Shri Vaishnav Vidyapeeth Vishwavidyalaya, Indore

Department of Computer Science and Engineering

Lab File

Subject: - Computer Graphics And Multimedia

Year: - 3rd

Semester: - V

Subject Code: - BTCS503

Submitted by: - Idrak Sheikh

Submitted to: - Ms Manorama Chauhan

	Index Computer Graphics & Multimedia (Practical)		Code BTCS-503
S No.	**Date**	**Name of Experiment**	**Sign**
1	20-09-2021	Explain any 10 graphics functions according to their initialization and declaration.	
2	27-9-2021	Implement an image using putpixel function.	
3	9-10-2021	Implement an image using graphics function.	
4	9-10-2021	Rasterize a line using DDA algorithm.	

Experiment – 1

- **Explain any 10 graphics functions according to their initialization and declaration.**

1. **Putpixel**
2. **Closegraph**
3. **Initgraph**
4. **Setcolor**
5. **Getcolor**

The Graphics.h is a library contain functions for drawing lines, circle and many other functions.

There are so many functions in graphics.h library. These are 10 graphics functions: -

1. Putpixel:-

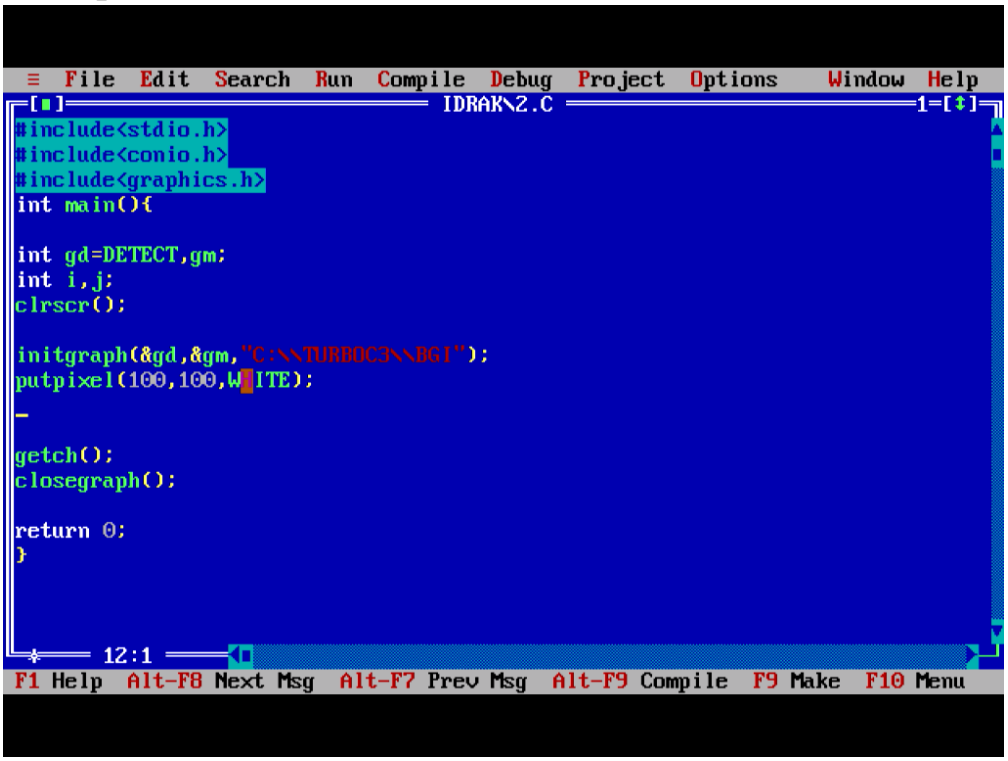
Putpixel () plots a pixel at a specified points.

- **Syntax: -**

  putpixel (int x, int y, int color);

It plots a point in the color defined at color in (x, y).it does not return.

- **Example: -**



- **Output: -**

2. Closegraph: -

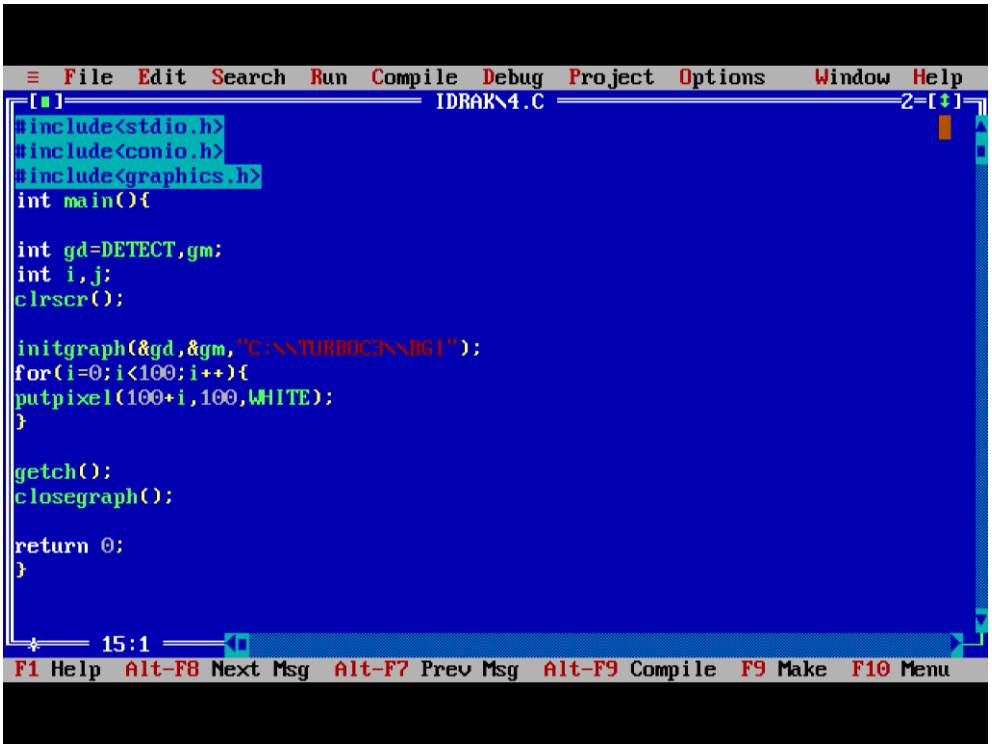
It shut downs the graphics system.

- **Syntax: -**

  closegraph ();

  It deallocates all the memory allocated by the graphics system.

- **Example: -**



Output: -

3. Intigraph: -

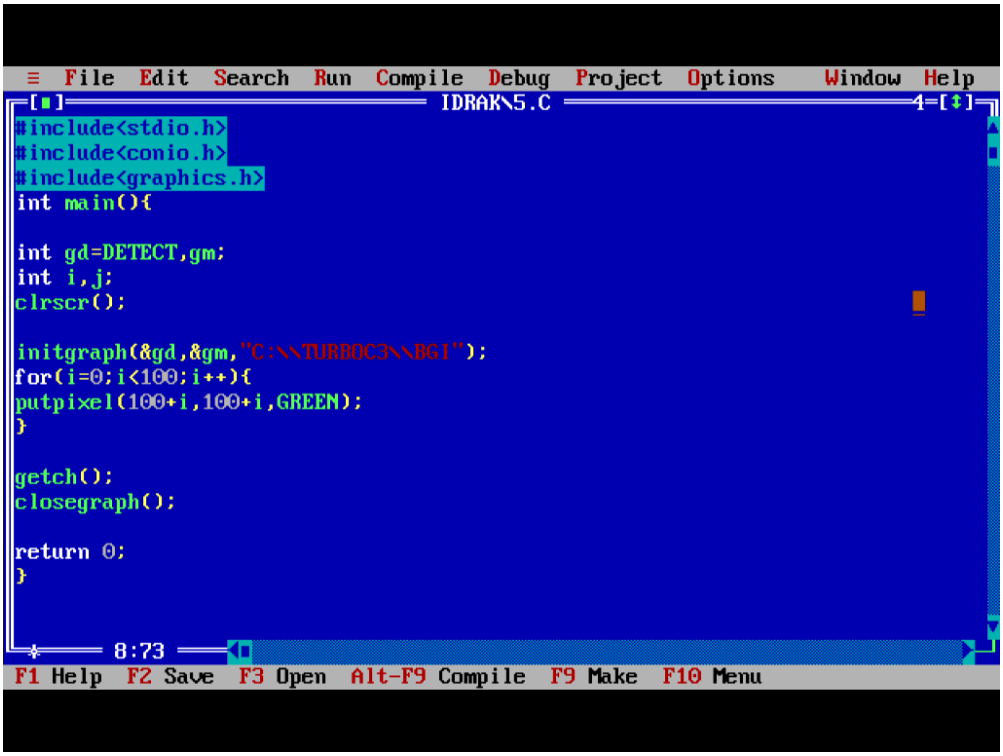
Initialize the graphics system.

