**Q. Write a program for drawing line, circle, ellipse, arc, sector, bar, etc. using inbuilt functions.**

**Coding:**

#include<graphics.h>

void main()

{

int gd,gm,errorcode,ch,a,b,c,d,e,f;

gd=DETECT;

initgraph(&gd,&gm,"C:\\tc\\bgi");

errorcode=graphresult();

if(errorcode!=grOk)

{ printf("Graphics Error: %s\n",grapherrormsg(errorcode));

printf("Press any key to halt");

getch();

exit(1); }

setbkcolor(15); setcolor(2);

do

{ printf("Choose the option:\n1. Line\n2. Circle\n3. Ellipse\n4. Arc\n5. Sector\n6. Bar\n7.Exit\n");

scanf("%d",&ch);

switch(ch)

{ case 1:

printf("\nEnter the co-ordinates of first point.\n");

scanf("%d %d",&a,&b);

printf("\nEnter the co-ordinates of second point.\n");

scanf("%d %d",&c,&d);

line(a,b,c,d);

break;

case 2:

printf("\nEnter the co-ordinates of center.\n");

scanf("%d %d",&a,&b);

printf("\nEnter the radius.\n");

scanf("%d",&c);

circle(a,b,c);

break;

case 3:

printf("\nEnter the co-ordinates of center.\n");

scanf("%d %d",&a,&b);

printf("\nEnter the start angle and end angle.\n");

scanf("%d %d",&c,&d);

printf("\nEnter the x-radius and y-radius.\n");

scanf("%d %d",&e,&f);

ellipse(a,b,c,d,e,f);

break;

case 4:

printf("\nEnter the co-ordinates of center.\n");

scanf("%d %d",&a,&b);

printf("\nEnter the start angle and end angle.\n");

scanf("%d %d",&c,&d);

printf("\nEnter the radius.\n");

scanf("%d",&e);

arc(a,b,c,d,e);

break;

case 5:

printf("\nEnter the co-ordinates of center.\n");

scanf("%d %d",&a,&b);

printf("\nEnter the start angle and end angle.\n");

scanf("%d %d",&c,&d);

printf("\nEnter the x-radius and y-radius.\n");

scanf("%d %d",&e,&f);

sector(a,b,c,d,e,f);

break;

case 6:

printf("\nEnter value of left and right.\n");

scanf("%d %d",&a,&b);

printf("\nEnter value of top and bottom.\n");

scanf("%d %d",&c,&d);

bar(a,c,b,d);

break;

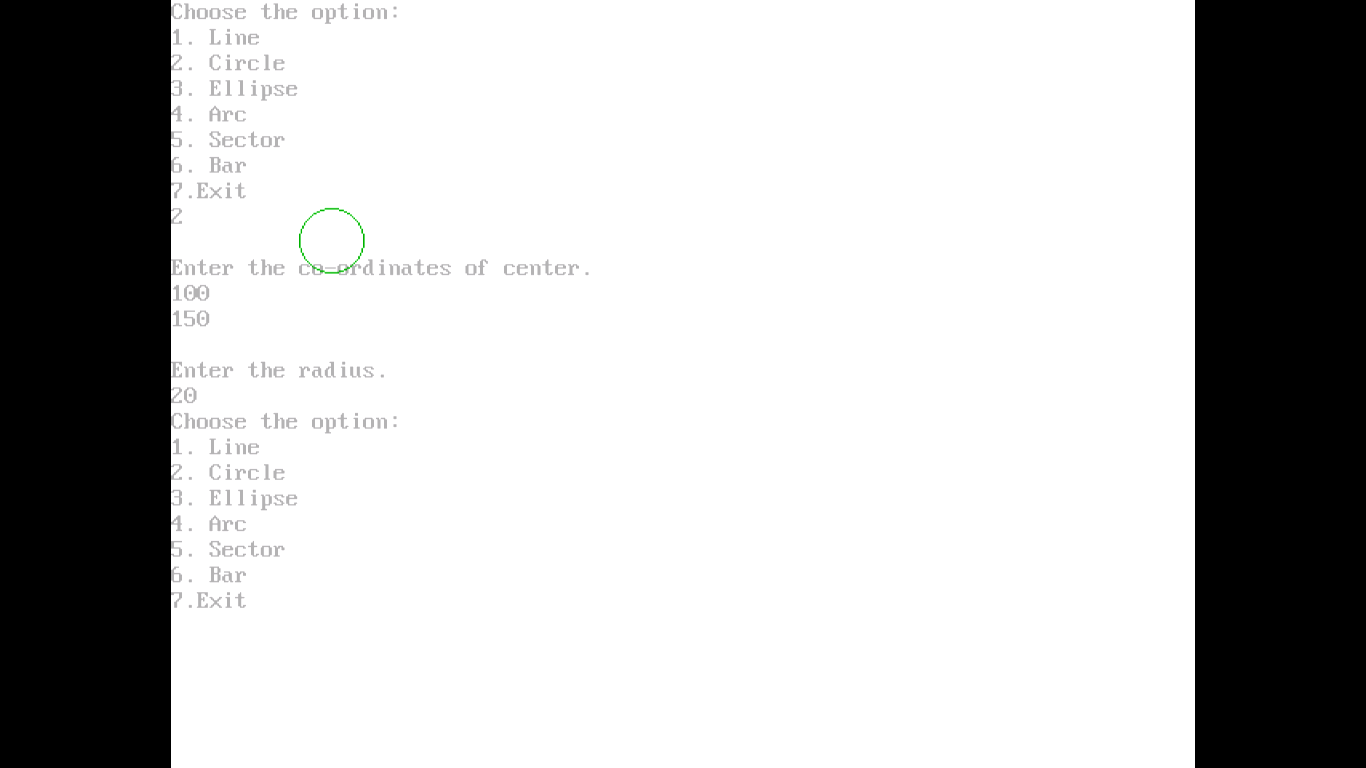
}

}while(ch<7);

getch();

closegraph();

}



**Q. Write a program for drawing nested circle, nested rectangle, etc. using inbuilt functions.**

**Coding:**

#include<graphics.h>

void main()

{ int gd,gm,errorcode,a,b,c,d,e,ch;

gd=DETECT;

initgraph(&gd,&gm,"C:\\tc\\bgi");

errorcode=graphresult();

if(errorcode!=grOk)

{ printf("Graphics Error: %s\n",grapherrormsg(errorcode));

printf("Press any key to hault.");

getch();

exit(0); }

setbkcolor(15);

setcolor(4);

do{ printf("Choose Option:\n1. Nested Rectangle\n2. Nested Arc\n3. Nested Circle\n4. Exit\n");

scanf("%d",&ch);

switch(ch)

{ case 1:

printf("Enter value of left, top, right, bottom.\n");

scanf("%d %d %d %d",&a,&b,&c,&d);

rectangle(a,b,c,d);

rectangle(a+5,b+5,c-5,d-5);

break;

case 2:

printf("Enter the co-ordinates of center.\n");

scanf("%d %d",&a,&b);

printf("Enter start angle and end angle.\n");

scanf("%d %d",&c,&d);

printf("Enter the radius.\n");

scanf("%d",&e);

arc(a,b,c,d,e);

arc(a,b,c,d,e-5);

break;

case 3:

printf("Enter the co-ordinates of center.\n");

scanf("%d %d",&a,&b);

printf("Enter the radius.\n");

scanf("%d",&c);

circle(a,b,c);

circle(a,b,c-5);

break;

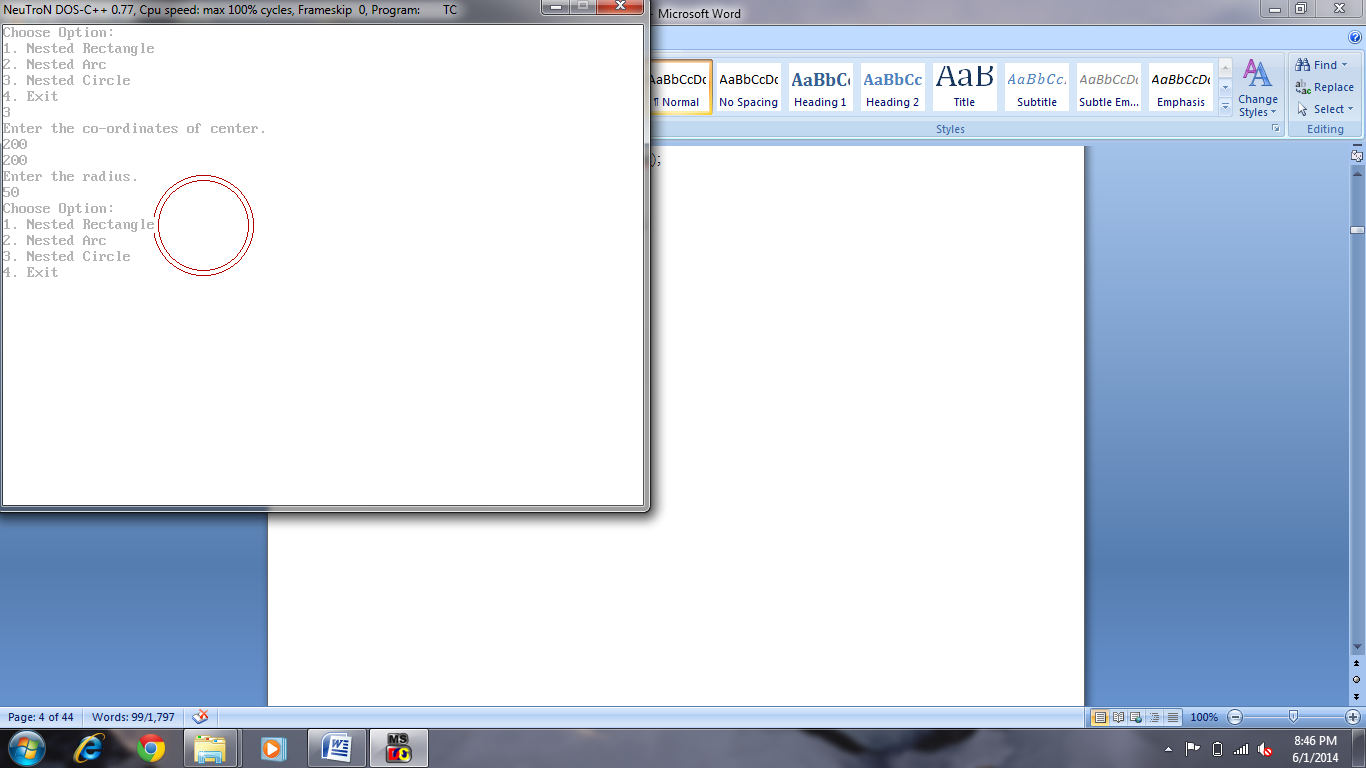
}

}while(ch<4);

getch();

closegraph();

}



**Q. Write a program for implementation of functions like setbkcolor(), setcolor(), setfillstyle, etc.**

**Coding:**

#include<graphics.h>

void main()

{ int gd,gm,errorcode;

gd=DETECT;

initgraph(&gd,&gm,"C:\\tc\\bgi");

errorcode=graphresult();

if(errorcode!=grOk)

{ printf("Graphics Error: %s\n",grapherrormsg(errorcode));

printf("Press any key to hault");

getch();

exit(0); }

setbkcolor(15);

setcolor(1);

setfillstyle(2,1);

rectangle(100,400,200,200);

rectangle(200,200,500,400);

rectangle(300,250,400,300);

rectangle(120,230,180,400);

line(100,200,150,100);

line(150,100,200,200);

line(150,100,450,100);

line(450,100,500,200);

circle(150,160,20);

floodfill(300,150,1);

setfillstyle(8,1);

floodfill(350,275,1);

setfillstyle(3,1);

