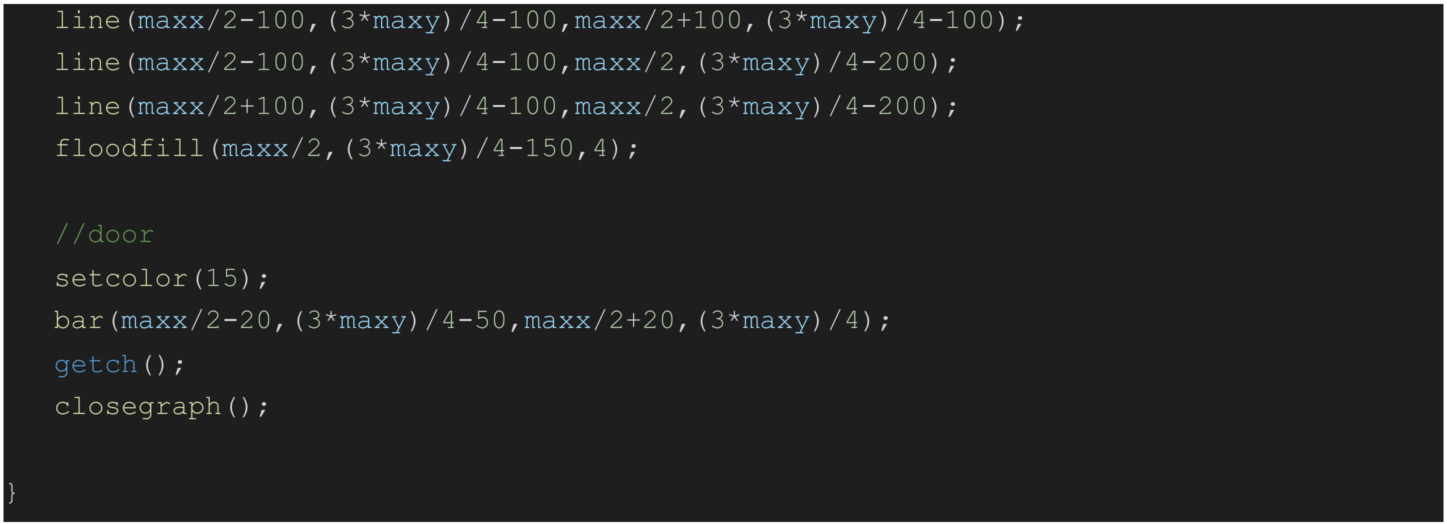
Computer Graphics Practicals

Assignment 3

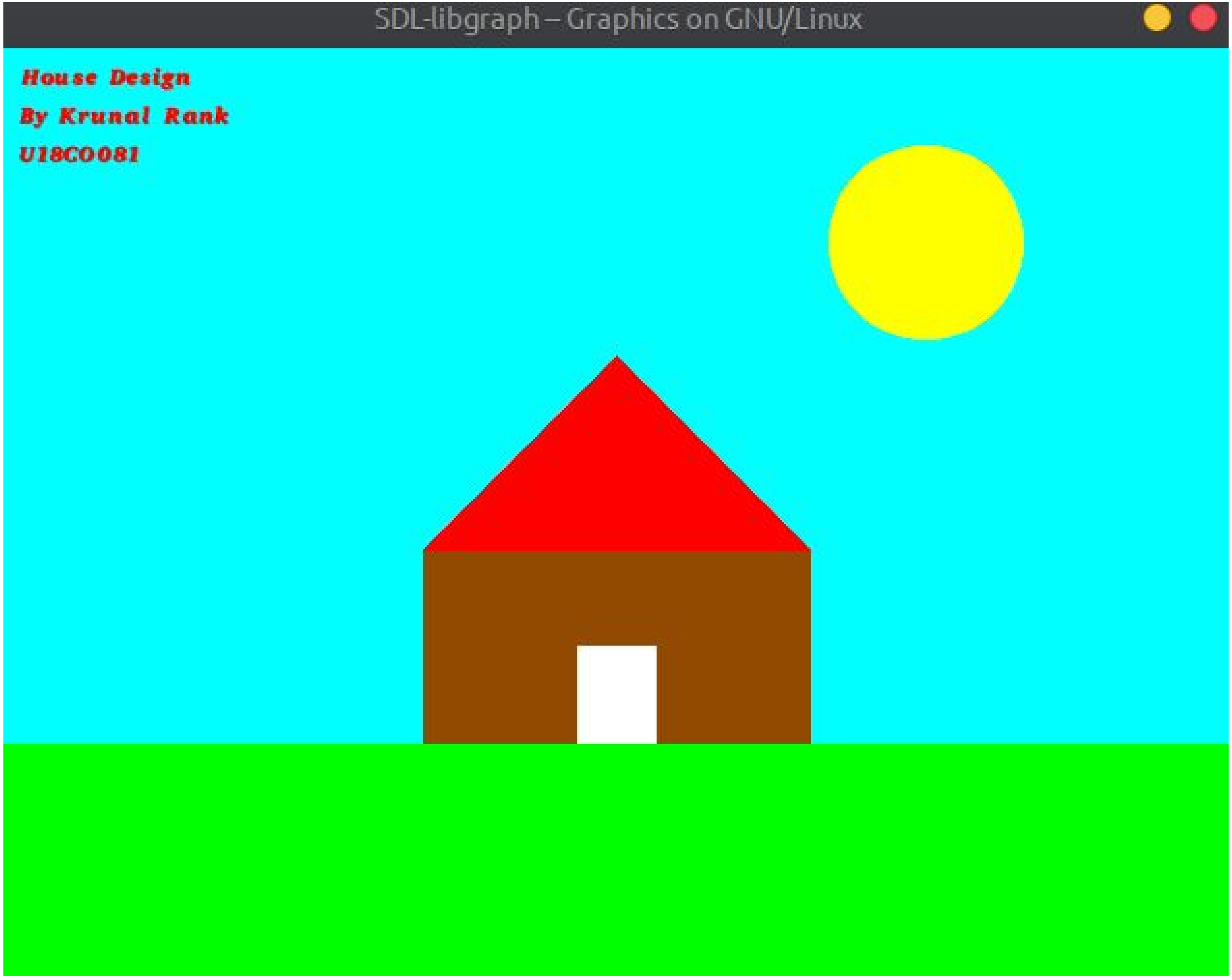
U18CO081
Krunal Rank

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

```
#include <graphics.h>
#include <bits/stdc++.h>

using namespace std;

int main(){
   int gdriver = DETECT;
   int gmode,errorcode;
   initgraph(&gdriver,&gmode,"");
   int maxx = getmaxx();
   int maxy = getmaxy();

   //grass
   setcolor(2);
   bar(0,(3*maxy)/4,maxx,maxy);

   //sky
   setcolor(3);
   bar(0,0,maxx,(3*maxy)/4);

   //tag
   setfontcolor(4);
   setcolor(1);
   outtextxy(10,10,"House Design");
   outtextxy(10,30,"By Krunal Rank");
