NAME: VENNELA G

REG NO: 20BDS0146

SUBJECT: ADVANCED C PROGRAMMING

SLOT: F1

ASSESSMENT NO: 1

1. **Discuss the differences between the vector graphics and pixel graphics with an example.**

Pixel Graphics	Vector Graphics
→ They are composed of pixels	→ They are composed of paths
→ Refresh process is independent of the complexity of image	→ Vector displays flicker when the number of primitives in image becomes too large
→ Graphic primitives are specified in terms of end points & must be scan converted into corresponding pixels	→ Scan conversion is not required
→ Pixel graphics can draw mathematical curves, polygons & boundaries of curved primitives only by pixel approximation	→ Vector graphics draw continuous & smooth lines
→ Pixel graphics cost less	→ Vector graphics cost more than pixel graphics
→ They occupy more space which depends on image quality	→ They occupy less space
→ File extensions: • BMP, • TIF, • GIF, • JPG	→ File extensions: • SVG, • EPS, • PDF, • AI, • DXF

2. Write a Program in C to draw the various shapes like line, circle, rectangle, square, arc, semi-circle, pie chart and bar chart using Switch statement. (include an output snapshot for each cases)

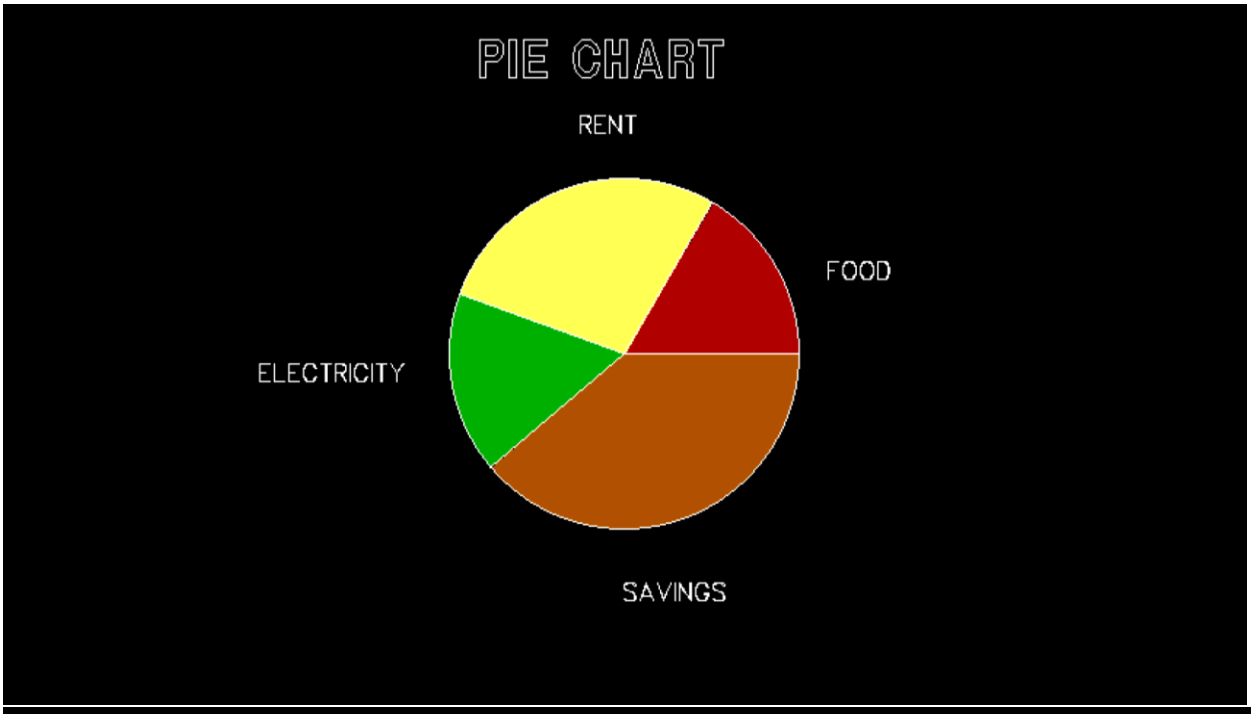
CODE:

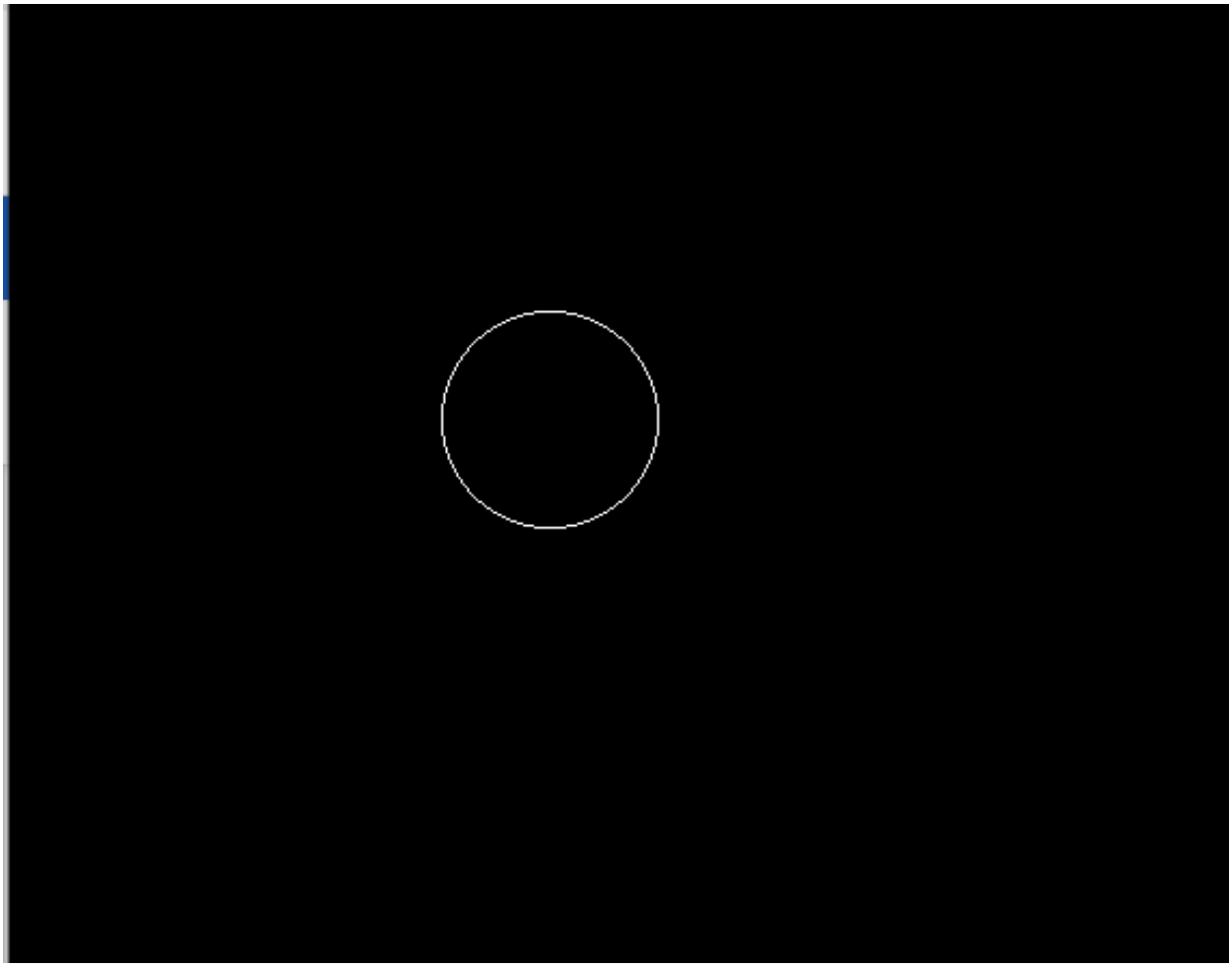
```
#include<graphics.h>
#include<conio.h>
#include<stdio.h>
int main()
{int k,userpattern=1,x_max,y_max,left=150,top=150,
right=450,bottom=450,x_start=100,y_start=120,x_end=140,y_e
nd=340,gd=DETECT,gm,x,y,i=0,j=5;scanf("%d",&k);
switch(k){case 1:  initgraph(&gd,&gm,"C:\\TURBOC3\\BGI");
settextstyle(BOLD_FONT,HORIZ_DIR,2);
outtextxy(220,10,"PIE CHART");
x=getmaxx()/2;
y=getmaxy()/2;
settextstyle(SANS_SERIF_FONT,HORIZ_DIR,1);
setfillstyle(SOLID_FILL,RED);
pieslice(x,y,0,60,120);
```

