**PROGRAM 1 : IMAGE ROTATION**

#include<stdio.h>

#include<math.h>

#include<graphics.h>

#include<conio.h>

int n=0,gd=DETECT,gm,xr,yr;

float a;

void main()

{

struct

{

int x;

int y;

}

pts[10],rpts[10];

int i;

clrscr();

printf("\n\n Enter no.of points to draw an object: ");

scanf("%d",&n);

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

{

printf("\n\n Enter(x%d,y%d)value: ",i+1,i+1);

scanf("%d,%d",&pts[i].x,&pts[i].y);

}

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

setbkcolor(BLUE);

setcolor(RED);

printf("\n Object before Rotation:");

fillpoly(n,pts);

printf("\n Enter an angle to rotate the object: ");

scanf("%f",&a);

a=a\*3.142/180;

printf("\n Enter the Reference point(x,y):");

scanf("%d,%d",&xr,&yr);

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

{

rpts[i].x=(int)((float)xr+(pts[i].x-xr)\*cos(a)-(pts[i].y-yr)\*sin(a));

rpts[i].y=(int)((float)yr+(pts[i].x-xr)\*sin(a)+(pts[i].y-yr)\*cos(a));

}

fillpoly(n,rpts);

printf("After rotation:");

getch();

closegraph();

}

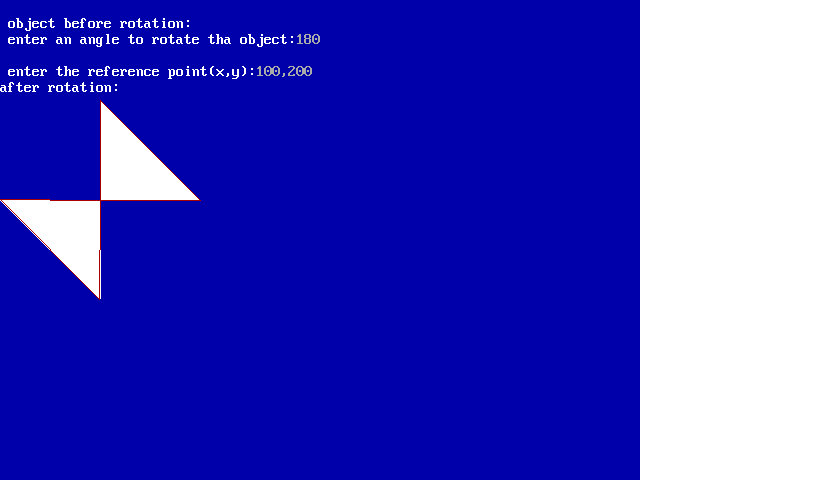
**OUTPUT**

**Enter number of points to draw an object : 3**

**Enter x1,y1 value : 100,100**

**Enter x2,y2 value : 200,200**

**Enter x3,y3 value : 100,200**



**PROGRAM 2 : TEXT FALL**

#include<graphics.h>

#include<stdio.h>

#include<string.h>

#include<conio.h>

int main(void)

{

int gd=DETECT,gm;

int i,j=0,x=10,k,l;

char s1[100],s2[20];

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

printf(" Enter a sentence:\n");

gets(s1);

l=strlen(s1);

cleardevice();

outtextxy(10,0,s1);

setviewport(0,15,639,464,0);

for(k=0;k<=l;k++)

{

if(s1[k]!=' '&&s1[k]!='\0')

{

s2[j++]=s1[k];

}

else

{

s2[j++]='\0';

j=0;

for(i=10;i<=460;i+=10)

{

outtextxy(x,i,s2);

delay(100);

}

clearviewport();

x=x+strlen(s2)\*7+10;

