

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:-

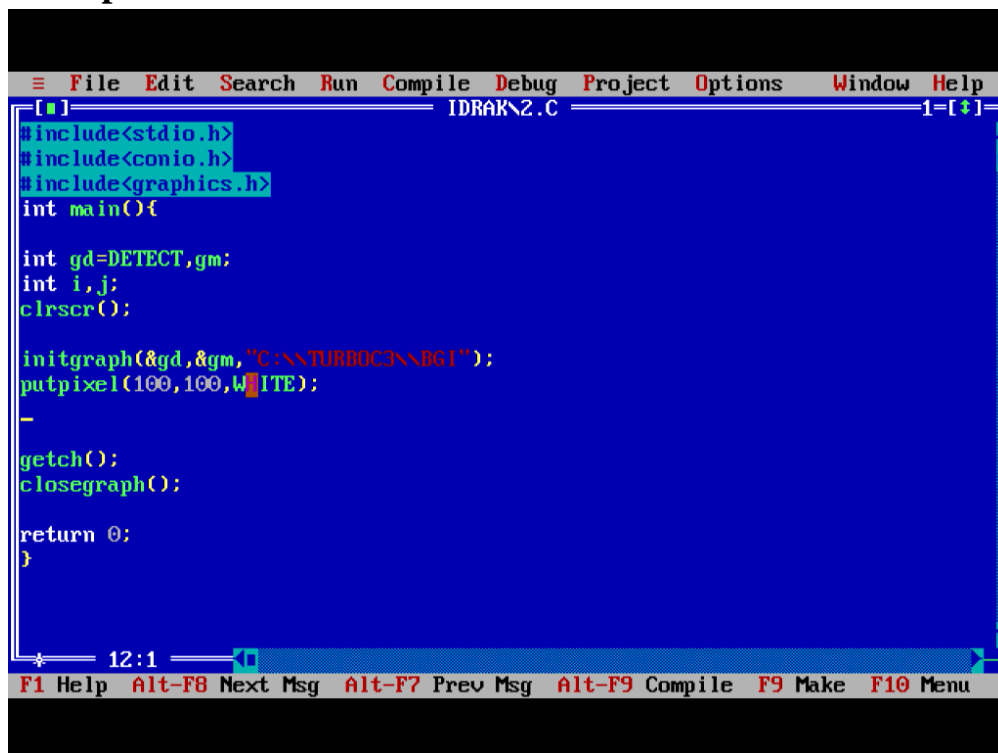
Putpixel () plots a pixel at a specified points.

- **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: -**



```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
int main()
{
    int gd=DETECT,gm;
    int i,j;
    clrscr();

    initgraph(&gd,&gm,"C:\\N\\TURBOC\\N\\BG1");
    putpixel(100,100,WHITE);

    getch();
    closegraph();

    return 0;
}
```

- **Output: -**



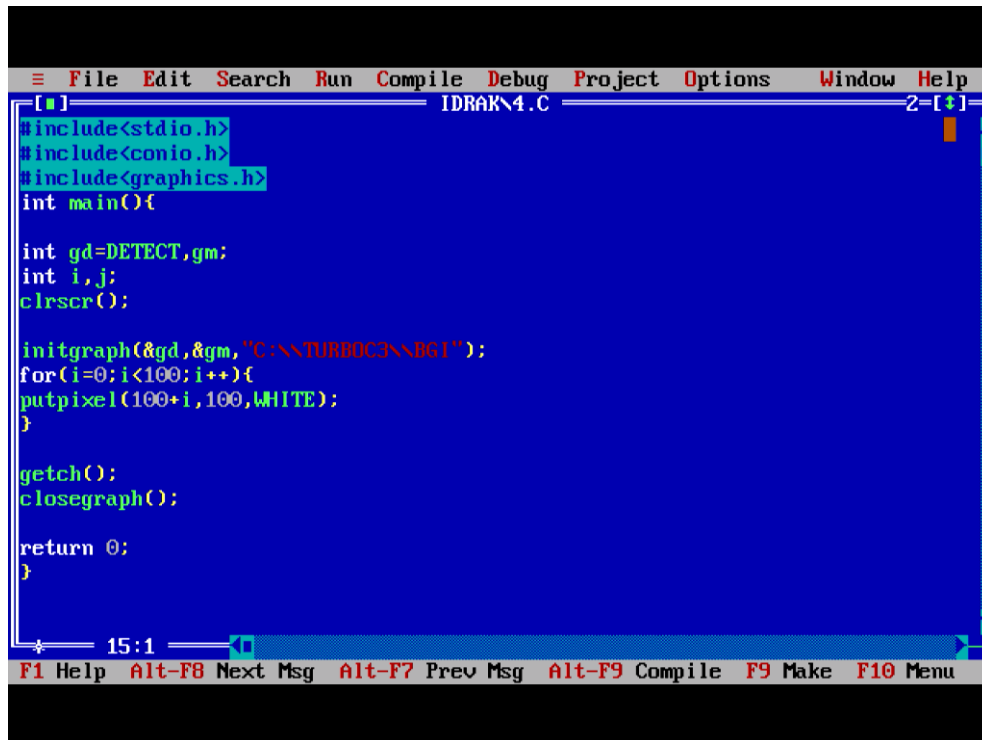
2. Closegraph: -

It shut downs the graphics system.

- **Syntax: -**
closegraph ();

It deallocates all the memory allocated by the graphics system.

- **Example: -**



```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
int main(){
    int gd=DETECT,gm;
    int i,j;
    clrscr();

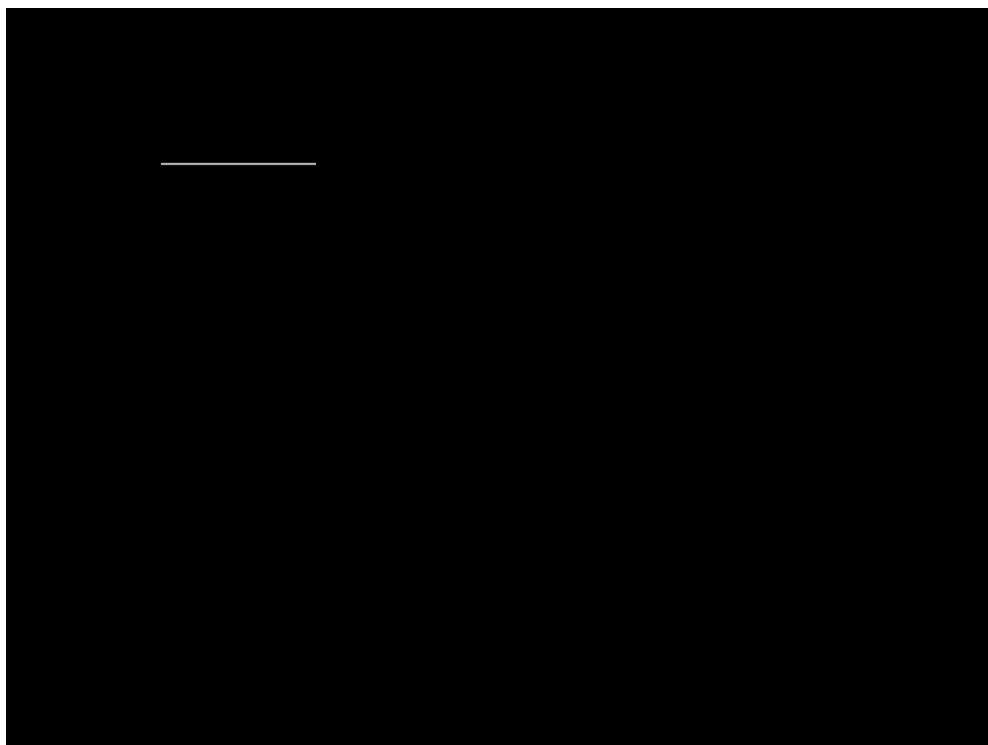
    initgraph(&gd,&gm,"C:\\\\TURBOC3\\\\BGI");
    for(i=0;i<100;i++){
        putpixel(100+i,100,WHITE);
    }

    getch();
    closegraph();

    return 0;
}
```

The screenshot shows the Turbo C++ IDE with the file IDRAK\\4.C open. The code is a C program that initializes the graphics system, draws a horizontal line of 100 white pixels, and then calls closegraph() to shut down the graphics system. The status bar at the bottom shows the cursor is at line 15, column 1.

