**DIGITAL ASSINGMENT-2**

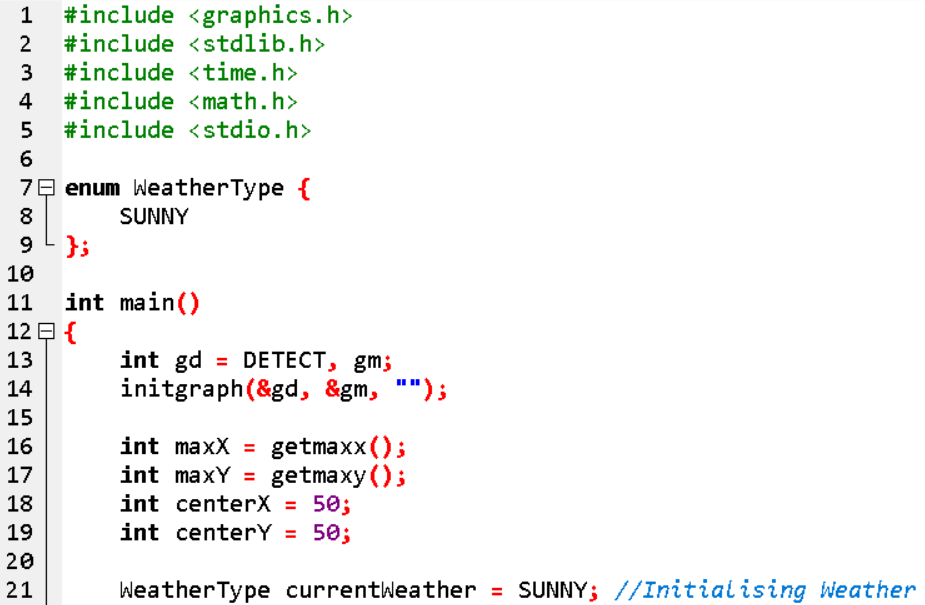
**Tridib Chatterjee(22BCE5142)**

**Anurag Chandra(22BCE5022)**

**Dynamic Car**

**Explanation: -**

We have used the software graphics.h to draw a dynamic moving car with sunny weather as Background.

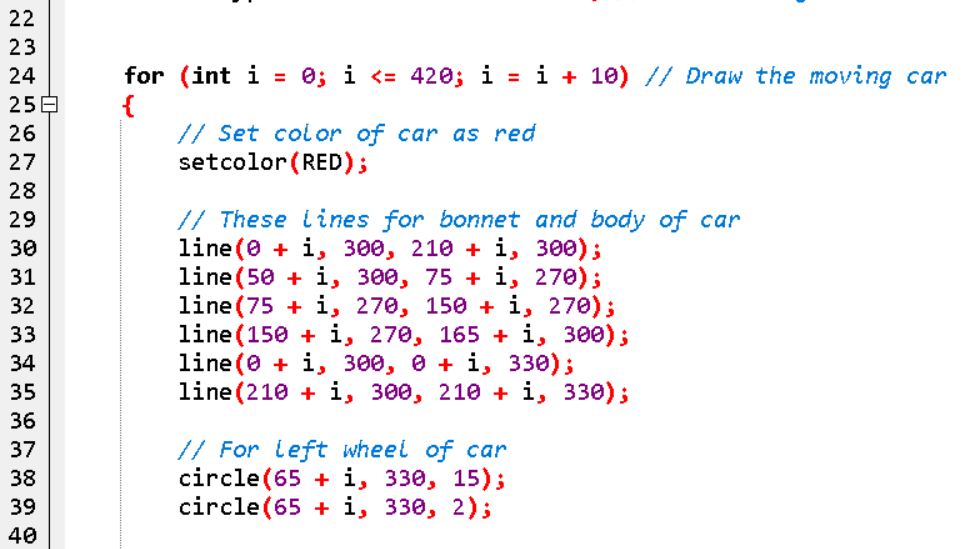


Lines 1-5

Declaration of libraries necessary for the execution of the program.

