**ROTATE AN IMAGE**

**EX.NO: 01**

**DATE:**

**AIM:**

To write a program to rotate an image.

**ALGORITHM:**

**STEP 1:** Start the process.

**STEP 2:** Initialize the graphics mode, graphics, drive and path.

**STEP 3:** Call rot() function.

**STEP 4:** Get four values for x & y.

**STEP 5:** Draw the shape for reference point.

**STEP 6:** Get the value for reference point.

**STEP 7:** Get the angle for rotation.

**STEP 8:** Rotate the given shape using the formula **t = r \* 3.14 / 180.0** floor function.

**STEP 9:** Draw the shape using line function.

**STEP 10:** Stop the process.

**CODING:**

#include<stdio.h>

#include<conio.h>

#include<graphics.h>

#include<math.h>

void rot();

void main()

{

int gd=DETECT,gm;

clrscr();

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

cleardevice();

rot();

getch();

}

void rot()

{

int x1,y1,x2,y2,x3,y3,x4,y4;

int x11,y11,x22,y22,x33,y33,x44,y44;

int xr,yr,r;

float t;

cleardevice();

settextstyle(7,0,5);

outtextxy(320,20,"2D -ROTATION");

gotoxy(1,5);

printf("\n enter the xl&yl values:");

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

printf("\n enter the x2&y2 values:");

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

printf("\n enter the x3&y3 values:");

scanf("%d%d",&x3 ,&y3);

printf("\n enter the x4&y4 values:");

scanf("%d%d",&x4,&y4);

cleardevice();

settextstyle(7,0,5);

settextjustify(1,1);

outtextxy(320,20,"2D-ROTATION");

setlinestyle(1 ,1 ,1);

line(x1,y1,x2,y2);

line(x2,y2,x3,y3);

line(x3,y3,x4,y4);

line(x4,y4,x1,y1);

setcolor(5);

gotoxy(1 ,25);

printf("enter the reference point xr,yr:");

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

putpixel(xr,yr, 5);

printf("enter the angle:");

scanf("%d" ,&r);

t=r\*3.14/180.0;

x11 =floor(xr+(x1-xr)\*cos(t)-(y1-yr)\*sin(t));

y11 =floor(yr+(x1-xr)\*sin(t)-(y1-yr)\*cos(t));

x22 =floor(xr+(x2-xr)\*cos(t)-(y2-yr)\*sin(t));

y22 =floor(yr+(x2-xr)\*sin(t)-(y2-yr)\*cos(t));

x33 =floor(xr+(x3-xr)\*cos(t)-(y3-yr)\*sin(t));

y33 =floor(yr+(x3-xr)\*sin(t)-(y3-yr)\*cos(t));

x44 =floor(xr+(x4-xr)\*cos(t)-(y4-yr)\*sin(t));

y44 =floor(yr+(x4-xr)\*sin(t)-(y4-yr)\*cos(t));

setcolor(15);

setlinestyle(0,1 ,1);

line(x11,y11,x22,y22);

line(x22,y22,x33,y33);

line(x33,y33,x44,y44);

line(x44,y44,x11,y11);

}

**OUTPUT:**

Enter The x1 & y1 Values : 11 22

Enter The x2 & y2 Values : 33 44

Enter The x3 & y3 Values : 55 66

Enter The x4 & y4 Values : 77 88

Enter the reference point xr & yr : 33 44

Enter the angle : 90

**DROP A WORD**

**EX.NO: 02**

**DATE:**

**AIM:**

To write a program to drop each word in a sentence one by one.

**ALGORITHM:**

**STEP 1:** Start the process.

**STEP 2:** Initialize the graphics drive, graphics mode & path.

**STEP 3:** Get the sentence.

**STEP 4:** Calculate the length of sentence using **strlen( )** function.

**STEP 5:** Check if t[i]! = “ “ then increment \* & I value.

**STEP 6:** Otherwise increment j & i value.

**STEP 7:** Repeat steps 5 & 6 until the while condition is true.

**STEP 8:** Calculate text width & display text within “for” loop.

**STEP 9:** Drop the word using drop function.

**STEP 10:** Stop the process.

**CODING:**

#include<stdio.h>

#include<conio.h>

#include<process.h>

#include<string.h>

#include<graphics.h>

#include<math.h>

char t[200],str[10][20];

int x[10],y[10],j=0;

void drop(int k)

