

CG LAB 3

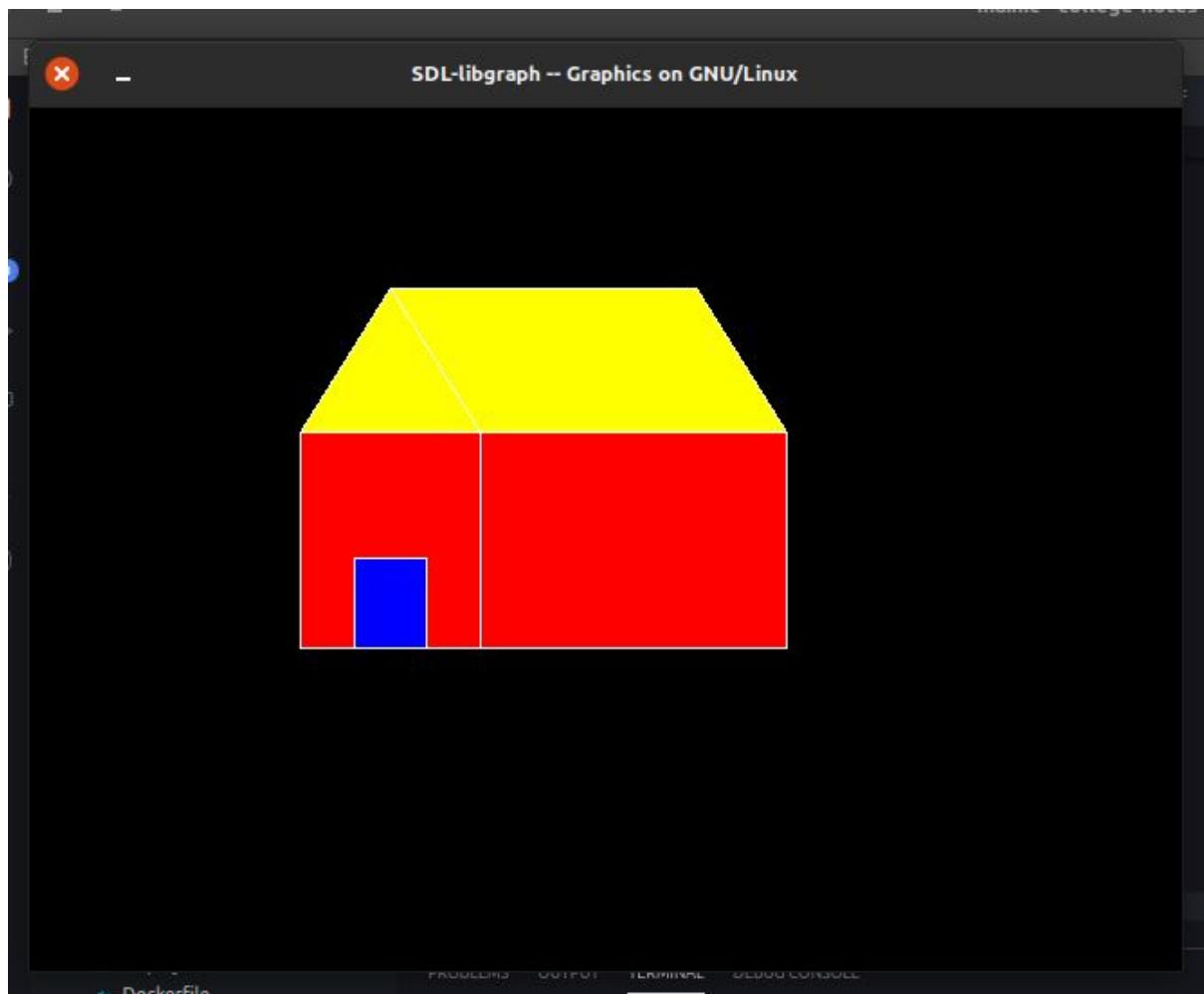
SAHIL BONDRE: U18CO021

1. Write a program to design a House and color it using predefined functions of graphics.h

```
#include <stdio.h>

#include "graphics.h"

int main() {
    int gd = DETECT, gm;
    initgraph(&gd, &gm, "");
    setcolor(WHITE);
    rectangle(150, 180, 250, 300);
    rectangle(250, 180, 420, 300);
    rectangle(180, 250, 220, 300);
    line(200, 100, 150, 180);
    line(200, 100, 250, 180);
    line(200, 100, 370, 100);
    line(370, 100, 420, 180);
    setcolor(RED);
    floodfill(152, 182, WHITE);
    floodfill(252, 182, WHITE);
    setcolor(BLUE);
    floodfill(182, 252, WHITE);
    setcolor(YELLOW);
    floodfill(200, 105, WHITE);
    floodfill(210, 105, WHITE);
    getch();
    closegraph();
    return 0;
}
```