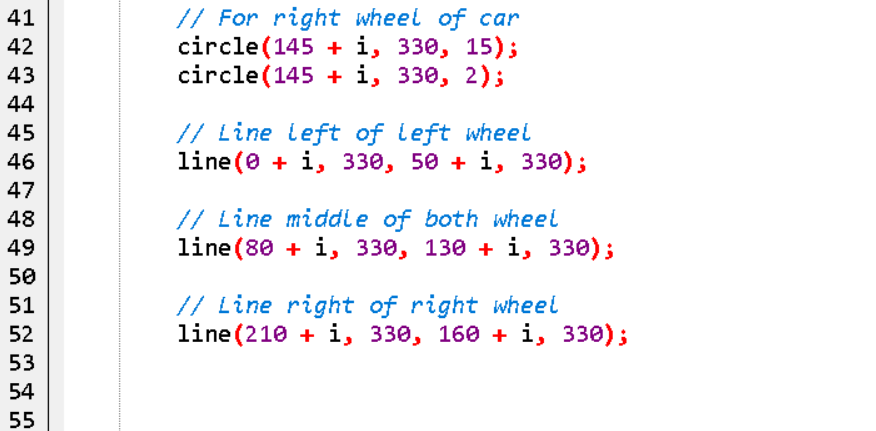
Line 7

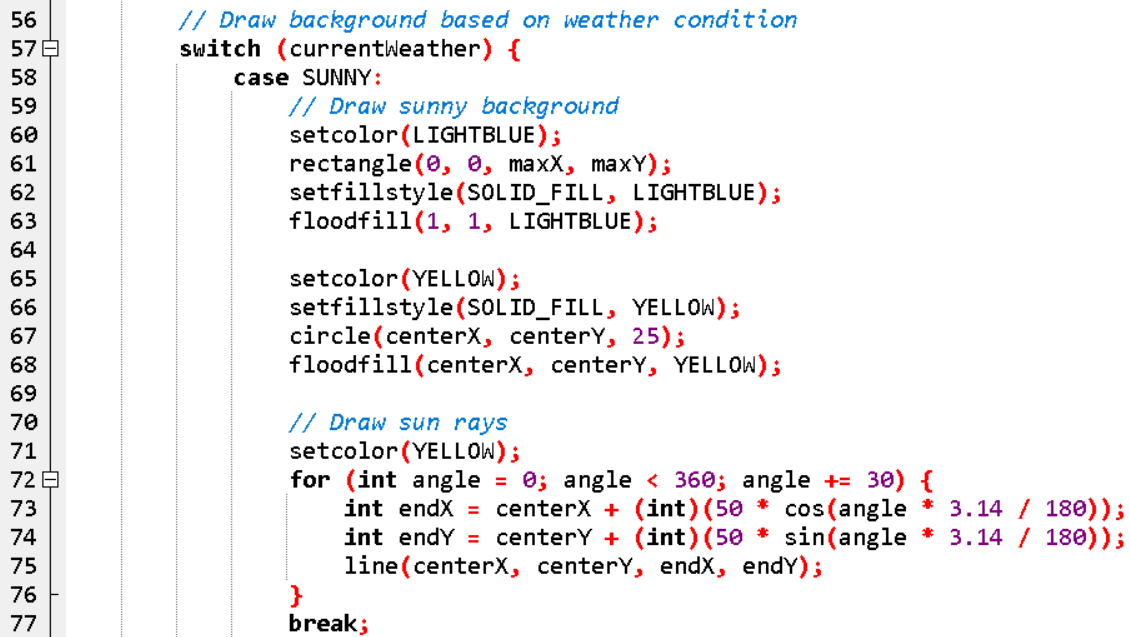
Declared a class type of Enum for the sunny background.