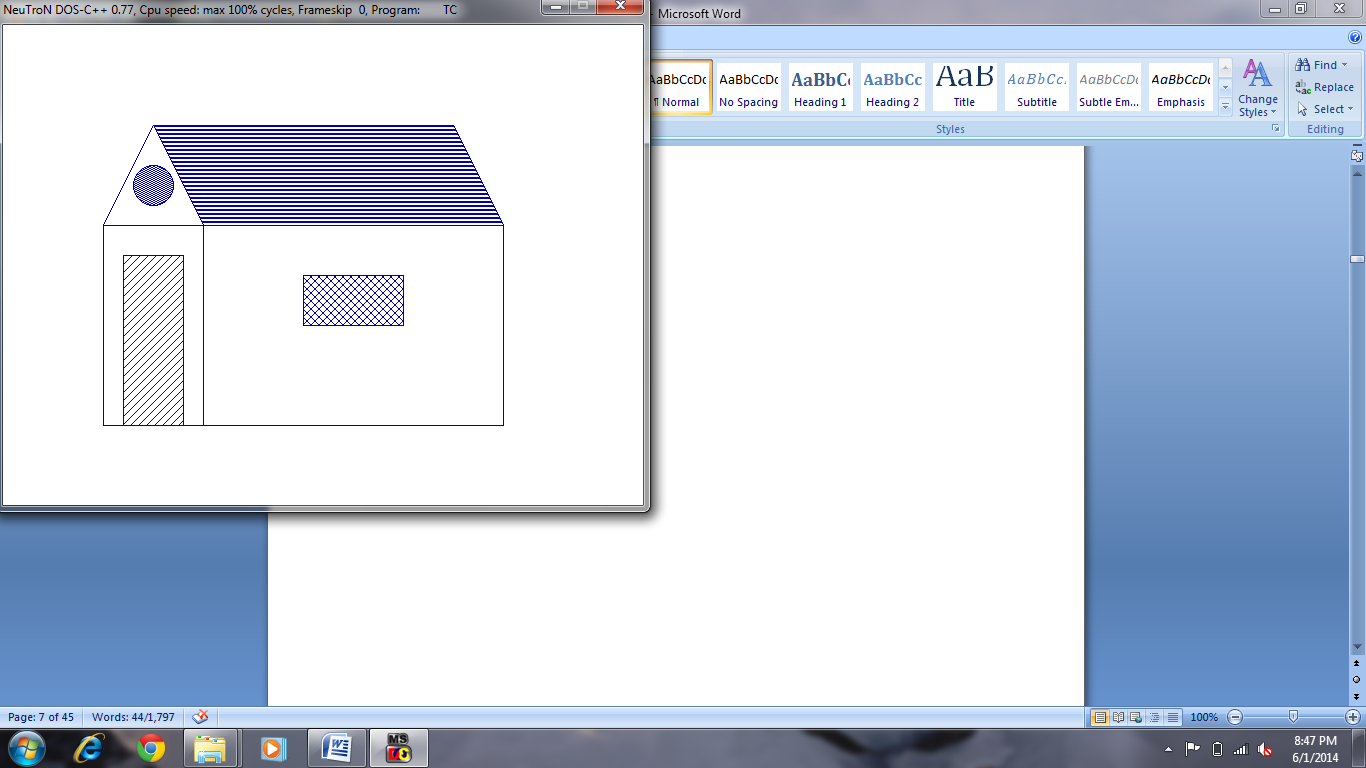
floodfill(150,250,1);

setfillstyle(9,1);

floodfill(160,170,1);

getch(); closegraph();

}



**Q. Write a program for scan conversion line using:**

* **DDA Algorithm**

**Coding:**

#include<graphics.h>

void main()

{ int gd,gm,errorcode,x1,x2,y1,y2,dx,dy,i;

float a,b,x,y,s;

gd=DETECT;

initgraph(&gd,&gm,"C:\\tc\\bgi");

errorcode=graphresult();

if(errorcode!=grOk)

{ printf("Graphics Error: %s\n",grapherrormsg(errorcode));

printf("Press any key to hault.");

getch();

exit(0); }

setbkcolor(15);

printf("Enter the co-ordinates of first point.\n");

scanf("%d %d",&x1,&y1);

printf("\nEnter the co-ordinates of second point.\n");

scanf("%d %d",&x2,&y2);

dx=x2-x1;

dy=y2-y1;

if(dx>=dy)

s=dx;

else

s=dy;

a=dx/s;

b=dy/s;

x=x1;

y=y1;

putpixel(x,y,5);

for(i=0;i<=s;i++)

{ x=x+a;

y=y+b;

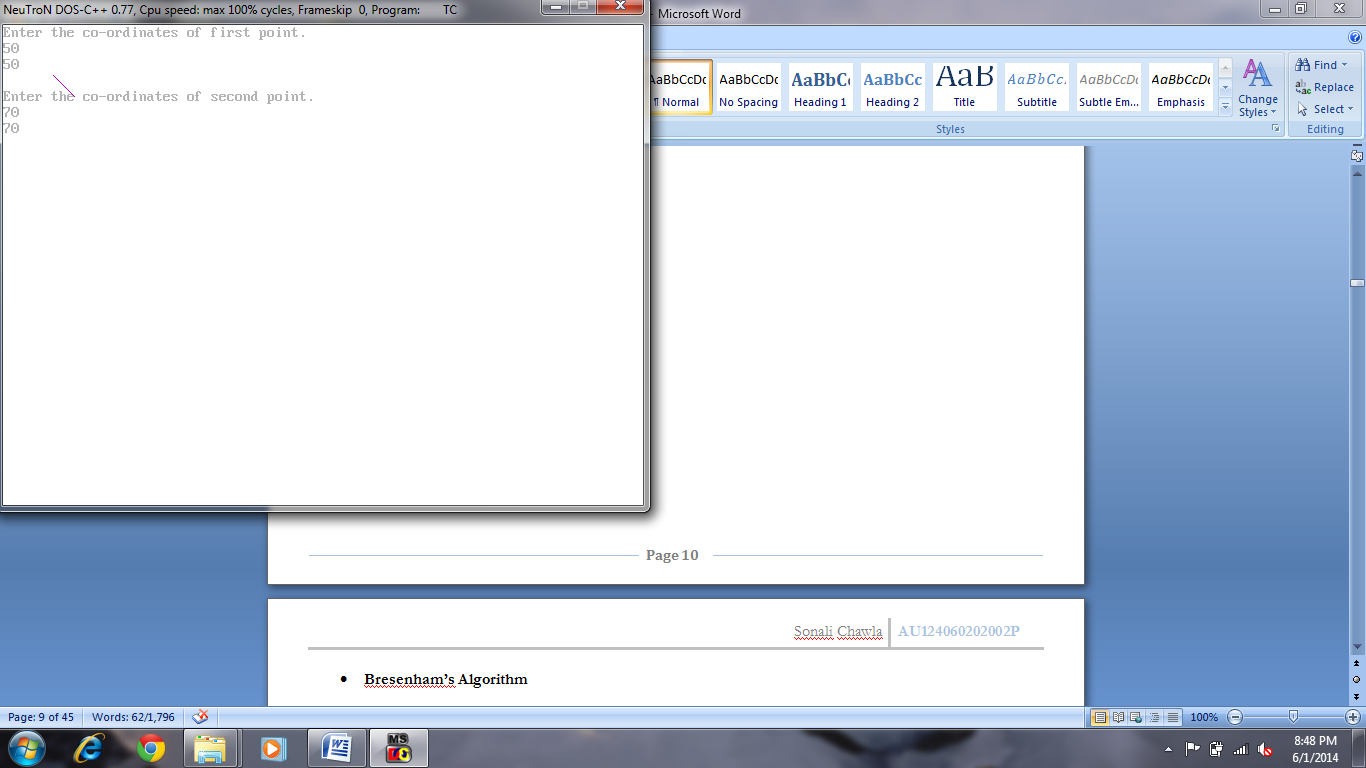
putpixel(x,y,5);

delay(30); }

getch();

closegraph();

}



* **Bresenham’s Algorithm**

**Coding:**

#include<graphics.h>

void main()

{ int gd,gm,errorcode,x,y,x1,y1,x2,y2,dx,dy,p,i;

gd=DETECT;

initgraph(&gd,&gm,"C:\\tc\\bgi");

errorcode=graphresult();

if(errorcode!=grOk)

{ printf("Graphics Error: %s\n",grapherrormsg(errorcode));

printf("Press any key to hault.");

getch();

exit(1); }

setbkcolor(15);

printf("Enter the start coordinates.\n");

scanf("%d %d",&x1,&y1);

printf("\nEnter the end coordinates.\n");

scanf("%d %d",&x2,&y2);

dx=x2-x1;

dy=y2-y1;

p=2\*dy-dx;

i=2\*dy;

x=x1;

y=y1;

do

{ putpixel(x,y,3);

delay(30);

x=x+1;

if(p<0)

p=p+i;

else

{ p=p+i-(2\*dx);

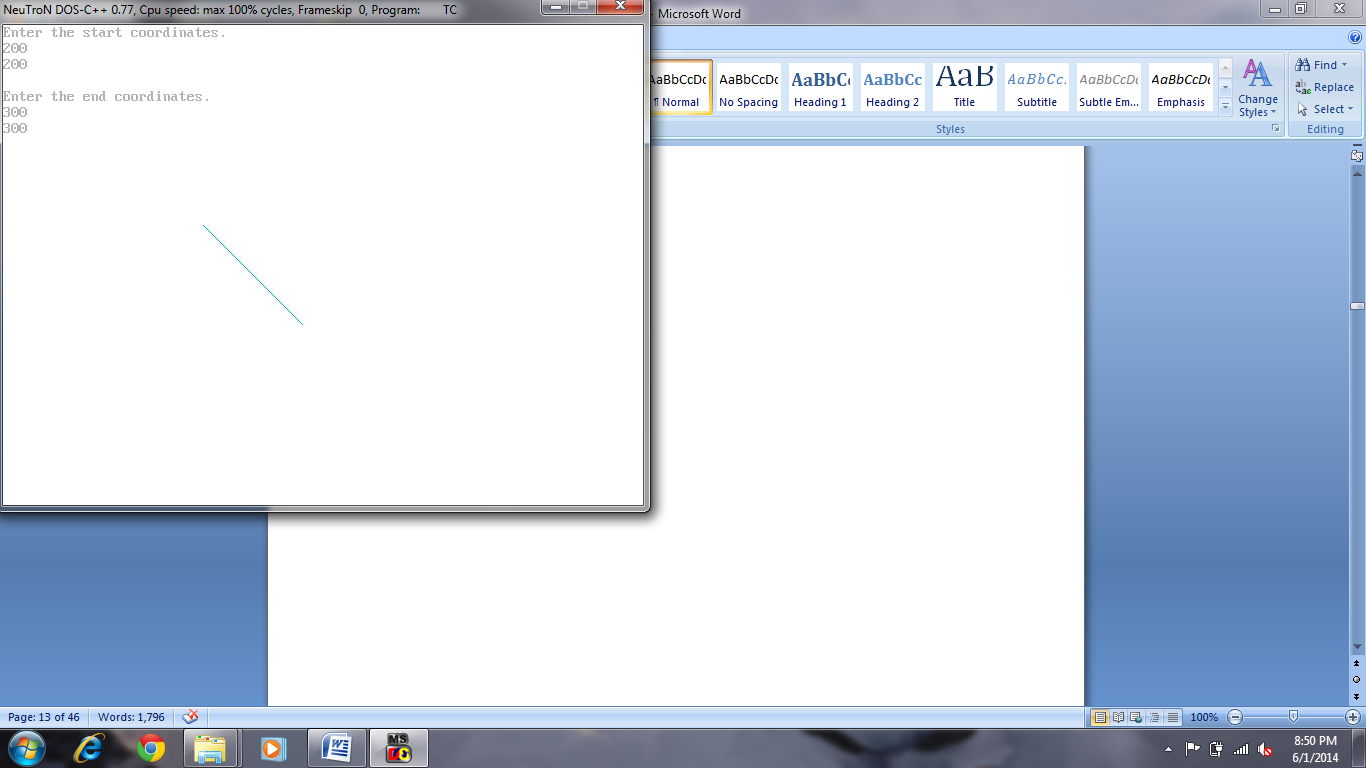
y=y+1; }

}while(x<x2);

getch();

closegraph();