2. Write a program to draw a Kite and color it using pre-defined functions of graphics.h

```
#include <graphics.h>

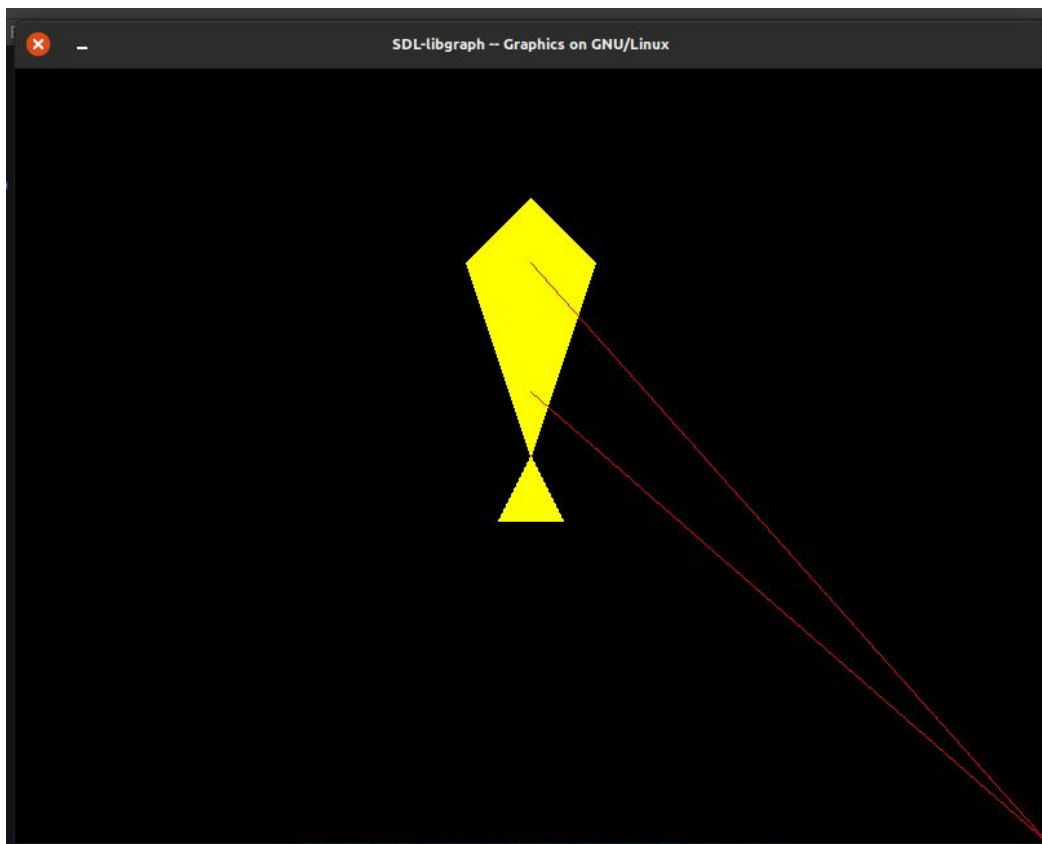
int main(int argc, char const *argv[]) {
    int gdriver = VGA;
    int gmode = VGAMAX;
    initgraph(&gdriver, &gmode, "");
    int xmid = getmaxx() / 2;
    int y1 = 100;
    int y2 = 150;
    int y3 = 300;
    int y4 = 350;
    int diff = 50;
    int roof[] = {
        xmid,
        y1,
        xmid + diff,
```

```

        y2,
        xmid,
        y3,
        xmid + diff / 2,
        y4,
        xmid - diff / 2,
        y4,
        xmid,
        y3,
        xmid - diff,
        y2,
        xmid,
        y1,
    };
    setcolor(YELLOW);
    fillpoly(8, roof);
    setcolor(RED);
    line(xmid, y2, getmaxx(), getmaxy());
    line(xmid, y3 - 50, getmaxx(), getmaxy());

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