}

}

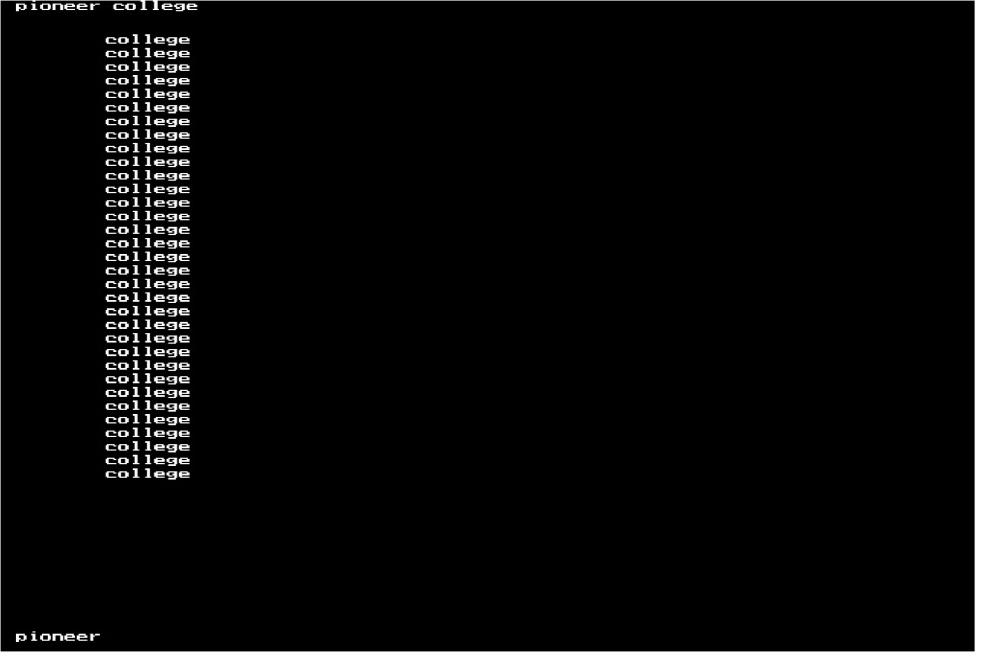
getch();

closegraph();

return(0);

}

**OUTPUT**



**PROGRAM 3 : PROGRAM TO DRAW A LINE USING DDA ALGORITHM**

#include<graphics.h>

#include<stdio.h>

#include<stdlib.h>

#include<conio.h>

void main()

{

int gd=DETECT,gm;

int x1,y1,x2,y2,dx,dy,k,steps;

float xinc,yinc;

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

printf("\t program to draw a line using dda algorithm ");

printf("\n\t\t Enter the value of x1,y1:");

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

printf("\n\t\t Enter the value of x2,y2:");

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

dx=x2-x1;

dy=y2-y1;

putpixel(x1,y1,13);

if(dx>dy)

steps=abs(dx);

else

steps=abs(dy);

xinc=dx/(float)steps;

yinc=dy/(float)steps;

for(k=0;k<=steps;k++)

{

x1+=xinc;

y1+=yinc;

putpixel(x1,y1,13);

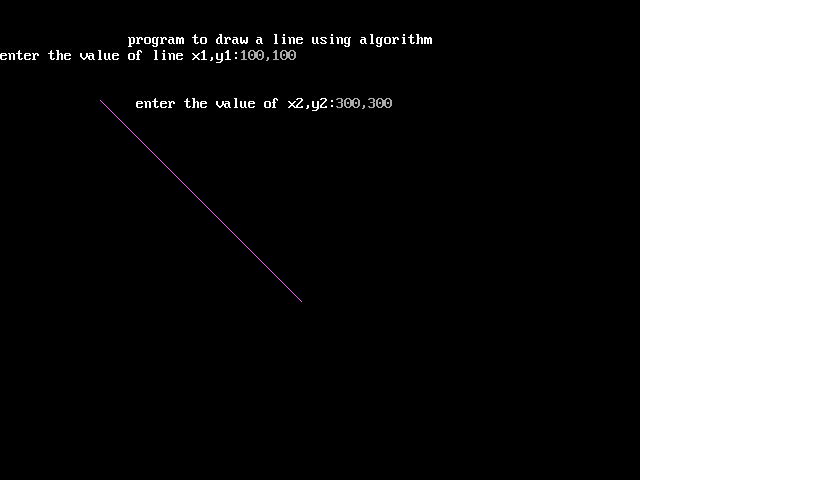
}

getch();

closegraph();

}

**OUTPUT**

****

**PROGRAM 4 : MOVE A CAR WITH SOUND EFFECT**

#include<stdio.h>

#include<graphics.h>

#include<stdlib.h>

#include<conio.h>

void draw\_car()