   outtextxy(10,50,"U18CO081");


   //sun
   setcolor(14);
   circle((3*maxx)/4,100,50);
   floodfill((3*maxx)/4,100,14);

   //house
   setcolor(6);
   bar(maxx/2-100,(3*maxy)/4-100,maxx/2+100,(3*maxy)/4);

   //roof
   setcolor(4);
```



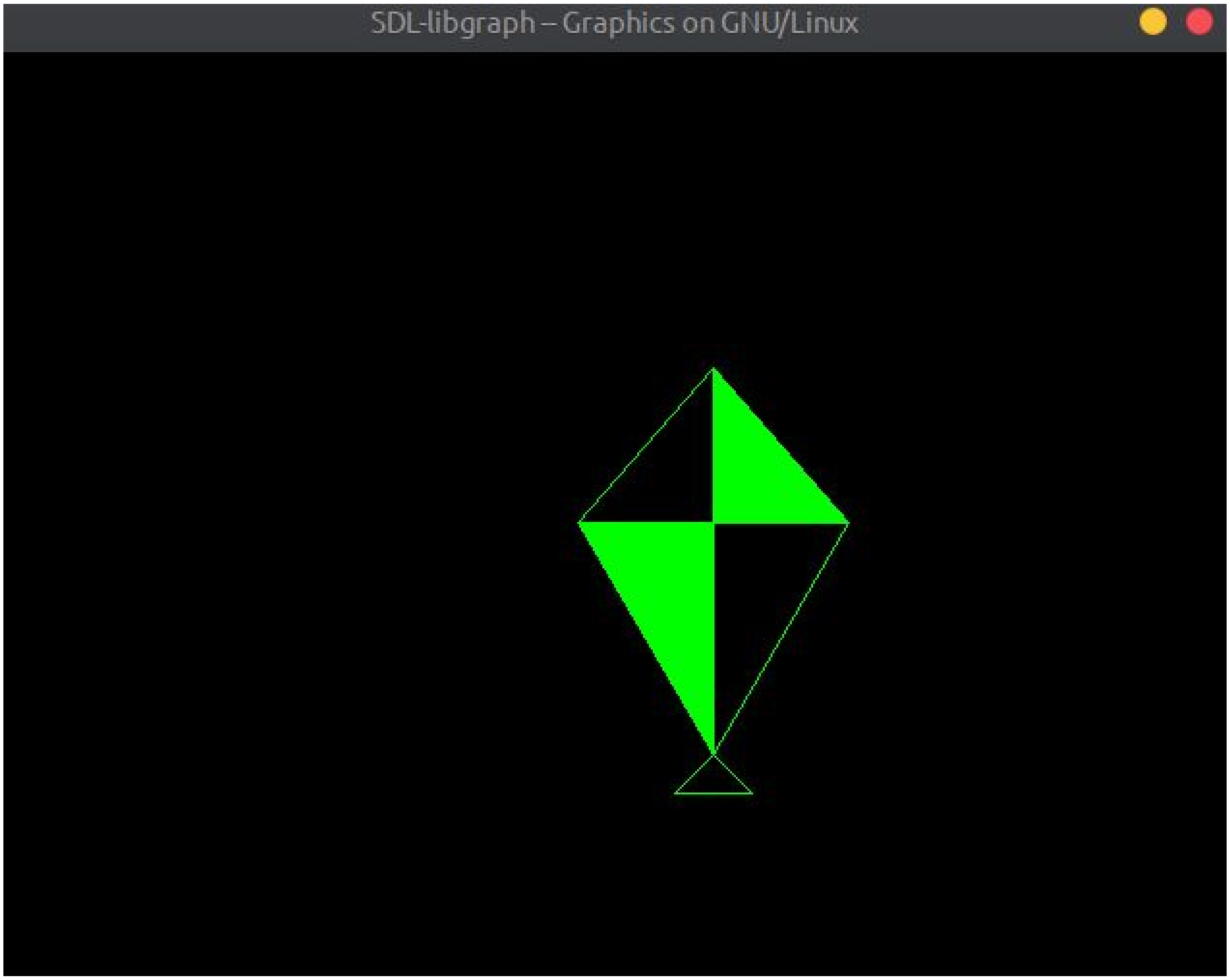
```
    line(maxx/2-100,(3*maxy)/4-100,maxx/2+100,(3*maxy)/4-100);
    line(maxx/2-100,(3*maxy)/4-100,maxx/2,(3*maxy)/4-200);
    line(maxx/2+100,(3*maxy)/4-100,maxx/2,(3*maxy)/4-200);
    floodfill(maxx/2,(3*maxy)/4-150,4);

    //door
    setcolor(15);
    bar(maxx/2-20,(3*maxy)/4-50,maxx/2+20,(3*maxy)/4);
    getch();
    closegraph();

}
```



```
House Design
By Krunal Rank
U18CO081
```