```

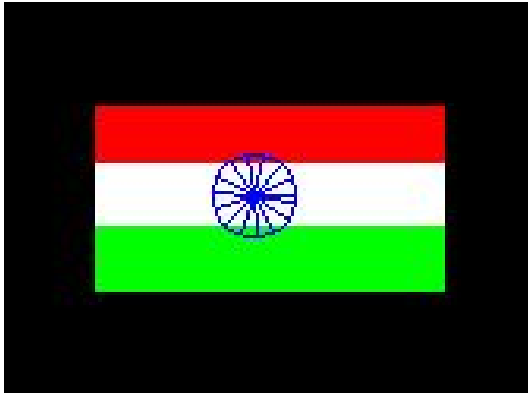


3. Write a program for drawing India's National Flag and Color it properly using pre-defined functions of graphics.h.

```
#include <math.h>
#include <stdio.h>

#include "graphics.h"
int main() {
    int gd = DETECT, gm, a, b, i, r, x, y;
    initgraph(&gd, &gm, "");
    // draw the top rectangle
    setcolor(RED);
    rectangle(110, 40, 220, 58);
    floodfill(111, 43, WHITE);
    setcolor(WHITE);
    rectangle(110, 58, 220, 78);
    floodfill(111, 59, WHITE);
    setcolor(GREEN);
    rectangle(110, 78, 220, 98);
    floodfill(111, 79, WHITE);

    a = 160;
    b = 68;
    r = 13;
    setcolor(BLUE);
    circle(a, b, r);
    for (i = 0; i <= 360; i = i + 25) {
        x = r * cos(i * 3.14 / 180);
        y = r * sin(i * 3.14 / 180);
        line(a, b, a + x, b - y);
    }
    getch();
    return 0;
}
```