}



**Q. Write a program for scan converting circle using:**

* **Polynomial Method**

**Coding:**

#include<graphics.h>

#include<math.h>

void main()

{ int gd=DETECT,gm,errorcode;

int r,h,k;

float xend,x=0,y=0,x1,y1,z;

initgraph(&gd,&gm,"C:\\tc\\bgi");

errorcode=graphresult();

if(errorcode!=grOk)

{ printf("Graphics Error: %s\n",grapherrormsg(errorcode));

printf("Press any key to hault");

getch();

exit(0); }

setbkcolor(15);

printf("Enter the center of circle\n");

scanf("%d %d", &h,&k);

printf("\nEnter the radius of the circle\n");

scanf("%d",&r);

xend=r/1.414;

while(x<=xend)

{ x++;

z=((r\*r)-(x\*x));

y=sqrt(z);

putpixel(x+h,y+k,4);

putpixel(y+h,x+k,4);

putpixel(-y+h,x+k,4);

putpixel(-x+h,y+k,4);

putpixel(-x+h,-y+k,4);

putpixel(-y+h,-x+k,4);

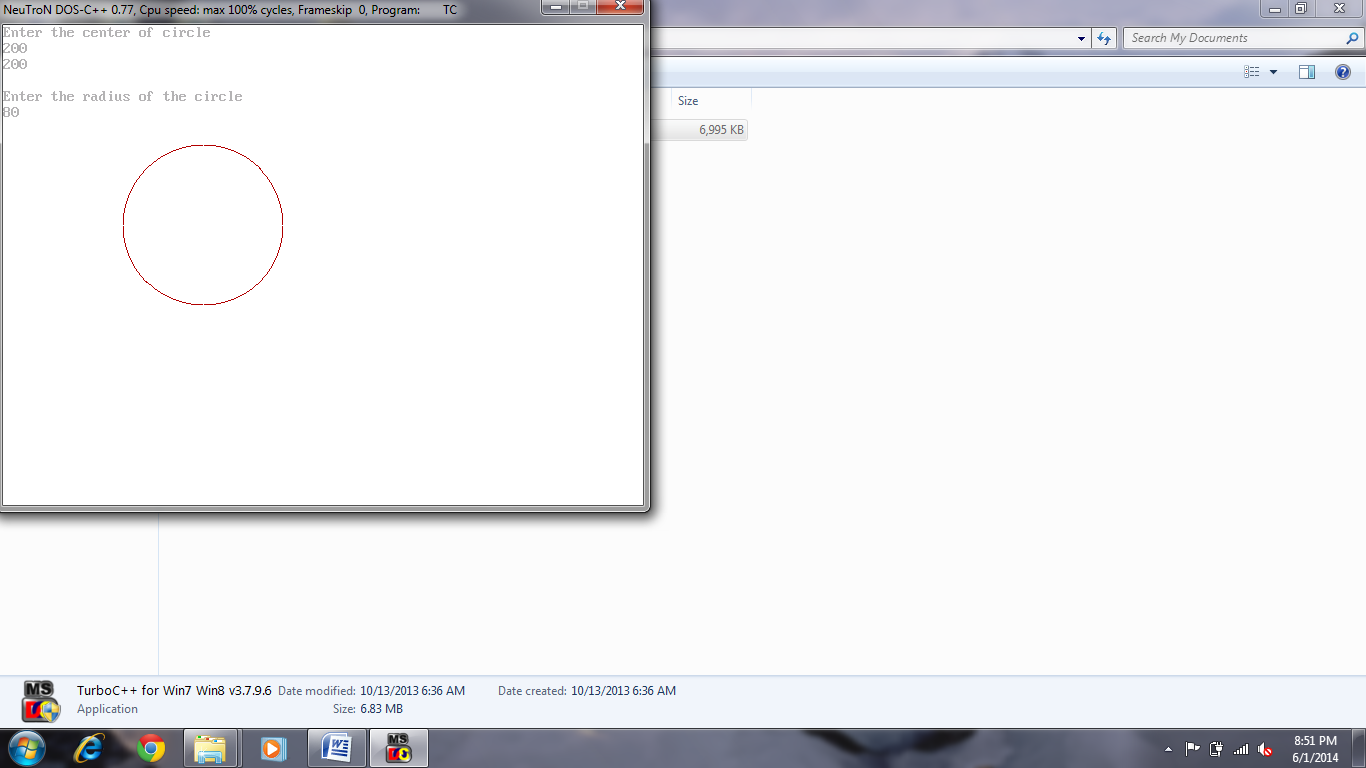
putpixel(y+h,-x+k,4);

putpixel(x+h,-y+k,4);

}getch();

closegraph();

}



* **Trigonometric Method**

**Coding:**

#include<graphics.h>

#include<math.h>

void main()

{ int gd=DETECT,gm,errorcode;

int r,h,k;

float xend,x,y,temp,tinc,t,tend,a;

initgraph(&gd,&gm,"C:\\tc\\bgi");

errorcode=graphresult();

if(errorcode!=grOk)

{ printf("Graphics Error: %s\n",grapherrormsg(errorcode));

printf("Press any key to hault.");

getch();

exit(1); }

setbkcolor(15);

printf("Enter the center of circle\n");

scanf("%d %d", &h,&k);

printf("\nEnter the radius of the circle\n");

scanf("%d",&r);

temp=3.141/180;

tinc=1.0/r;

t=0;

tend=45;

while(t<=tend)

{ a=temp\*t;

x=r\*cos(a);

y=r\*sin(a);

putpixel(x+h,y+k,9);

putpixel(y+h,x+k,9);

putpixel(-y+h,x+k,9);

putpixel(-x+h,y+k,9);

putpixel(-x+h,-y+k,9);

putpixel(-y+h,-x+k,9);

putpixel(y+h,-x+k,9);

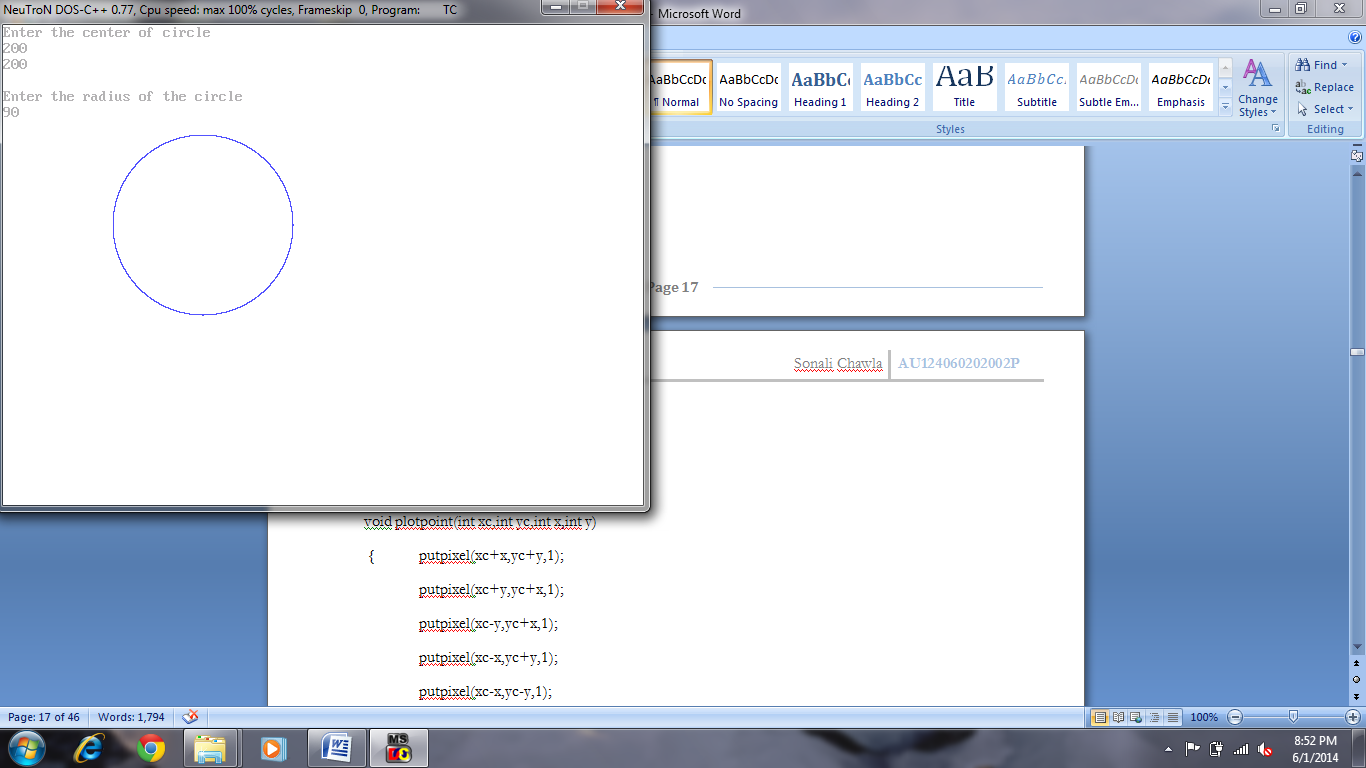
putpixel(x+h,-y+k,9);

t=t+tinc;

}getch();

closegraph();

}



* **Mid-point Method**

**Coding:**

#include<graphics.h>

void plotpoint(int xc,int yc,int x,int y)

{ putpixel(xc+x,yc+y,1);

putpixel(xc+y,yc+x,1);

putpixel(xc-y,yc+x,1);

putpixel(xc-x,yc+y,1);

putpixel(xc-x,yc-y,1);

putpixel(xc-y,yc-x,1);

putpixel(xc+y,yc-x,1);

putpixel(xc+x,yc-y,1);

delay(20);

}

void main()

{ int gd=DETECT,gm,xc=300,errorcode,yc=250,r,x,y,Pk;

initgraph(&gd,&gm,"C:\\tc\\bgi");

errorcode=graphresult();

if(errorcode!=grOk)

{ printf("Graphics Error: %s\n",grapherrormsg(errorcode));

printf("Press any key to hault");

getch();

exit(1); }

setbkcolor(15);

printf("Enter the Radius\n");

scanf("%d",&r);

x=0;

y=r;

Pk=1-r;

plotpoint(xc,yc,x,y);

while(x<y)

{ if(Pk<0)

{ x=x+1;

Pk=Pk+(2\*x)+1; }

else

{ x=x+1;

y=y-1;