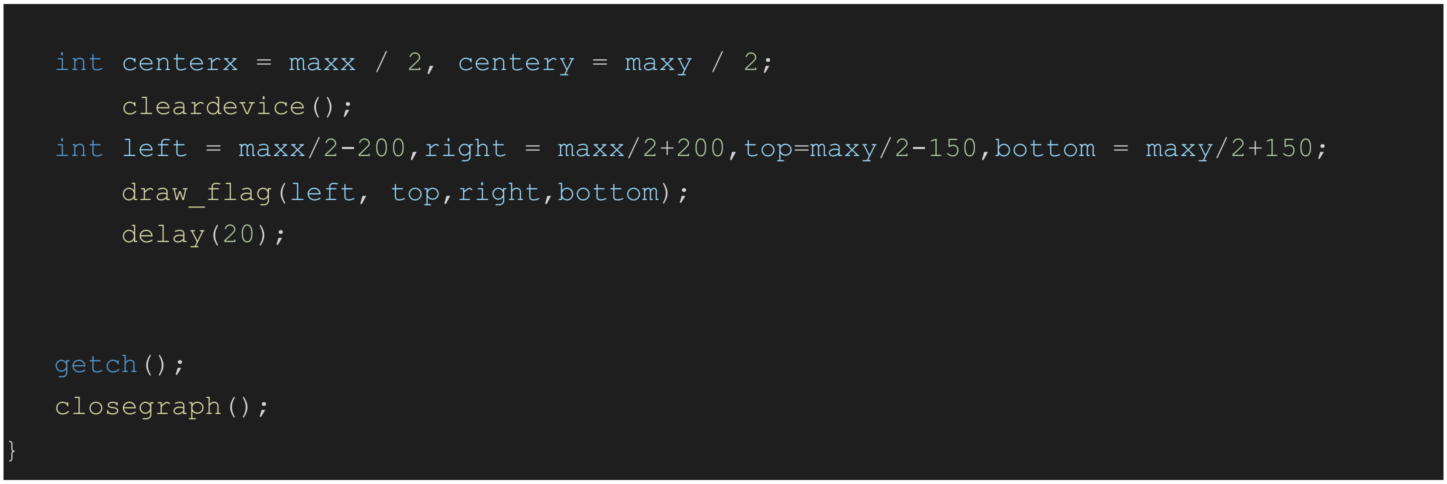
2. Write a program to draw a Kite and color it using predefined functions of graphics.h.

```
#include <graphics.h>
#include <bits/stdc++.h>
#include <signal.h>

using namespace std;

void draw_kite(int centerx, int centery)
{

    setcolor(2);
    line(centerx, centery + 100, centerx, centery - 100);
    line(centerx - 70, centery - 20, centerx + 70, centery - 20);
    line(centerx, centery + 100, centerx - 70, centery - 20);
    line(centerx, centery + 100, centerx + 70, centery - 20);
    line(centerx, centery - 100, centerx - 70, centery - 20);
    line(centerx, centery - 100, centerx + 70, centery - 20);
    line(centerx, centery + 100, centerx - 20, centery + 120);
    line(centerx - 20, centery + 120, centerx + 20, centery + 120);
