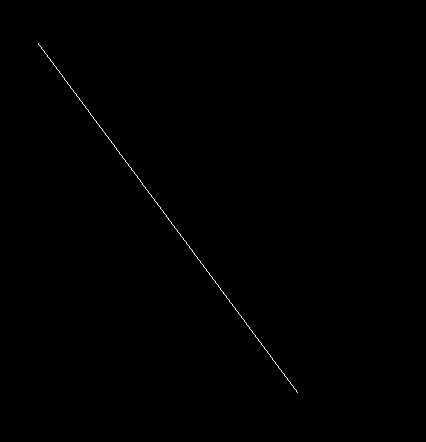
    return 0;

}

**Result :**

Text

Description automatically generated with medium confidence



**Que 5). Write a program to implement Bresenham’s Circle drawing algorithm.**

**Code :**

#include <stdio.h>

#include <dos.h>

#include <graphics.h>

#include <conio.h>

void plotPoints(int cx, int cy, int x, int y){

    putpixel(cx + x, cy + y, YELLOW);

    putpixel(cx - x, cy + y, YELLOW);

    putpixel(cx + x, cy - y, YELLOW);

    putpixel(cx - x, cy - y, YELLOW);

    putpixel(cx + y, cy + x, YELLOW);

    putpixel(cx - y, cy + x, YELLOW);

    putpixel(cx + y, cy - x, YELLOW);

    putpixel(cx - y, cy - x, YELLOW);

}

int main(){

    int cx, cy, x = 0, y, r, p;

    int gd = DETECT, gm;

    printf("Enter the coordinates of centre of the circle: ");

    scanf("%d %d", &cx, &cy);

    printf("Enter radius of : ");

    scanf("%d", &r);

    y = r;

    p = 3 - 2 \* r;

    initgraph(&gd, &gm,  "C:\\Turboc3\\BGI");

    cleardevice();

    while (x < y){

    plotPoints(cx, cy, x, y);

    x++;

    if (p < 0)

        p = p + 4 \* x + 6;

    else{

        y--;

        p = p + 4 \* (x - y) + 10;

    }

    plotPoints(cx, cy, x, y);

    delay(200);

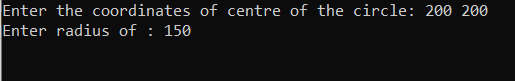
    }

    getch();

    return 0;

}

**Result :**



**Shape, circle

Description automatically generated**

**Que 6). Write a program to draw an Ellipse using Mid-Point algorithm.**

**Code :**

#include <stdio.h>

#include <graphics.h>

#include <conio.h>

#include <dos.h>

void main(){

    long x, y, x\_center, y\_center;

    long a\_sqr, b\_sqr, fx, fy, d, a, b, tmp1, tmp2;

    int g\_driver = DETECT, g\_mode;

    clrscr();

    printf("\*\*\*\*\*\*\*\*\* MID POINT ELLIPSE ALGORITHM \*\*\*\*\*\*\*\*\*");

    printf("\n\n Enter coordinate x and y = ");

    scanf("%ld%ld", &x\_center, &y\_center);

    printf("\n Now enter constants a and b = ");

    scanf("%ld%ld", &a, &b);

    initgraph(&g\_driver, &g\_mode, "C:\\TURBOC3\\BGI");

    cleardevice();

    x = 0;

    y = b;

    a\_sqr = a \* a;

    b\_sqr = b \* b;

    fx = 2 \* b\_sqr \* x;

    fy = 2 \* a\_sqr \* y;

    d = b\_sqr - (a\_sqr \* b) + (a\_sqr \* 0.25);

    do{

        putpixel(x\_center + x, y\_center + y, 1);

        putpixel(x\_center - x, y\_center - y, 1);

        putpixel(x\_center + x, y\_center - y, 1);

        putpixel(x\_center - x, y\_center + y, 1);

        if (d < 0){

            d = d + fx + b\_sqr;

        }

        else{

            y = y - 1;

            d = d + fx + -fy + b\_sqr;

            fy = fy - (2 \* a\_sqr);

        }

        x = x + 1;

        fx = fx + (2 \* b\_sqr);

        delay(10);

    } while (fx < fy);

    tmp1 = (x + 0.5) \* (x + 0.5);

    tmp2 = (y - 1) \* (y - 1);

    d = b\_sqr \* tmp1 + a\_sqr \* tmp2 - (a\_sqr \* b\_sqr);

    do{

        putpixel(x\_center + x, y\_center + y, 1);

        putpixel(x\_center - x, y\_center - y, 1);

        putpixel(x\_center + x, y\_center - y, 1);

        putpixel(x\_center - x, y\_center + y, 1);

        if (d >= 0)

            d = d - fy + a\_sqr;

        else{

            x = x + 1;

            d = d + fx - fy + a\_sqr;

            fx = fx + (2 \* b\_sqr);

        }

        y = y - 1;

        fy = fy - (2 \* a\_sqr);

    } while (y > 0);

    getch();

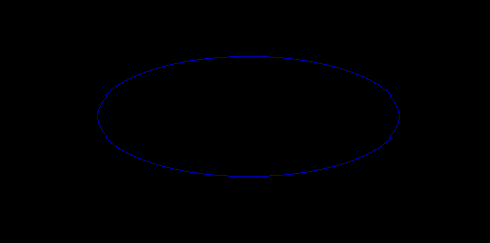
    closegraph();

}

**Result :**

**Graphical user interface, application

Description automatically generated with medium confidence**

****