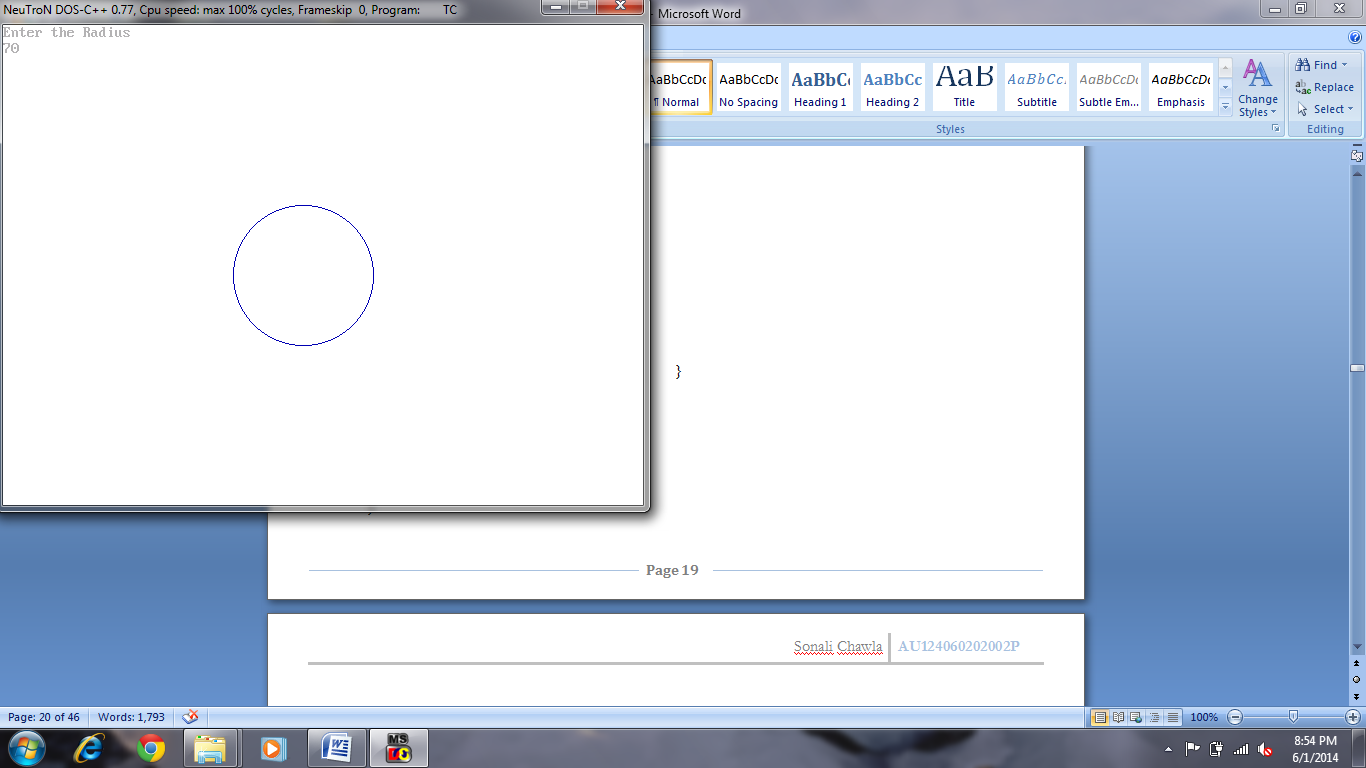
Pk=Pk+(2\*x)-(2\*y)+1; }

plotpoint(xc,yc,x,y);

} getch();

closegraph();

}



* **Bresenham’s Method**

**Coding:**

# include<graphics.h>

void main()

{ int gd,gm,errorcode,r,x,y,p,xc=320,yc=240;

gd=DETECT;

initgraph(&gd,&gm,"C:\\TC\\BGI");

errorcode=graphresult();

if(errorcode!=grOk)

{ printf("Graphics Error: %s\n",grapherrormsg(errorcode));

printf("Press any key to hault.");

getch();

exit(1); }

setbkcolor(15);

printf("Enter the radius\n");

scanf("%d",&r);

x=0;

y=r;

putpixel(xc+x,yc-y,6);

delay(20);

p=3-(2\*r);

for(x=0;x<=y;x++)

{ if(p<0)

p=(p+(4\*x)+6);

else

{ y=y-1;

p=p+((4\*(x-y)+10)); }

putpixel(xc+x,yc-y,6);

putpixel(xc-x,yc-y,6);

putpixel(xc+x,yc+y,6);

putpixel(xc-x,yc+y,6);

putpixel(xc+y,yc-x,6);

putpixel(xc-y,yc-x,6);

putpixel(xc+y,yc+x,6);

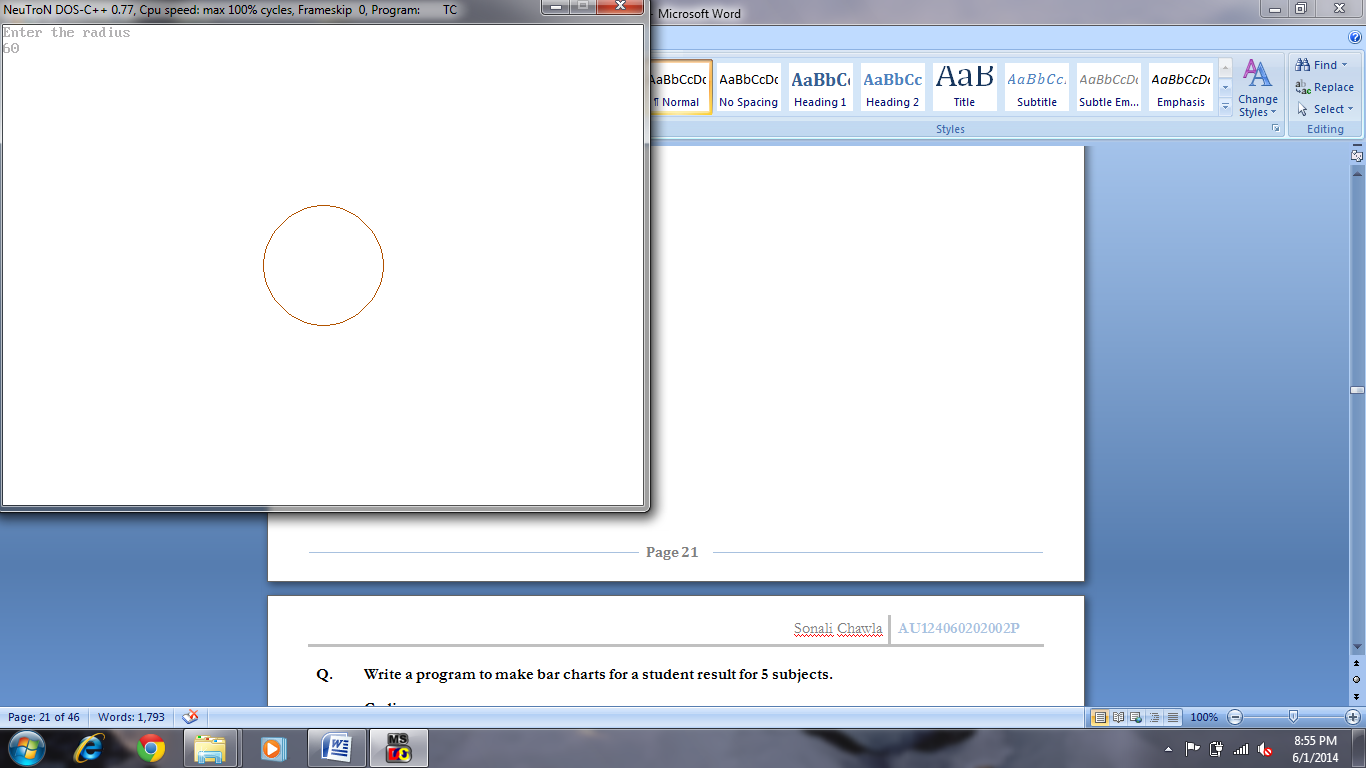
putpixel(xc-y,yc+x,6);

delay(20);

} getch();

closegraph();

}



**Q. Write a program to make bar charts for a student result for 5 subjects.**

**Coding:**

#include<graphics.h>

#include<stdlib.h>

void main()

{ int gd,gm,errorcode,a[5],i,d[5],u[5],y1,cnt=0;

char b[5];

gd=DETECT;

initgraph(&gd,&gm,"c:\\tc\\bgi");

errorcode = graphresult();

if (errorcode != grOk)

{ printf("Graphics error: %s\n", grapherrormsg(errorcode));

printf("Press any key to halt:");

getch();

exit(1); }

setbkcolor(15);

setcolor(5);

printf("Enter 5 marks:\n");

for(i=0;i<5;i++)

{ d[i]=1;

printf("Marks %d:",(i+1));

scanf("%d",&a[i]);

u[i]=460-(a[i]\*5); }

y1=460;

while(1)

{ if(d[0]){line(100,y1,150,y1);}

if(d[1]){line(200,y1,250,y1);}

if(d[2]){line(300,y1,350,y1);}

if(d[3]){line(400,y1,450,y1);}

if(d[4]){line(500,y1,550,y1);}

delay(50);

for(i=0;i<5;i++)

{ if(u[i]==y1)

{ outtextxy((100\*(i+1)),(u[i]-15),itoa(a[i],b,10));

d[i]=0;

cnt++; }

}

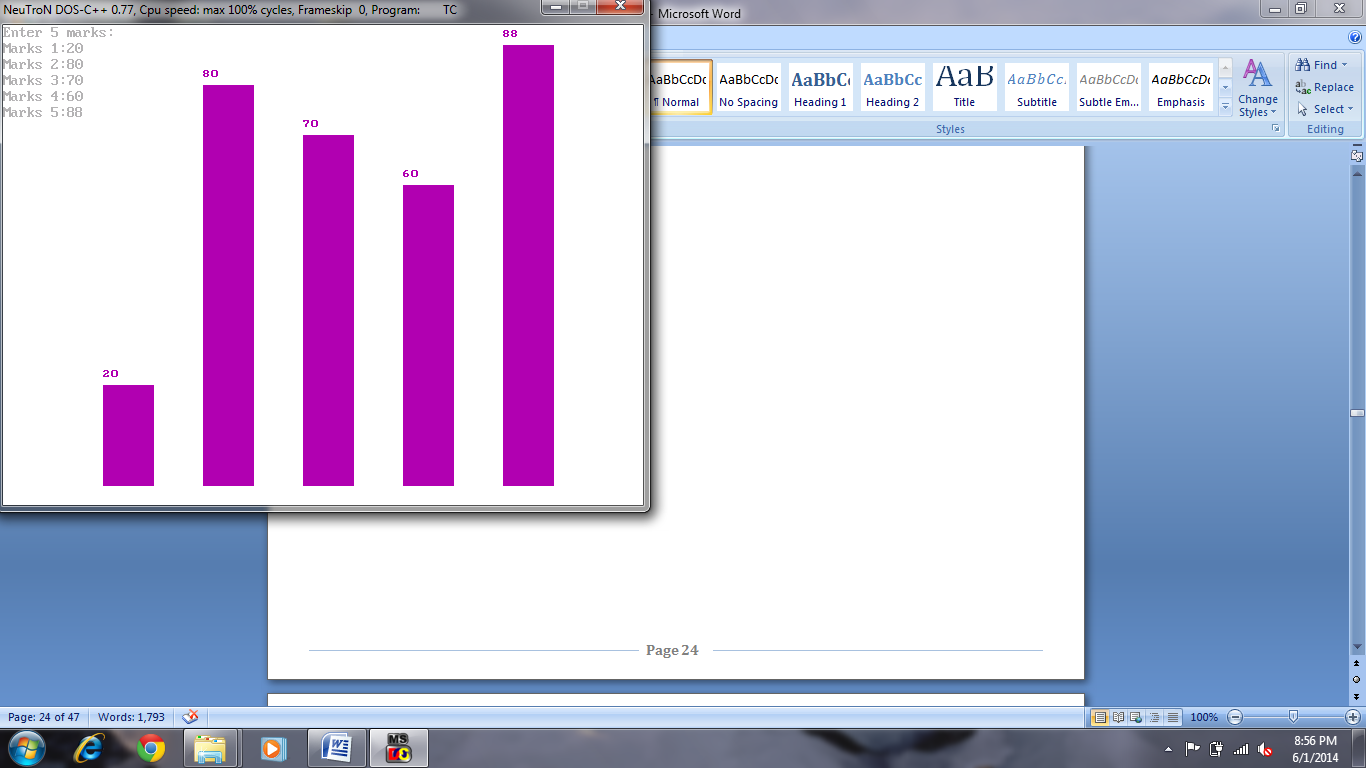
y1--;

if(cnt==5){break;}

} getch();

closegraph();

}



**Q. Write a program to make a pie cart for a student’s result for 5 subjects.**

**Coding:**

#include<graphics.h>

#include<stdlib.h>

void main()

{ int gd,gm,errorcode,i,a[5],u[5],s=0,c=0,y=220;

char d[10];

gd=DETECT;

initgraph(&gd,&gm,"C:\\tc\\bgi");

errorcode = graphresult();

if (errorcode != grOk)