- **Syntax: -**

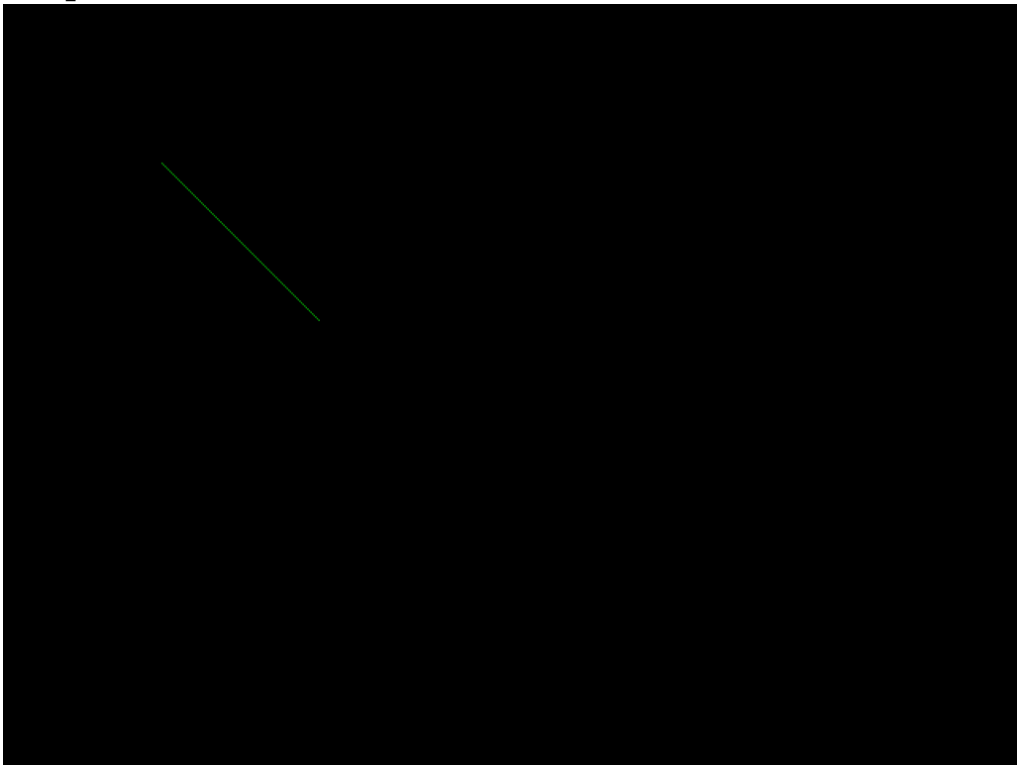
  initgraph (int *graphdriver, int *graphmode, char *pathtodriver);

  To start the graphics system, you must first call the initgraph().

- **Example: -**



- **Output: -**

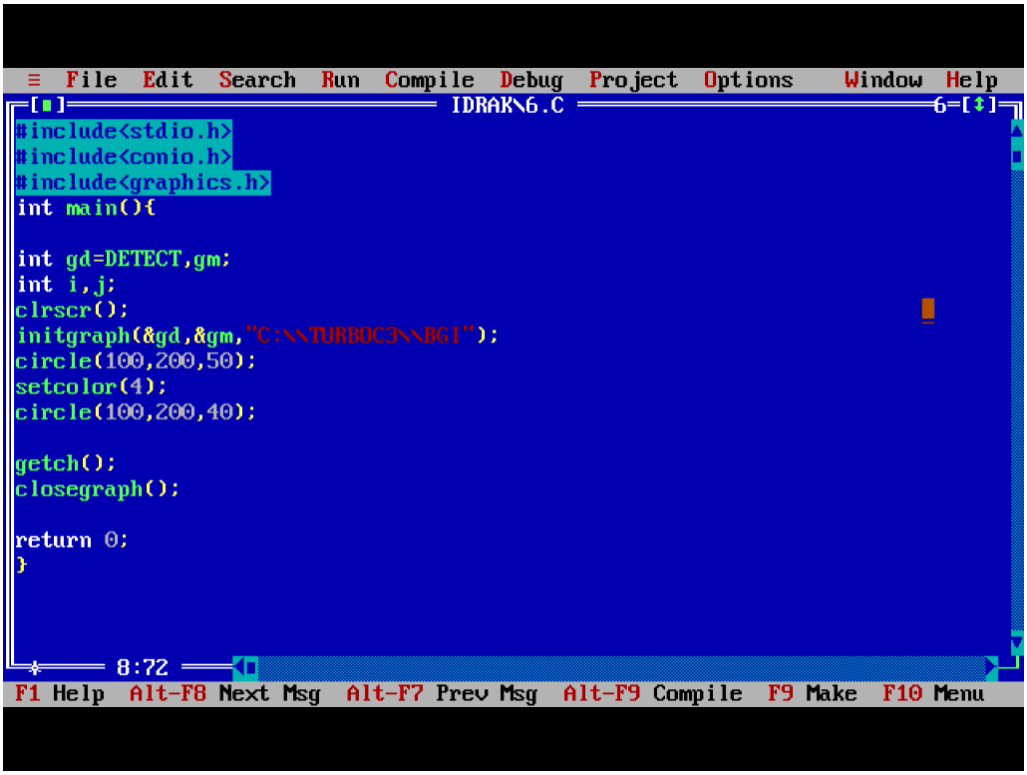


4. Setcolor: -

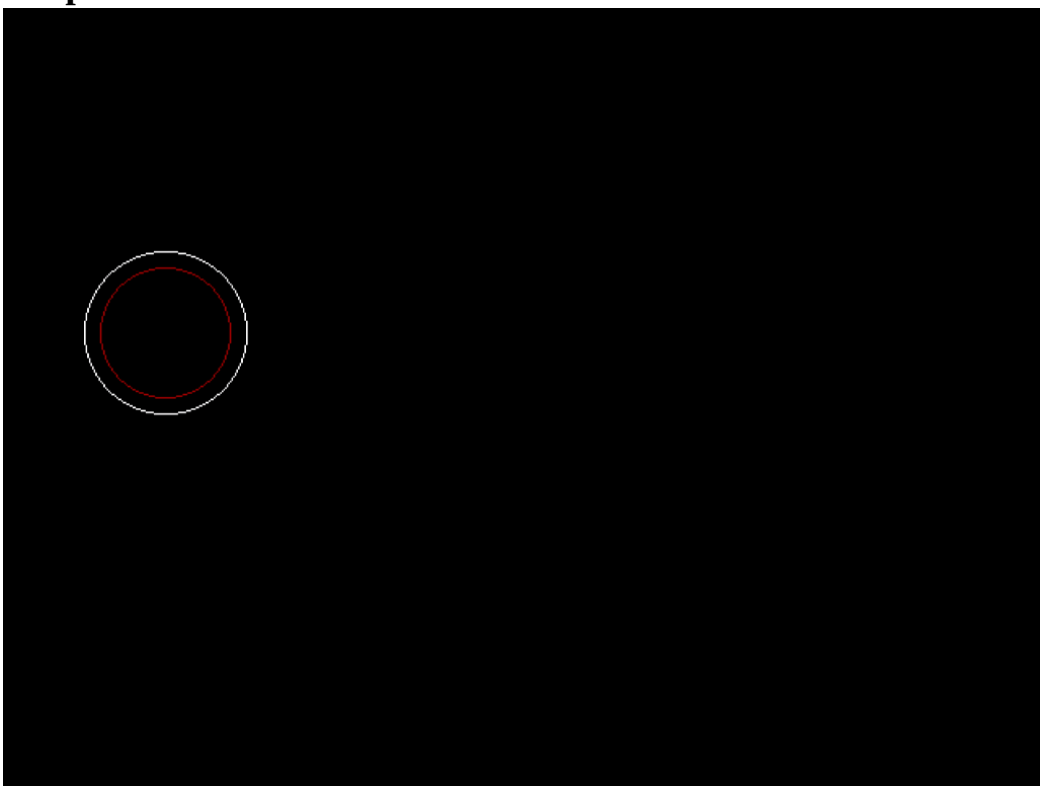
Setcolor sets the current drawing color.

- **Syntax: -**
  setcolor(int color);

  setcolor sets the current drawing color to color, which can range from 0 to getmaxcolor.
- **Example: -**



- **Output: -**



5. Getcolor: -

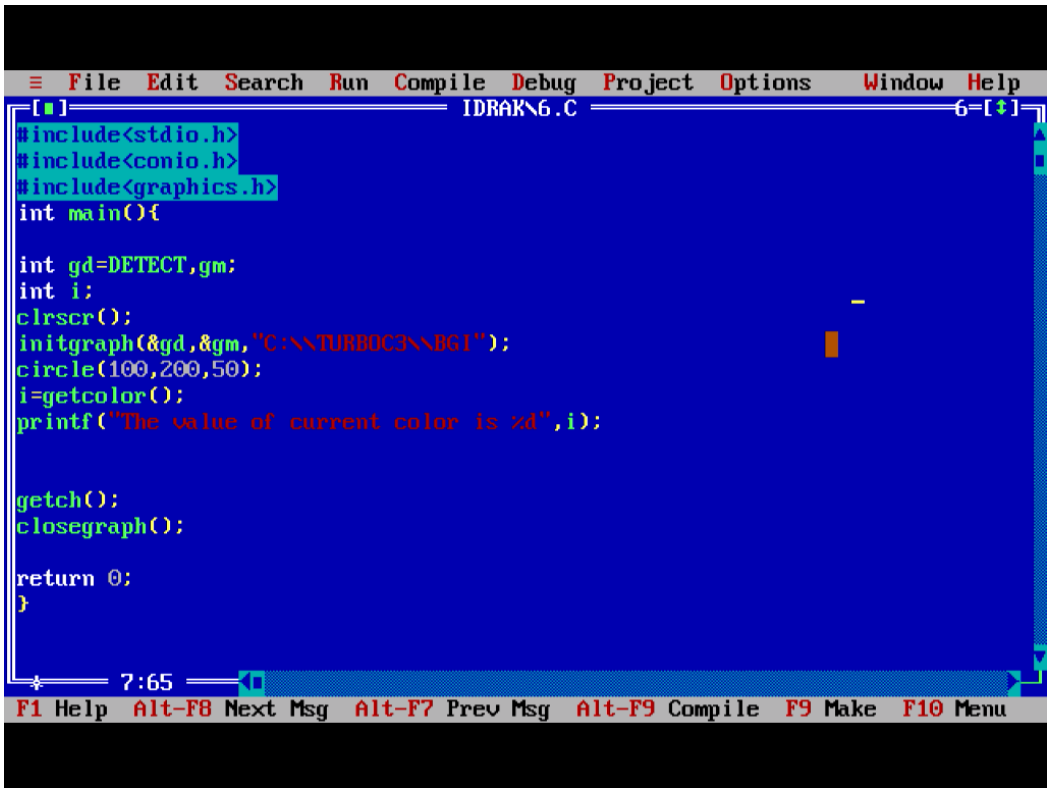
Getcolor returns the current drawing color.