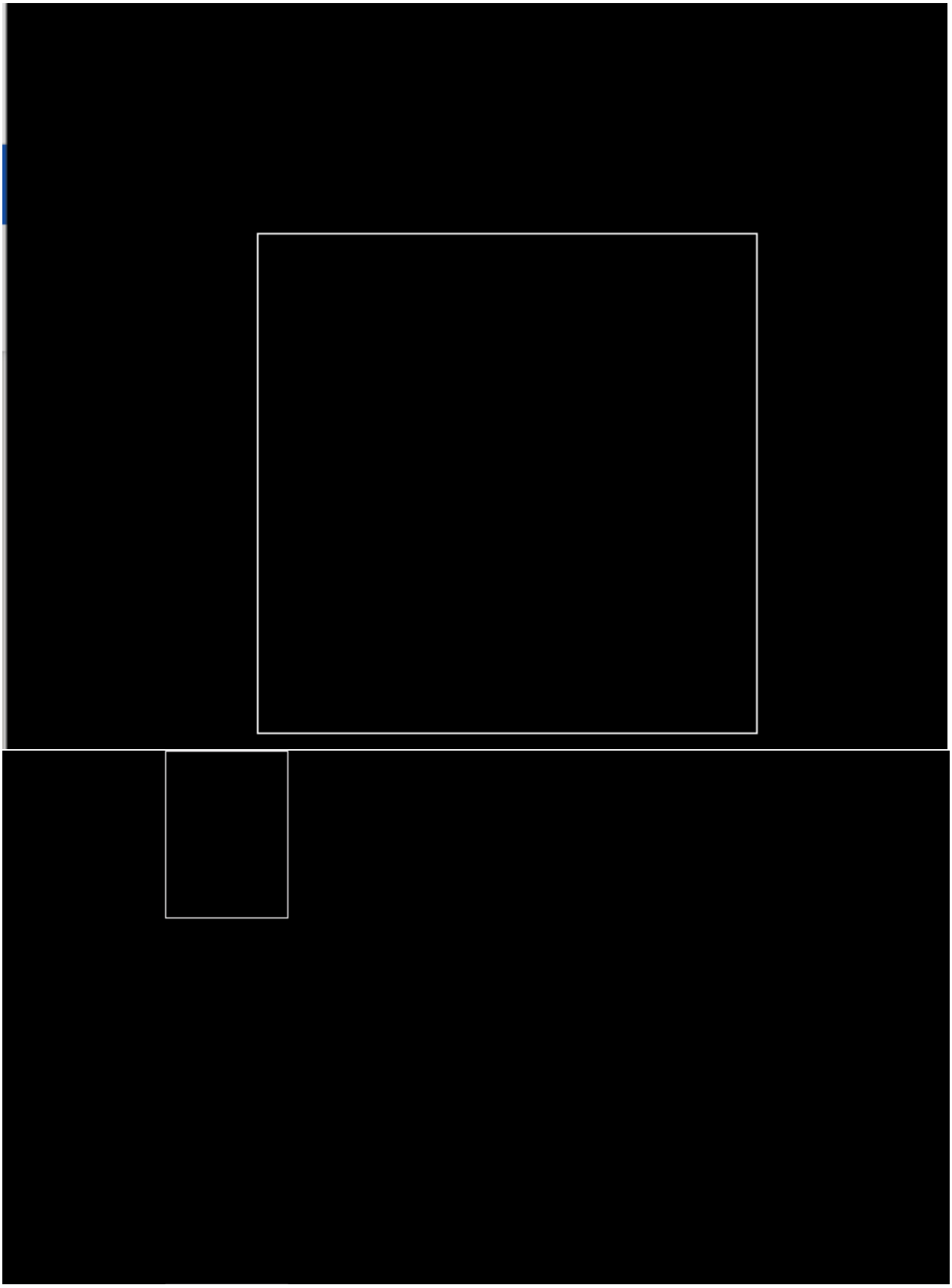
```
outtextxy(x+140,y-70,"FOOD");
setfillstyle(SOLID_FILL,YELLOW);
pieslice(x,y,60,160,120);
outtextxy(x-30,y-170,"RENT");
setfillstyle(SOLID_FILL,GREEN);
pieslice(x,y,160,220,120);
outtextxy(x-250,y,"ELECTRICITY");
setfillstyle(SOLID_FILL,BROWN);
pieslice(x,y,220,360,120);
outtextxy(x,y+150,"SAVINGS");break;
case 2:
      initgraph(&gd,&gm,"C:\\TURBOC3\\BGI");setlinestyle(DOTTED_LINE,userpattern,3);
x=100;y=70;x_max=350;y_max=70;
line(x,y,x_max,y_max);break;
case 3:initgraph(&gd,&gm,"C:\\TURBOC3\\BGI");circle(250,200,50);break;
case 4:initgraph(&gd,&gm,"C:\\TURBOC3\\BGI");rectangle(left,top,right,bottom);break;
```