{

int i,p,s;

for(p=60;p<=440;p++)

{

delay(50);

cleardevice();

outtextxy(200,10,"TEXT ANIMATION");

outtextxy(200,30,"\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

for(s=k+1;s<=j;s++)

for(i=k+1;i<=s;i++)

outtextxy(x[i],50,str[i]);

outtextxy(x[k],p,str[k]);

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

for(i=0;i<=s-1;i++)

outtextxy(x[i],440,str[i]);

}

}

void main()

{

int gd=DETECT, gm=0;

int i=0,k=0,m=0,m1,in=1,l;

int p;

clrscr();

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

settextstyle(1,0,2);

printf("\n ENTER THE SENTENCE:");

gets(t);

l=strlen(t);

while(i<l)

{

if(t[i]!=' ')

{

str[j][k]=t[i];

k++;

i++;

}

else

{

j++;

i++;

k=0;

}

}

cleardevice();

x[0]=m;

y[0]=50;

outtextxy(200,10,"TEXT ANIMATION");

outtextxy(200,30,"\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

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

{

m1=textwidth(str[i]);

outtextxy(m,50,str[i]);

m=m+m1+10;

x[in]=m;

y[in]=10;

in++;

}

settextstyle(6,0,1);

outtextxy(0,100,"PRESS ANY KEY TO DROP...");

getch();

settextstyle(1,0,2);

for(i=0;1<=j;i++)

drop(i);

getch();

closegraph();

}

**OUTPUT:**

**DROPING A WORD**

Enter the sentence: BACHELOR OF COMPUTER SCIENCE

TEXT ANIMATION

---------- -----------------

BACHELOR OF COMPUTER SCIENCE

Press any key to drop

SCIENCE

COMPUTER

OF

BACHELOR

BACHELOR OF COMPUTER SCIENCE

**DDA LINE DRAWING ALGORITHM**

**EX.NO: 03**

**DATE:**

**AIM:**

To write a program to drop a line using DDA algorithm.

**ALGORITHM:**

**STEP 1:** Start the process.

**STEP 2:** Get the value of starting point & ending point.

**STEP 3:** Initialize the graphics mode, drive and path.

**STEP 4:** Calculate dx, dy value based on points.

**STEP 5:** If dx > = dy then length = dx, otherwise length = dy.

**STEP 6:** Again calculate dx, dy value based on length.

**STEP 7:** Display the pixel using while loop.

**STEP 8:** Increment the starting value.

**STEP 9:** Repeat steps 7 & 8 until the while condition is true.

**STEP 10:** Stop the process.

**CODING:**

#include<stdio.h>

#include<conio.h>

#include<graphics.h>

#include<math.h>

#include<string.h>

void main()

{

float x,y,xl,yl,x2,y2,dx,dy,length;

int i,gd,gm;

clrscr();

printf("\nENTER THE VALUE OF(x 1 &yl ):");

scanf("%f%f",&xl ,&yl);

printf("\nENTER THE VALUE OF(x2&y2):");

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

detectgraph(&gd ,&gm);

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

dx=abs(x2-xl);

dy=abs(y2-yl);

if(dx<=dy)

length=dx;

else

length=dy;

dx=(x2-xl)/length;

dy=(y2-yl)/length;

x=xl+0.5;

y=yl+0.5;

i=1;

while(i<length)

{

putpixel(x,y,15);

x=x+dx;

y=y+dy;

i=i+1;

delay(100);

}

getch();

closegraph();

}

**OUTPUT:**

Enter The Value : 200 250

Enter The Value : 300 500

**MOVING A CAR**

**EX.NO: 04**

**DATE:**

**AIM:**

To write a program to move a car with sound effect.