- **Syntax: -**

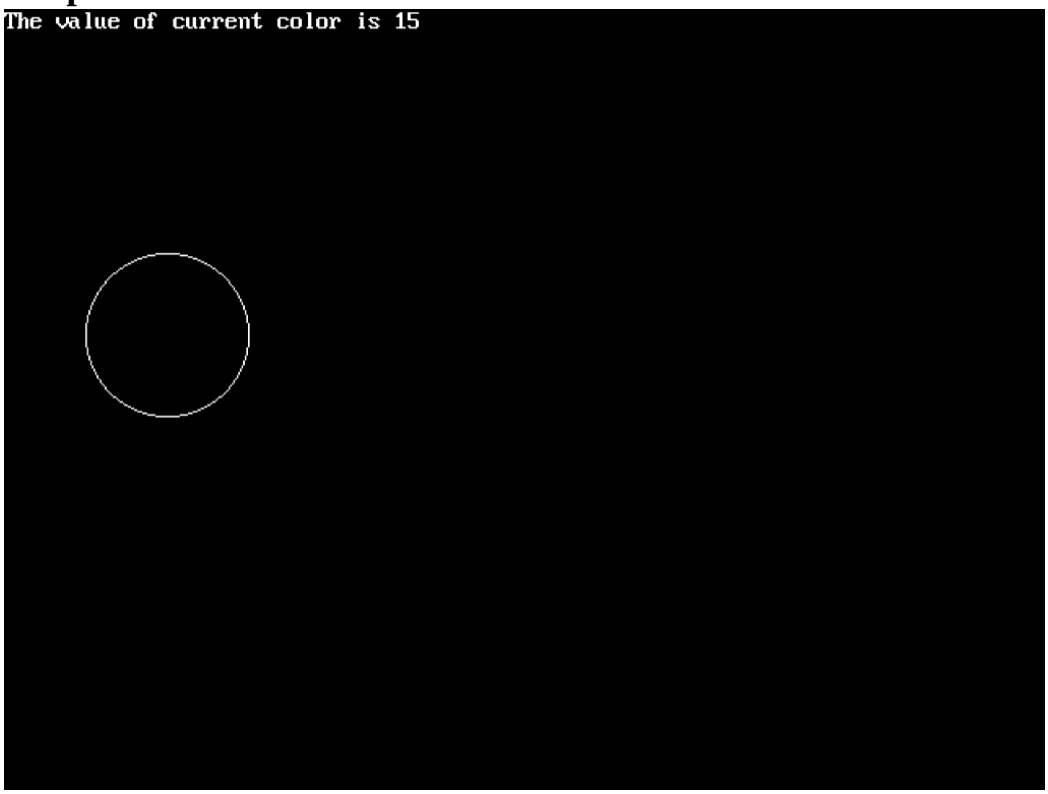
  getcolor();

  it returns the current drawing color.

- **Example: -**



- **Output: -**



6. Arc: -

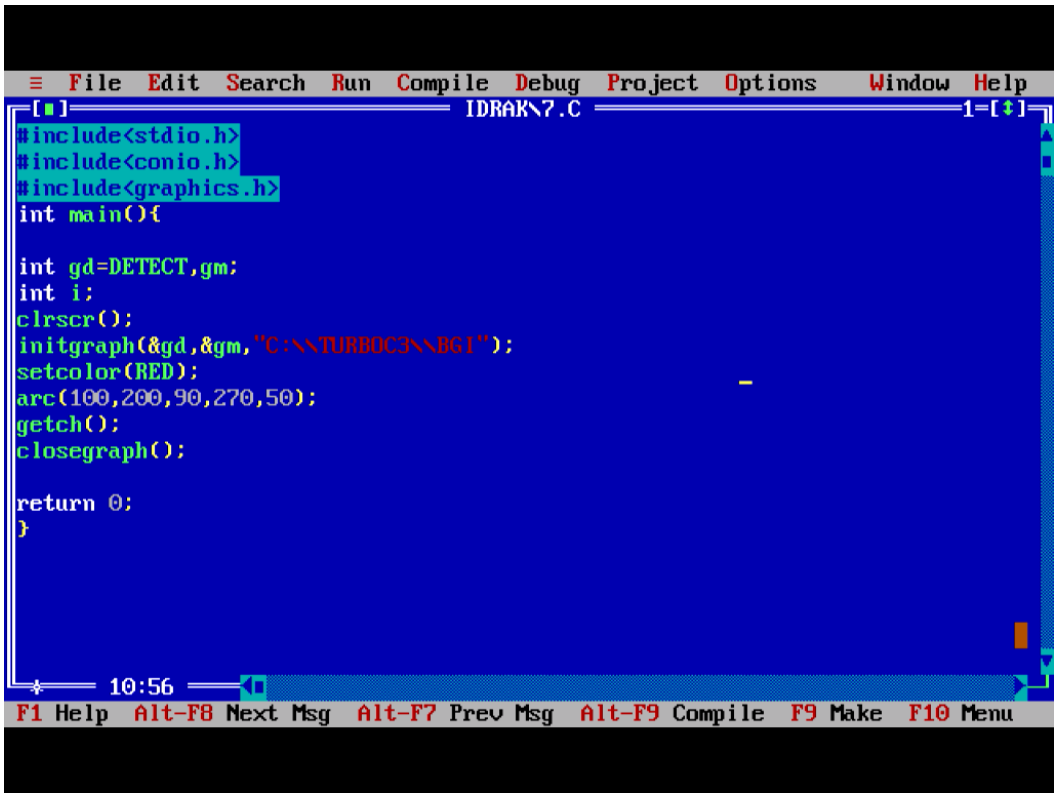
Arc draw a circular arc.

- **Syntax: -**

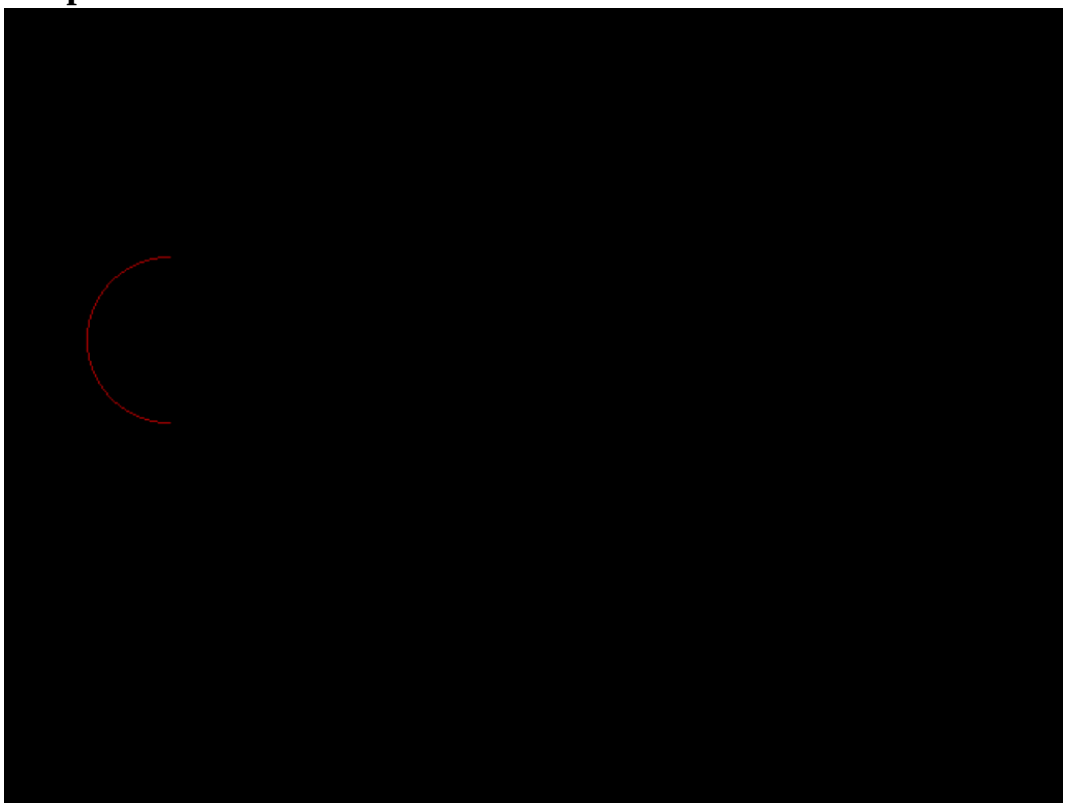
  arc(int x, int y, int startangle, int endangle, int radius);

  arc draws a circular arc in the current drawing color.