{ printf("Graphics error: %s\n", grapherrormsg(errorcode));

printf("Press any key to halt:");

getch();

exit(1); }

setbkcolor(15);

setcolor(5);

printf("Enter 5 marks of student.\n");

for(i=0;i<5;i++)

{ printf("\nMarks %d: ",(i+1));

scanf("%d",&a[i]);

s=s+a[i]; }

for(i=0;i<5;i++)

{ u[i]=(a[i]\*360)/s;

setfillstyle((i+1),(i+1));

pieslice(250,250,c,(u[i]+c),100);

fillellipse(500,y,10,10);

outtextxy(515,y,itoa(a[i],d,10));

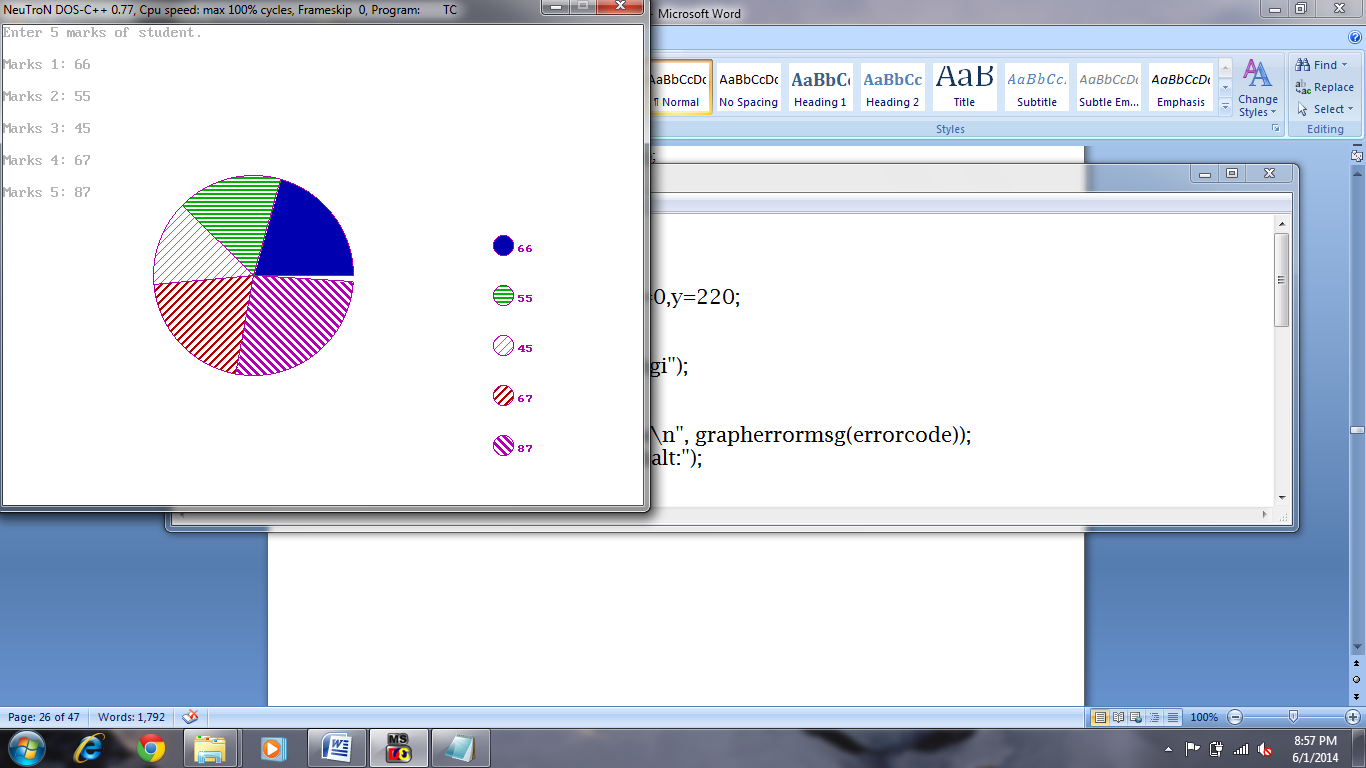
y=y+50;

c=c+u[i]; }

getch();

closegraph();

}



**Q. Write a program to get the translation vector from the user and translate the triangle accordingly.**

**Coding:**

#include<graphics.h>

void main()

{ int gd,gm,errorcode,a[8],i,x,y;

gd=DETECT;

initgraph(&gd,&gm,"C:\\tc\\bgi");

errorcode=graphresult();

if(errorcode!=grOk)

{ printf("Graphics Error: %s\n",grapherrormsg(errorcode));

printf("Press any key to hault.");

getch();

exit(1);

}

setbkcolor(15);

setcolor(1);

printf("Enter coordinates for points of triangle.\n");

for(i=0;i<6;i++)

{ scanf("%d",&a[i]);

if(i==0)

a[6]=a[0];

else if(i==1)

a[7]=a[1]; }

drawpoly(4,a);

printf("\n Enter the translation factor for x and y.\n");

scanf("%d %d",&x,&y);

for(i=0;i<8;i++)

{ if(i%2==0)

a[i]=a[i]+x;

else

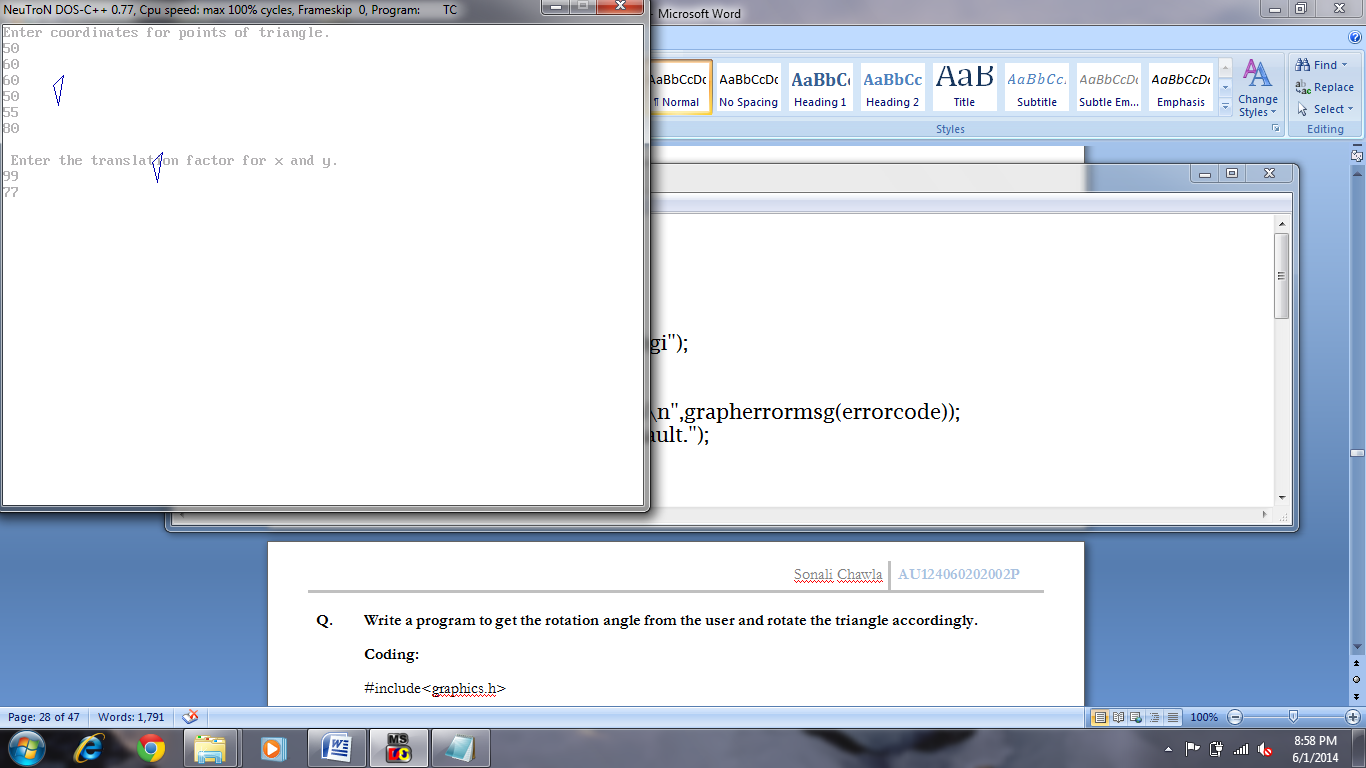
a[i]=a[i]+y; }

drawpoly(4,a);

getch();

closegraph();

}



**Q. Write a program to get the rotation angle from the user and rotate the triangle accordingly.**

**Coding:**

#include<graphics.h>

#include<math.h>

void main()

{ int gd=DETECT,gm,errorcode;

int x1,x2,x3,y1,y2,y3,tx,ty,ch;

float a;

initgraph(&gd,&gm,"C:\\tc\\bgi");

errorcode=graphresult();

if(errorcode!=grOk)

{ printf("Graphics Error: %s\n",grapherrormsg(errorcode));

printf("Press any key to hault");

getch();

exit(1);

}

setbkcolor(15);

setcolor(6);

printf("Enter coordinates of the triangle\n");

scanf("%d%d%d%d%d%d",&x1,&y1,&x2,&y2,&x3,&y3);

line(x1,y1,x2,y2);

line(x2,y2,x3,y3);

line(x3,y3,x1,y1);

printf("\nEnter the angle of rotation\n");

scanf("%f",&a);

a=(a\*3.14)/180;

x1=x1\*cos(a)-y1\*sin(a);

y1=x1\*sin(a)+y1\*cos(a);

x2=x2\*cos(a)-y2\*sin(a);

y2=x2\*sin(a)+y2\*cos(a);

x3=x3\*cos(a)-y3\*sin(a);

y3=x3\*sin(a)+y3\*cos(a);

setcolor(3);

line(x1,y1,x2,y2);

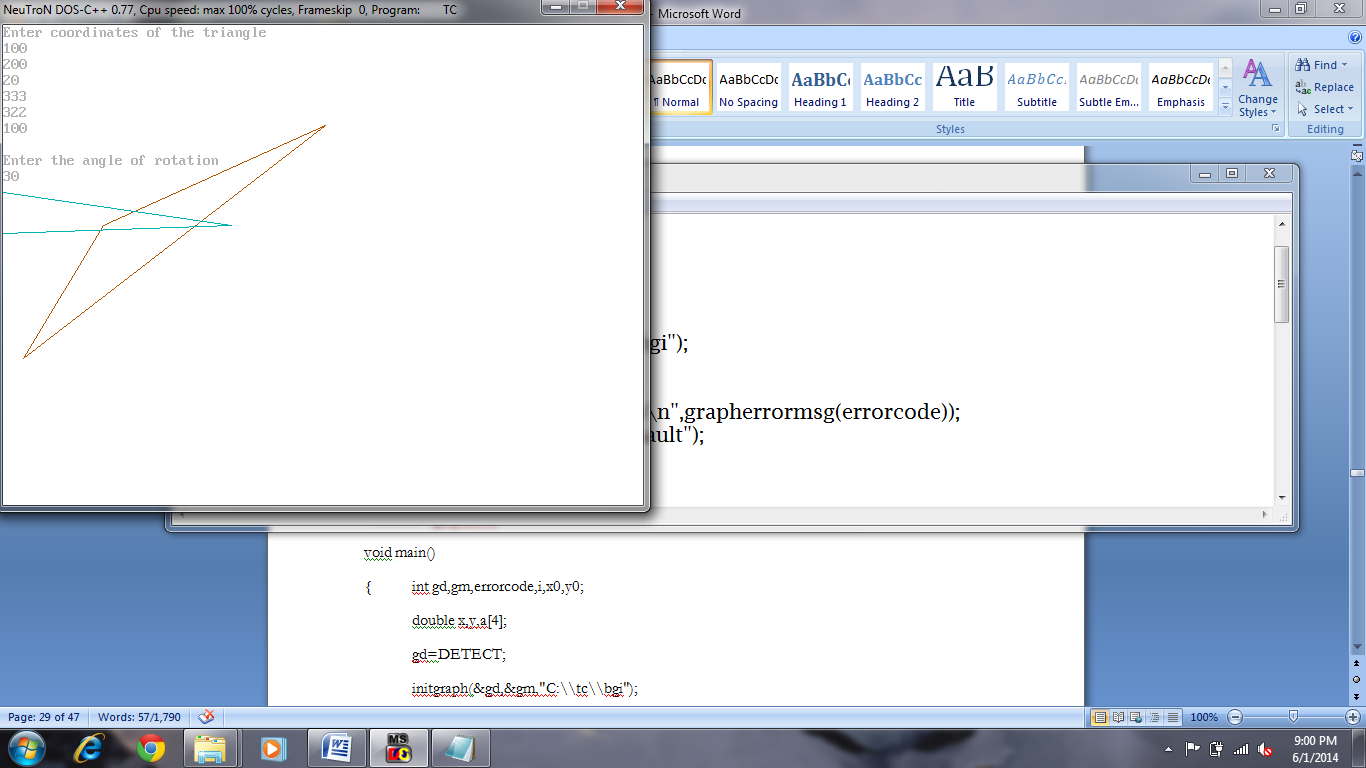
line(x2,y2,x3,y3);

line(x3,y3,x1,y1);

getch();

closegraph();