4. Write a program for displaying a Moving Car Using predefined functions of graphics.h

```
#include <stdio.h>

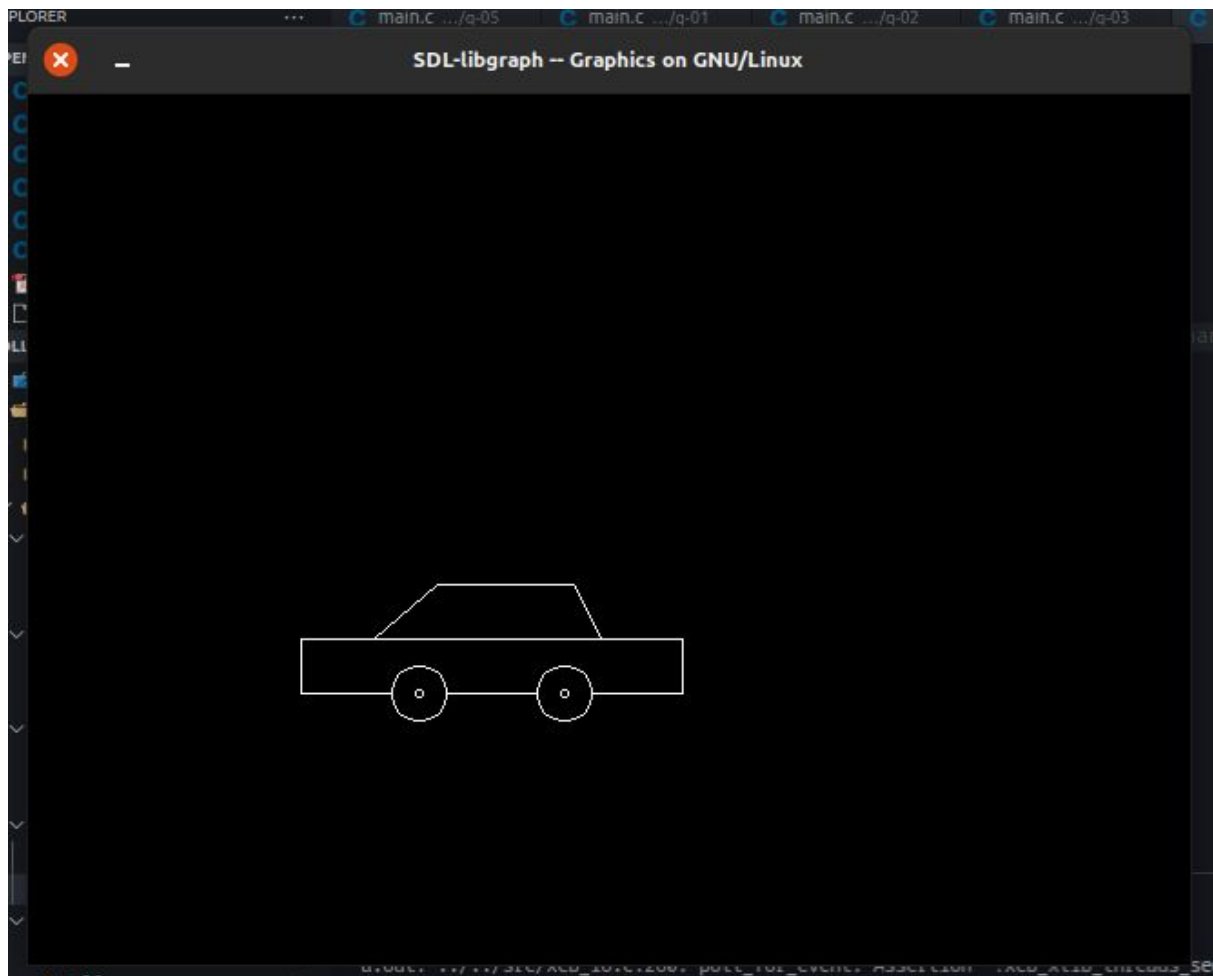
#include "graphics.h"
void draw_moving_car() {
    int i, j = 0, gd = DETECT, gm;
    initgraph(&gd, &gm, "");
    while (1)
        for (i = 0; i <= 420; i = i + 10) {
            setcolor(WHITE);
            line(0 + i, 300, 210 + i, 300);
            line(75 + i, 270, 150 + i, 270);
            line(75 + i, 270, 40 + i, 300);
            line(150 + i, 270, 165 + i, 300);
            line(0 + i, 300, 0 + i, 330);
            line(210 + i, 300, 210 + i, 330);
            circle(65 + i, 330, 15);
            circle(65 + i, 330, 2);
            circle(145 + i, 330, 15);
            circle(145 + i, 330, 2);
            line(0 + i, 330, 50 + i, 330);
            line(80 + i, 330, 130 + i, 330);
            line(210 + i, 330, 160 + i, 330);
            delay(100);
            setcolor(BLACK);
            line(0 + i, 300, 210 + i, 300);
            line(75 + i, 270, 40 + i, 300);
            line(50 + i, 300, 75 + i, 270);
            line(75 + i, 270, 150 + i, 270);
            line(150 + i, 270, 165 + i, 300);
            line(0 + i, 300, 0 + i, 330);
            line(210 + i, 300, 210 + i, 330);
            circle(65 + i, 330, 15);
```

```

        circle(65 + i, 330, 2);
        circle(145 + i, 330, 15);
        circle(145 + i, 330, 2);
        line(0 + i, 330, 50 + i, 330);
        line(80 + i, 330, 130 + i, 330);
        line(210 + i, 330, 160 + i, 330);
    }
    getch();
    closegraph();
}

int main() {
    draw_moving_car();
    return 0;
}