- **Example: -**



- **Output: -**



7. Circle: -

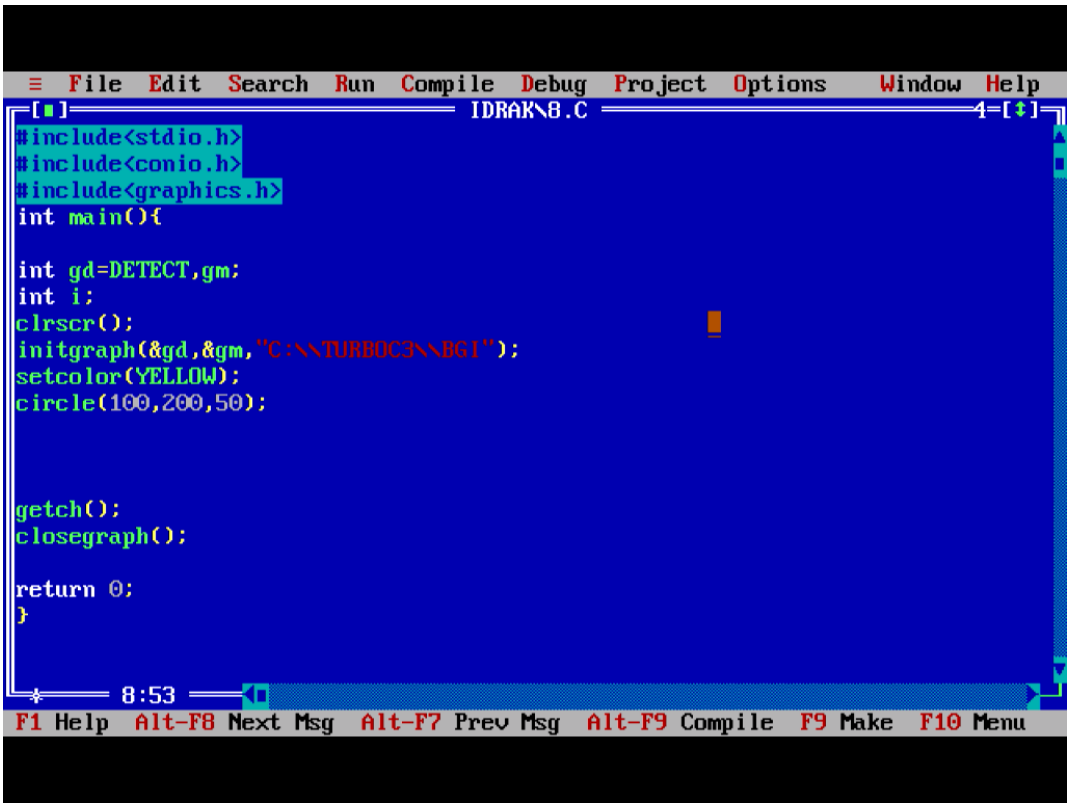
Circle draws a circle.

- **Syntax: -**

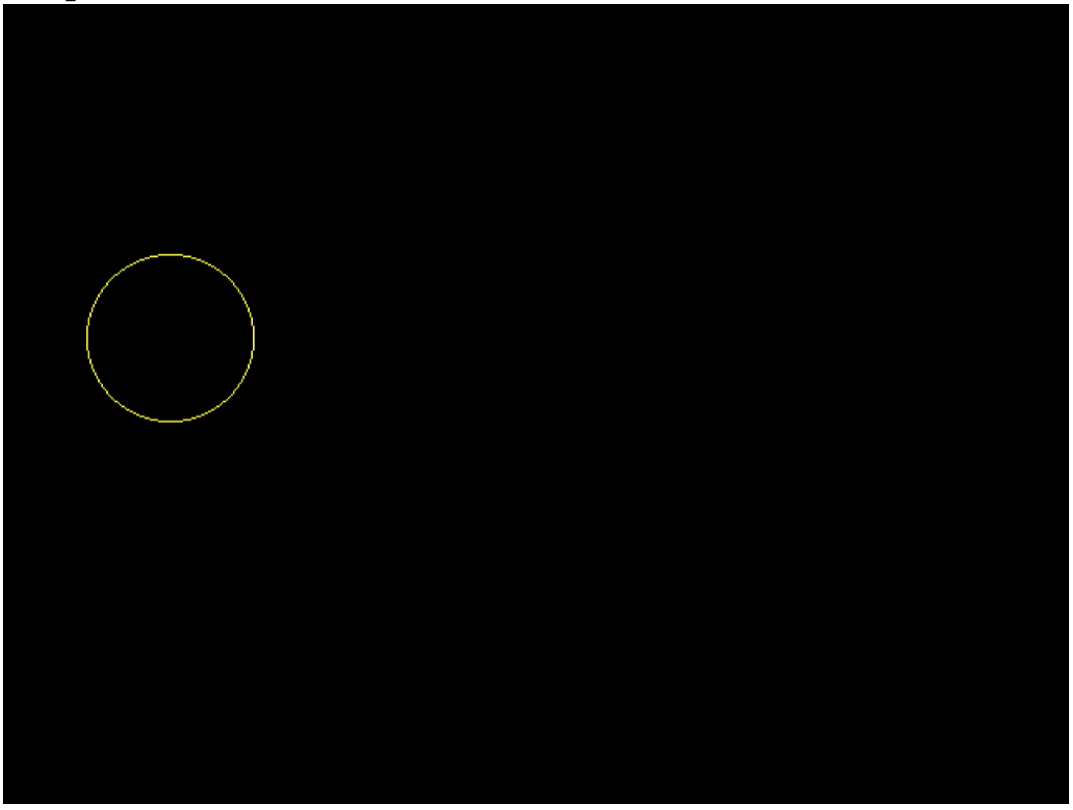
  circle(int x, int y, int radius);

  circle draws a circle in the current drawing color.