Output: -



3. Intigraph: -

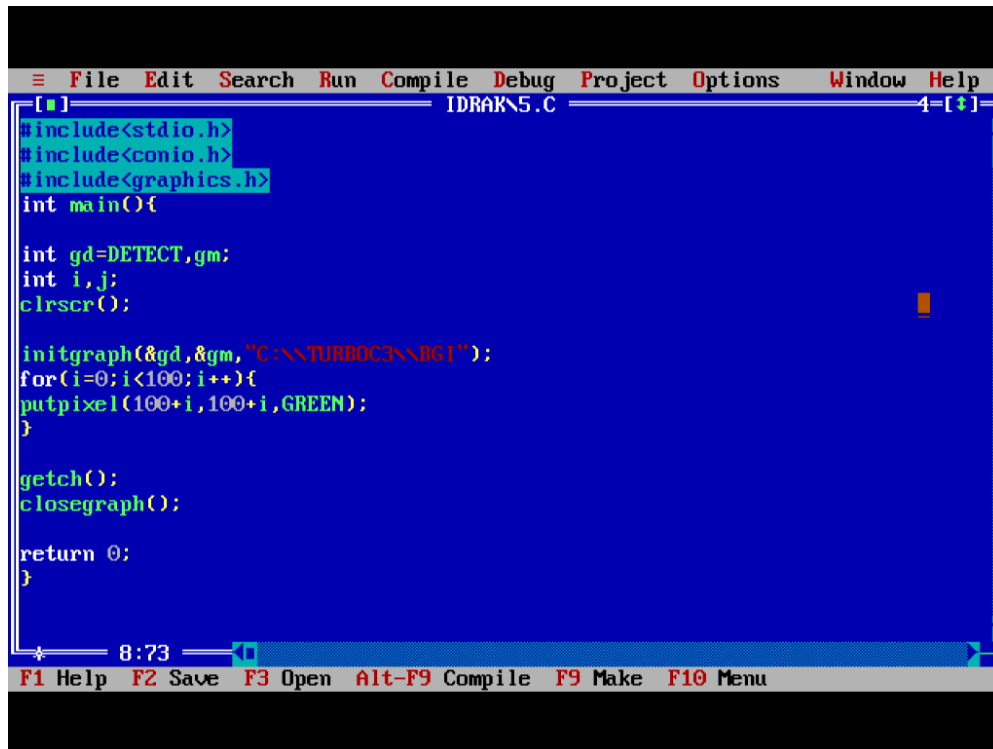
Initialize the graphics system.

- **Syntax: -**

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

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

- **Example: -**



```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
int main(){

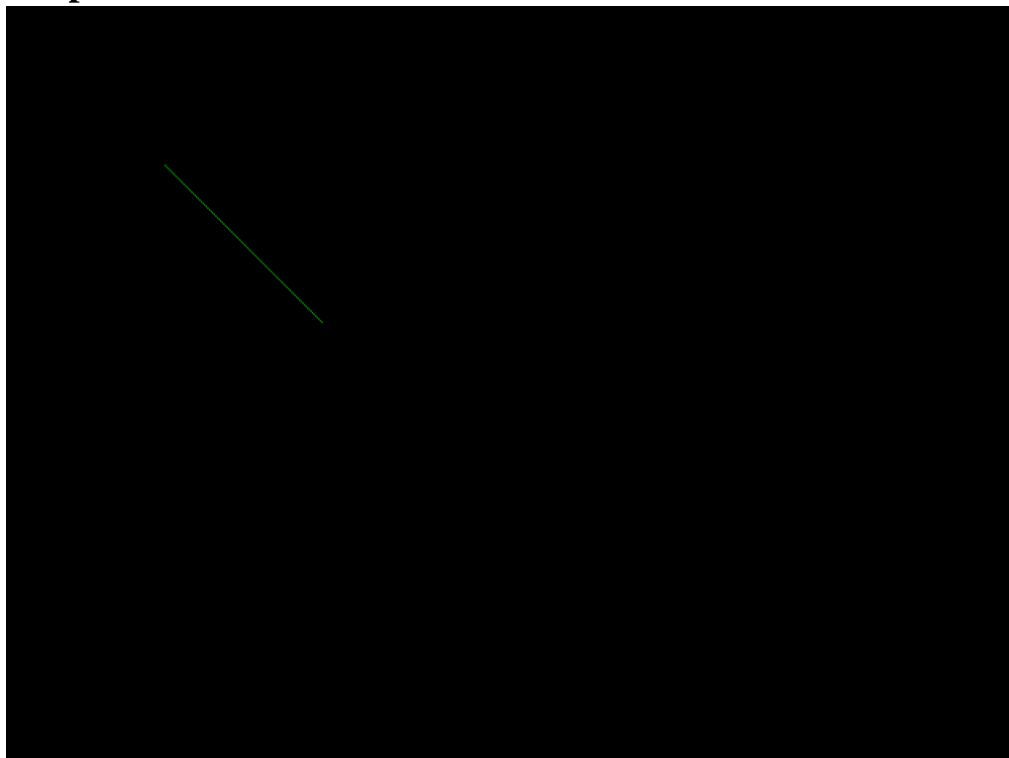
int gd=DETECT,gm;
int i,j;
clrscr();

initgraph(&gd,&gm,"C:\\N\\TURBOC3\\BGI");
for(i=0;i<100;i++){
putpixel(100+i,100+i,GREEN);
}

getch();
closegraph();

return 0;
}
```