**ALGORITHM:**

**STEP 1:** Start the process.

**STEP 2:** Initialize the graphics mode, graphics drive and path.

**STEP 3:** Calculate mid x and mid y value.

**STEP 4:** Draw the line using line function.

**STEP 5:** Set the background color using setbkcolor function.

**STEP 6:** Call **displaycar( )** function within while loop.

**STEP 7:** Increment the K value within while loop.

**STEP 8:** Move the car with sound effect using sound functon.

**STEP 9:** Repeat steps 6 & 7 until pressing any key.

**STEP 10:** Stop the process.

**CODING:**

#include<stdio.h>

#include<conio.h>

#include<graphics.h>

int midx,midy,maxx,maxy;

void displaycar(int c,int k)

{

setcolor(c);

ellipse(50+k,midy+40,0,360,10,10);

ellipse(100+k,midy+40,0,360,10,10);

line(60+ k,midy+40,90+k,midy+40);

line(20+k,midy+40,40+k,midy+40);

line(110+k,midy+40,130+k,midy+40);

line(20+k,midy+40,20+k,midy+20);

line(130+k,midy+40,130+k,midy+20);

line(20+k,midy+20,30+k,midy+20);

line(130+k,midy+20,100+k,midy+20);

line(30+k,midy+20,30+k,midy);

line(100+k,midy+20,100+k,midy);

line(30+k,midy,100+k,midy);

}

void main()

{

int gd=DETECT,gm;

int k=20;

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

midx=(maxx=getmaxx())/2;

midy=(maxy=getmaxy()) /2;

line(0,midy-50, maxx,midy- 50);

line(0,midy+51 ,maxx,midy+51);

setbkcolor(0);

while(!kbhit())

{

displaycar(5,k);

sound(700);

delay(50);

displaycar(0,k-5);

k+=5;

if(k>=maxx)

k=-20;}

nosound();

getch();

}

**OUTPUT:**

**MOVING A CAR**

**BOUNCE A BALL**

**EX.NO: 05**

**DATE:**

**AIM:**

To write a program to bounce a ball with sound effect.

**ALGORITHM:**

**STEP 1:** Start the process.

**STEP 2:** Initialize the graphics mode, graphics drive and path.

**STEP 3:** Draw the line using line function.

**STEP 4:** Calculate y = cos ( ( ( i \* 22 / 7 ) / 180 ) ) / j.

**STEP 5:** Calculate Y value within “for” loop.

**STEP 6:** Check if ( y > 0 ) then increment the X values.

**STEP 7:** Draw the circle using circle function.

**STEP 8:** Increment the count values.

**STEP 9:** Repeat the step 5, 6, 7 until the condition is true.

**STEP 10:** Bounce a ball with sound effect using sound function.

**STEP 11:** Stop the process.

**CODING:**

#include<dos.h>

#include<graphics.h>

#include<math.h>

#include<conio.h>

void main()

{

int gd=DETECT,gm,k;

float x=1 ,y=120.00000,j=.5,count=.1;

float r=10,i;

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

setcolor(10);

line(0,415,650,415);

sleep(1);

for(k=0;k<=7;k++)

{

for(i=90;i<270;i+=10)

{

y=cos(((i\*22/7)/180))/j;

if(y>0)

y=-y;

x+=5;

setcolor(10);

setfillstyle(1,10);

circle(x,y\*100+400,10);

delay(100);

nosound();

setcolor(0);

setfillstyle(1 ,10);

circle(x,y\*100+400,r);

floodfill(x,y\*100+400,0);

}

sound(150);

j+=count;

count+=.1;

}

nosound();

getch();

closegraph();

}

**OUTPUT:**

**CHECK PIXEL POSITION IN POLYGON**

**EX.NO: 06**

**DATE:**

**AIM:**

To write a program to test whether given pixel is inside (or) outside (or) on the polygon.

**ALGORITHM:**

**STEP 1:** Start the process

**STEP 2:** Initialize the graphics mode, graphics drive and path.

**STEP 3:** Get the values of input n.

**STEP 4:** Get the co-ordinate value within for loop.

**STEP 5:** Draw the polygon shape using fillpoly.

**STEP 6:** Set the border color using **setcolor( )** function & display the pixel using **putpixel( )** function.

**STEP 7:** Get the any pixel value.

**STEP 8:** Get the color of particular pixel using **getpixel( )** function.

**STEP 9:** If b = 15 display inside the polygon.

**STEP 10:** If b = 10 display on the polygon.

**STEP 11:** Otherwise display outside the polygon.

**STEP 12:** Stop the process.

**CODING:**

#include<stdio.h>

#include<graphics.h>

#include<conio.h>

void main()