- **Example: -**



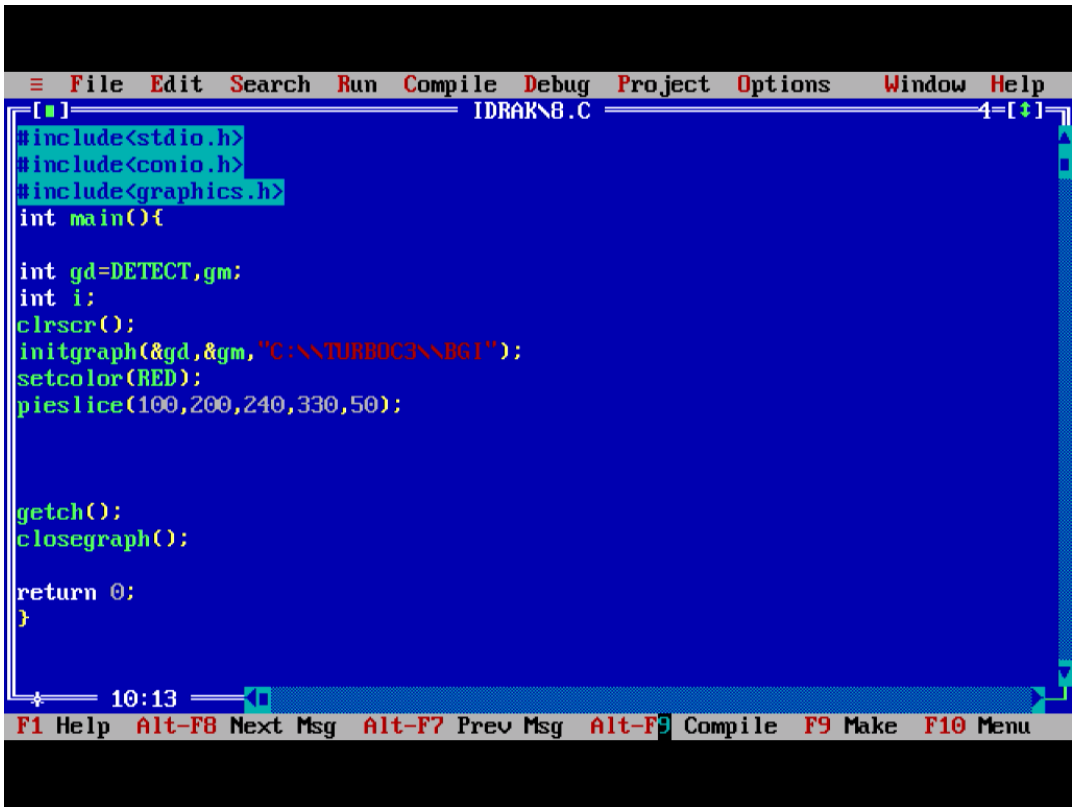
- **Output: -**



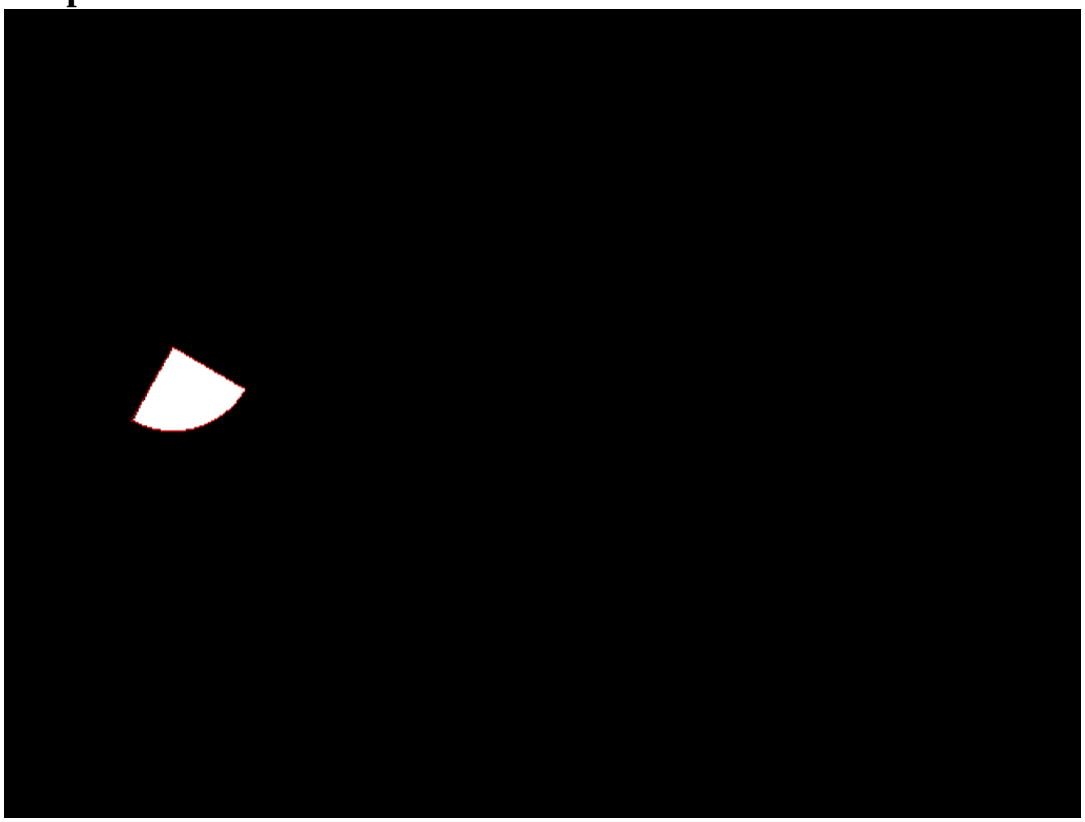
8. Pieslice: -

Pieslice draws and fill a circular pie slice.

- **Syntax: -**
  pieslice(int x, int y, int startangle, int endangle, int radius);
  pieslice draws a pie slice in the current drawing color, then fills it using the current fill pattern and fill color.
- **Example: -**



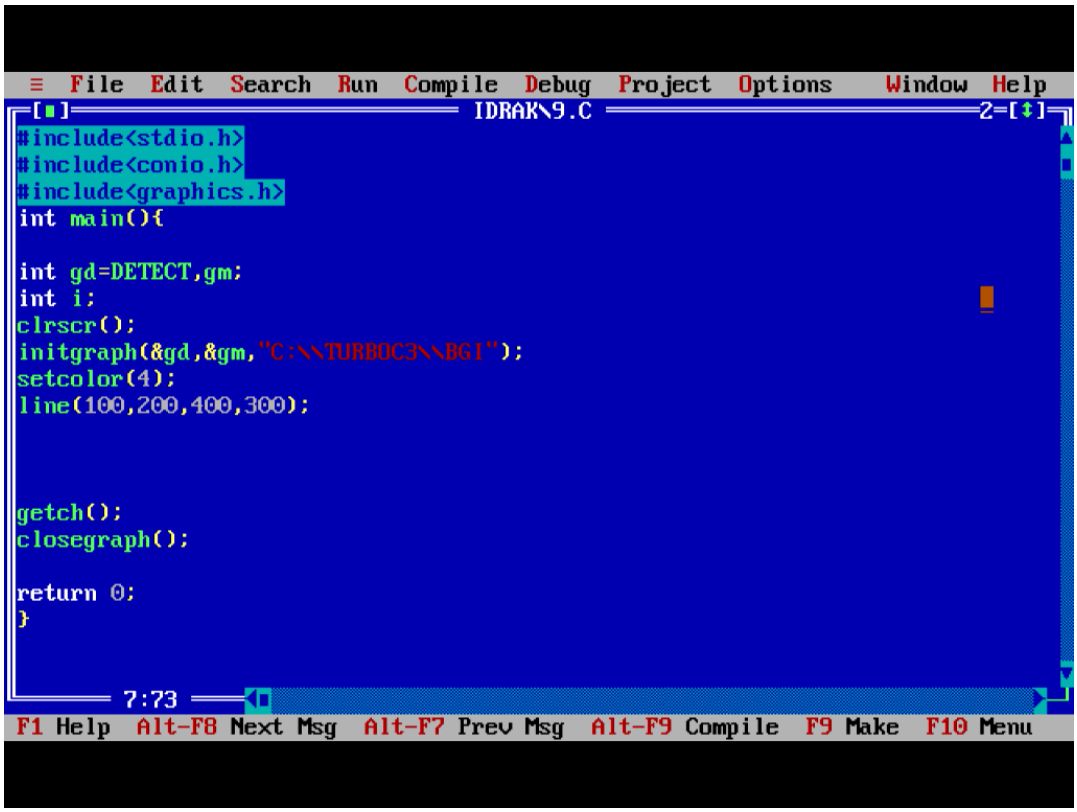
- **Output: -**



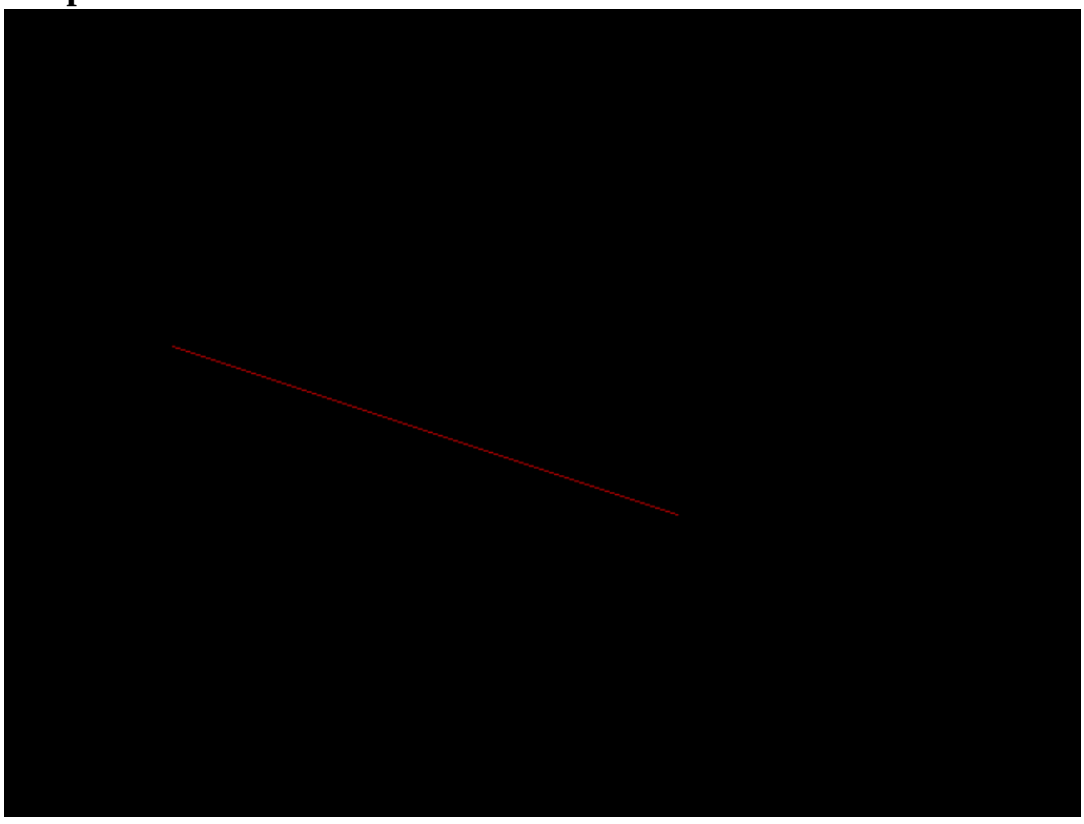
9. Line: -

Line function draw line between two specified points.