}



**Q. Write a program to get the scaling factor from the user and scale the triangle accordingly.**

**Coding:**

#include<graphics.h>

void main()

{ int gd,gm,errorcode,i,x0,y0;

double x,y,a[4];

gd=DETECT;

initgraph(&gd,&gm,"C:\\tc\\bgi");

errorcode=graphresult();

if(errorcode!=grOk)

{ printf("Graphics Error: %s\n",grapherrormsg(errorcode));

printf("Press any key to hault.");

getch();

exit(1); }

setbkcolor(15);

setcolor(1);

printf("Enter the coordinates of rectangle.\n");

for(i=0;i<4;i++)

scanf("%lf",&a[i]);

rectangle(a[0],a[1],a[2],a[3]);

x0=(a[0]+a[2])/2;

y0=(a[1]+a[3])/2;

printf("\nEnter scaling factor for x and y.\n");

scanf("%lf %lf",&x,&y);

a[0]=x0-(x\*(x0-a[0]));

a[1]=y0-(y\*(y0-a[1]));

a[2]=x0+(x\*(a[2]-x0));

a[3]=y0+(y\*(a[3]-y0));

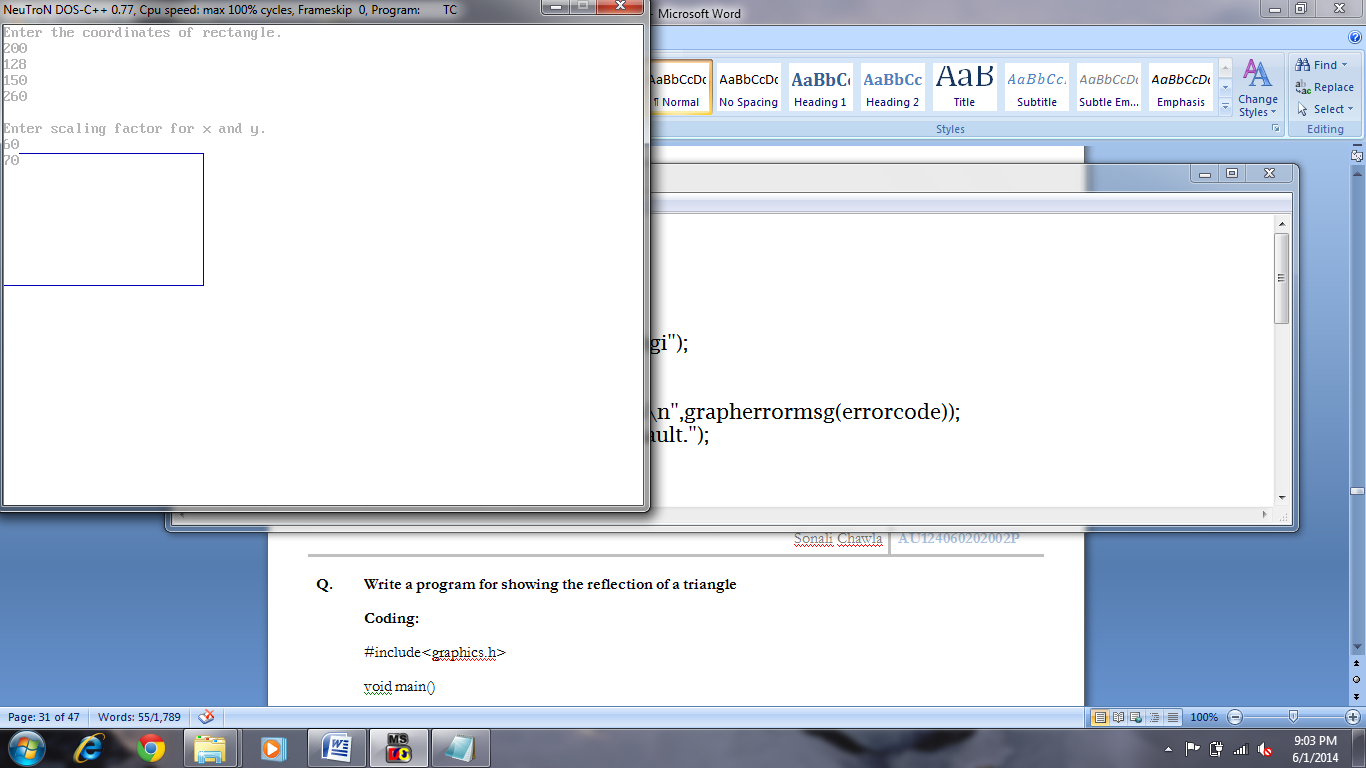
setcolor(3);

rectangle(a[0],a[3],a[2],a[1]);

getch();

closegraph();

}



**Q. Write a program for showing the reflection of a triangle**

**Coding:**

#include<graphics.h>

void main()

{ int gd=DETECT,gm,errorcode;

int x1,x2,x3,y1,y2,y3,xn1,xn2,xn3,yn1,yn2,yn3,c;

float a;

initgraph(&gd,&gm,"C:\\tc\\bgi");

errorcode=graphresult();

if(errorcode!=grOk)

{ printf("Graphics Error: %s\n",grapherrormsg(errorcode));

printf("Press any key to hault.");

getch();

exit(1); }

setbkcolor(15);

setcolor(10);

printf("Enter the coordinates of triangle\n");

scanf("%d%d%d%d%d%d",&x1,&y1,&x2,&y2,&x3,&y3);

line(x1,y1,x2,y2);

line(x2,y2,x3,y3);

line(x3,y3,x1,y1);

printf("Enter the axis of reflection\n");

printf("\n1.about x axis");

printf("\n2.about y axis\n");

scanf("%d",&c);

line(x1,y1,x2,y2);

line(x2,y2,x3,y3);

line(x3,y3,x1,y1);

if(c==1)

{ xn1=x1; yn1=180-y1;

xn2=x2; yn2=180-y2;

xn3=x3; yn3=180-y3; }

if(c==2)

{ xn1=180-x1; yn1=y1;

xn2=180-x2; yn2=y2;

xn3=180-x3; yn3=y3; }

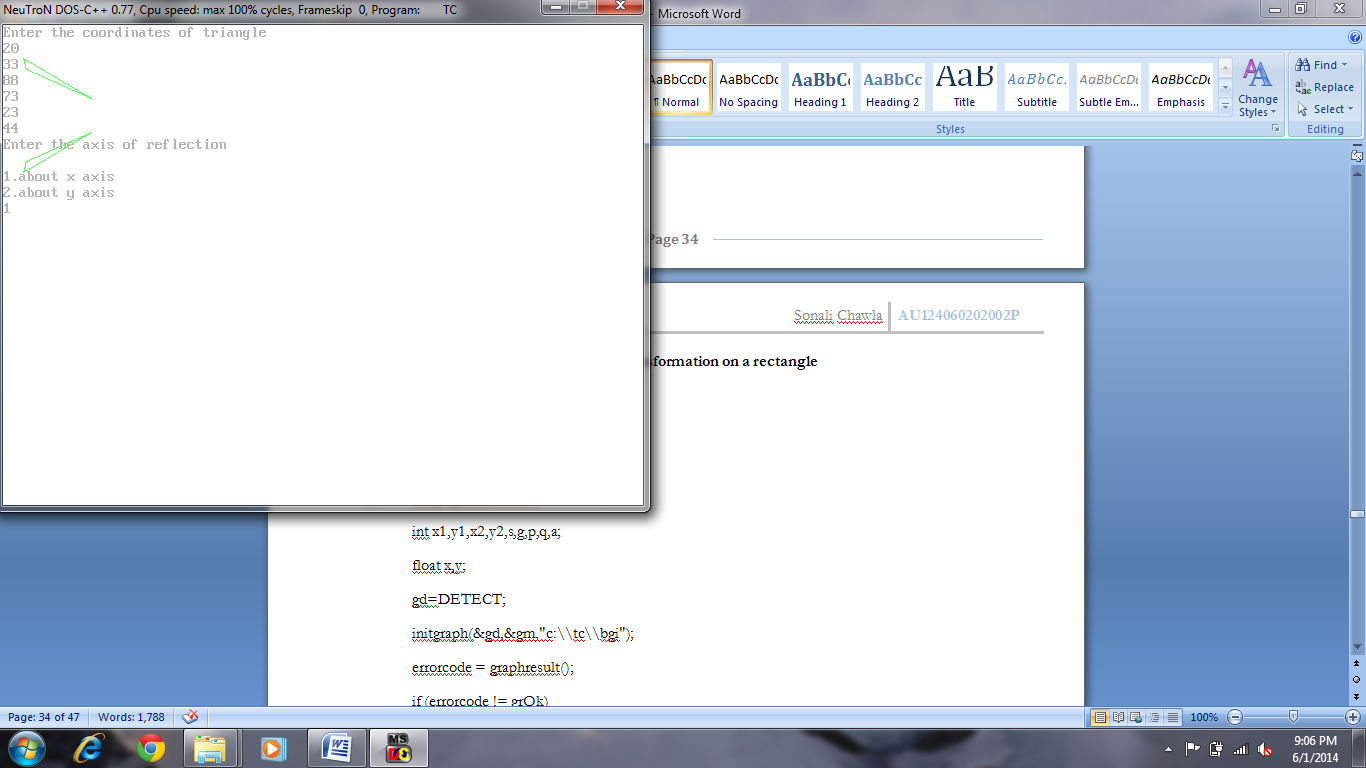
line(xn1,yn1,xn2,yn2);

line(xn2,yn2,xn3,yn3);

line(xn3,yn3,xn1,yn1);

getch(); closegraph();

}



**Q. Write a program to perform shearing transformation on a rectangle**

**Coding:**

#include<graphics.h>

void main()

{ int gd,gm,errorcode;

int x1,y1,x2,y2,s,g,p,q,a;

float x,y;

gd=DETECT;

initgraph(&gd,&gm,"c:\\tc\\bgi");

errorcode = graphresult();

if (errorcode != grOk)

{ printf("Graphics error: %s\n", grapherrormsg(errorcode));

printf("Press any key to halt:");

getch();

exit(1); }

setbkcolor(15);

setcolor(1);

printf("Enter the coordinates\n");

scanf("%d%d%d%d",&x1,&y1,&x2,&y2);

line(x1,y1,x2,y1);

line(x2,y2,x1,y2) ;

line(x2,y1,x2,y2);

line(x1,y2,x1,y1);

printf("\nEnter the shearing factor along the x-axis\n");

scanf("%f",&x);

printf("\nEnter trhe scaling factor to the y-axis\n");

scanf("%f",&y);

a=((x\*y1)+0.5);

s=x1+a;

a=((y\*x1)+0.5);

p=y1+a;

a=((x\*y2)+0.5);

g=x2+a;

a=((y\*x2)+0.5);

q=y2+a;

setcolor(2);

line(x1,y1,x2,p);

line(g,q,s,y2) ;