    line(centerx, centery + 100, centerx + 20, centery + 120);
    floodfill(centerx + 50, centery - 21, 2);
    floodfill(centerx - 10, centery + 10, 2);
}

int main()
{
    int gd = DETECT, gm;
    int x = 10, y = 480;
    initgraph(&gd, &gm, "");
    int gdriver = DETECT;
    int gmode, errorcode;
    initgraph(&gdriver, &gmode, "");
    int maxx = getmaxx();
    int maxy = getmaxy();

    int centerx = maxx / 2, centery = maxy / 2;
    while (true)
    {
        cleardevice();


        int dirx = rand() % 3;
        int diry = rand() % 3;
        dirx = dirx == 0 ? -4 : (dirx == 1) ? 0 : 4;
        diry = diry == 0 ? -4 : (diry == 1) ? 0 : 4;
        centerx += dirx;
        centery += diry;
        draw_kite(centerx, centery);
        delay(20);
    }

    getch();
    closegraph();
}
```

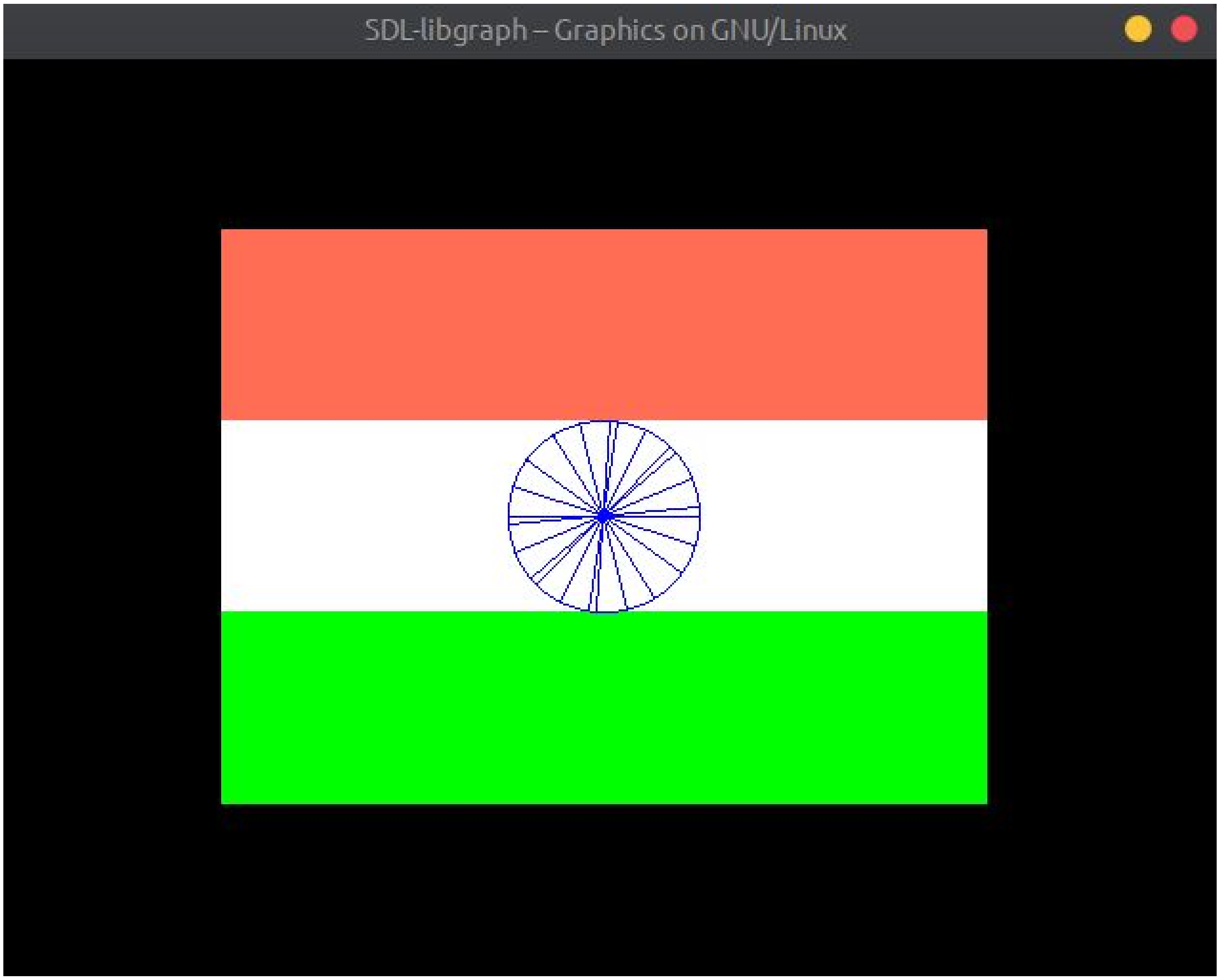