**Lines 24-52**

Drawing the body of the car



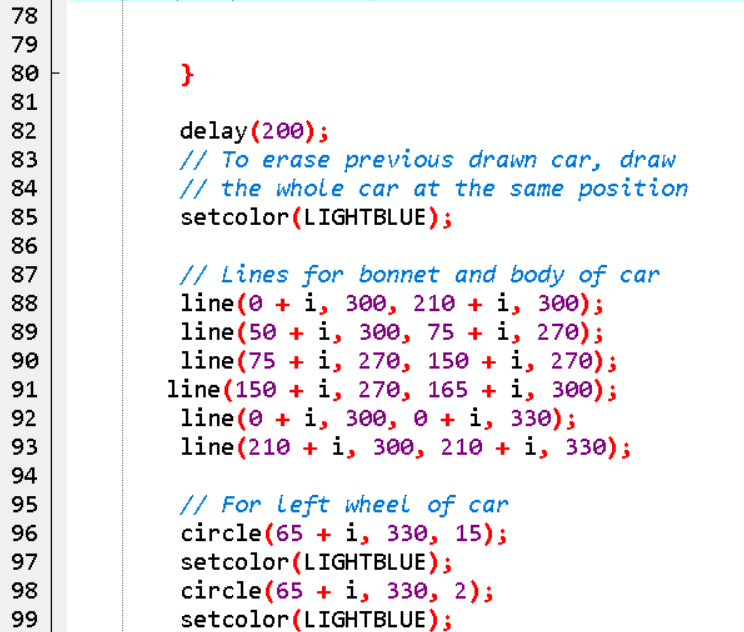


**Lines 56-68**

Draw the Sunny background

**Lines 70-77**

Draw the Sun-Rays

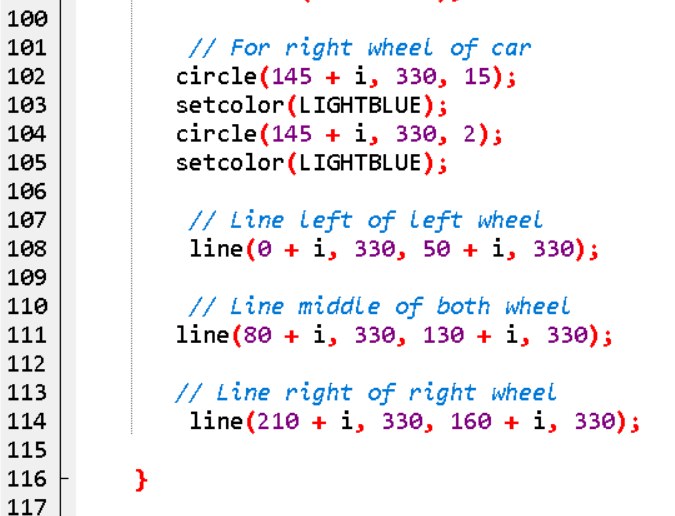


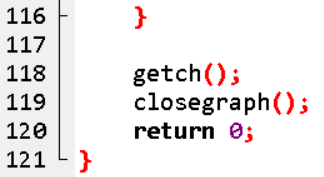
**Line 82**

To move the car forward

**Lines 85-114**

To erase the previously drawn car and draw the same car at the next position.





**THE ACTUAL CODE and THE OUTPUT**

#include <graphics.h>

#include <stdlib.h>

#include <time.h>

#include <math.h>

#include <stdio.h>

enum WeatherType {

SUNNY

};

int main()

{

int gd = DETECT, gm;

initgraph(&gd, &gm, "");

int maxX = getmaxx();

int maxY = getmaxy();

int centerX = 50;

int centerY = 50;

WeatherType currentWeather = SUNNY; //Initialising Weather

for (int i = 0; i <= 420; i = i + 10) // Draw the moving car

{

// Set color of car as red

setcolor(RED);

// These lines for bonnet and body of car

line(0 + i, 300, 210 + 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);

// For left wheel of car

circle(65 + i, 330, 15);

circle(65 + i, 330, 2);

// For right wheel of car

circle(145 + i, 330, 15);

circle(145 + i, 330, 2);

// Line left of left wheel

line(0 + i, 330, 50 + i, 330);

// Line middle of both wheel

line(80 + i, 330, 130 + i, 330);

// Line right of right wheel

line(210 + i, 330, 160 + i, 330);

// Draw background based on weather condition

switch (currentWeather) {

case SUNNY:

// Draw sunny background

setcolor(LIGHTBLUE);

rectangle(0, 0, maxX, maxY);

setfillstyle(SOLID\_FILL, LIGHTBLUE);

floodfill(1, 1, LIGHTBLUE);

setcolor(YELLOW);

setfillstyle(SOLID\_FILL, YELLOW);

circle(centerX, centerY, 25);

floodfill(centerX, centerY, YELLOW);

// Draw sun rays

setcolor(YELLOW);

for (int angle = 0; angle < 360; angle += 30) {

int endX = centerX + (int)(50 \* cos(angle \* 3.14 / 180));

int endY = centerY + (int)(50 \* sin(angle \* 3.14 / 180));

line(centerX, centerY, endX, endY);