- **Syntax: -**
  line(int x1, int y1, int x2, int y2);
  line draws a line from (x1, y1) to (x2, y2) using current.
- **Example: -**



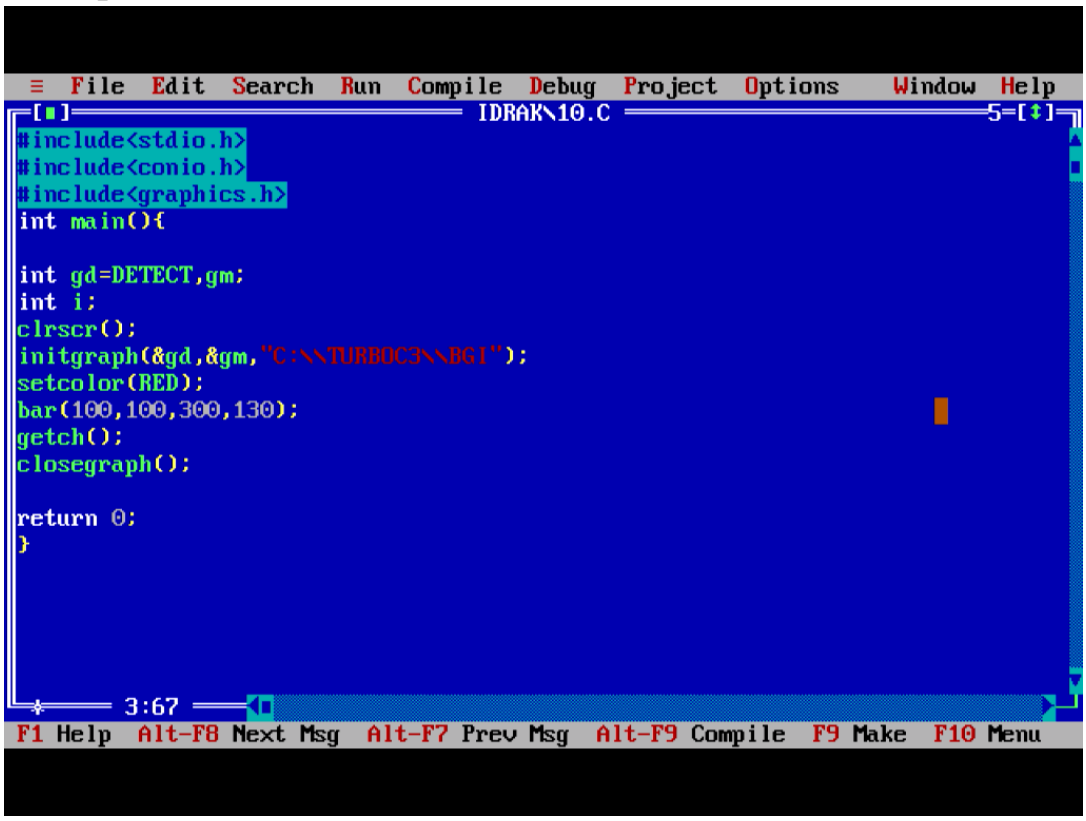
- **Output: -**



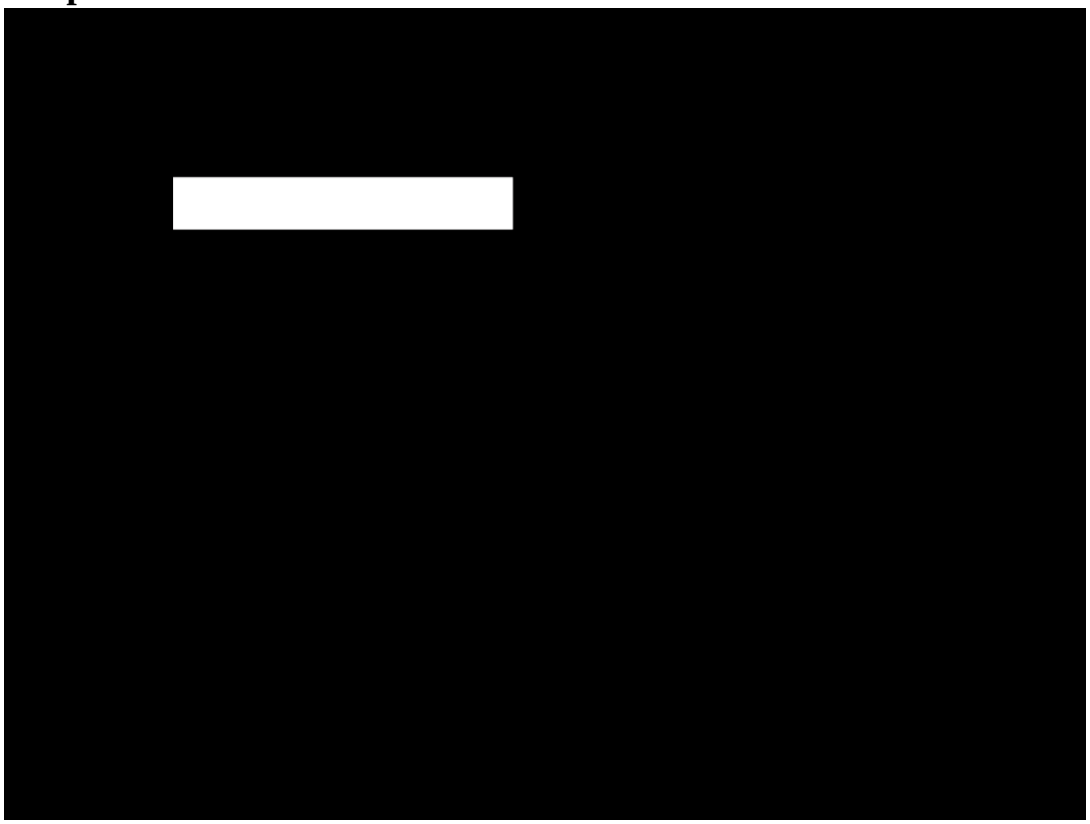
10. Bar: -

Draw a bar.

- **Syntax: -**
  bar(int left, int top, int right, int bottom);
  bar draws a filled-in, rectangular 2d bar.
- **Example: -**