3. Write a program for drawing India's National Flag and Color it properly using predefined functions of graphics.h.

```
#include <graphics.h>
#include <bits/stdc++.h>
#include <signal.h>

using namespace std;

void draw_flag(int left, int top,int right,int bottom)
{
   assert(bottom>top);
   assert(right>left);
   int diffy = bottom - top;
   int del = diffy/3;
   setcolor(12);
   bar(left,top,right,top+del);
   setcolor(15);
   bar(left,top+del,right,top+2*del);
   setcolor(2);
   bar(left,top+2*del,right,bottom);

   int ccx = (left+right)/2,ccy = (top+bottom)/2,rad = del/2;
   setcolor(1);
   circle(ccx,ccy,rad);

   for(int i = 0;i<180;i+=15){
       double x1 = ccx + (double)rad*cos((double)i);
       double x2 = ccx - (double)rad*cos((double)i);
       double y1 = ccy + (double)rad*sin((double)i);
       double y2 = ccy - (double)rad*sin((double)i);
       line(x1,y1,x2,y2);
   }

}

int main()
{
   int gd = DETECT, gm;
   int x = 10, y = 480;
   initgraph(&gd, &gm, "");
   int gdriver = DETECT;
   int gmode, errorcode;
   initgraph(&gdriver, &gmode, "");
   int maxx = getmaxx();
   int maxy = getmaxy();
```

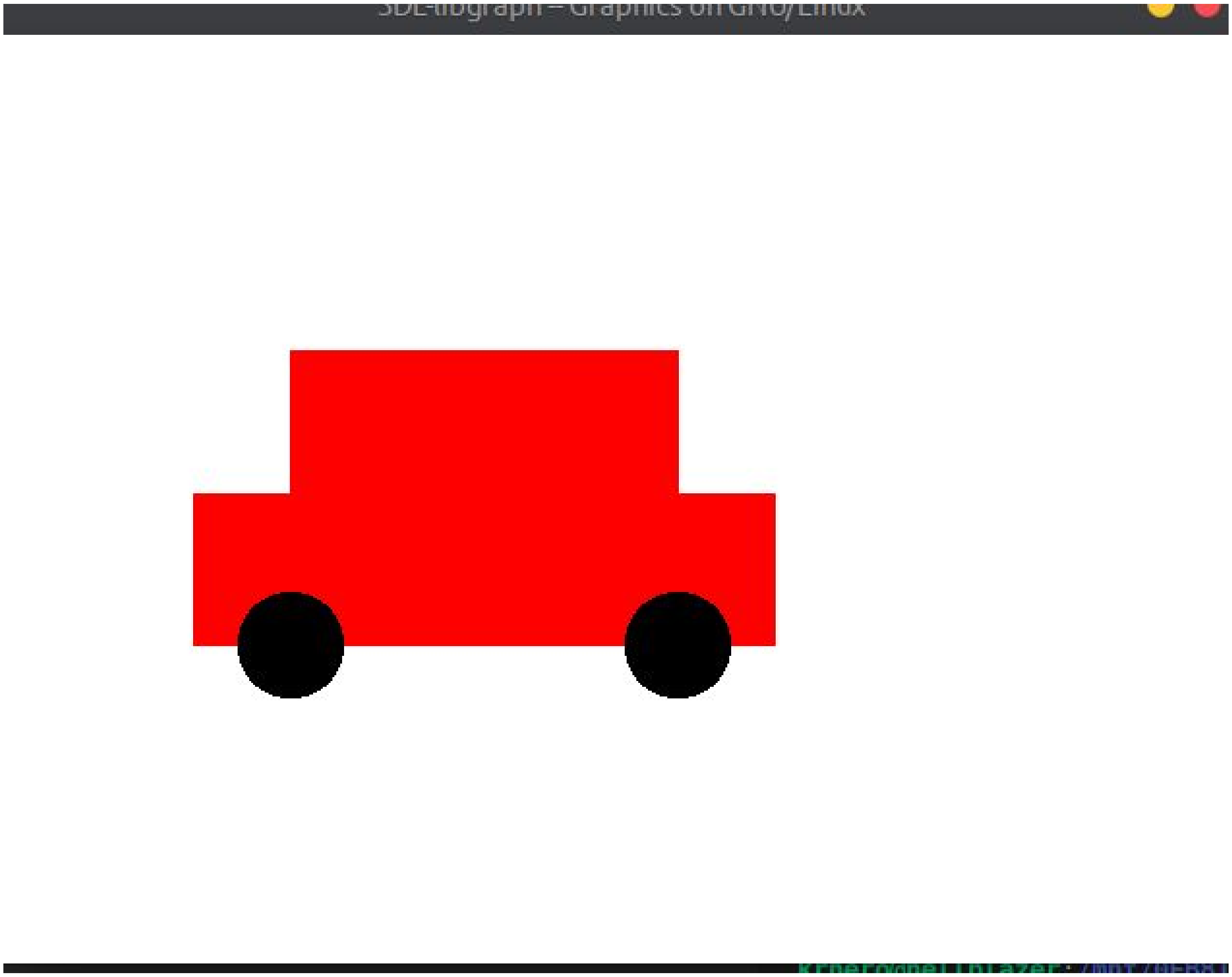


```
   int centerx = maxx / 2, centery = maxy / 2;