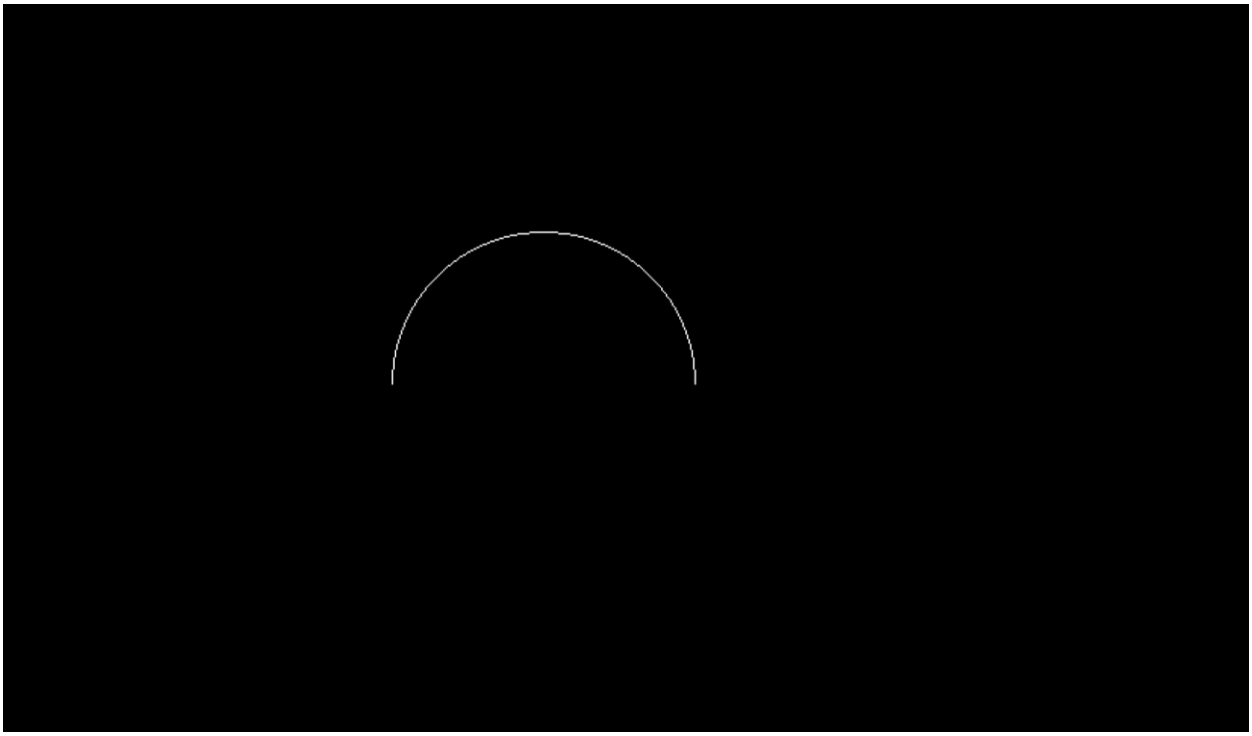
```
case
5:initgraph(&gd,&gm,"C:\\TURBOC3\\BGI");rectangle(150,150,
40,0);break;
case
6:initgraph(&gd,&gm,"C:\\TURBOC3\\BGI");arc(250,250,0,180,
100);break;
case
7:initgraph(&gd,&gm,"C:\\TURBOC3\\BGI");setfillstyle(SOLID_FI
LL,GREEN);
while(i<10)
{x_start=x_start+40+j;
x_end=x_end+40+j;
y_start=y_start+20;
bar(x_start,y_start,x_end,y_end);
i++;
}
break;
}
getch();
closegraph();
return 0;
}
```