- **Output: -**



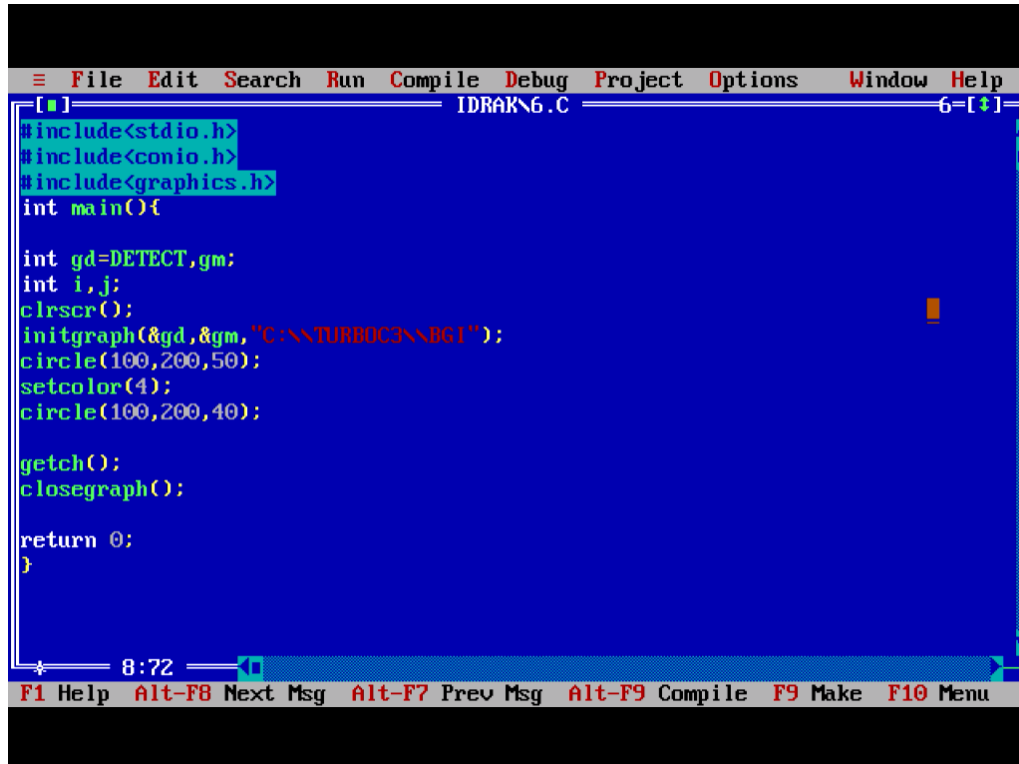
4. Setcolor: -

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: -**



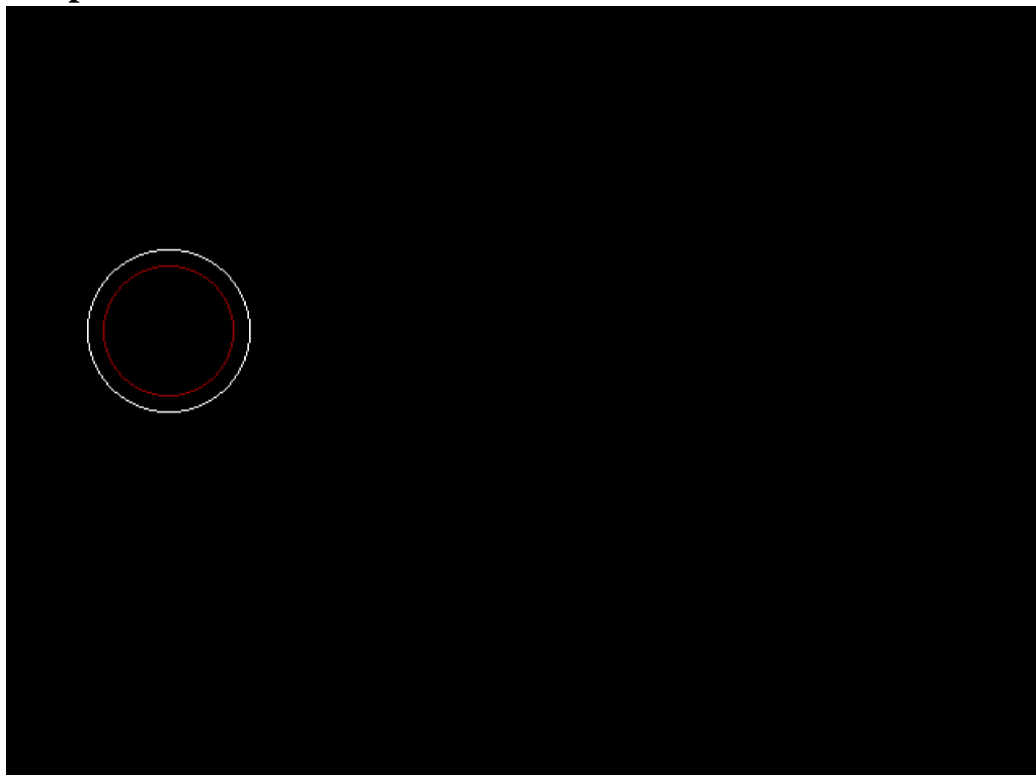
```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
int main(){

int gd=DETECT,gm;
int i,j;
clrscr();
initgraph(&gd,&gm,"C:\\\\TURBOC3\\\\\\BGI");
circle(100,200,50);
setcolor(4);
circle(100,200,40);

getch();
closegraph();

return 0;
}
```

- **Output: -**



5. Getcolor: -

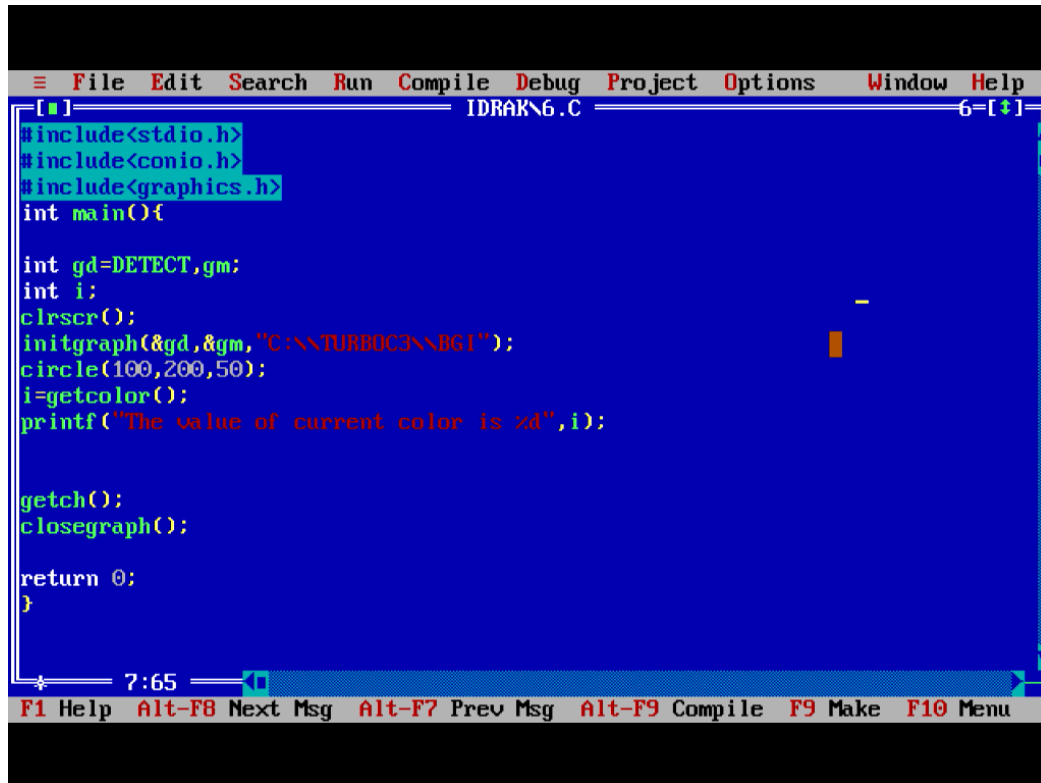
Getcolor returns the current drawing color.

- **Syntax: -**

getcolor();

it returns the current drawing color.

- **Example: -**



```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
int main()
{
    int gd=DETECT,gm;
    int i;
    clrscr();
    initgraph(&gd,&gm,"C:\\N\\TURBOC3\\N\\BGI");
    circle(100,200,50);
    i=getcolor();
    printf("The value of current color is %d",i);

    getch();
    closegraph();

    return 0;
}
```

- **Output: -**



6. Arc: -

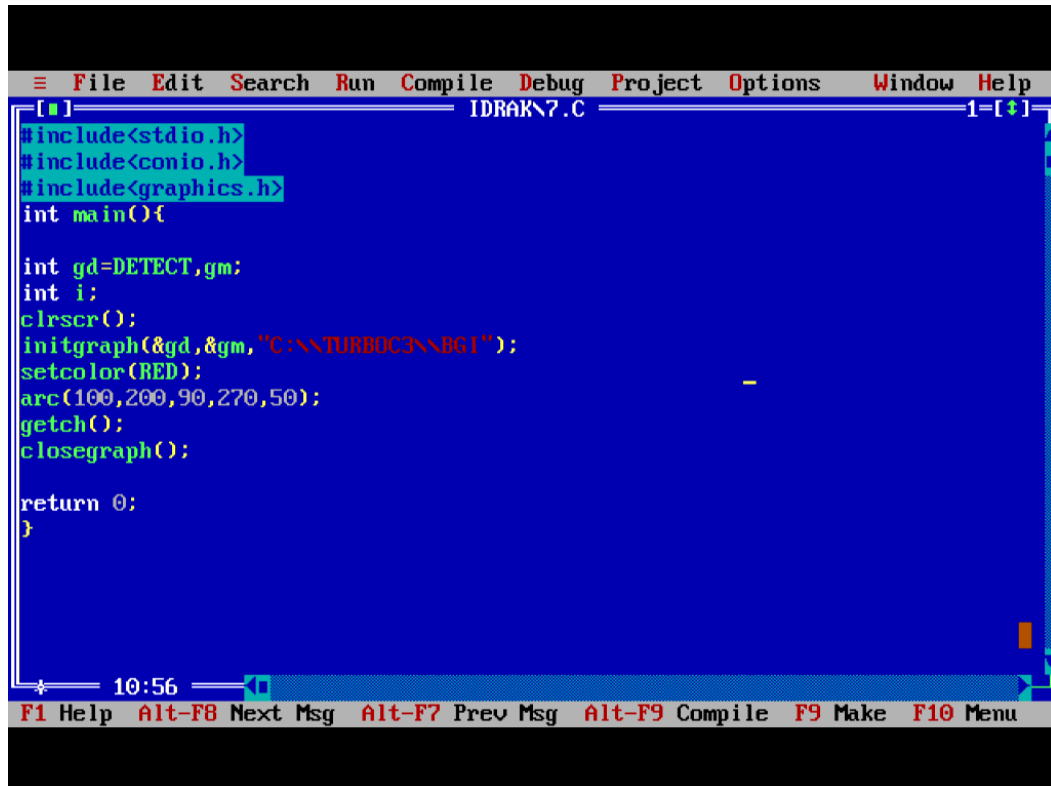
Arc draw a circular arc.

- **Syntax: -**

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

arc draws a circular arc in the current drawing color.