       cleardevice();
   int left = maxx/2-200,right = maxx/2+200,top=maxy/2-150,bottom = maxy/2+150;
       draw_flag(left, top,right,bottom);
       delay(20);



   getch();
   closegraph();
}
```



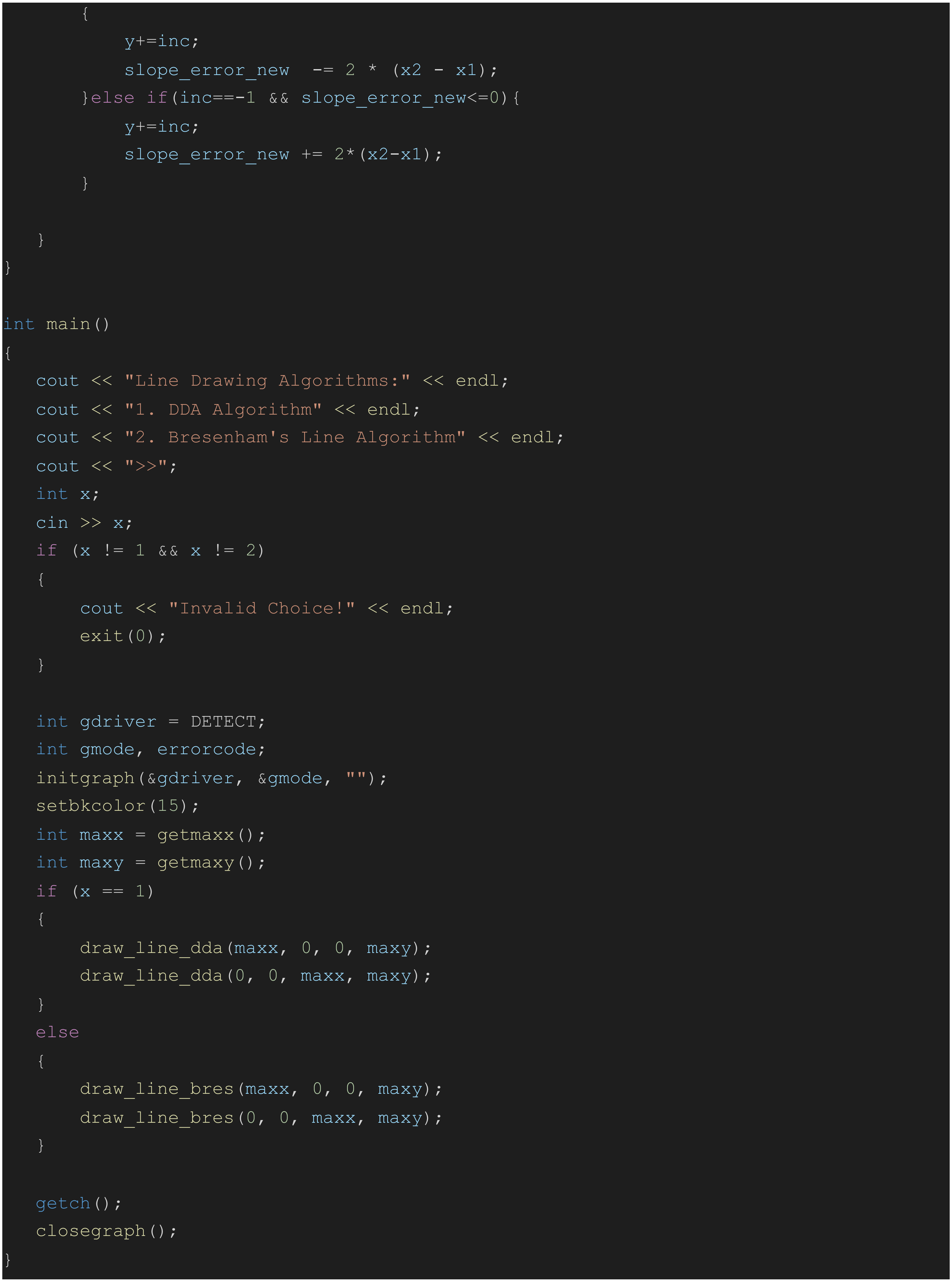
4. Write a program for displaying a Moving Car using predefined functions of graphics.h.

```
#include <graphics.h>
#include <bits/stdc++.h>
#include <signal.h>

using namespace std;

void draw_car(int left, int top,int right,int bottom)
{
    assert(bottom>top);
    assert(right>left);

    int unitx = (right-left)/8;
    int unity = (bottom-top)/8;
    setcolor(4);
    bar(left+unitx,top+4*unity,right-unitx,bottom-2*unity);
    bar(left+2*unitx,top+2*unity,right-2*unitx,top+4*unity);

    setcolor(0);
    circle(left+2*unitx,bottom-2*unity,3*(unity)/4);
    floodfill(left+2*unitx,bottom-2*unity,0);
    circle(right-2*unitx,bottom-2*unity,3*(unity)/4);
    floodfill(right-2*unitx,bottom-2*unity,0);




}

int main()
{
    int gd = DETECT, gm;
    int x = 10, y = 480;
    initgraph(&gd, &gm, "");
    int gdriver = DETECT;
    int gmode, errorcode;
    initgraph(&gdriver, &gmode, "");
    setbkcolor(15);
    int maxx = getmaxx();
    int maxy = getmaxy();