}

break;

}

delay(200);

// To erase previous drawn car, draw

// the whole car at the same position

setcolor(LIGHTBLUE);

// Lines for bonnet and body of car

line(0 + i, 300, 210 + 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);

// For left wheel of car

circle(65 + i, 330, 15);

setcolor(LIGHTBLUE);

circle(65 + i, 330, 2);

setcolor(LIGHTBLUE);

// For right wheel of car

circle(145 + i, 330, 15);

setcolor(LIGHTBLUE);

circle(145 + i, 330, 2);

setcolor(LIGHTBLUE);

// Line left of left wheel

line(0 + i, 330, 50 + i, 330);

// Line middle of both wheel

line(80 + i, 330, 130 + i, 330);

// Line right of right wheel

line(210 + i, 330, 160 + i, 330);

}

getch();

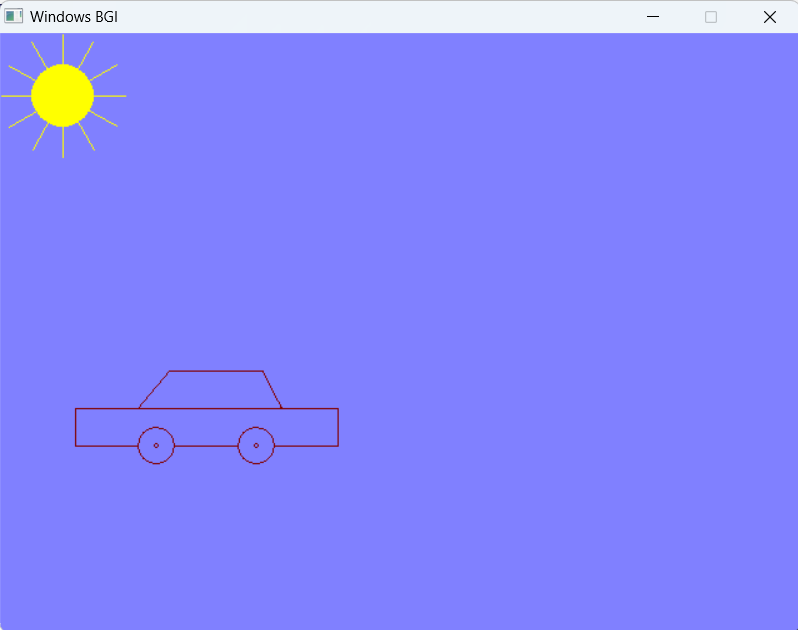
closegraph();

return 0;

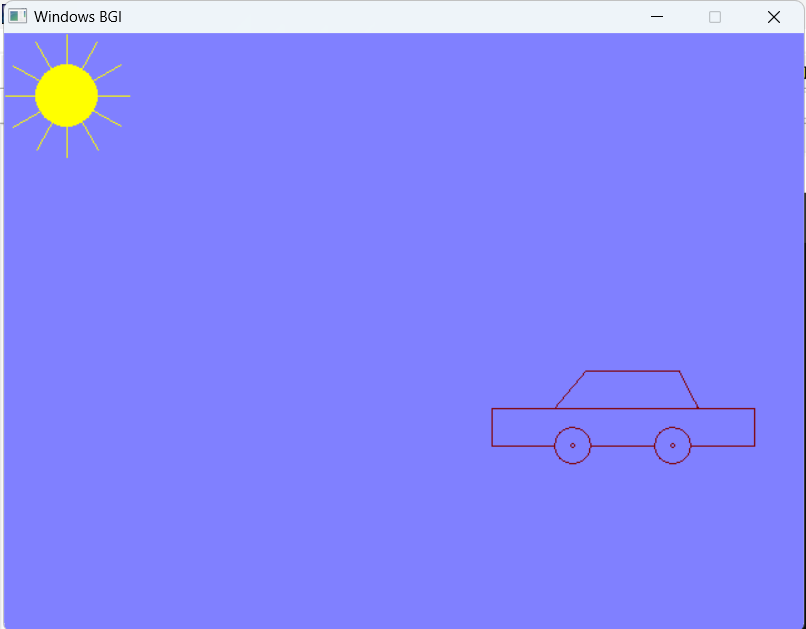
}

**OUTPUT**

**Initial moment(starting)**

****

**Final moment(ending)**

****