{

int gdrive=DETECT,gmode;

int a[10],i,j,n,x,y,b;

clrscr();

initgraph(&gdrive, &gmode,"c:\\tc\\bgi");

printf("\n\tENTER THE NO. OF POINTS USED:");

scanf("%d",&n);

for(i=0,j=0;i<n;i++,j+=2)

{

printf("\n\tENTER THE COORDINATE POINTS:");

scanf("%d%d",&a[j],&a[j+1]);

}

setcolor(10);

fillpoly(n,a);

printf("\n\tENTER THE FINDING PIXEL CO-ORDINATE VALUES:");

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

b=getpixel(x,y);

putpixel(x,y,2);

printf("\n%d",&b);

if(b==15)

printf("\n\tGIVEN PIXEL IS INSIDE THE POLYGEN");

else

if(b==10)

printf("\n\tGIVEN PIXEL IS ON THE POLYGEN");

else

printf("\n\tGIVEN PIXEL IS OUTSIDE THE POLYGEN");

getch();

closegraph();

}

**OUTPUT:**

CHECK PIXEL POSITION IN POLYGON

Enter the No.Of Points Used: 4

Enter The Co-Ordinate Points: 150 200

Enter The Co-Ordinate Points: 150 250

Enter The Co-Ordinate Points: 300 250

Enter The Co-Ordinate Points: 300 200

Enter The Finding Pixel Co-Ordinate Values: 180 280

b = 15

Given Pixel Is Inside the Polygon

Enter the No.Of Points Used : 4

Enter The Co-Ordinate Points: 150 200

Enter The Co-Ordinate Points: 150 250

Enter The Co-Ordinate Points: 300 250

Enter The Co-Ordinate Points: 300 200

Enter The Finding Pixel Co-Ordinate Values: 100 100

b = 0

Given Pixel Is Outside the Polygon

**SUN FLOWER**

**EX.NO: 07**

**DATE:**

**AIM:**

To create a sun flower image using **Adobe Photoshop.**

**ALGORITHM:**

**STEP 1:** Start the process.

**STEP2:** To open a new file, select the file and then click on new, select the size **640\*480** and then click ok.

**STEP 3:** Select custom shape tool in the tool box.

**STEP 4:** Select any flower shape and then draw in the work area.

**STEP 5:** Select any stem and leaf shape and then draw in the work area.

**STEP 6:** Draw an ellipse in the center of the flower.

**STEP 7:** Draw the grass using brush tool by changing brush size to 112.

**STEP 8:** Apply desired color from the color palette.

**STEP 9:** Finally we select the menu bar, click layer-> flatten image.

**STEP 10:** Save the image in **speg4** file format.

**STEP 11:** Stop the process.

**OUTPUT:**

****

**PLASTIC SURGERY**

**EX.NO: 08**

**DATE:**

**AIM:**

To create a plastic surgery using **Adobe photo shop**

**ALGORITHM:**

**STEP 1:** Start the process.

**STEP 2:** Open the existing image.

**STEP 3:** Select the filter option in menu bar.

**STEP 4:** Select liquefy from filter option separate window will open.

**STEP 5:** Adjust the brush size as necessary.

**STEP 6:** Use then pucker tool or bloat tool to do surgery.

**STEP 7:** Alter the change will be occur, save the image in the **jpeg** file format.

**STEP 8:** Stop the process.

**OUTPUT: BEFORE**



**AFTER**



**SEE THROUGH TEXT**

**EX.NO: 09**

**DATE:**

**AIM:**

To create a see through text using **Adobe photo shop**

**ALGORITHM:**

**STEP 1:** Start the process.

**STEP 2:** Click on layer Menu, click new fill layer and then pattern.

**STEP 3:** Select the layer background and press key “D” foreground white and background black.

**STEP 4:** Type any text using text tool in tool box, the text will be appear in white color.

**STEP 5:** In layer gradient choose style, the dialog box will be displayed.

**STEP 6:** In blending option click blend mode =>Multiply, Capacity 75% click, Drop shadow => Angle 120, Distance 5, Spread 0, Size 5.