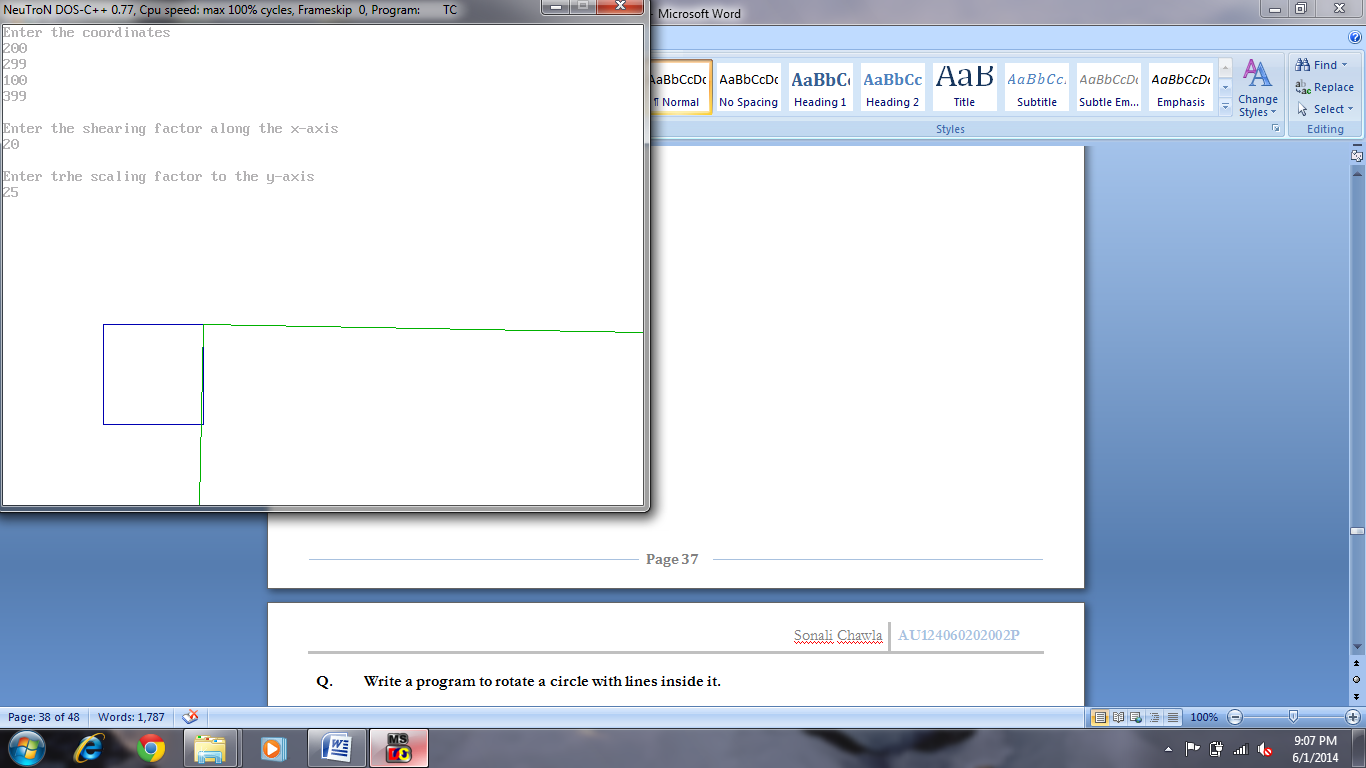
line(x2,p,g,q);

line(s,y2,x1,y1);

getch();

closegraph();

}



**Q. Write a program to rotate a circle with lines inside it.**

**Coding:**

#include<graphics.h>

#include<math.h>

void main()

{ int gd,gm,errorcode,x1,a=1;

gd=DETECT;

initgraph(&gd,&gm,"C:\\tc\\bgi");

errorcode=graphresult();

if(errorcode!=grOk)

{ printf("Graphics Error: %s\n",grapherrormsg(errorcode));

printf("Press any key to hault");

getch();

exit(1); }

setbkcolor(15);

setcolor(5);

x1=100;

do

{ cleardevice();

line(50,300,550,300);

circle(x1,250,50);

if(a%2==0)

line(x1,200,x1,300);

else

line(x1-50,250,x1+50,250);

delay(300);

x1=x1+50;

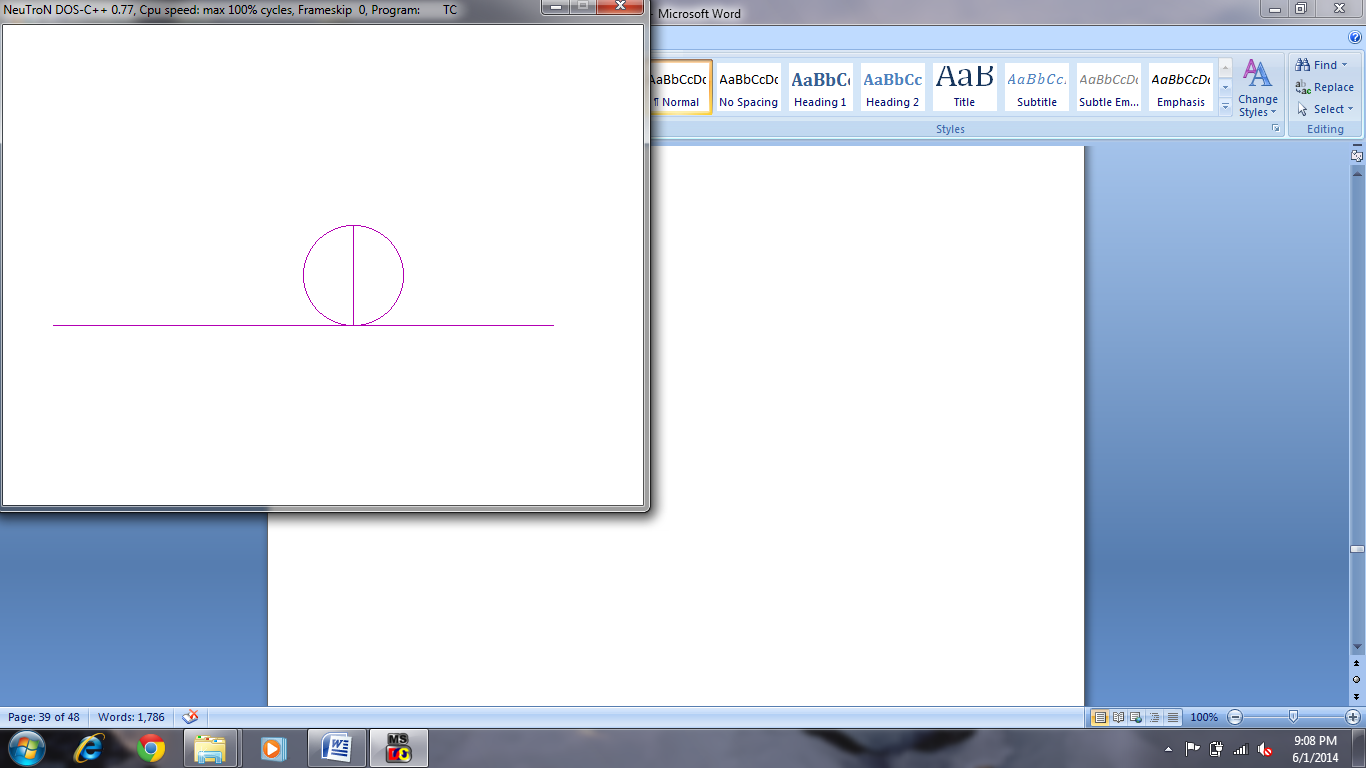
a++;

}while(a<10);

getch();

closegraph();

}



**Q. Write a program to switch on and off the fan. This program includes a fan having switches to increase the speed and switch off the fan.**

**Coding:**

#include<graphics.h>

void main()

{ int gd,gm,errorcode,s=0,e=45,c,r=50,x=200,y=300,sp,d=1000;

gd=DETECT;

initgraph(&gd,&gm,"c:\\tc\\bgi");

errorcode = graphresult();

if (errorcode != grOk)

{ printf("Graphics error: %s\n", grapherrormsg(errorcode));

printf("Press any key to halt:");

getch(); exit(1); }

setbkcolor(15); setcolor(4);

pieslice(x,y,s,e,r);

pieslice(x,y,s+135,e+135,r);

pieslice(x,y,s+245,e+255,r);

printf("Press down arrow key to switch off the fan");

printf("\nPress space bar to increase speed of the fan");

delay(1000);

while(!kbhit())

{ s=0; e=45;

for(c=0;c<20;c++)

{ pieslice(x,y,s,e,r);

pieslice(x,y,s+135,e+135,r);

pieslice(x,y,s+245,e+255,r);

s=s+2; e=e+2;

delay(d);

if((s>=360)||(e>=360))

break;

if(kbhit())

{ sp=getch();

if(sp==32)

d=500;

if(sp==80)

{ outtextxy(170,400,"fan stopped");

pieslice(x,y,s,e,r);

pieslice(x,y,s+135,e+135,r);

pieslice(x,y,s+245,e+255,r);

break; }

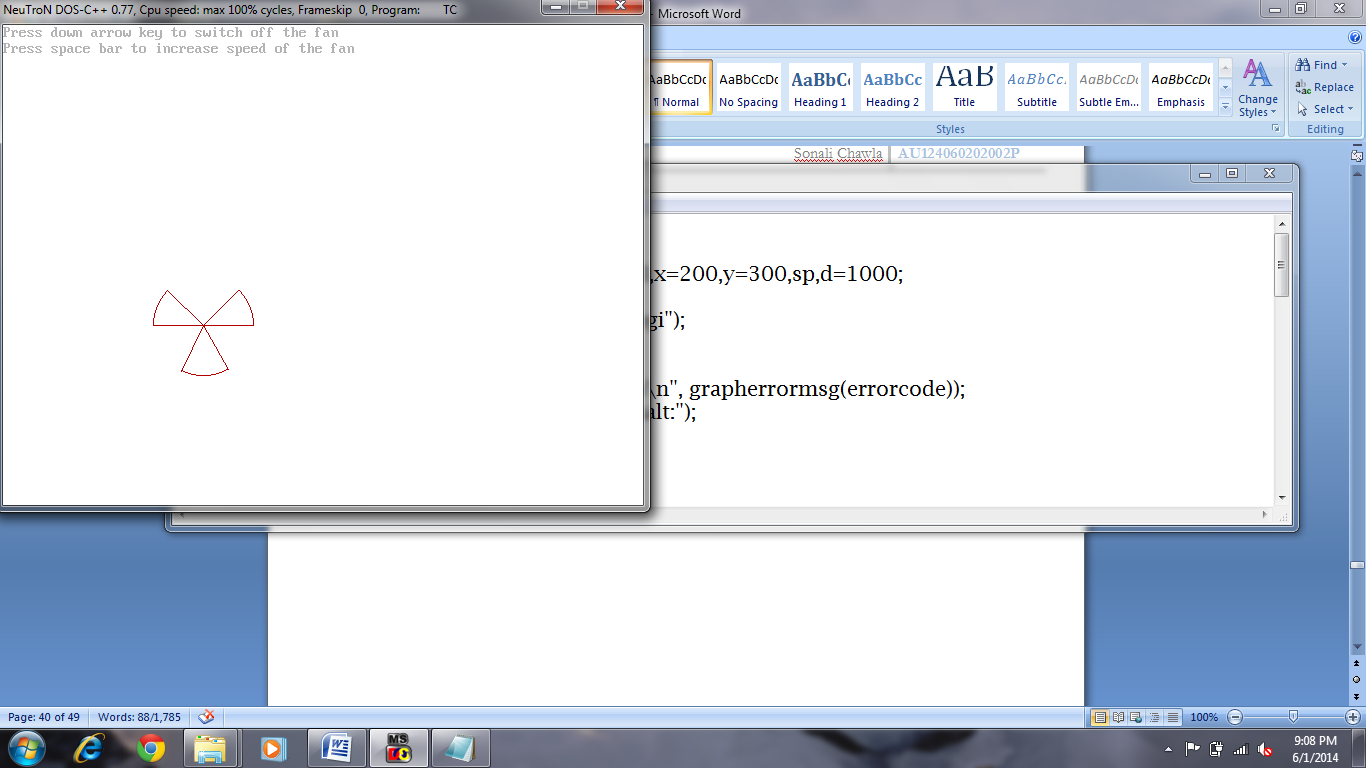
} cleardevice();