{

int poly1[10]={0,200,0,160,260,160,260,200,0,200};

int poly2[10]={40,160,70,130,170,130,200,160,40,160};

setfillstyle(SOLID\_FILL,5);

fillpoly(5,poly1);

setfillstyle(SOLID\_FILL,25);

fillpoly(5,poly2);

setfillstyle(SOLID\_FILL,20);

fillellipse(50,200,20,20);

fillellipse(200,200,20,20);

}

void main()

{

int gd=DETECT,gm,i,j;

void\*bitmap;

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

draw\_car();

bitmap=malloc(imagesize(0,130,270,230));

getimage(0,130,270,230,bitmap);

for(i=0;!kbhit();i++)

{

if(i>500)i=0;

putimage(i,130,bitmap,OR\_PUT);

setbkcolor(WHITE);

sound(100);

delay(8);

putimage(i,130,bitmap,XOR\_PUT);

nosound();

}

closegraph();

getch();

}

**OUTPUT**

**PROGRAM 5 : BOUNCE A BALL**

#include<graphics.h>

#include<stdio.h>

#include<conio.h>

void main()

{

int gd=DETECT,gm;

int i,x=0,y=0,ymin=0;

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

for(i=2;i<=10;i++)

{

setfillstyle(SOLID\_FILL,20);

for(y=ymin;y<=480;y+=2)

{

if(y%5==0)x=x+3;

fillellipse(x,y,10,10);

delay(20);

}

ymin=480-(480/i);

for(y=480;y>=ymin;y-=2)

{

if(y%5==0)x=x+3;

fillellipse(x,y,10,10);

delay(20);

}

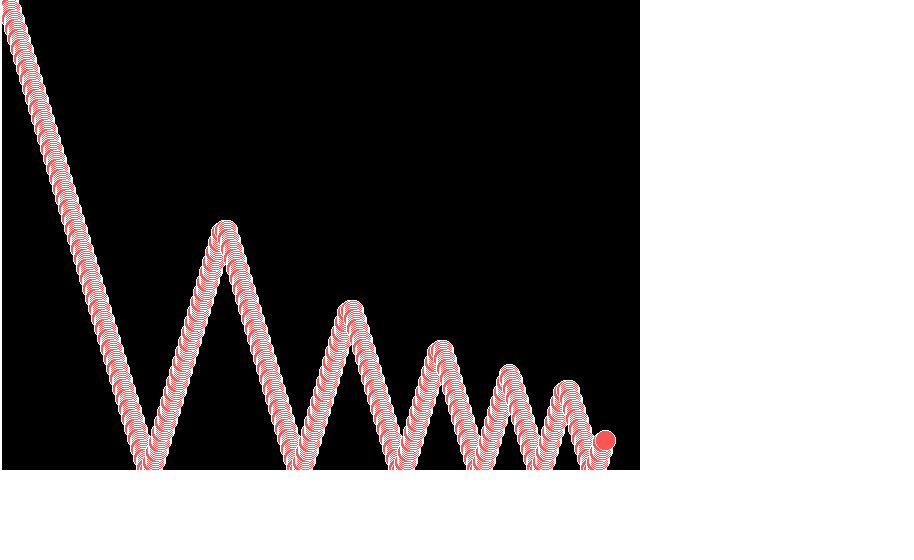
if(x>=640)break;

}

closegraph();

}

**OUTPUT**

****

**PROGRAM 6 : POLYGON CLIPPING**

#include<stdio.h>

#include<conio.h>

#include<graphics.h>

void main() {

int gd=DETECT, gm:

int xcmin, ycmin, xcmax, ycmax;

int x,y,i;

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

printf ("Enter the clipmin value:");

scanf ("%d,%d",&xcmin, &ycmin):

printf ("Enter the clipmax value:");

scanf("%d, %d",&xcmax, &ycmax);

rectangle(xcmin,ycmin,xcmax,ycmax):

printf ("\n Enter the x & y value:");

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

putpixel(x,y,20);

circle(x,y,3);

if((xcmin< x&& x<xcmax) && (ycmin<y && y<ycmax ))

{

printf ("Given point is inside clip boundry");

}

else if ((xcmin > x ||x>xcmax) | (ycmin > y || y>ycmax))

{

printf ("Given point is outside clip boundry");

}

else

{

printf ("Given point is on clip boundary");

}

getch();

closegraph();

}

**OUTPUT**

****