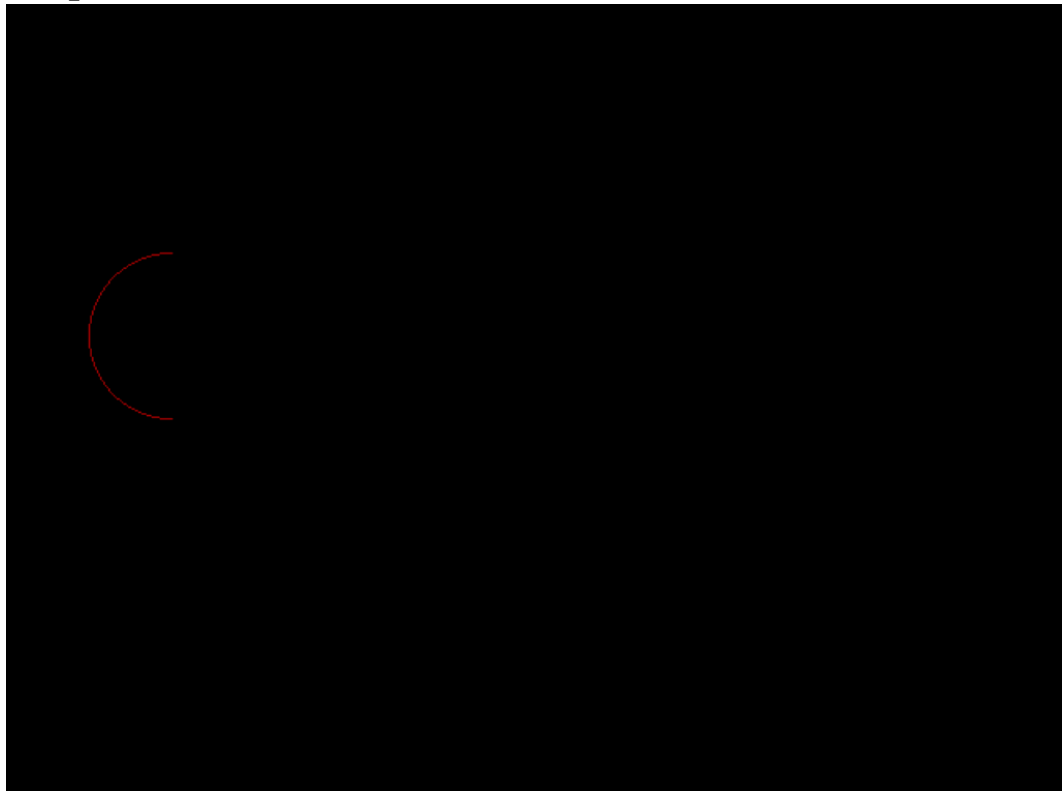
- **Example: -**



```
File Edit Search Run Compile Debug Project Options Window Help
IDRAKN7.C 1=[+]  
#include<stdio.h>  
#include<conio.h>  
#include<graphics.h>  
int main()  
{  
    int gd=DETECT,gm;  
    int i;  
    clrscr();  
    initgraph(&gd,&gm,"C:\\N\\TURBOC3\\BGI");  
    setcolor(RED);  
    arc(100,200,90,270,50);  
    getch();  
    closegraph();  
  
    return 0;  
}
```

10:56
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9 Make F10 Menu

- **Output: -**



7. Circle: -

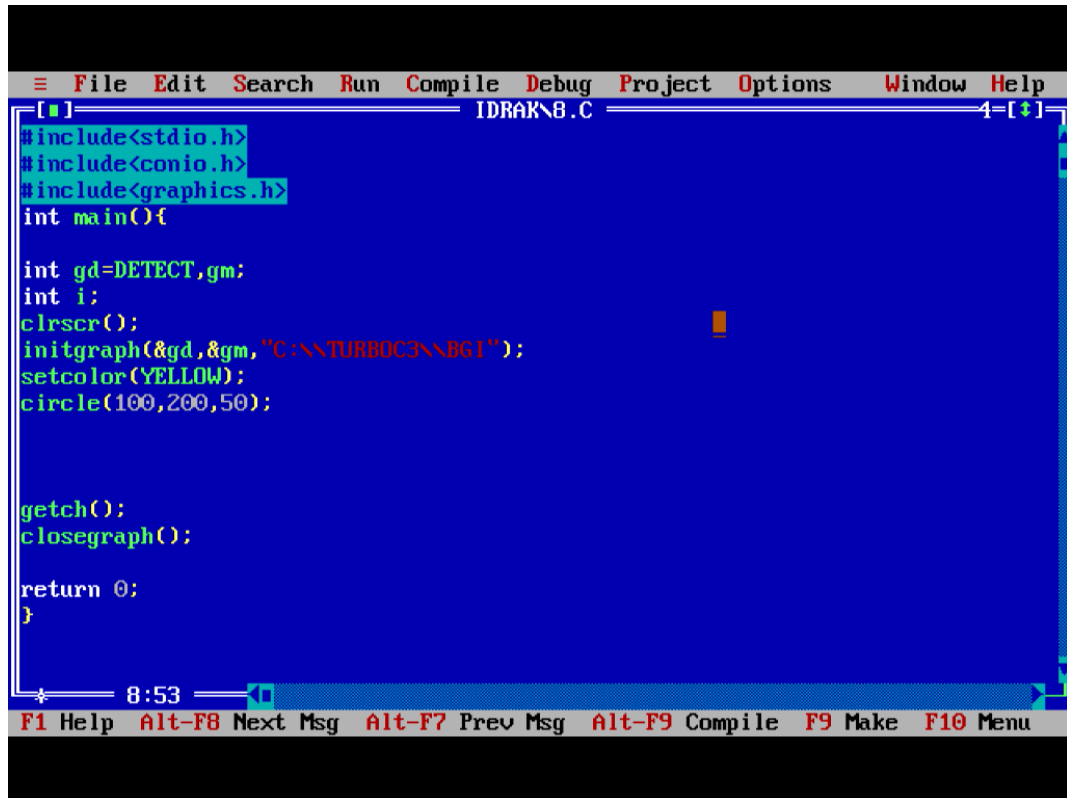
Circle draws a circle.

- **Syntax: -**

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

circle draws a circle in the current drawing color.

- **Example: -**



```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
int main(){

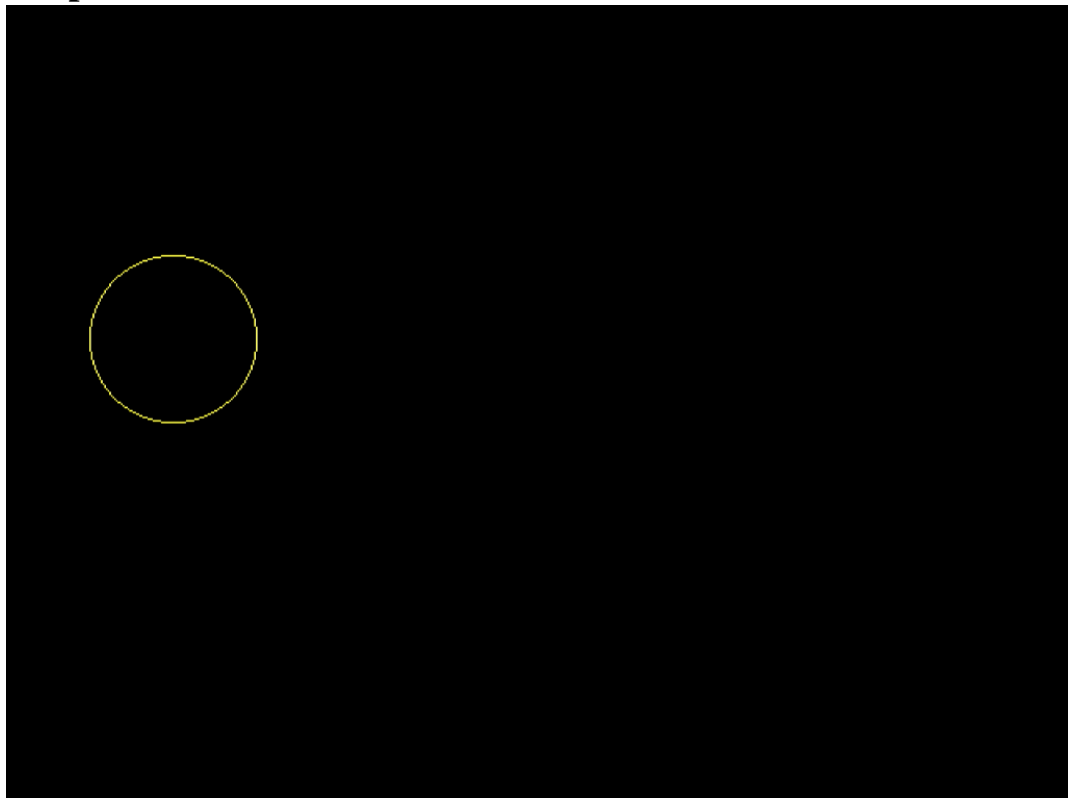
int gd=DETECT,gm;
int i;
clrscr();
initgraph(&gd,&gm,"C:\\\\TURBOC3\\\\BGI");
setcolor(YELLOW);
circle(100,200,50);

getch();
closegraph();

return 0;
}
```

The screenshot shows the Turbo C++ IDE with the file 'IDRAK\\8.C' open. The code in the editor includes headers for stdio, conio, and graphics, and uses the circle function to draw a yellow circle at coordinates (100, 200) with a radius of 50. The status bar at the bottom shows the time as 8:53 and various function key shortcuts.