```



5. Write a menu driven program for following line drawing algorithms.

1. DDA Algorithm

2. Bresenham's Line Algorithm

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

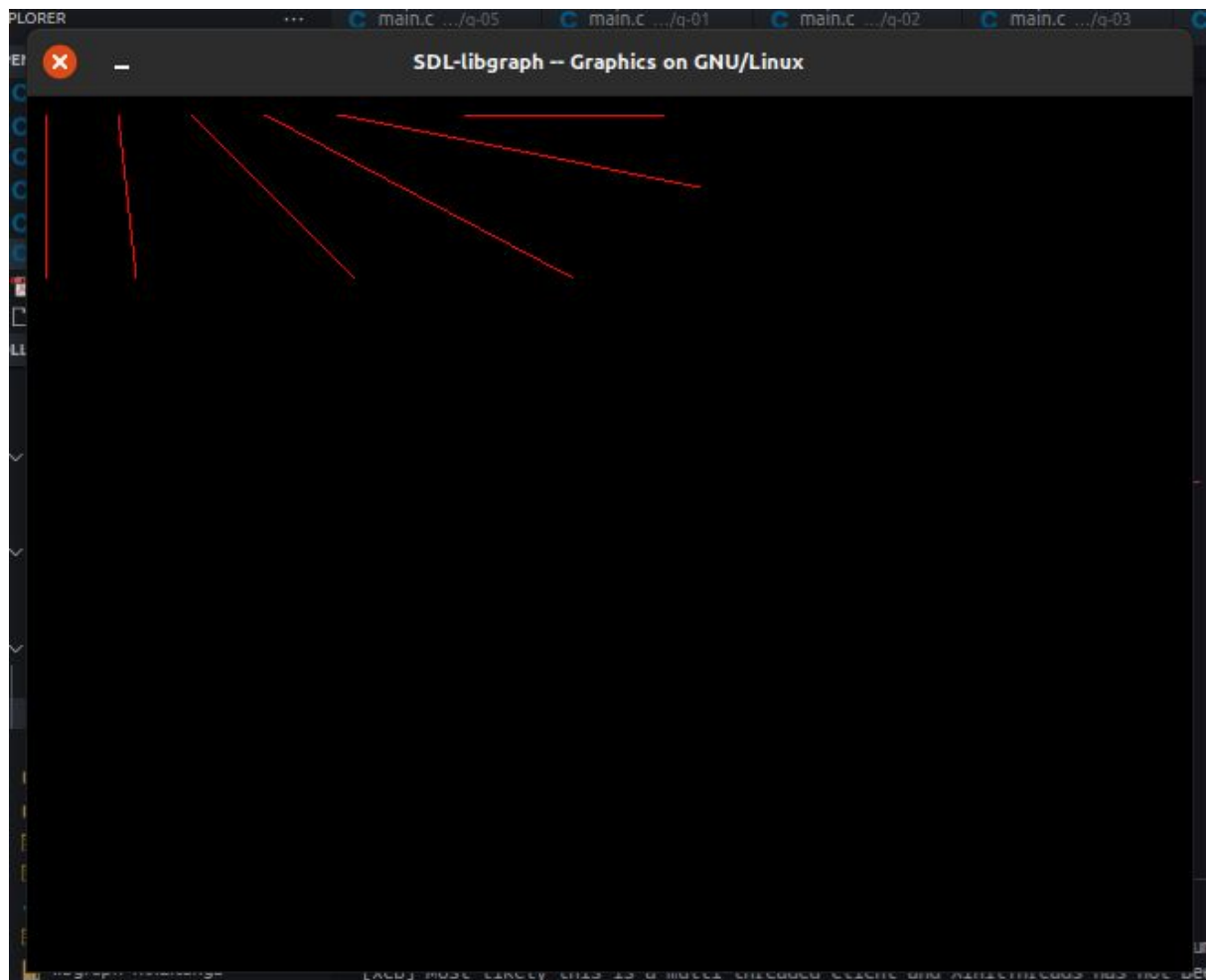
void dda(int x0, int y0, int x1, int y1) {

    double dx = (double)(x1 - x0);
    double dy = (double)(y1 - y0);
    double steps;
    if (abs(dx) >= abs(dy)) {
        steps = abs(dx);
    } else {
        steps = abs(dy);
    }
    double inc_x = dx / steps;
    double inc_y = dy / steps;
    double x = x0;
    double y = y0;
    int i = 1;
    while (i <= steps) {
        putpixel(x, y, RED);
        x += inc_x;
        y += inc_y;
        i = i + 1;
    }
}

int main(int argc, char const *argv[]) {
    int gd = DETECT, gm, i;
    int x0, x1, y0, y1;
    printf("Enter (x0, y0): ");
    scanf("%d %d", &x0, &y0);
    printf("Enter (x1, y1): ");
    scanf("%d %d", &x1, &y1);

    initgraph(&gd, &gm, "");
    dda(x0, y0, x1, y1);
    getch();
    closegraph();

    return 0;
}
```



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

void bresenham(int x0, int y0, int x1, int y1) {
    int dx, dy, p, x, y;
    dx = x1 - x0;
    dy = y1 - y0;
    x = x0;
    y = y0;
    p = 2 * dy - dx;
    while (x < x1) {
        if (p >= 0) {
            putpixel(x, y, RED);
            y = y + 1;
            p = p + 2 * dy - 2 * dx;
        } else {
```



```

        putpixel(x, y, RED);
        p = p + 2 * dy;
    }
    x = x + 1;
}
}

int main() {
    int gd = DETECT, gm, i;
    int x0, x1, y0, y1;
    printf("Enter (x0, y0): ");
    scanf("%d %d", &x0, &y0);
    printf("Enter (x1, y1): ");
    scanf("%d %d", &x1, &y1);
    initgraph(&gd, &gm, "");
    bresenhams(x0, y0, x1, y1);
    getch();
    closegraph();
    return 0;
}

```