} break;

} getch();

closegraph();

}



**Q. Write a program to design traffic signal with continuous alternate light display until a key is pressed.**

**Coding:**

#include<graphics.h>

void main()

{ int gd,gm,errorcode;

gd=DETECT;

initgraph(&gd,&gm,"c:\\tc\\bgi");

errorcode = graphresult();

if (errorcode != grOk)

{ printf("Graphics error: %s\n", grapherrormsg(errorcode));

printf("Press any key to halt:");

getch(); exit(1); }

rectangle(170,200,250,400); circle(210,240,25);

circle(210,300,25); circle(210,360,25);

outtextxy(200,450,"PRESS ANY KEY");

getch();

while (1)

{ cleardevice();

rectangle(170,200,250,400); circle(210,240,25);

circle(210,300,25); circle(210,360,25);

setfillstyle(1,RED);

floodfill(210,240,WHITE);

outtextxy(200,240,"STOP");

delay(2500); cleardevice();

rectangle(170,200,250,400); circle(210,300,25);

circle(210,240,25); circle(210,360,25);

setfillstyle(1,YELLOW); floodfill(210,300,WHITE);

setcolor(BLUE);

outtextxy(190,300,"READY");

setcolor(WHITE);

delay(2500); cleardevice();

rectangle(170,200,250,400); circle(210,360,25);

circle(210,240,25); circle(210,300,25);

setfillstyle(1,GREEN); floodfill(210,360,WHITE);

outtextxy(200,360,"GO"); delay(2500);

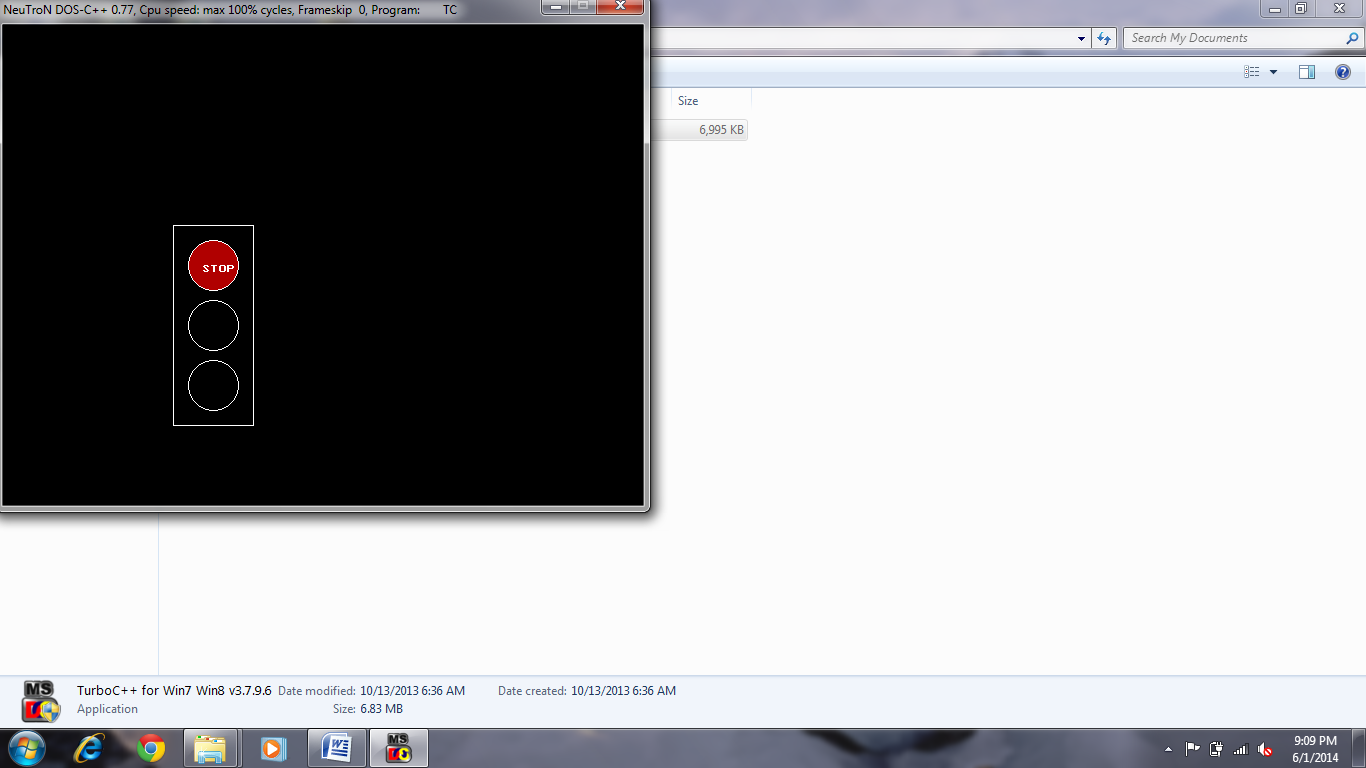
if(kbhit())

break;

}getch();

closegraph();

}



**Q. Write a program to design the output of ball hitting the corners of window.**

**Coding:**

#include<graphics.h>

void main()

{ int gd,gm,errorcode,x,y=50,r=50;

gd=DETECT;

initgraph(&gd,&gm,"c:\\tc\\bgi");

errorcode = graphresult();

if (errorcode != grOk)

{ printf("Graphics error: %s\n", grapherrormsg(errorcode));

printf("Press any key to halt:");

getch();

exit(1); }

setbkcolor(15);

setcolor(5);

while(!kbhit())

{ if(kbhit())

exit(0);

else

for(x=50;x<=600;x++)

{ circle(x,y,r);

delay(5);

cleardevice(); }

if(kbhit())

exit(0);

else

for(y=50;y<=430;y++)

{ circle(x,y,r);

x--;

delay(5);

cleardevice(); }

if(kbhit())

exit(0);

else

for(y=430;x>=50;y=y-2)

{ circle(x,y,r);

x--;

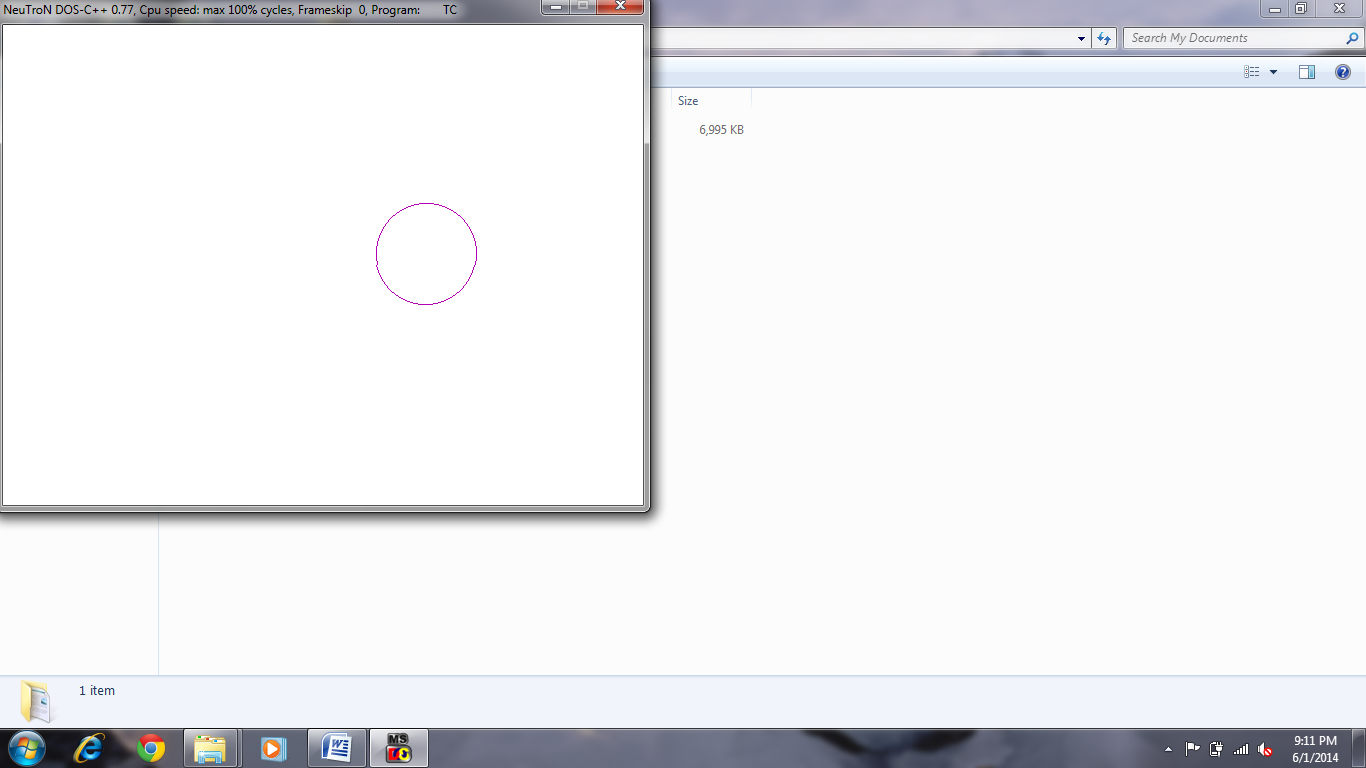
delay(5);

cleardevice(); }

} getch();

closegraph();

}



**Q. Write a program to revolve a coin on a table.**

**Coding:**

#include<graphics.h>

void main()

{ int gd,gm,errorcode,x=250,y=250,s=0,e=360,a=25,b=40;

gd=DETECT;

initgraph(&gd,&gm,"c:\\tc\\bgi");

errorcode = graphresult();

if (errorcode != grOk)

{ printf("Graphics error: %s\n", grapherrormsg(errorcode));

printf("Press any key to halt:");

getch(); exit(1); }

setbkcolor(15);

setcolor(1);

while(!kbhit())

{ rectangle(100,290,400,310); rectangle(100,310,120,400);

rectangle(380,310,400,400); circle(x,y,40);

delay(200); cleardevice();

rectangle(100,290,400,310); rectangle(100,310,120,400);

rectangle(380,310,400,400); ellipse(x,y,s,e,a,b);

delay(200); cleardevice();

rectangle(100,290,400,310); rectangle(100,310,120,400);

rectangle(380,310,400,400); ellipse(x,y,s,e,a-10,b);

delay(200); cleardevice();

rectangle(100,290,400,310); rectangle(100,310,120,400);

rectangle(380,310,400,400); line(250,290,250,210);

delay(200); cleardevice();

rectangle(100,290,400,310); rectangle(100,310,120,400);

rectangle(380,310,400,400); ellipse(x,y,s,e,a-10,b);

delay(200); cleardevice();

rectangle(100,290,400,310); rectangle(100,310,120,400);

rectangle(380,310,400,400); ellipse(x,y,s,e,a,b);

delay(200); cleardevice();

rectangle(100,290,400,310); rectangle(100,310,120,400);

rectangle(380,310,400,400);

}

getch();

closegraph();

}