- **Output: -**



8. Pieslice: -

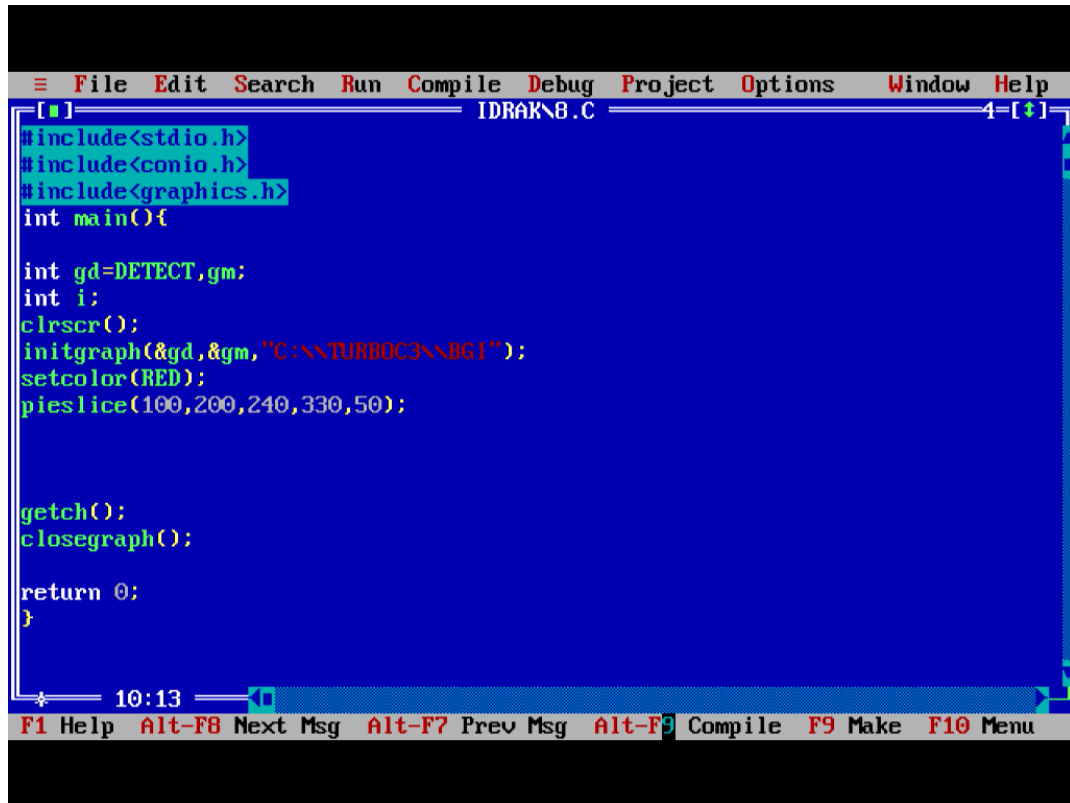
Pieslice draws and fill a circular pie slice.

- **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: -**



```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
int main(){

int gd=DETECT,gm;
int i;
clrscr();
initgraph(&gd,&gm,"C:\\\\TURBOC3\\\\BGI");
setcolor(RED);
pieslice(100,200,240,330,50);

getch();
closegraph();

return 0;
}
```

- **Output: -**



9. Line: -

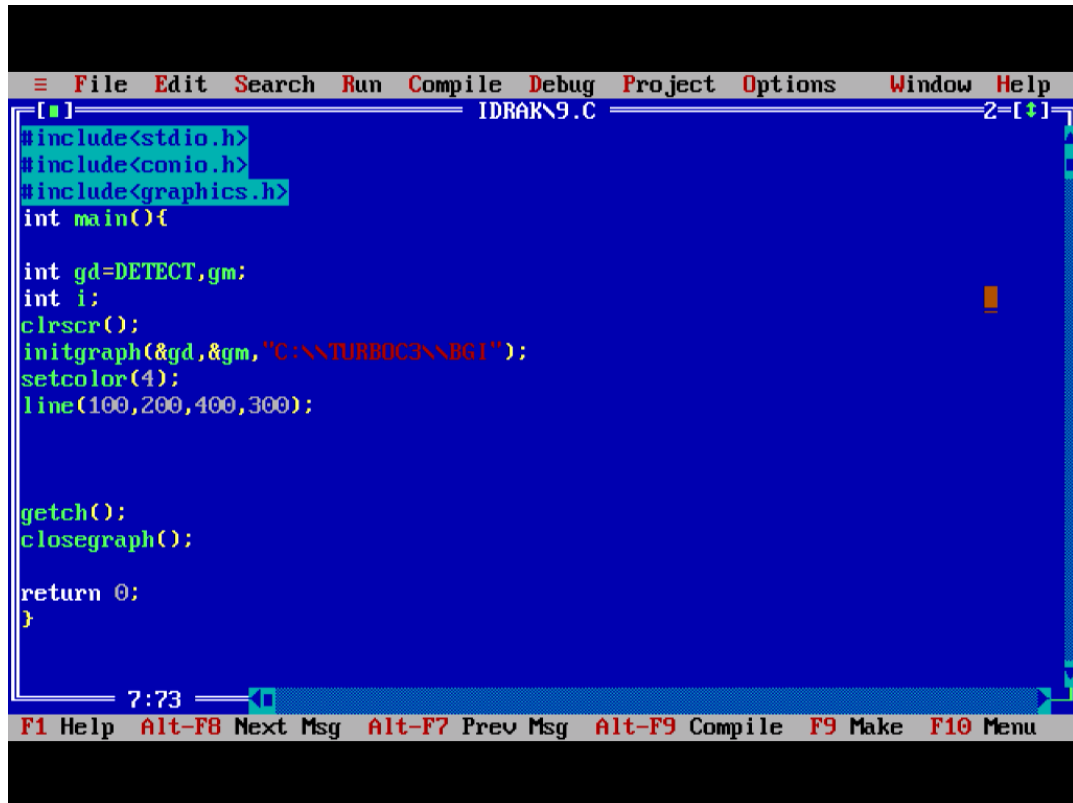
Line function draw line between two specified points.

- **Syntax: -**

line(int x1, int y1, int x2, int y2);

line draws a line from (x1, y1) to (x2, y2) using current.