**STEP 7:** Click be level and emboss -> size 1-16, select any number between 1to16.

**STEP 8:** Save the project in **JPEG** file format.

**STEP 9:** Stop the process.

**OUTPUT:**



**BLACK AND WHITE IMAGE INTO COLOR IMAGE**

**EX.NO: 10**

**DATE:**

**AIM:**

To create a black and white image and convert it into color image using **Adobe photo shop.**

**ALGORITHM:**

**STEP 1:** Start the process.

**STEP 2:** Open the gray scale image.

**STEP 3:** Choose the magic wand tool select the particular area from the image.

**STEP 4:** Select the buckettool from tool box and pick the appropriate color from palette and apply in the selected area to colorize the picture.

**STEP 5:** Save the image in **JPEG** file format.

**STEP 6:** Stop the process.

**OUTPUT:**

****

**PLANE ANIMATION**

**EX.NO: 11**

**DATE:**

**AIM:**

To create a plane animation using **Adobe photo shop**

**ALGORITHM:**

**STEP 1: S**tart the process.

**STEP 2:** Open the plane image.

**STEP 3:** Select polygonal lasso tool to cut the plane image, edit-> copy.

**STEP 4:** Next open clouds image, edit-> paste.

**STEP 5:** Save the image as separate file.

**STEP 6:** Select layer option -> channels -> red.

**STEP 7:** Ctrl + A to select the picture.

**STEP 8:** Ctrl + C to copy the picture.

**STEP 9:** Ctrl + D to deselect the picture.

**STEP 10:** Select the RGB in channel.

**STEP 11:** Switch over to layer tab and select add layer mask button.

**STEP 12:** Press Alt key and click the layer mask thumbnail in layer tab.

**STEP 13:** Ctrl + V to paste the image.

**STEP 14:** Ctrl + D to deselect the image.

**STEP 15:** Remove the link that appears between layer and layer mask thumbnail.

**STEP 16:** Cloud image to select.

**STEP 17:** Select red in the channels.

**STEP 18:** Ctrl + A to select all image.

**STEP 19:** Ctrl + C to copy the image.

**STEP 20:** Ctrl + D to deselect the image.

**STEP 21:** Click layer mask thumbnail window in layer tab which contains plane.

**STEP 22:** Ctrl + V to paste the image.

**STEP 23:** Ctrl + D to deselect the image.

**STEP 24:** Ctrl + I to invert the image.

**STEP 25:** Ctrl + L to set the layer level.

**STEP 26:** Change the layer input to the 70, 1.00, 160.

**STEP 27:** Click the layer thumbnail which contain the plane channel -> RGB.

**STEP 28:** Switch over to image ready by selecting edit in image ready option from tool box.

**STEP 29:** Select window -> animation to view animation window.

**STEP 30:** Click on duplicate current frame to create duplicate fame.

**STEP 31:** According to frame, adjust the plane position.

**STEP 32:** Click on the explorer icon in tool box to view plane animation.

**STEP 33:** Save the web page.

**STEP 34:** Stop the process.

**OUTPUT:**





**WEB PAGE**

**EX.NO: 12**

**DATE:**

**AIM:**

To create a web page using **Adobe photo shop**

**ALGORITHM:**

**STEP 1:** Start the process.

**STEP 2:** Open a new document.

**STEP 3**: Select layer -> new fill layer -> pattern click.

**STEP 4:** Select horizontal tab tool used to type the text.

**STEP 5:** Apply blending effect to typed text.

**STEP 6:** Using round rectangle tool, draw buttons.

**STEP 7:** Again click layer properties of button to apply blending effect to them.

**STEP 8:** Name the buttons using horizontal type tool and apply blending effect to it.

**STEP 9:** Use the slice tool and select the buttons.

**STEP 10:** Create web pages using html and copy the address.

**STEP 11:** Select the edit slice option by applying right click on the button.

**STEP 12:** Paste the copied address of web page on the **URL** and **TARGET** text box to view the web page.

**STEP 13:** Switch over the image ready and click on explorer icon in the tool box to view the webpage.

**STEP 14:** Click on the button to see the appropriate results.

**STEP 15:** Stop the process.

**OUTPUT:**

****