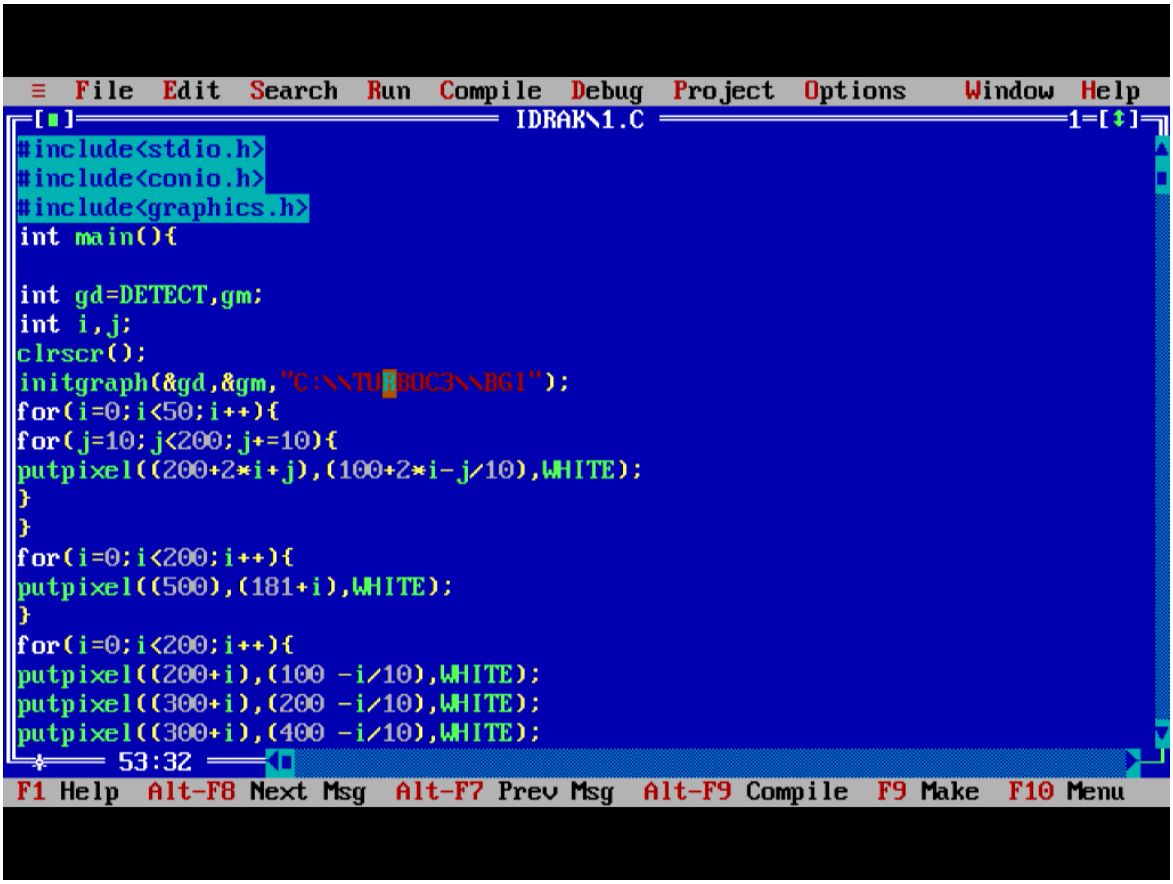
- **Output: -**

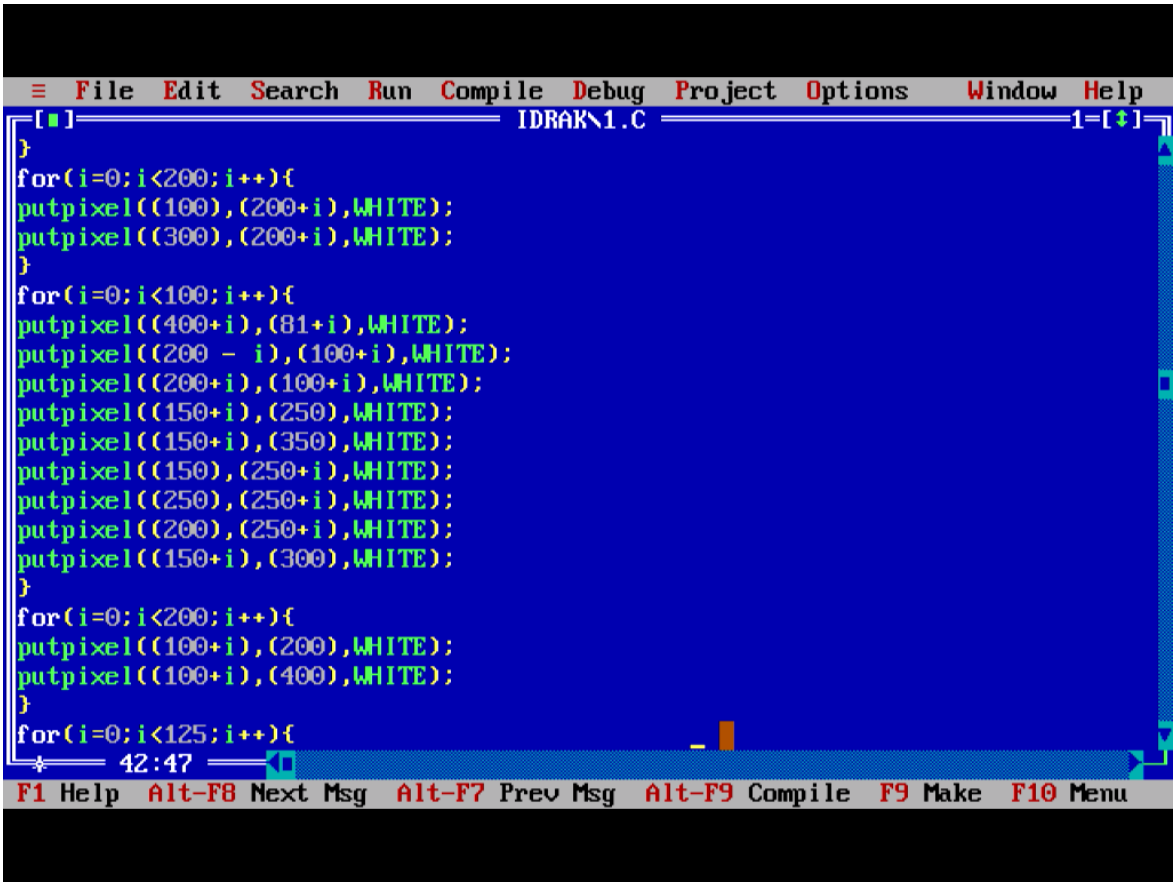


Experiment – 2

➢ Implement an image using putpixel function.

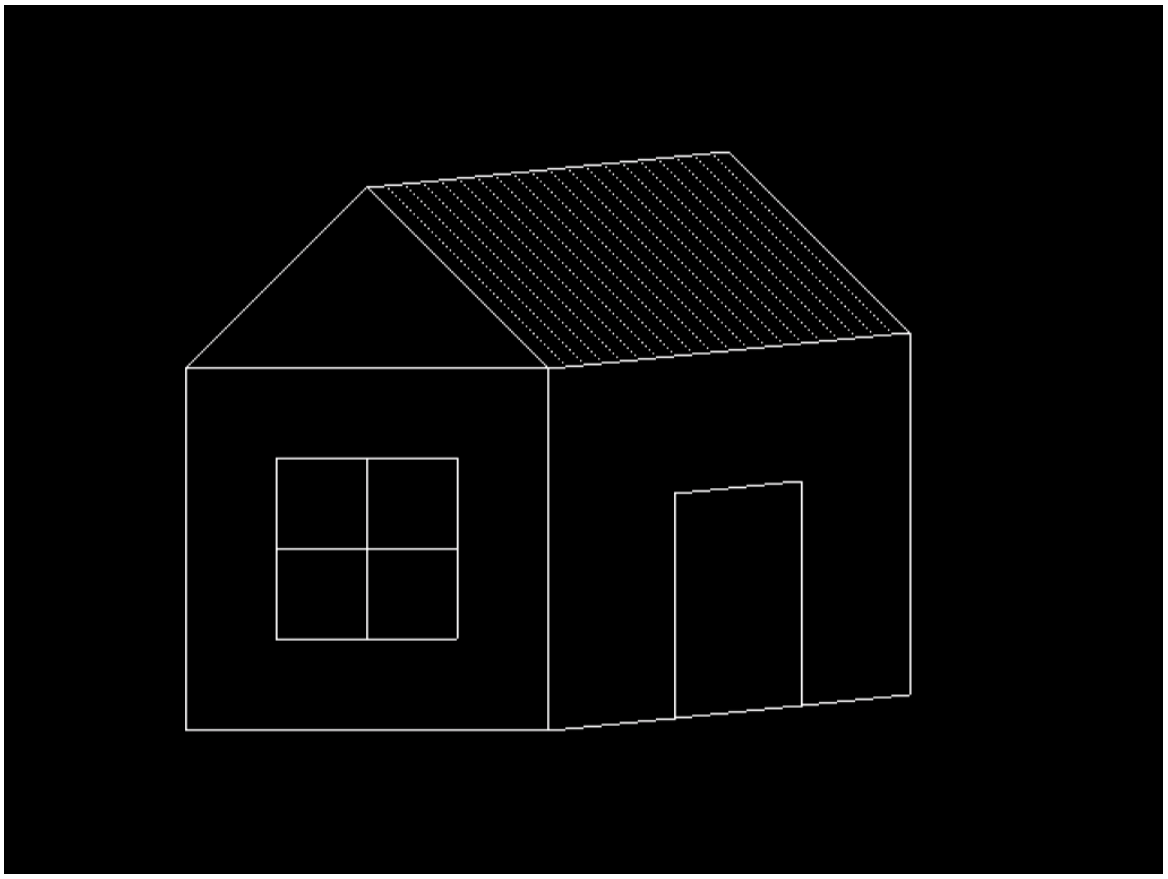
Code:-





```
putpixel((150),(250+i),WHITE);
putpixel((250),(250+i),WHITE);
putpixel((200),(250+i),WHITE);
putpixel((150+i),(300),WHITE);
}
for(i=0;i<200;i++){
putpixel((100+i),(200),WHITE);
putpixel((100+i),(400),WHITE);
}
for(i=0;i<125;i++){
putpixel((370),(270+i),WHITE);
putpixel((440),(263+i),WHITE);
//putpixel((500),(275+i),WHITE);
}
for(i=0;i<70;i++){
putpixel((370+i),(269 -i/10),WHITE);
}
getch();
closegraph();
return (0);
}
```

Output: -



Experiment – 3

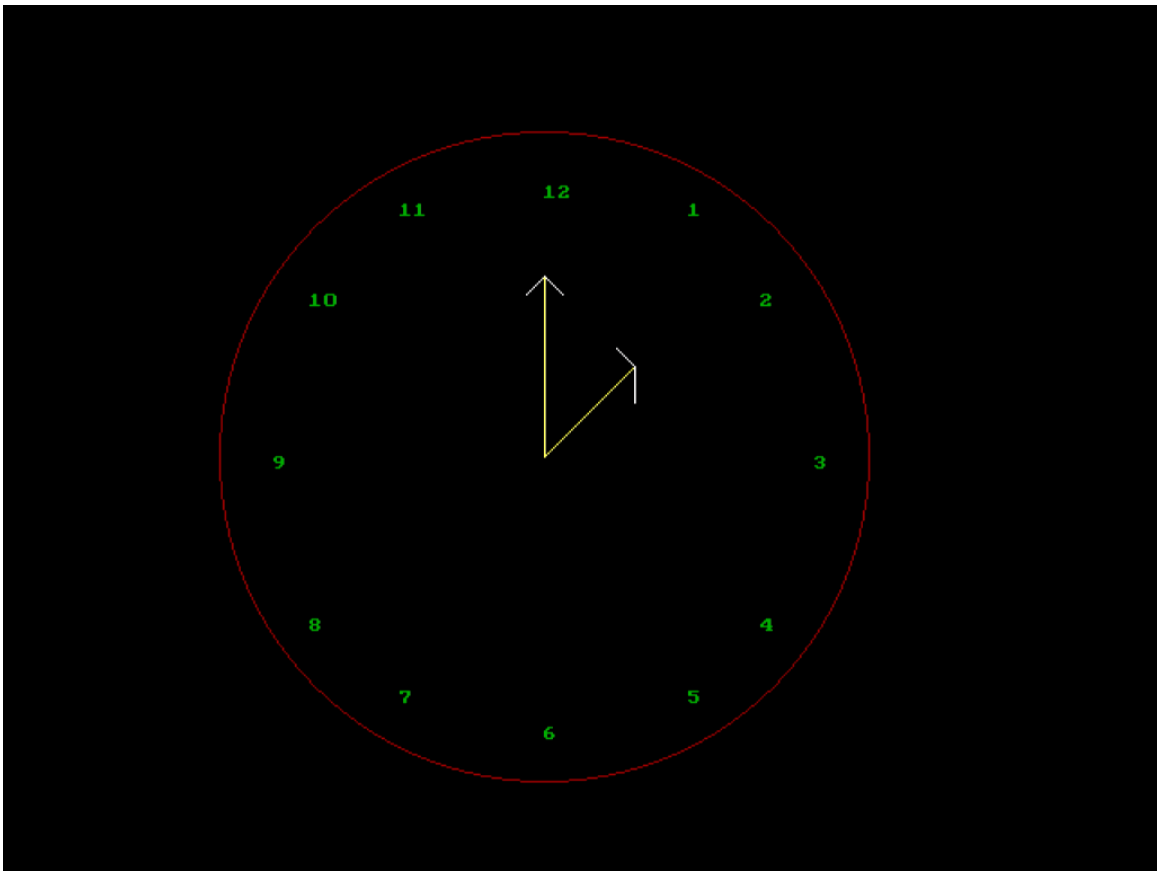
- **Implement an image using graphics function.**