- **Example: -**



```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
int main(){

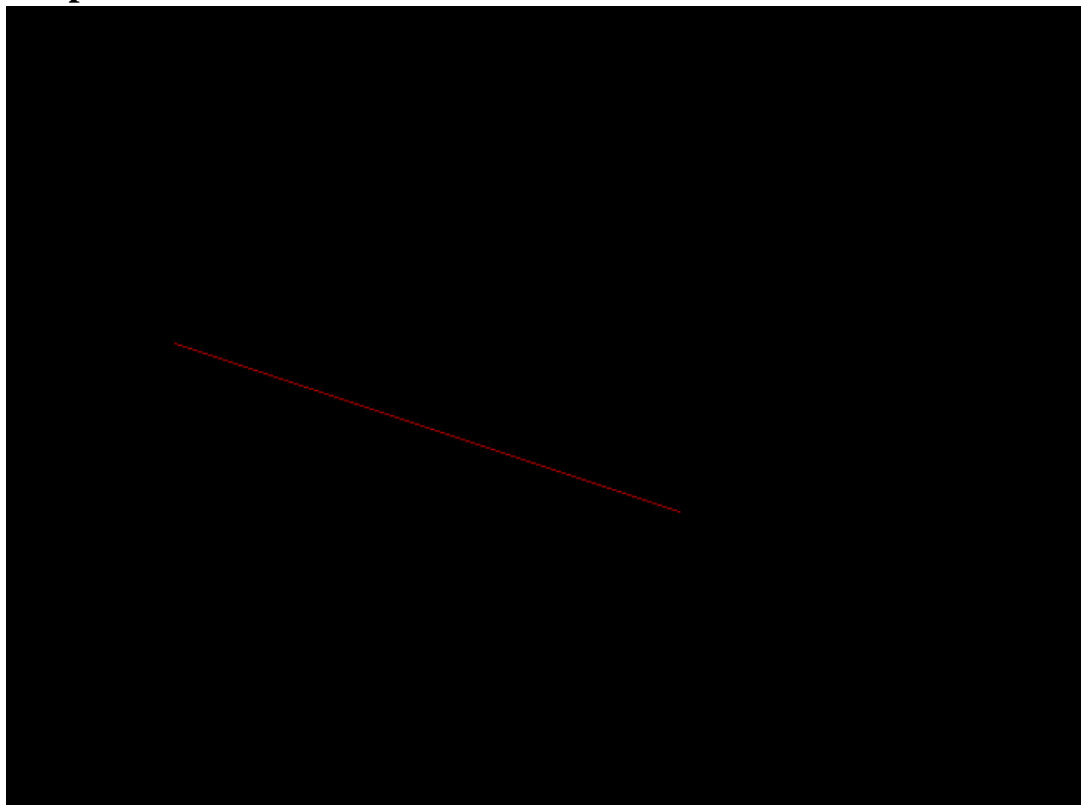
int gd=DETECT,gm;
int i;
clrscr();
initgraph(&gd,&gm,"C:\\\\TURBOC3\\\\BG1");
setcolor(4);
line(100,200,400,300);

getch();
closegraph();

return 0;
}
```

The screenshot shows the Turbo C++ IDE with the file 'IDRAK9.C'. The code defines a main function that initializes the graphics mode, clears the screen, sets the background to black, and draws a red line from (100, 200) to (400, 300). The status bar at the bottom shows function key shortcuts like F1 Help, Alt-F8 Next Msg, etc.

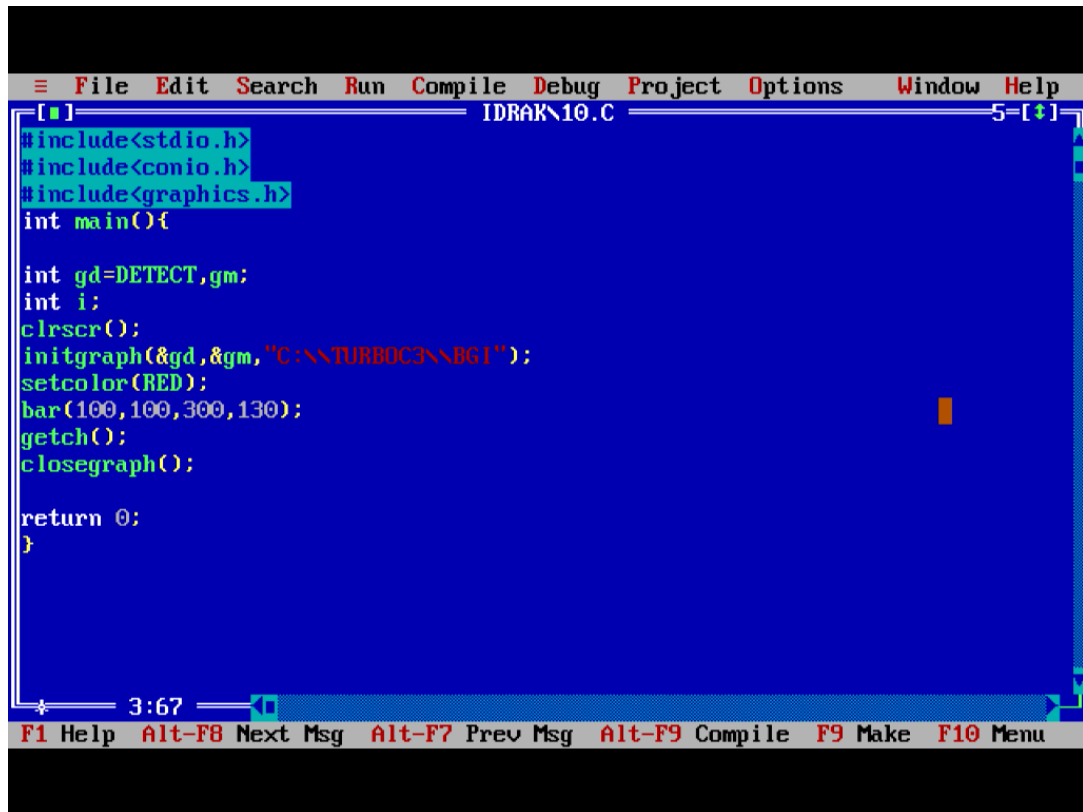
- **Output: -**



10. Bar: -

Draw a bar.

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



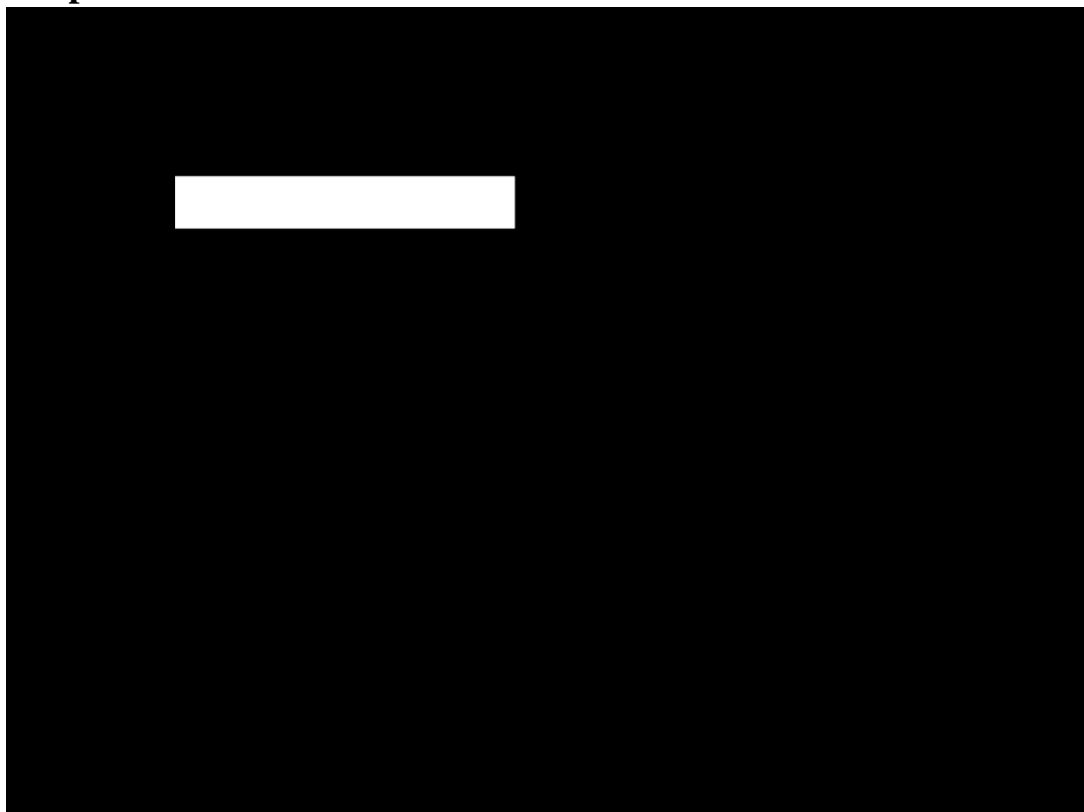
```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
int main(){

int gd=DETECT,gm;
int i;
clrscr();
initgraph(&gd,&gm,"C:\\N\\TURBODC3\\BGI");
setcolor(RED);
bar(100,100,300,130);
getch();
closegraph();

return 0;
}
```

The screenshot shows a Turbo C++ IDE window titled 'IDRAK\\10.C'. The code defines a main function that initializes the graphics mode, sets the color to red, and draws a bar from (100, 100) to (300, 130). The IDE interface includes a menu bar (File, Edit, Search, Run, Compile, Debug, Project, Options, Window, Help) and a status bar at the bottom with function key shortcuts (F1 Help, Alt-F8 Next Msg, Alt-F7 Prev Msg, Alt-F9 Compile, F9 Make, F10 Menu).