    int centerx = maxx / 2, centery = maxy / 2;
    int del = 10;
    int left = maxx/2-200,right = maxx/2+200,top=maxy/2-150,bottom = maxy/2+150;
    while(true){
        cleardevice();
        if(right+del>maxx) del = -10;
        else if(left+del<0) del = 10;
        left += del;
        right += del;
        draw_car(left, top,right,bottom);
        delay(30);
    }


    getch();
    closegraph();
}
```

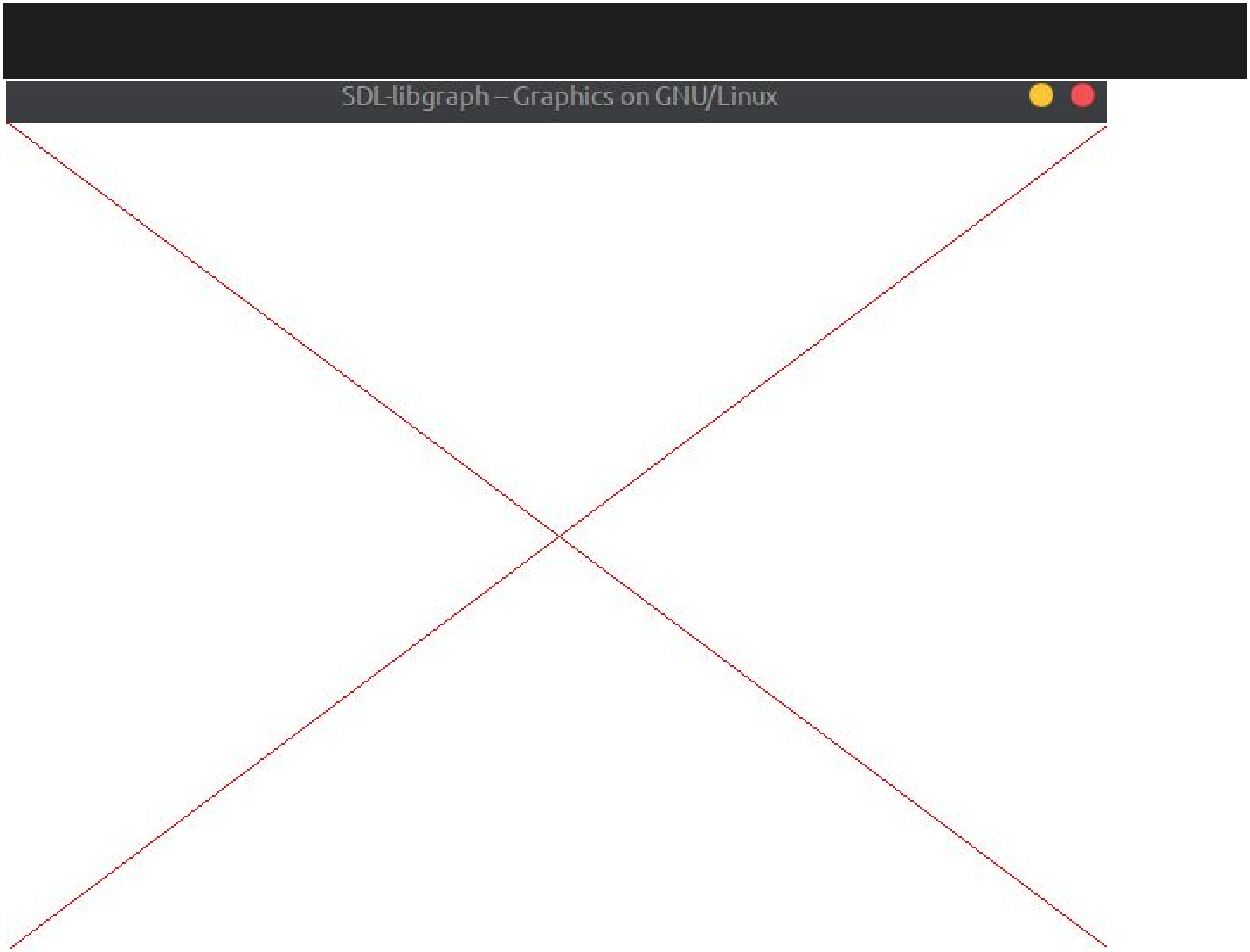


5. Write a menu driven program for following line drawing algorithms.
I. DDA Algorithm
II. Bresenham's Line Algorithm.

```
#include <graphics.h>
#include <bits/stdc++.h>
#include <signal.h>

using namespace std;

void draw_line_dda(int x1, int y1, int x2, int y2)
{
    double dx = x2 - x1, dy = y2 - y1;
    double step;
    if (abs(dx) >= abs(dy))
        step = abs(dx);
    else
        step = abs(dy);
    double xin = dx / step, yin = dy / step;
    double x = x1, y = y1;
    putpixel(x, y, 4);
    for (int i = 0; i < step; i++)
    {
        x += xin;
        y += yin;
        putpixel(x, y, 4);
    }
}

void draw_line_bres(int x1, int y1, int x2, int y2)
{
    if(x1>x2){
        swap(x1,x2);
        swap(y1,y2);
    }
    int inc = y1>y2?-1:1;