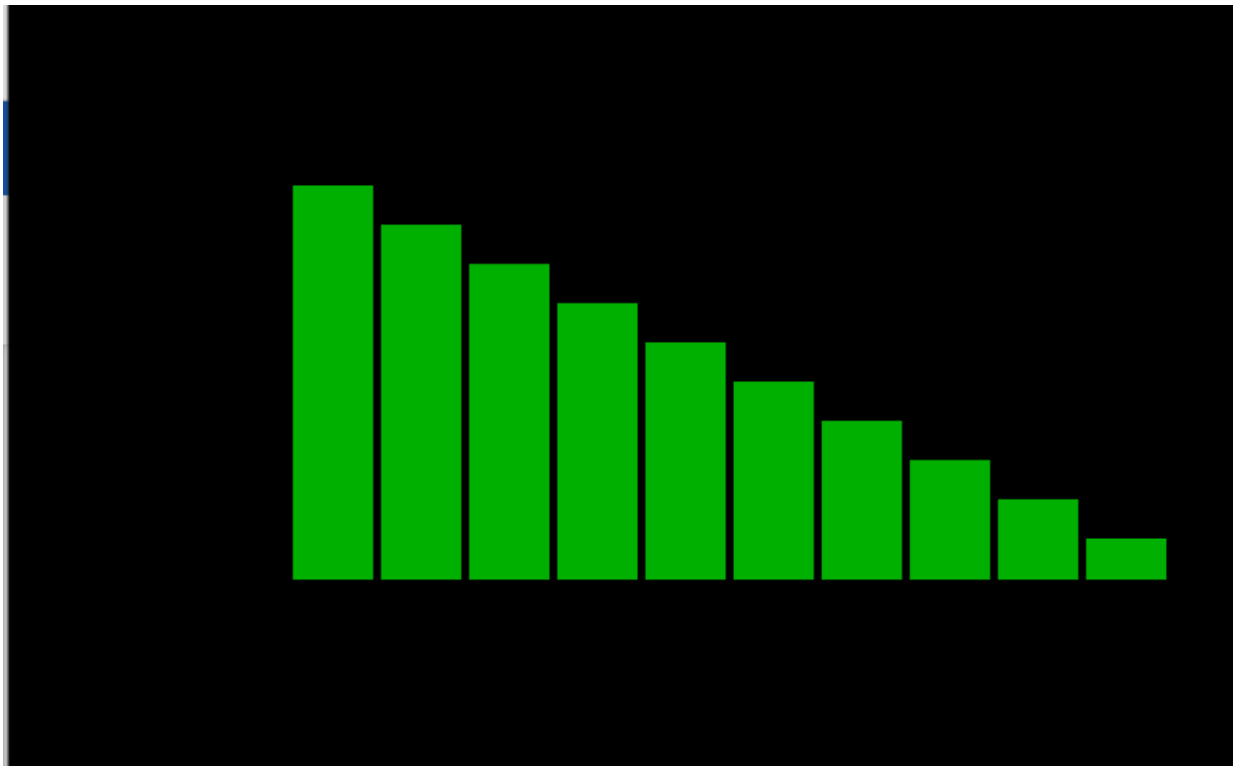
OUTPUT:











3. Using different graphics functions available for text formatting in C Language, write a C program for displaying text in different sizes, different colors and different font styles. (include the output snapshot for your registration number as text)

CODE:

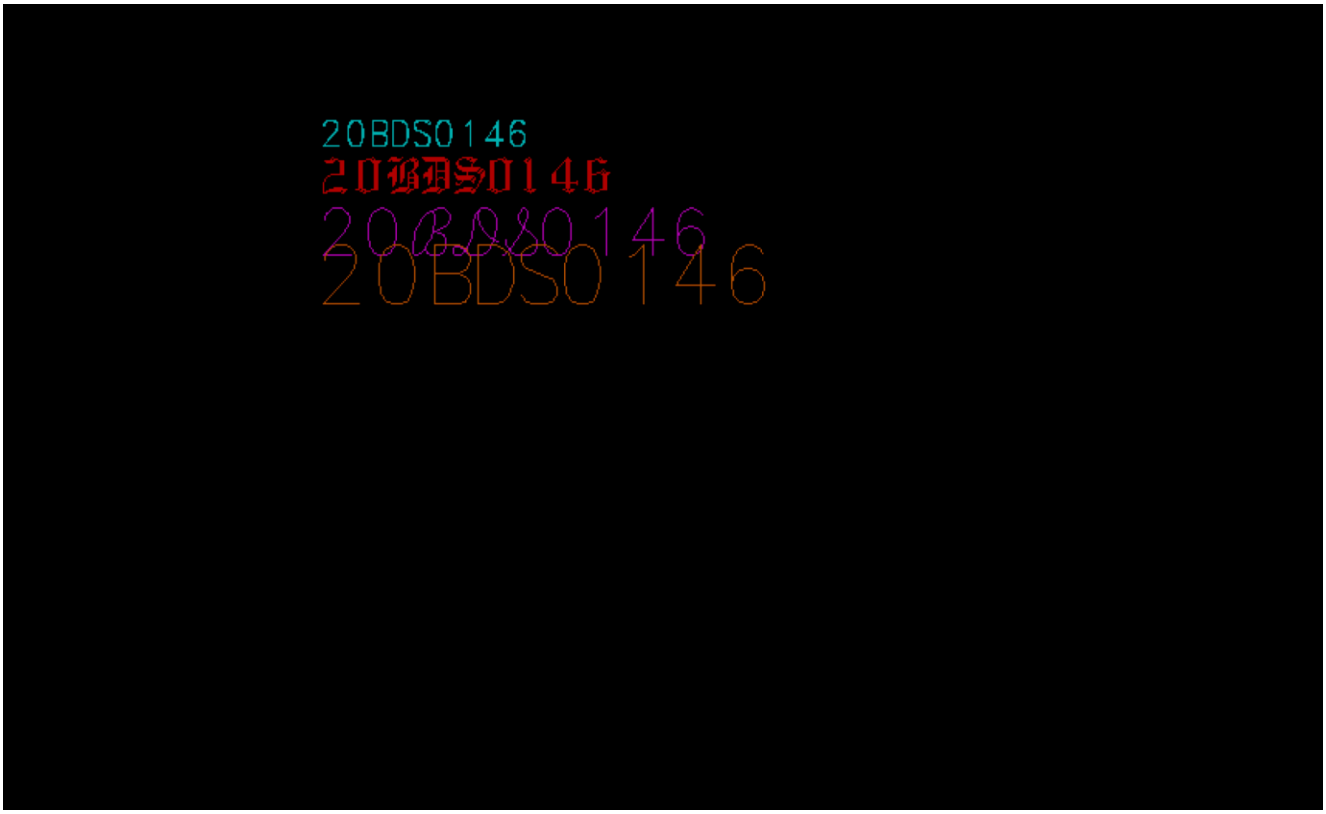
```
#include<stdio.h>
#include<graphics.h>
#include<dos.h>

void printMsg()
{
int gd=DETECT,gm,i;
initgraph(&gd,&gm,"C:\\TURBOC3\\BGI");
for(i=3;i<7;i++)
{
setcolor(i);
settextstyle(i,0,i);
```

```
outtextxy(100,20*i,"20BDS0146");
delay(500);
}
delay(2000);
}

int main()
{
printMsg();
return 0;
}
```