Code: -

```
≡ File  Edit  Search  Run  Compile  Debug  Project  Options    Window  Help
[■]═══════════════ IDRAK\11.C ═══════════════2=[↕]
#include<conio.h>
#include<stdio.h>
#include<graphics.h>
int main(){

int gd=DETECT,gm;
int i;
clrscr();
initgraph(&gd,&gm,"C:\\TURBOC3\\BGI");
setcolor(RED);
circle(300,250,180);

putpixel(300,250,RED);
setcolor(YELLOW);
line(300,250,300,150);
line(300,250,350,200);

setcolor(WHITE);
line(300,150,290,160);
line(300,150,310,160);
line(350,200,340,190);
═══ 2:15 ═══
F1 Help  Alt-F8 Next Msg  Alt-F7 Prev Msg  Alt-F9 Compile  F9 Make  F10 Menu
```

```
≡ File  Edit  Search  Run  Compile  Debug  Project  Options    Window  Help
[■]═══════════════ IDRAK\11.C ═══════════════2=[↕]
line(350,200,340,190);
line(350,200,350,220);
setcolor(GREEN);
outtextxy(300,100,"12");
outtextxy(450,250,"3");
outtextxy(300,400,"6");
outtextxy(150,250,"9");
outtextxy(170,160,"10");
outtextxy(220,110,"11");
outtextxy(380,110,"1");
outtextxy(420,160,"2");
outtextxy(380,380,"5");
outtextxy(420,340,"4");
outtextxy(220,380,"7");
outtextxy(170,340,"8");
getch();
closegraph();

return 0;
}
═══ 41:1 ═══
F1 Help  Alt-F8 Next Msg  Alt-F7 Prev Msg  Alt-F9 Compile  F9 Make  F10 Menu
```

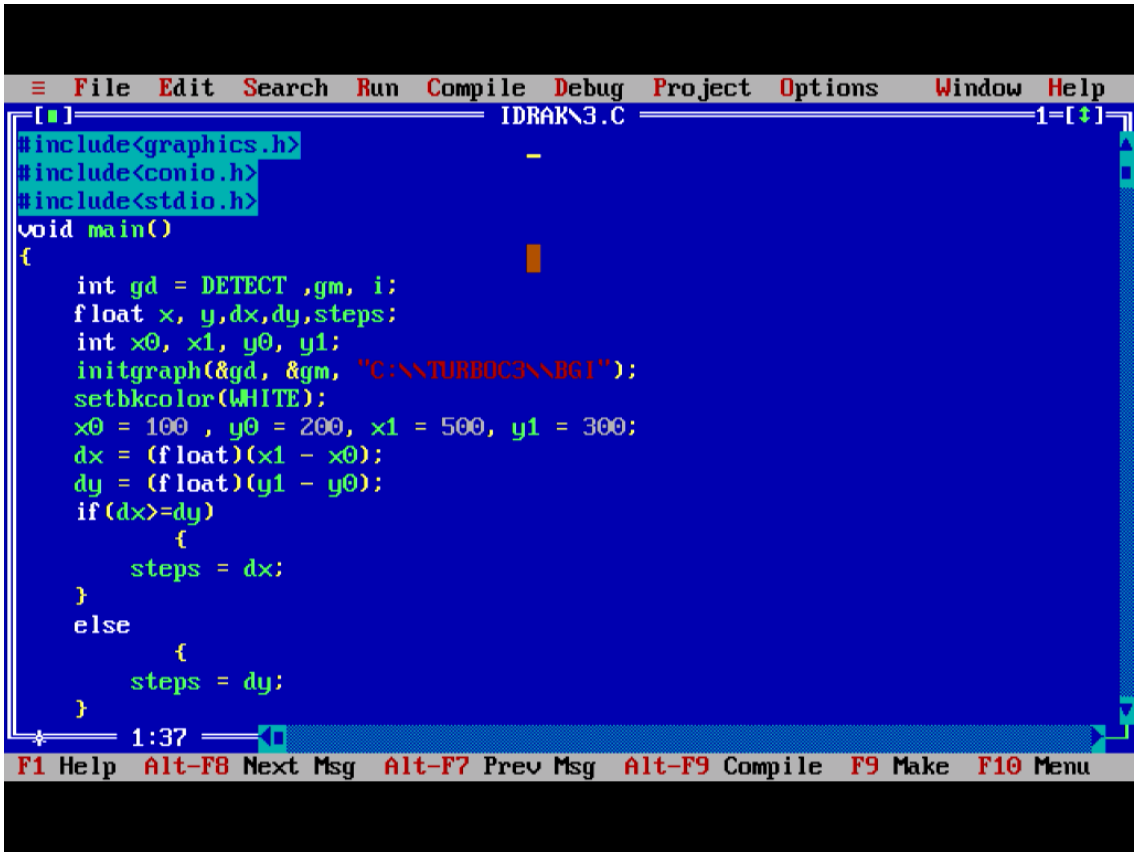
Output: -

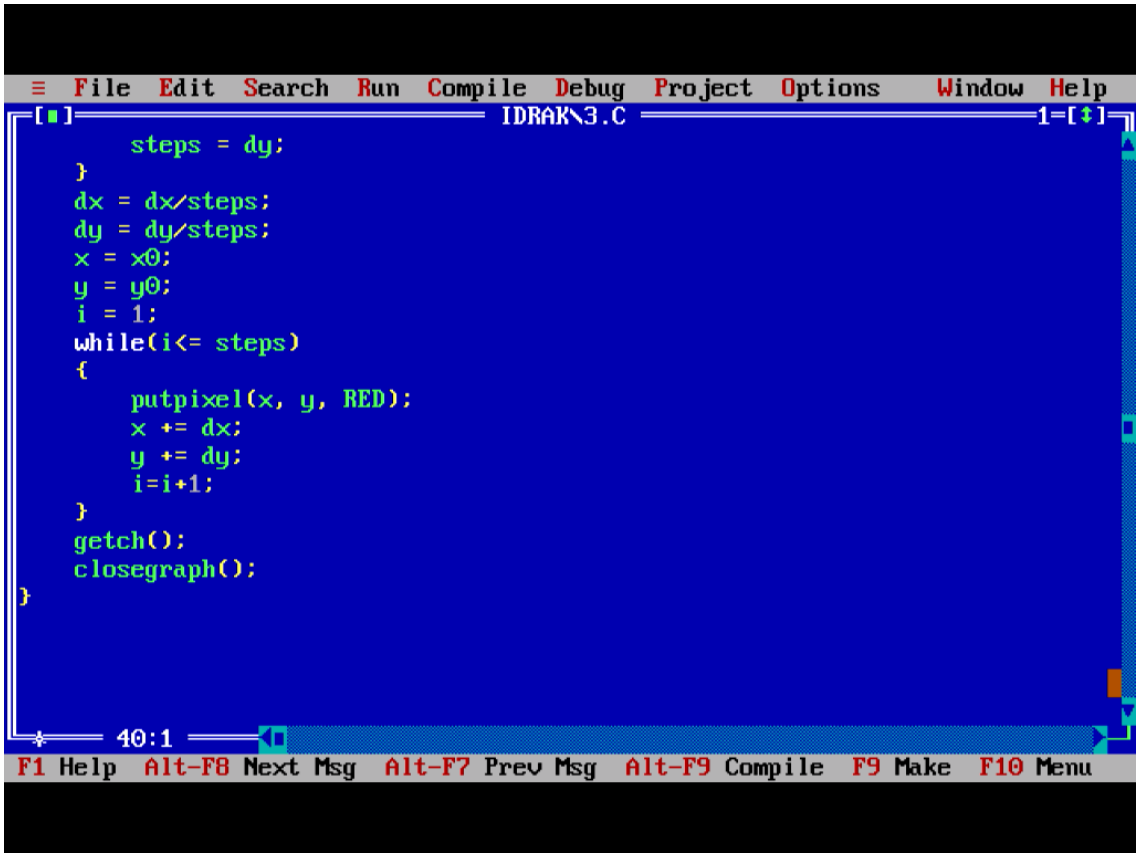


Experiment – 4

- **Rasterize a line using DDA algorithm.**

Code: -





Output: -