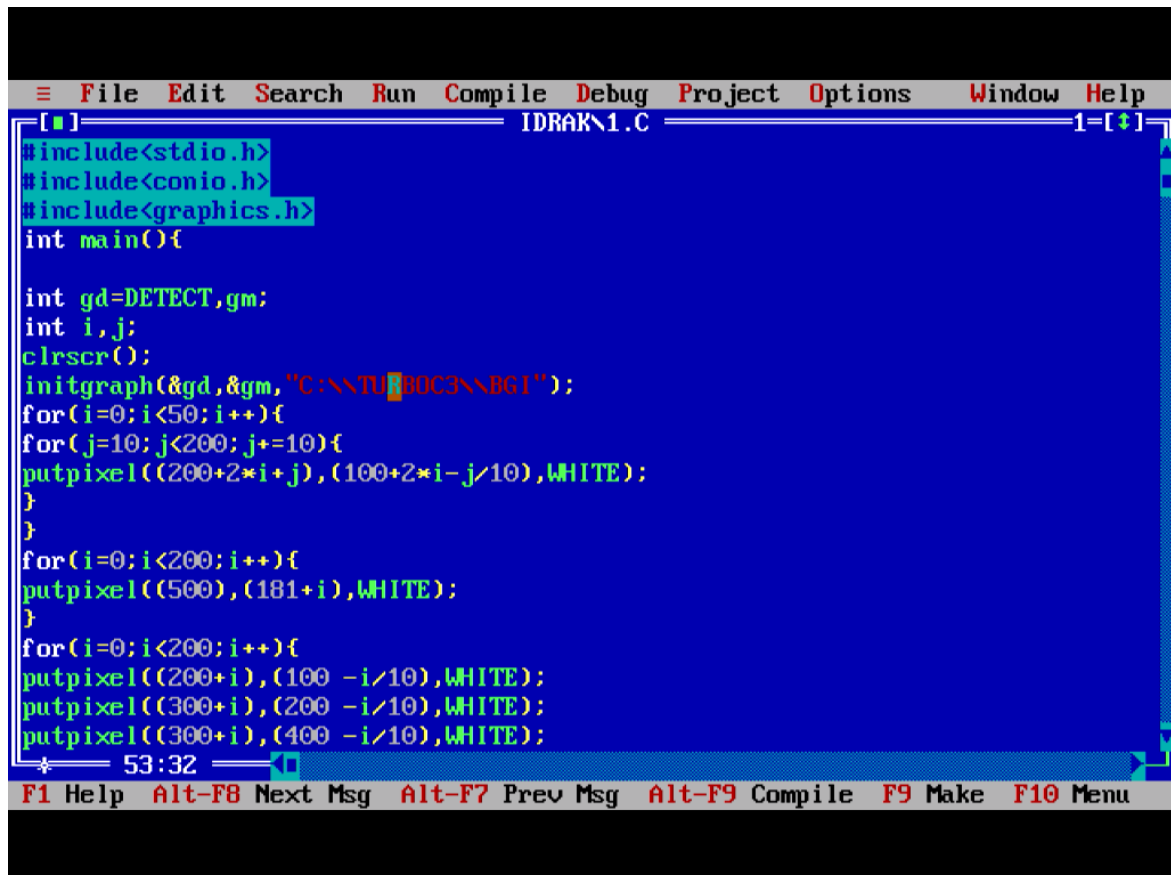
- **Output: -**



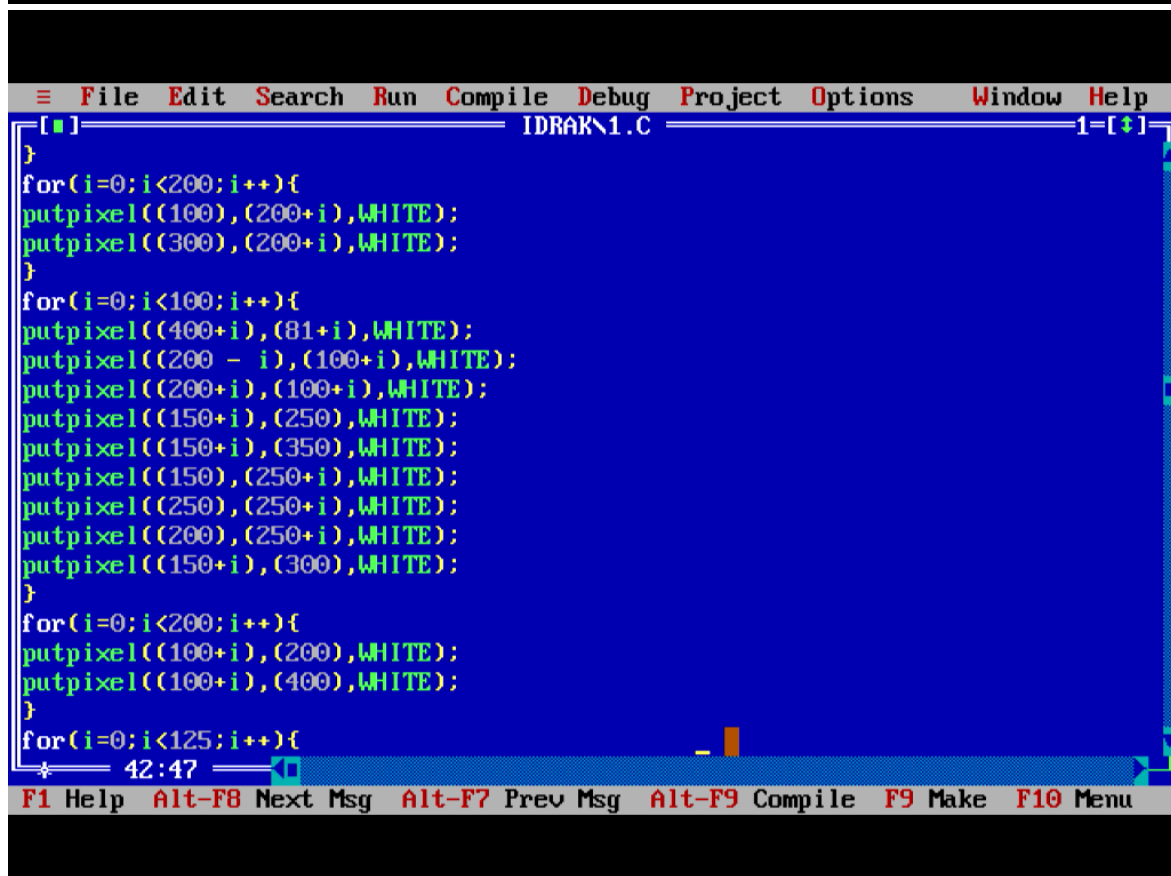
Experiment – 2

- Implement an image using putpixel function.

Code:-



```
File Edit Search Run Compile Debug Project Options Window Help
IDRAK\1.C 1=[+/-]
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
int main(){
    int gd=DETECT,gm;
    int i,j;
    clrscr();
    initgraph(&gd,&gm,"C:\\N\\TU\\BOC3\\BGI");
    for(i=0;i<50;i++){
        for(j=10;j<200;j+=10){
            putpixel((200+2*i+j),(100+2*i-j/10),WHITE);
        }
    }
    for(i=0;i<200;i++){
        putpixel((500),(181+i),WHITE);
    }
    for(i=0;i<200;i++){
        putpixel((200+i),(100 -i/10),WHITE);
        putpixel((300+i),(200 -i/10),WHITE);
        putpixel((300+i),(400 -i/10),WHITE);
    }
    * 53:32 *
```



```
File Edit Search Run Compile Debug Project Options Window Help
IDRAK\1.C 1=[+/-]
    }
    for(i=0;i<200;i++){
        putpixel((100),(200+i),WHITE);
        putpixel((300),(200+i),WHITE);
    }
    for(i=0;i<100;i++){
        putpixel((400+i),(81+i),WHITE);
        putpixel((200 - i),(100+i),WHITE);
        putpixel((200+i),(100+i),WHITE);
        putpixel((150+i),(250),WHITE);
        putpixel((150+i),(350),WHITE);
        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++){
    * 42:47 *
```

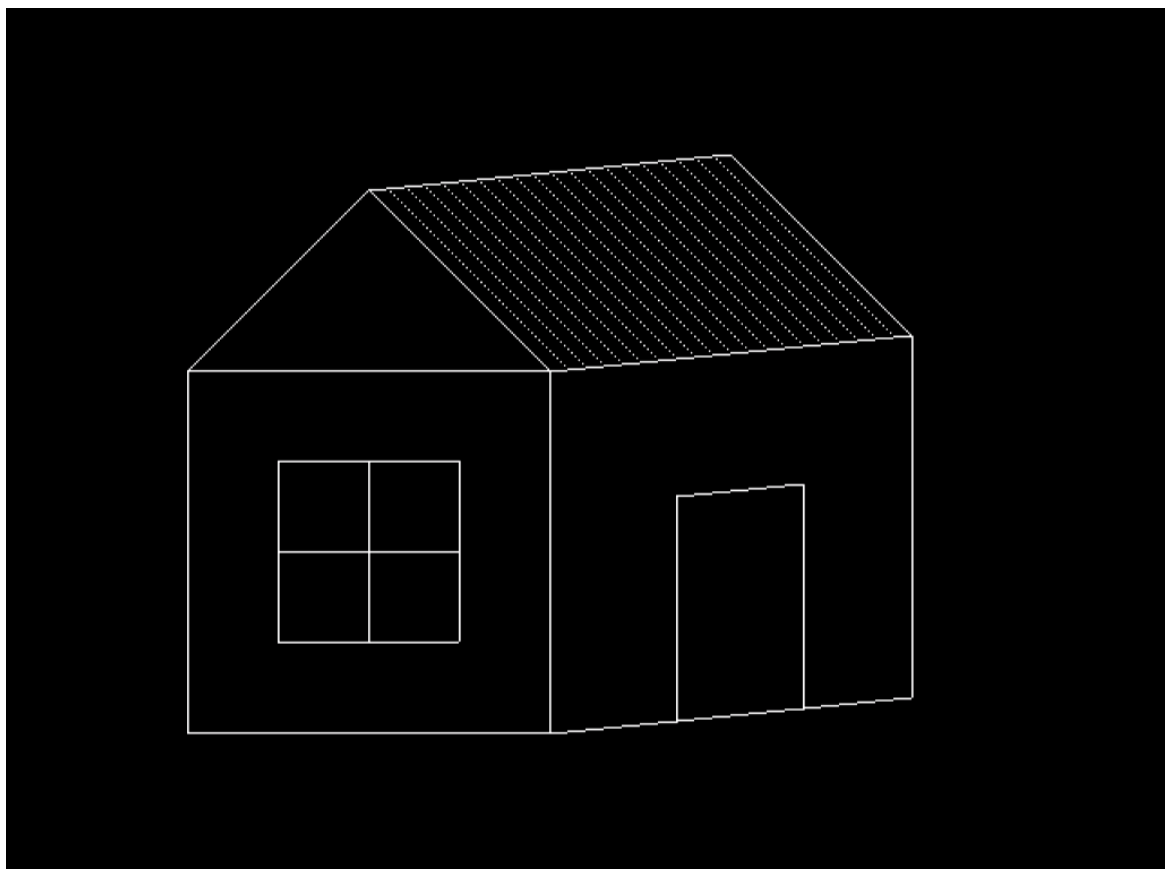
```
≡ File Edit Search Run Compile Debug Project Options Window Help
IDRAK\1.C 1-[+]
```

```
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);
}
```

```
* 53:47 *
```

```
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9 Make F10 Menu
```

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:\\NTURBOC3\\BGI");
setcolor(RED);
circle(300,250,180);

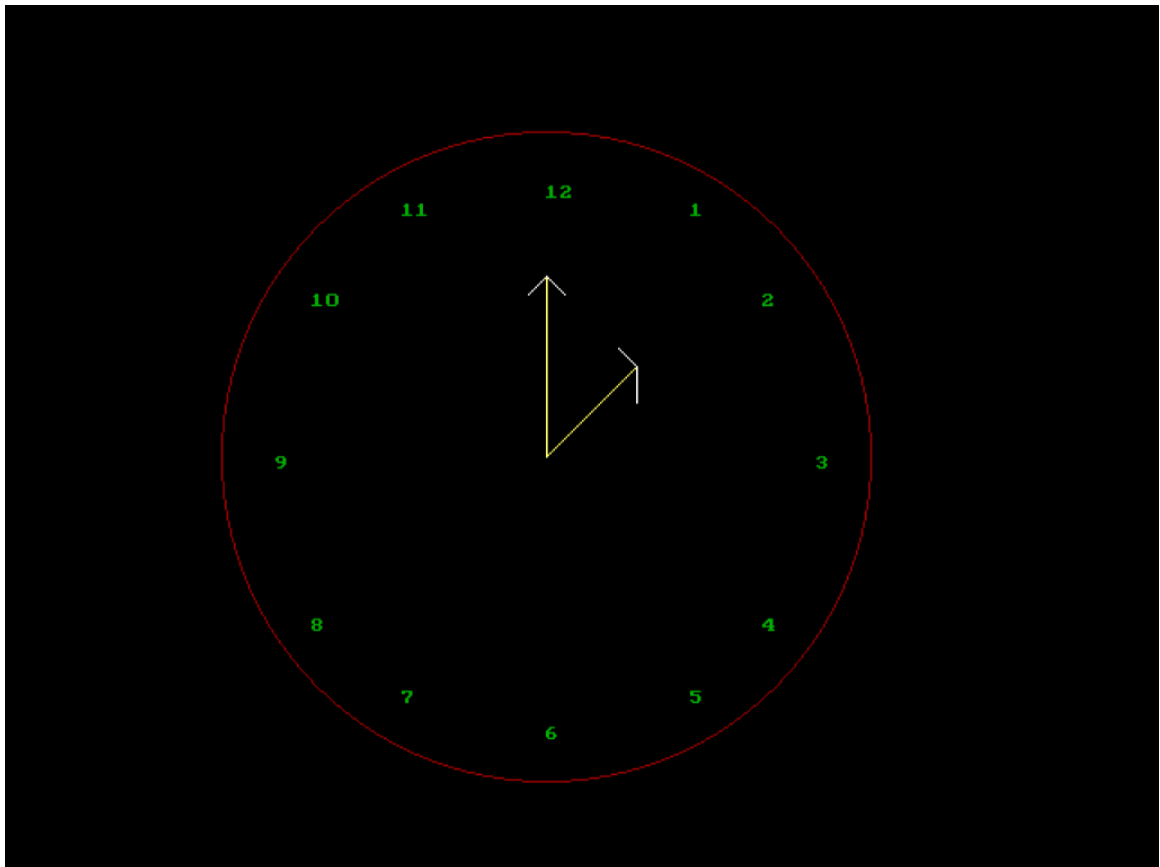
putpixel(300,250,RED);
setcolor(YELLOW);
line(300,250,300,150);
line(300,250,350,200);

setcolor(WHITE);
line(300,150,290,160);
line(300,150,310,160);
line(350,200,340,190);
* 2:15 *
```

```
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 *
```


Output: -



Experiment – 4

➤ Rasterize a line using DDA algorithm.

Code: -

```
File Edit Search Run Compile Debug Project Options Window Help
IDRAK\3.C 1=[+]
#include<graphics.h>
#include<conio.h>
#include<stdio.h>
void main()
{
    int gd = DETECT ,gm, i;
    float x, y,dx,dy,steps;
    int x0, x1, y0, y1;
    initgraph(&gd, &gm, "C:\\\\TURBOC3\\\\BGI");
    setbkcolor(WHITE);
    x0 = 100 , y0 = 200, x1 = 500, y1 = 300;
    dx = (float)(x1 - x0);
    dy = (float)(y1 - y0);
    if(dx>=dy)
    {
        steps = dx;
    }
    else
    {
        steps = dy;
    }
    * 1:37 *
```

```
File Edit Search Run Compile Debug Project Options Window Help
IDRAK\3.C 1=[+]
        steps = dy;
    }
    dx = dx/steps;
    dy = dy/steps;
    x = x0;
    y = y0;
    i = 1;
    while(i<= steps)
    {
        putpixel(x, y, RED);
        x += dx;
        y += dy;
        i=i+1;
    }
    getch();
    closegraph();
}
* 40:1 *
```

Output: -