OUTPUT:



4. Different graphic Functions are available in C-Language for filling a given object with colors. Using the graphic functions, write a C program for filling various closed objects with different colors. . (include an output snapshot for each cases).

CODE:

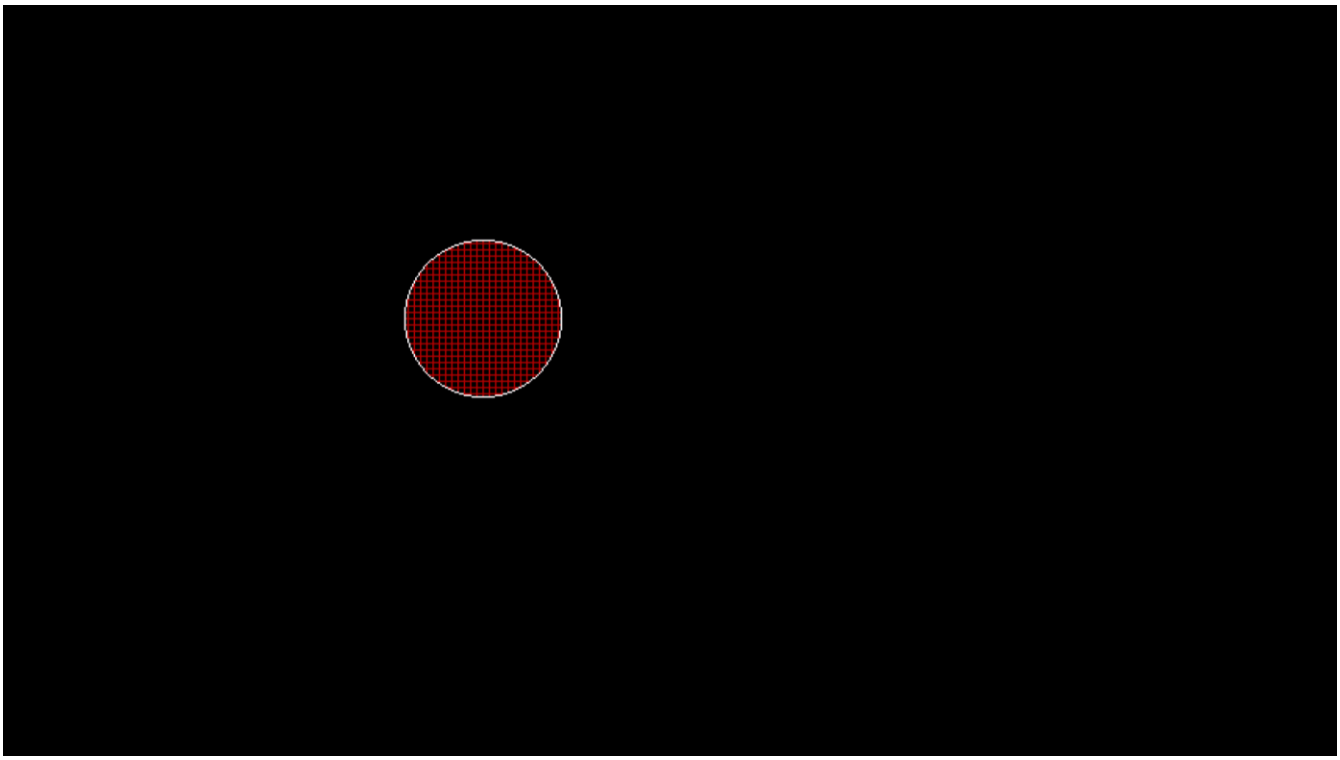
```
#include<graphics.h>


int main()
{
```

```
int gd=DETECT,gm;
int x=200;
int y=200;
int radius=50;
int
border_color=WHITE;initgraph(&gd,&gm,"C:\\TURBOC3\\BGI");
setfillstyle(HATCH_FILL,RED);
circle(x,y,radius);
floodfill(x,y,border_color);
getch();

closegraph();
initgraph(&gd,&gm,"C:\\TURBOC3\\BGI");
setfillstyle(HATCH_FILL,RED);
rectangle(200,200,450,450);
floodfill(x,y,border_color);
getch();
closegraph();
return 0;
}
```

OUTPUT:



5. Write a C program for simulating a traffic signal activity. Show the output snapshot for the different activity.

CODE:

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
void car(int x)
{
int y=350,r=25;
cleardevice();
```

```
line(x-249,y,x-219,y);