    int m_new = 2 * (y2 - y1);
    int slope_error_new = m_new + inc==1?-(x2 - x1):(x2-x1);
    for (int x = x1, y = y1; x <= x2; x++)
    {
        putpixel(x, y, 4);

        slope_error_new += m_new;

        if (inc==1 && slope_error_new >= 0)
```



```
        {
            y+=inc;
            slope_error_new  -= 2 * (x2 - x1);
        }else if(inc==-1 && slope_error_new<=0){
            y+=inc;
            slope_error_new += 2*(x2-x1);
        }

    }
}

int main()
{
    cout << "Line Drawing Algorithms:" << endl;
    cout << "1. DDA Algorithm" << endl;
    cout << "2. Bresenham's Line Algorithm" << endl;
    cout << ">>";
    int x;
    cin >> x;
    if (x != 1 && x != 2)
    {
        cout << "Invalid Choice!" << endl;
        exit(0);
    }

    int gdriver = DETECT;
    int gmode, errorcode;
    initgraph(&gdriver, &gmode, "");
    setbkcolor(15);
    int maxx = getmaxx();
    int maxy = getmaxy();
    if (x == 1)
    {
        draw_line_dda(maxx, 0, 0, maxy);
        draw_line_dda(0, 0, maxx, maxy);
    }
    else
    {
        draw_line_bres(maxx, 0, 0, maxy);
        draw_line_bres(0, 0, maxx, maxy);
    }

    getch();
    closegraph();
}
```


SDL-libgraph – Graphics on GNU/Linux