sector(x-169,y,0,180,50,50);
line(x-119,y,x-89,y);
circle(x-219,y+25,r);
circle(x-119,y+25,r);
}

int main()
{
int gd=DETECT,gm,x;
char ch;
initgraph(&gd,&gm,"C:\\TURBOC3\\BGI");
x=getmaxx();
ch='y';
while(ch!='e')
{
cleardevice();
rectangle(50,150,100,50);
line(75,150,75,450);
line(85,150,85,450);
circle(75,75,10);
circle(75,100,10);
```

```
circle(75,125,10);
car(x);
switch(ch)
{
case 'r':
setfillstyle(SOLID_FILL,RED);
circle(75,75,10);
floodfill(75,75,WHITE);
circle(75,100,10);
circle(75,125,10);
break;
case 'g':
setfillstyle(SOLID_FILL,GREEN);
circle(75,75,10);
circle(75,100,10);
floodfill(75,100,WHITE);
circle(75,125,10);
while(!kbhit())
{x--;
if(x<290)
{x=getmaxx();
}
```

```
car(x);
circle(75,75,10);
circle(75,100,10);
floodfill(75,100,WHITE);
circle(75,125,10);
delay(50);
}
break;

case 'y':
setfillstyle(SOLID_FILL,YELLOW);
circle(75,75,10);
circle(75,100,10);
circle(75,125,10);
floodfill(75,125,WHITE);
break;
default: printf("THANK YOU");
break;
}
delay(50);
ch=getch();
}
```

```
getch();
return 0;
}
```

OUTPUT: