

LIST OF EXPERIMENTS

S.NO.	TITLE	PAGE.NO.
1.	Rotating an image.	
2.	Dropping a word	
3.	Line drawing using DDA	
4.	Moving a car	
5.	Bouncing a ball	
6.	Polygon Clipping	
7.	Create Sun Flower using Photoshop.	
8.	Animate Plane flying in the Clouds using Photoshop.	
9.	Create Plastic Surgery for the Nose using Photoshop.	
10.	Create See-through text using Photoshop.	
11.	Create a Web Page using Photoshop.	
12.	Convert Black and White Photo to Color Photo using Photoshop.	

EX:NO:01

ROTATING AN IMAGE

AIM:

To write a program to rotate an image.

ALGORITHM:

Step 1:Start the process.

Step2: Declare variables x1,y1,x2,y2,x3,y3 and also declare the functions.

Step3: Declare gdriver=DETECT, mode and Initialize the graphic mode with the path location in TurboC3 folder.

Step4: Load the values of the variables x1,y1,x2,y2,x3,y3 into the frame buffer.

Step5: Using the function **triangle(x1,y1,x2,y2,x3,y3)** draw a triangle.

Step6: Load the value of the variable angle into the frame buffer.

Step7: Using Rotate function rotates the image to certain angle value.

Step8: Close thegraph and stop the process.

SOURCE CODE:

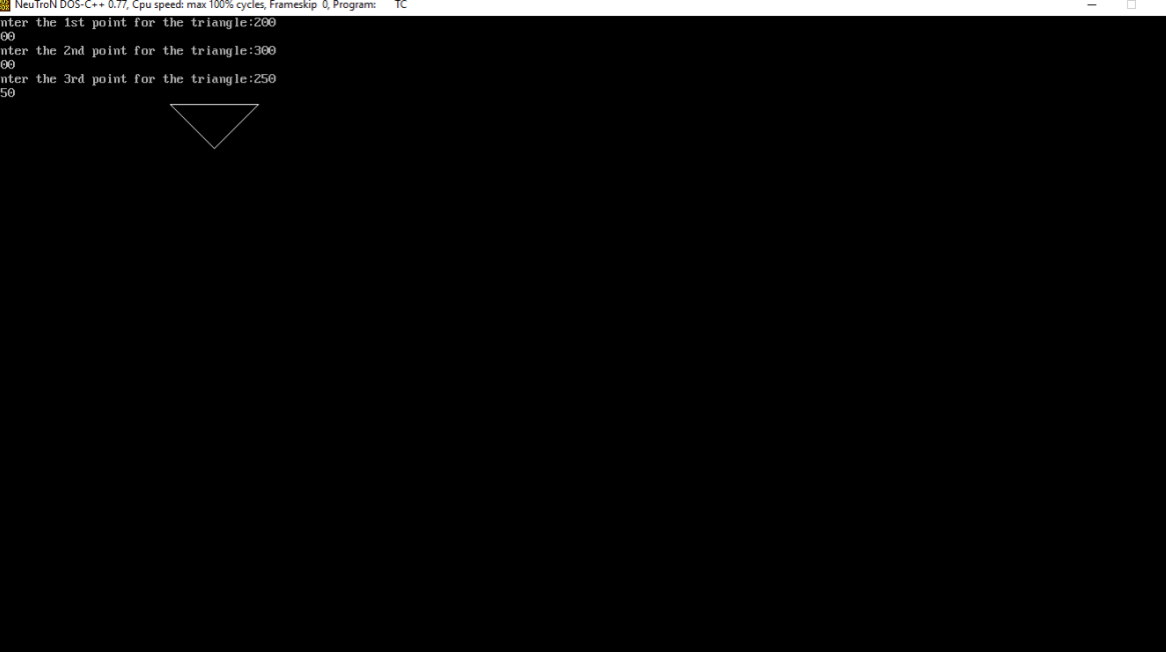
```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
#include<process.h>
#include<math.h>
void triangle(int x1,int y1,int x2,int y2,int x3,int y3);
void Rotate(int x1,int y1,int x2,int y2,int x3,int y3);
void main()
{
    int gd=DETECT,gm;
    int x1,y1,x2,y2,x3,y3;
    initgraph(&gd,&gm,"..\\bgi");
    printf("Enter the 1st point for the triangle:");
    scanf("%d%d",&x1,&y1);
    printf("Enter the 2nd point for the triangle:");
    scanf("%d%d",&x2,&y2);
    printf("Enter the 3rd point for the triangle:");
    scanf("%d%d",&x3,&y3);
    triangle(x1,y1,x2,y2,x3,y3);
    getch();
    cleardevice();
    Rotate(x1,y1,x2,y2,x3,y3);
    setcolor(1);
    triangle(x1,y1,x2,y2,x3,y3);
    getch();
}
void triangle(int x1,int y1,int x2,int y2,int x3,int y3)
{
    line(x1,y1,x2,y2);
```

```

        line(x2,y2,x3,y3);
        line(x3,y3,x1,y1);
    }
void Rotate(int x1,int y1,int x2,int y2,int x3,int y3)
{
    int x,y,a1,b1,a2,b2,a3,b3,p=x2,q=y2;
    float Angle;
    printf("Enter the angle for rotation:");
    scanf("%f",&Angle);
    cleardevice();
    Angle=(Angle*3.14)/180;
    a1=p+(x1-p)*cos(Angle)-(y1-q)*sin(Angle);
    b1=q+(x1-p)*sin(Angle)+(y1-q)*cos(Angle);
    a2=p+(x2-p)*cos(Angle)-(y2-q)*sin(Angle);
    b2=q+(x2-p)*sin(Angle)+(y2-q)*cos(Angle);
    a3=p+(x3-p)*cos(Angle)-(y3-q)*sin(Angle);
    b3=q+(x3-p)*sin(Angle)+(y3-q)*cos(Angle);
    printf("Rotate");
    triangle(a1,b1,a2,b2,a3,b3);
}

```

OUTPUT:

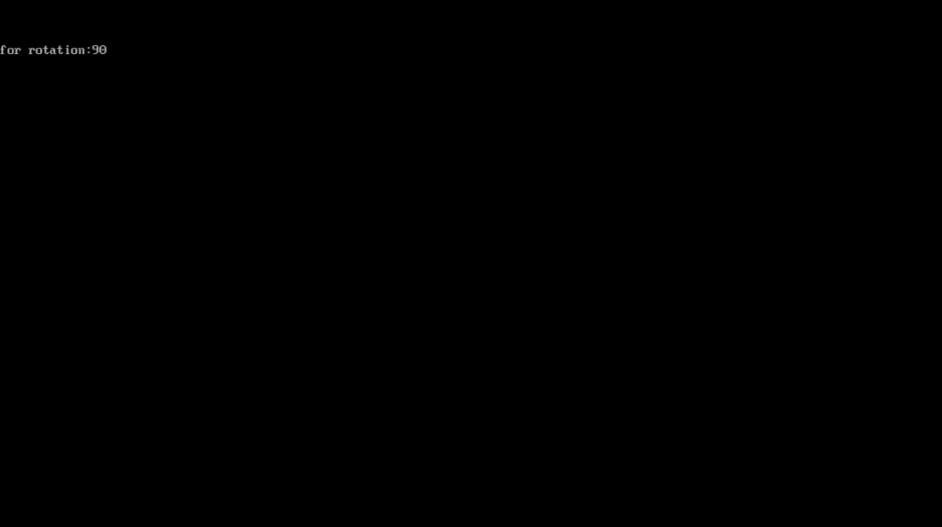



NeuTrON DOS-C++ 0.77, Cpu speed: max 100% cycles, Frameskip 0, Program TC

Enter the 1st point for the triangle:200
100

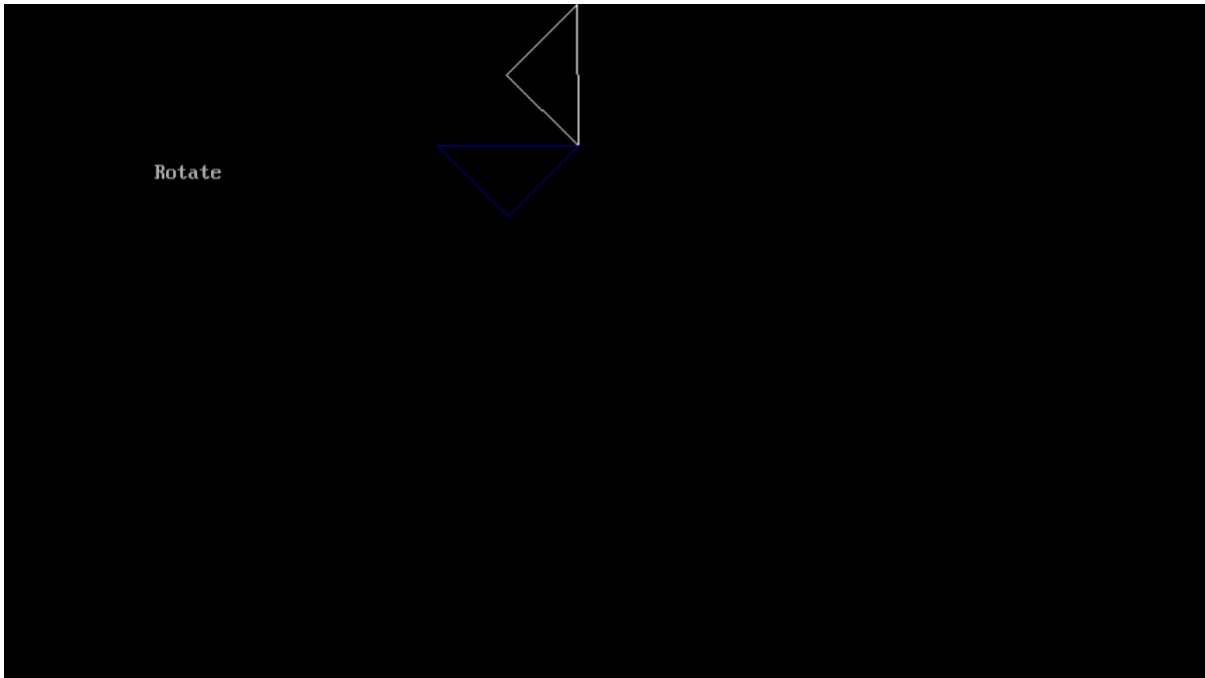
Enter the 2nd point for the triangle:300
100

Enter the 3rd point for the triangle:250
150



NeuTroN DOS-C++ 0.77, Cpu speed: max 100% cycles, Frameskip 0, Program: TC

Enter the angle for rotation:90



RESULT:

This the program has been executed successfully and result is verified

EX:NO: 02

DROPPING A WORD

AIM:

To write a program to drop each word of a sentence one by one from the top.

ALGORITHM:

Step1:Start the process.

Step2: Declare variables i.

Step3: Set the background color of the screen to be white.

Step4: Using the function OUTTEXTXY() print the text on the screen.

Step5: Set the color of the text to be in BLUE color.

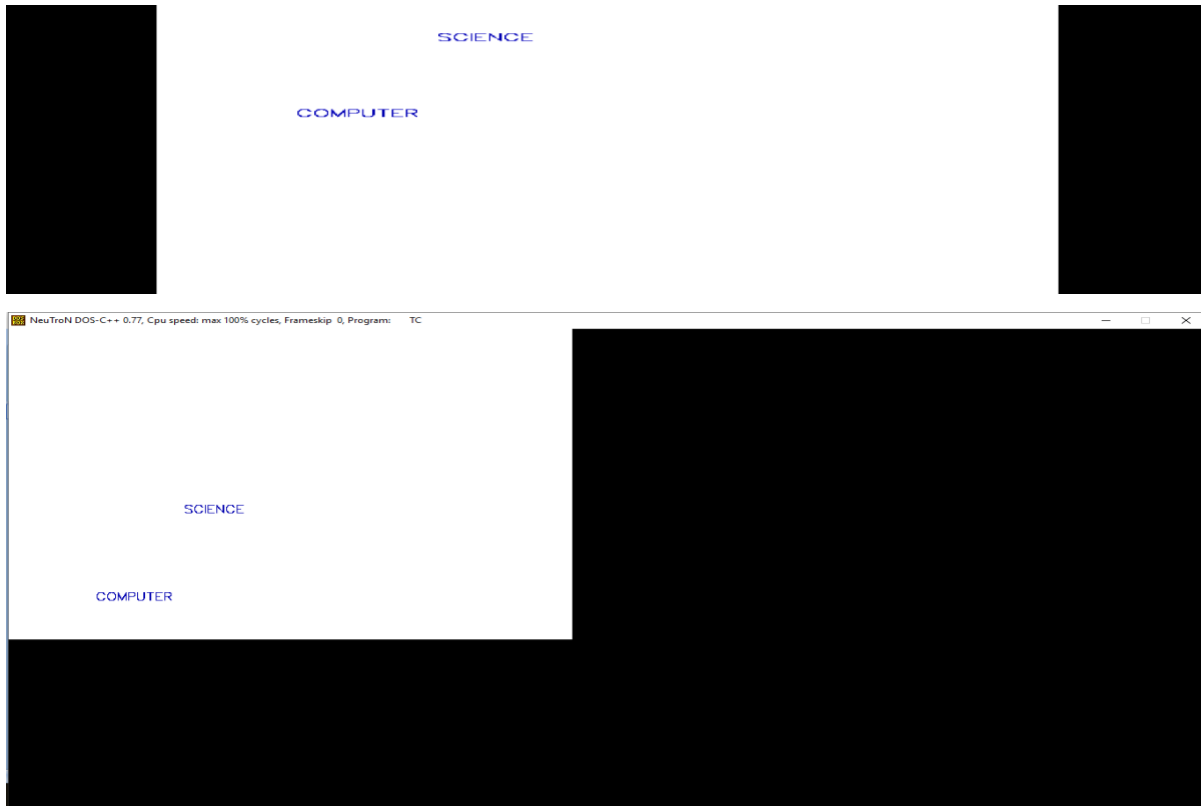
Step6: Drop the word one by one in the text by incrementing the Y coordinates value and using delay function.

Step7:Close Graph and stop.

SOURCE CODE:

```
#include<iostream.h>
#include<conio.h> #include<graphics.h>
#include<dos.h>
void main()
{
    int gd=DETECT,gm;
    initgraph(&gd,&gm,"..\\bgi");
    int i;
    setbkcolor(WHITE); // Set the background color
    for(i=0;i<350;i++)
    {
        settextstyle(3,0,1);
        outtextxy(200,40,"SCIENCE");
        setcolor(BLUE);
        outtextxy(100,40+i,"COMPUTER");
        delay(25);
        cleardevice();
    }
    for(i=0;i<350;i++)
    {
        outtextxy(100,400,"COMPUTER");
        setcolor(BLUE);
        outtextxy(200,40+i,"SCIENCE");
        delay(25);
        cleardevice();
    }
    outtextxy(100,400,"COMPUTER");
    outtextxy(200,400,"SCIENCE");
    getch();
    closegraph();}
```


OUTPUT:



RESULT:

This the program has been executed successfully and result is verified

EX:NO: 03

DRAWING A LINE USING DDA ALGORITHM

AIM:

To write a program to draw a line using DDA Algorithm.

ALGORITHM:

Step1: Start the process.

Step2: Declare variables $x, y, x_1, y_1, x_2, y_2, k, dx, dy, s, x_i, y_i$ and also declare $gdriver = DETECT$, mode.

Step3: Initialize the graphic mode with the path location in TurboC3 folder.

Step4: Input the two line end-points and store the left end-points in (x_1, y_1) .

Step5: Load (x_1, y_1) into the frame buffer; that is, plot the first point. put $x = x_1, y = y_1$.

Step 6: Calculate $dx = x_2 - x_1$ and $dy = y_2 - y_1$.

Step 7: If $\text{abs}(dx) > \text{abs}(dy)$, do $s = \text{abs}(dx)$.

Step8: Otherwise $s = \text{abs}(dy)$.

Step9: Then $x_i = dx/s$ and $y_i = dy/s$.

Step10: Start from $k=0$ and continuing till k

i. $x = x + x_i$.

ii. $Y = y + y_i$.

Step11: Plot pixels using `putpixel` at points (x, y) in specified colour.

Step12: Close Graph and stop.

SOURCE CODE:

```
#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, "..\\bgi");
    setbkcolor(WHITE);
    printf("Enter X0,Y0,X1,Y1:");
    scanf("%d %d %d %d",&x0,&y0,&x1,&y1);
    dx = (float)(x1 - x0);
    dy = (float)(y1 - y0);
    if(dx>=dy)
        steps = dx;
    else
        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();  
}
```

OUTPUT:



RESULT:

This the program has been executed successfully and result is verified

EX:NO: 04

MOVING A CAR

AIM:

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

ALGORITHM:

Step 1: Start the process.

Step 2: Create a function draw_car() to draw a car using line() and circle() commands.

Step 3: In the main function declare the variable gd and gm, auto detection of graphical driver is done using the variable gd.

Step 4: Initiate the graphical driver and graphic mode using init graph function.

Step 5: Invoke the draw_car function inside the main function.

Step 6: Using for loop move the car on the screen.

Step 7: Display the car with sound effects on the output screen using putimage() and sound() commands.

Step 8: Stop the process.

SOURCE CODE:

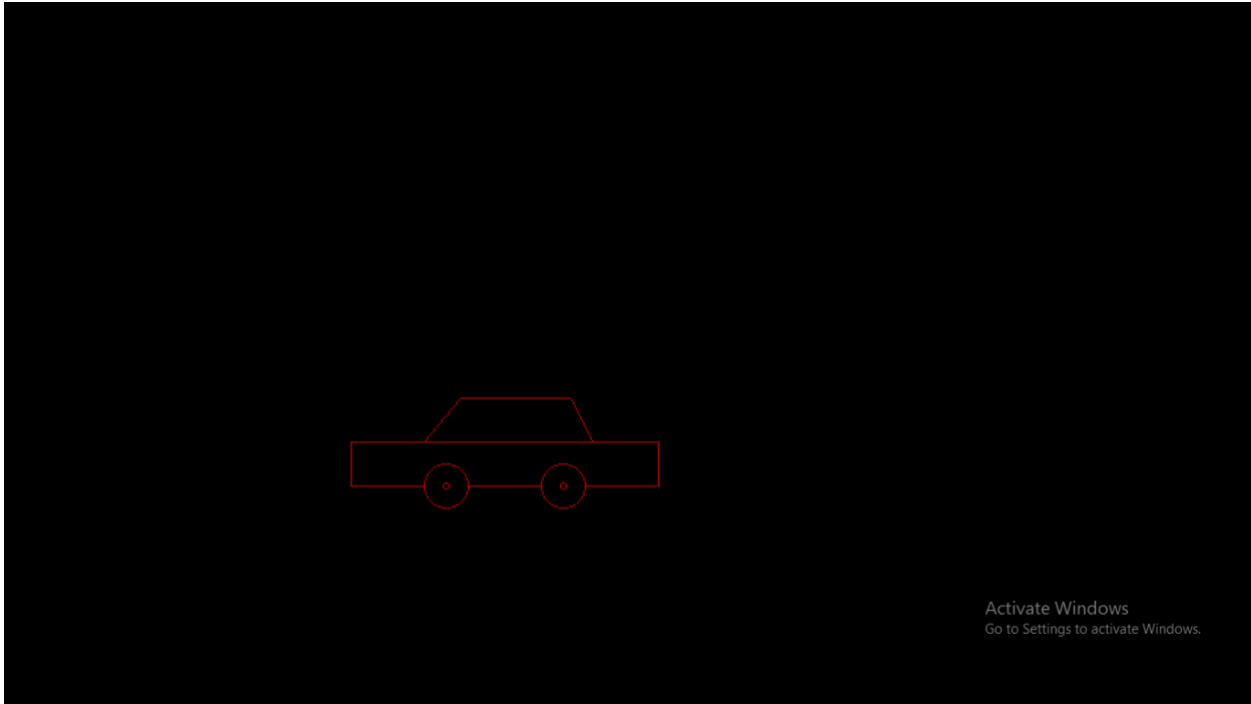
```
#include <graphics.h>
#include <stdio.h>

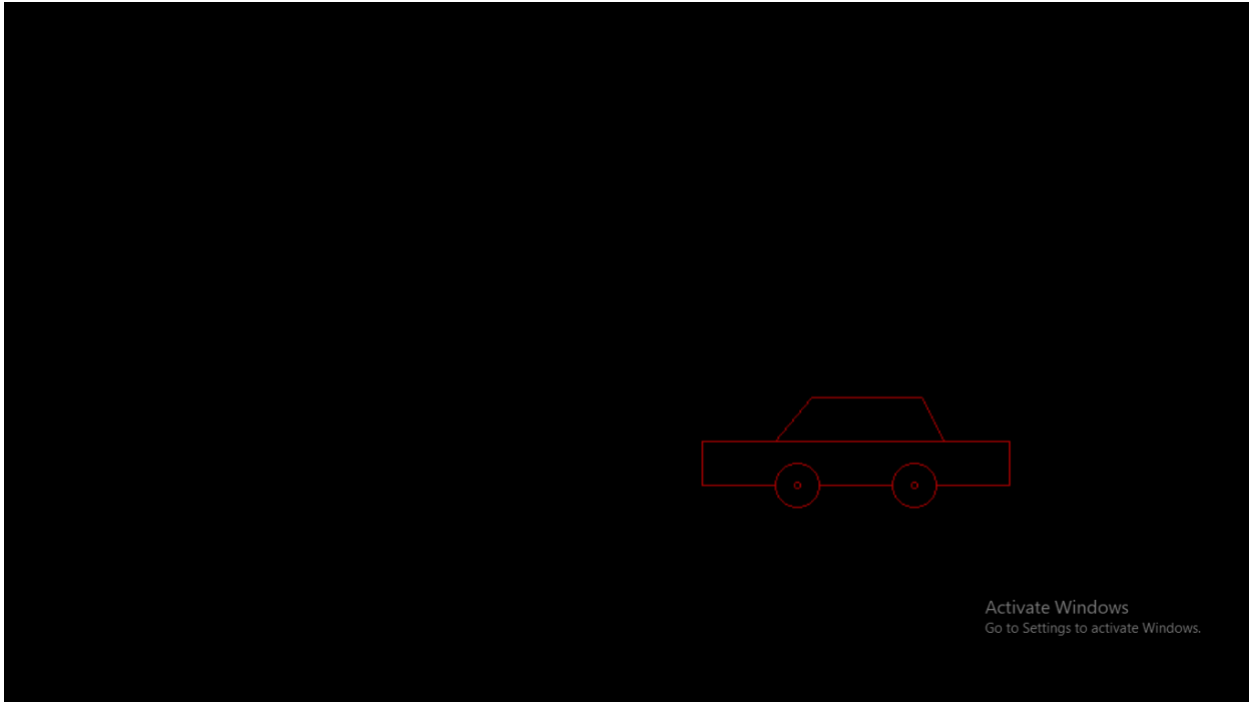
void draw_moving_car(void)
{
    int i, j = 0, gd = DETECT, gm;
    initgraph(&gd, &gm, "");
    for (i = 0; i <= 420; i = i + 10)
    {
        setcolor(RED);
        line(0 + i, 300, 210 + i, 300);
        line(50 + i, 300, 75 + i, 270);
        line(75 + i, 270, 150 + i, 270);
        line(150 + i, 270, 165 + i, 300);
        line(0 + i, 300, 0 + i, 330);
        line(210 + i, 300, 210 + i, 330);
        circle(65 + i, 330, 15);
        circle(65 + i, 330, 2);
        circle(145 + i, 330, 15);
        circle(145 + i, 330, 2);
        line(0 + i, 330, 50 + i, 330);
        line(80 + i, 330, 130 + i, 330);
        line(210 + i, 330, 160 + i, 330);
        delay(100);
        setcolor(BLACK);
        line(0 + i, 300, 210 + i, 300);
        line(50 + i, 300, 75 + i, 270);
        line(75 + i, 270, 150 + i, 270);
        line(150 + i, 270, 165 + i, 300);
        line(0 + i, 300, 0 + i, 330);
```

```
        line(210 + i, 300, 210 + i, 330);
        circle(65 + i, 330, 15);
        circle(65 + i, 330, 2);
        circle(145 + i, 330, 15);
        circle(145 + i, 330, 2);
        line(0 + i, 330, 50 + i, 330);
        line(80 + i, 330, 130 + i, 330);
        line(210 + i, 330, 160 + i, 330);
    }
    getch();
    closegraph();
}

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

OUTPUT:





RESULT:

This the program has been executed successfully and result is verified

EX:NO: 05

BOUNCING A BALL

AIM:

To write a program to bounce a ball and move it with the sound effect.

ALGORITHM:

Step 1: Start the process.

Step 2: Declare the variable gd and gm, auto detection of graphical driver is done using the variable gd.

Step 3: Initiate the graphical driver and graphic mode using init graph function.

Step 4: Declare the pointer variable ball as a global variable.

Step 5: Declare the function image and draw the ball using fillellipse() and fill the color for the ball using setfillstyle().

Step 6: Declare the necessary variable in the main function.

Step 7: Move the ball using the variable mx and my with sound effects.

Step 8: Stop the process.

SOURCE CODE:

```
#include<graphics.h>
#include<conio.h>
#include<alloc.h>
#include<dos.h>
#include<stdlib.h>
void *ball;
void image()
{
    setcolor(RED);
    setfillstyle(SOLID_FILL, GREEN);
    fillellipse(10,10,10,10);
    ball=malloc(imagesize(0,0,20,20));
    getimage(0,0,20,20,ball);
    cleardevice();
}
void main()
{
    int gm,gd=DETECT;
    initgraph(&gd,&gm,"..\\bgi");
    int mx=getmaxx()/2,my=0;
    int x=1,y=1,s=0,key=0,xstep=1,ystep=1;
    image();
    setbkcolor(WHITE);
    while(key!=27)
    {
        while(!kbhit())
        {
            putimage(mx,t,ball,XOR_PUT);
            delay(5);
            putimage(mx,t,ball,XOR_PUT);
```

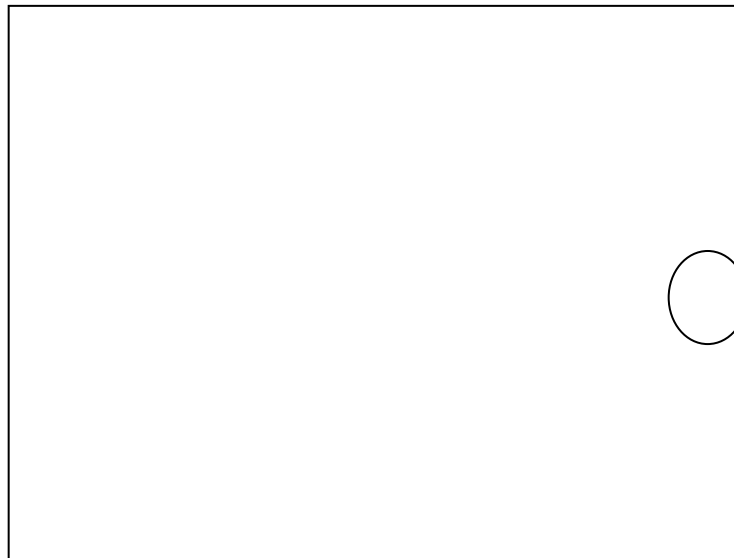
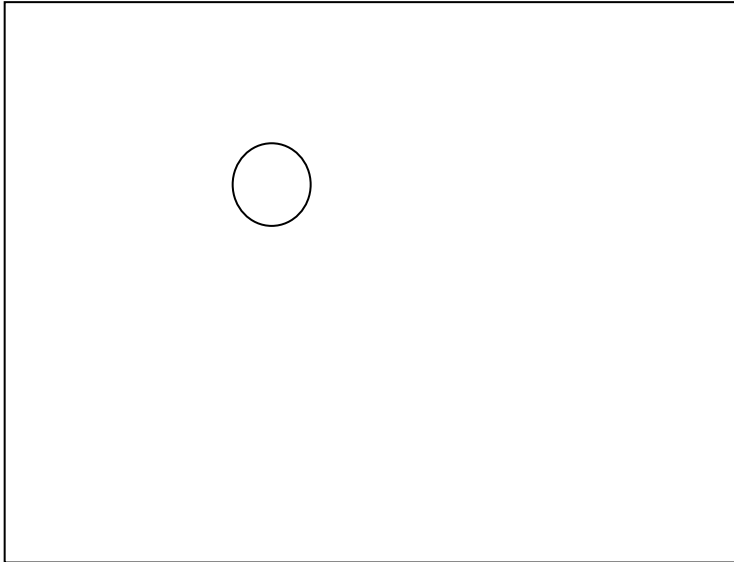
```

        if(mx>getmaxx()||mx<=0)
        {
            x*=-1;
            sound(1000);
            s=0;
            xstep=x*(random(4)+1);
            ystep=y*(random(3)+1);
            if(mx<=0)
            mx=0;
            else
            mx=getmaxx();
        }
        if(my>getmaxy()||my<=0)
        {
            y*=-1;
            sound(100);
            s=0;
            ystep=y*(random(4)+1);
            xstep=x*(random(3)+1);
            if(my<=0)
            my=0;
            else
            my=getmaxy();
        }
        mx+=x+xstep;
        my+=y+ystep;
        s++;
        if(s==5)
        {
            nosound();
        }
    }
}

```

```
        key=getch();  
    }  
    closegraph();  
}
```

OUTPUT:



RESULT:

This the program has been executed successfully and result is verified

EX:NO:6

POLYGON CLIPPING

AIM:

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

ALGORITHM:

Step 1: Start the process.

Step 2: Declare the variable gd and gm, auto detection of graphical driver is done using the variable gd.

Step 3: Draw the polygon with given points.

Step 4: Set the values for a pixel co-ordinate.

Step 5: Using a “pointInpoly()” checking the pixel position.

Step 6: Based on the boundary values, print the result as “Inside or outside the boundary.

Step 7: close the graphical mode.

Step 8: Stop the process.

SOURCE CODE:

```
#include<conio.h>
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
#include<graphics.h>
int polyx[6]={ 540,590,570,510,490,540};
int polyy[6]={ 220,270,320,320,270,220};
int polysides=5;
int x,y;
int pointInpoly();
void draw_polygon();
void main()
{
    int c,gd=DETECT,gm;
    initgraph(&gd,&gm,"..\\bgi");
    draw_polygon();
    printf("\nEnter the value of x:");
    scanf("%d",&x);
    printf("\n Enter the value of y:");
    scanf("%d",&y);
    putpixel(x,y,WHITE);
    c=pointInpoly();
    if(c==0)
        printf("\n point is outside the polygon");
    else
        printf("\n point is inside the polygon");
    getch();
}
void draw_polygon()
```

```

{
    int i,j;
    for(i=0;i<polysides;i++)
    {
        if(i==(polysides-1))
            line(polyX[i],polyY[i],polyX[0],polyY[0]);
        else
            line(polyX[i],polyY[i],polyX[i+1],polyY[i+1]);
    }
    int pointInpoly()
    {
        int i,j=polysides-1;
        int oddnodes=0;
        for(i=0;i<polysides;i++)
        {
            if(polyY[i]<y&&polyY[j]>=y||polyY[j]<y&&polyY[i]>=y)
            {
                if(polyX[i]+(y-polyY[i])/(polyY[j]-polyY[i]*(polyX[j]-polyX[i])<x))
                {
                    oddnodes=1;
                }
            }
            j=i;
        }
        return oddnodes;
    }
}

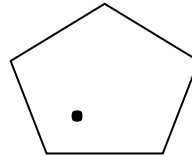
```


OUTPUT:

ENTER THE VALUE FOR X: 550

ENTER THE VALUE FOR Y: 300

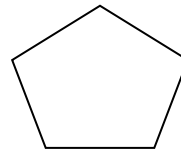
Inside



ENTER THE VALUE FOR X: 350

ENTER THE VALUE FOR Y: 400

Outside



RESULT:

This the program has been executed successfully and result is verified

EX:NO:7

CREATE SUN FLOWER USING PHOTOSHOP

AIM:

To Create a Sunflower using Photoshop.

ALGORITHM:

Step 1: Click the file menu and select a new Photoshop window.

Step 2: Select the Paint bucket tool from the toolbox and fill the background color to the window.

Step 3: Select the shape tool from the toolbox and draw the flower shape in the window and give the appropriate color to the sunflower.

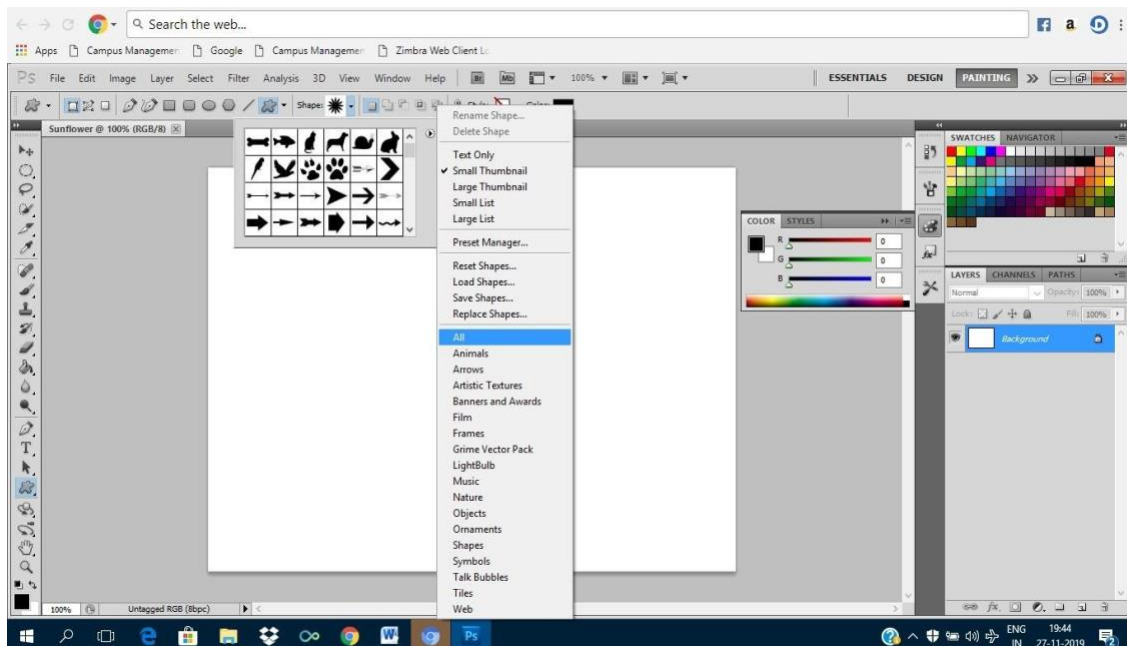
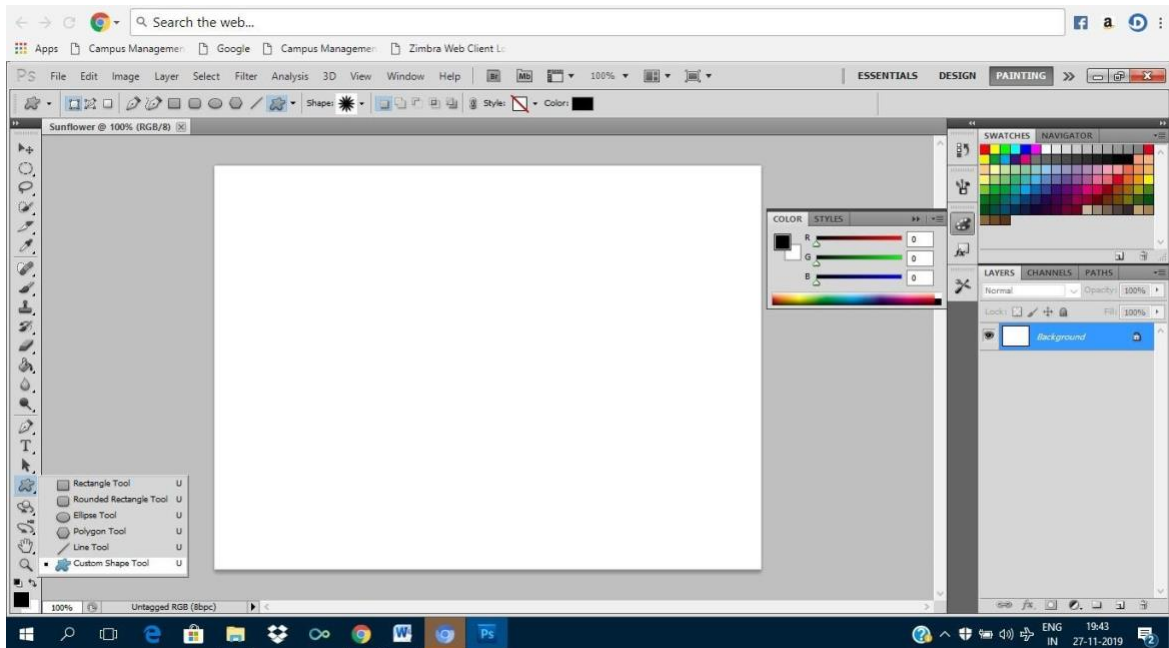
Step 4: Select the elliptical marquee tool from the toolbox and draw the circle shape in the middle of the flower and fill with appropriate color, and using the brush tool draw the seeds in the circle.

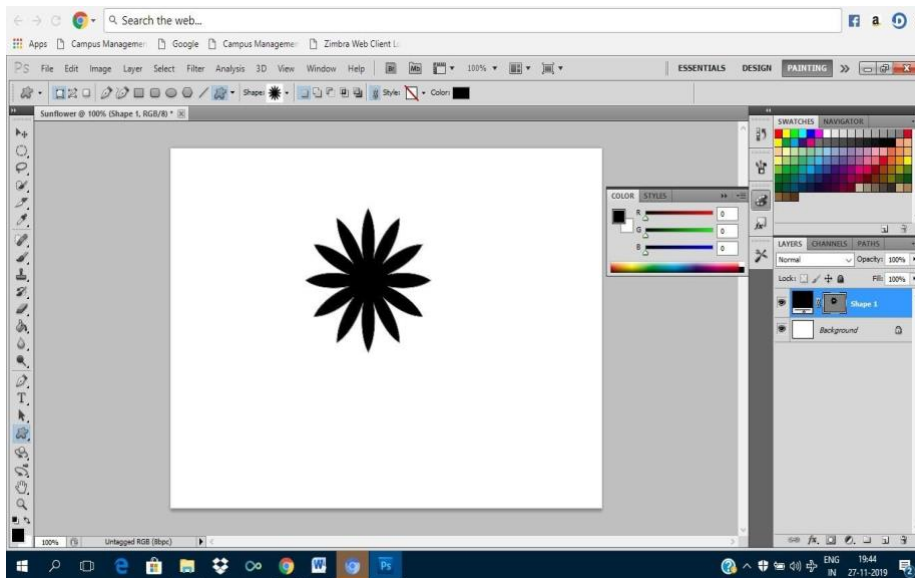
Step 5: Using the shape tool select the bended arrow, and place it below the flower like a stem. Hide the arrow behind the flower and fill with color.

Step 6: Using the shape tool select the leaf like structure, and by selecting the move tool, and check the transform control to change the direction and place it properly on the stem. **Step 7:** Goto file menu and click save.

Step 8: Save it as .jpg or .psd format.

OUTPUT:





RESULT:

The above program has been verified and executed successfully.

EX:NO: 8

ANIMATE A PLANE FLYING THE CLOUDS USING PHOTOSHOP

AIM:

To animate a plane flying the clouds using Photoshop.

ALGORITHM:

Step 1: Click the file menu and select a new Photoshop window.

Step 2: Make the background color as blue using paint bucket tool or fill color option.

Step 3: Make the clouds effects using the filter menu option.

Step 4: Create a new layer and the draw the rectangle using rectangular Marqueetool in the toolbar.

Step 5: Connect the marquee tool in front of the rectangle. Draw a line back of the rectangle using line tool.

Step 6: Then click the background layer and color it using the paint brush tool.

Step 7: Draw the wings of the plane and include a color.

Step 8: Then move it using move tool to the bottom of the layer.

Step 9: Choose window in the menu bar and then select the animation option.

Step 10: Create new layer and move the plane from top to bottom and bottom totop.

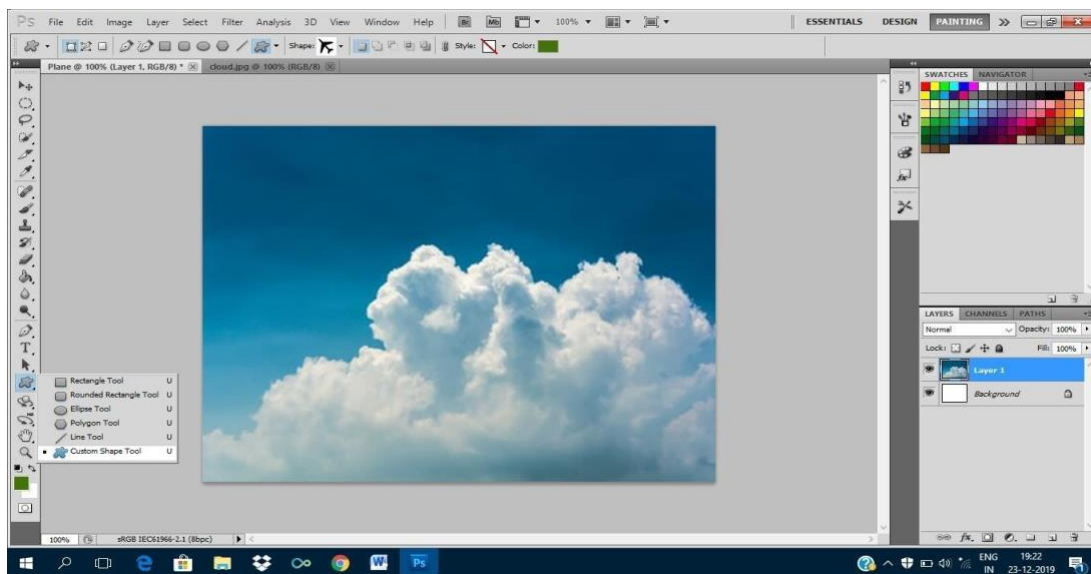
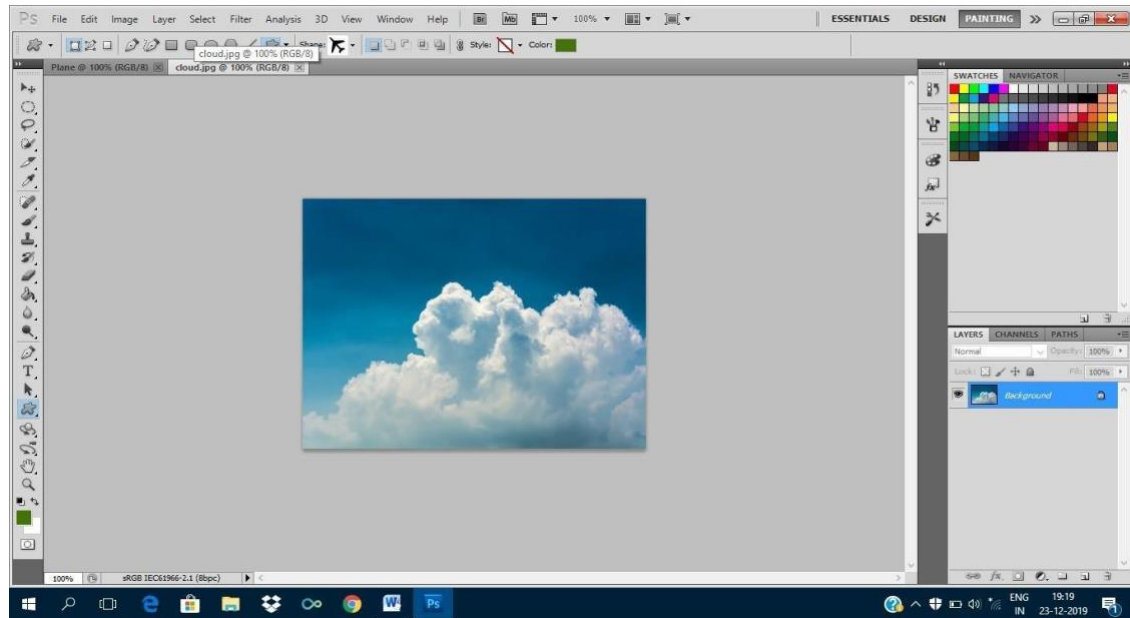
Step 11: This process is continued until the plane is reached at bottom or top.

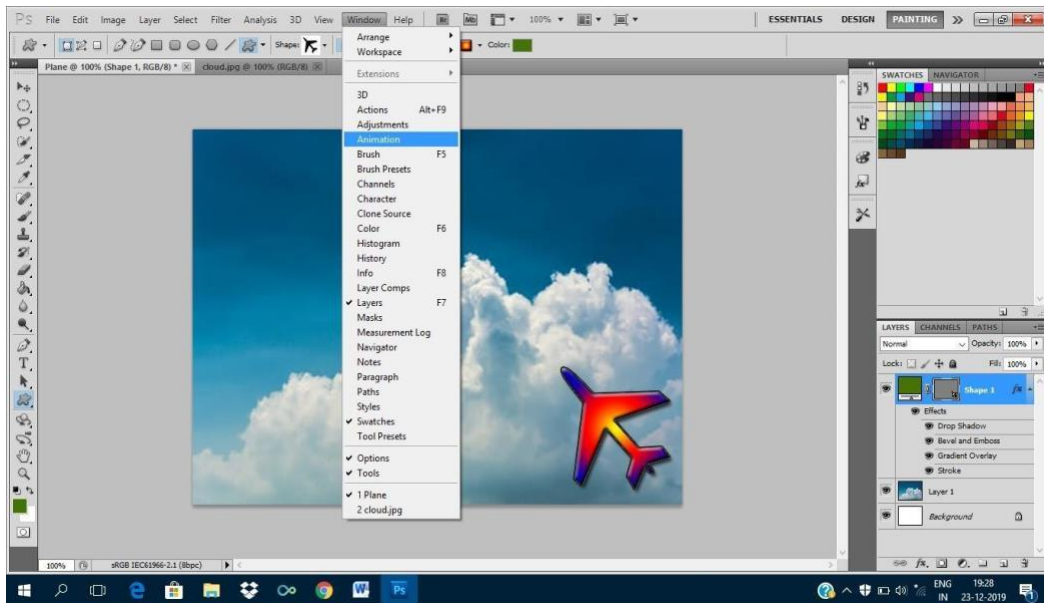
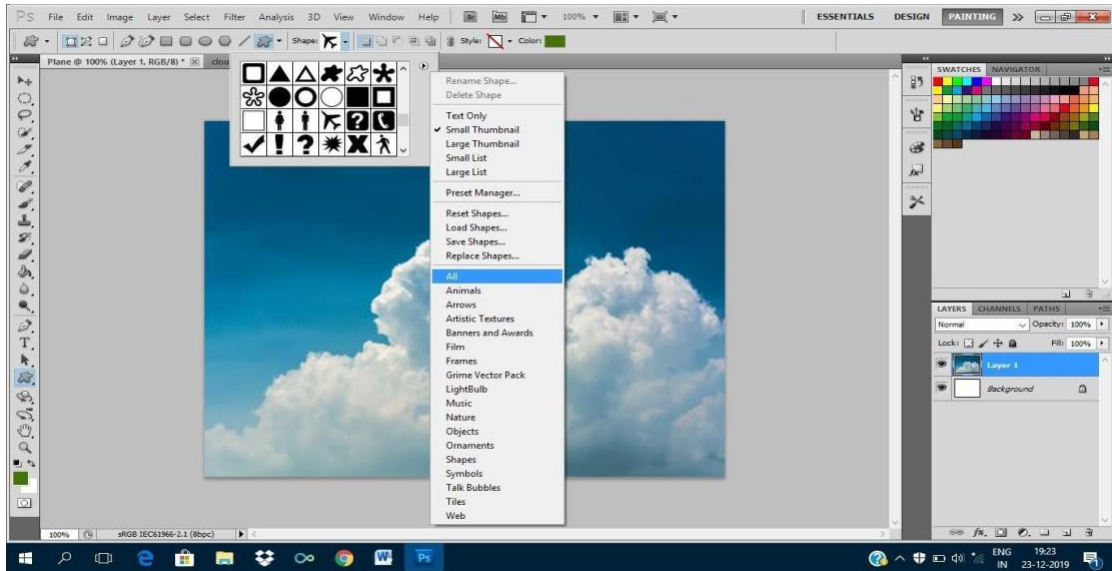
Step 12: Click the play button to play the animation scheme.

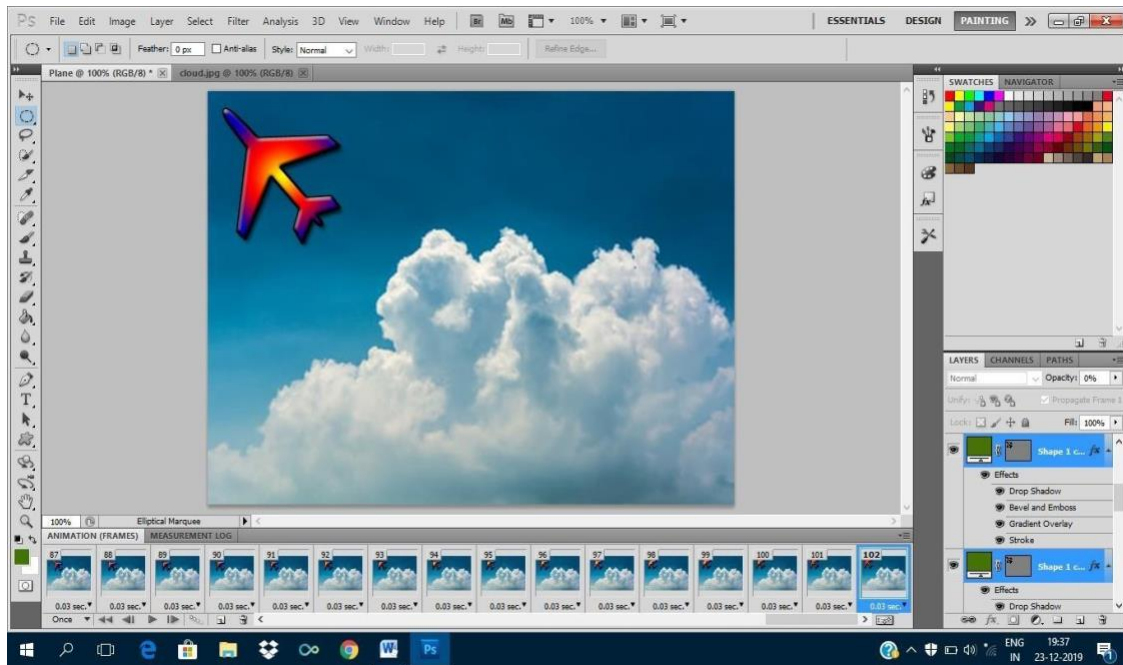
Step 13: Click the stop button.

Step 14: The image can be saved as .gif format.

OUTPUT:







RESULT:

The above program has been verified and executed successfully.

EX:NO:9 CREATE PLASTIC SURGERY FOR THE NOSE USING PHOTOSHOP

AIM :

To create plastic surgery for the nose using Photoshop.

ALGORITHM:

Step 1: Download the picture from internet and paste that picture in Photoshop file

Step 2: Draw a rough selection around the outside of the nose by using lasso tool.

Step 3: Select the feather option, using the keyboard shortcut as Alt+Ctrl+D.

Step 4: Enter the feather radius values of somewhere between 20-30 pixels

Step 5: Select the liquefy option under the image menu.

Step 6: Using the warp tool and pucker tool to adjust the nose

Step 7: Change the brush stroke as 65.

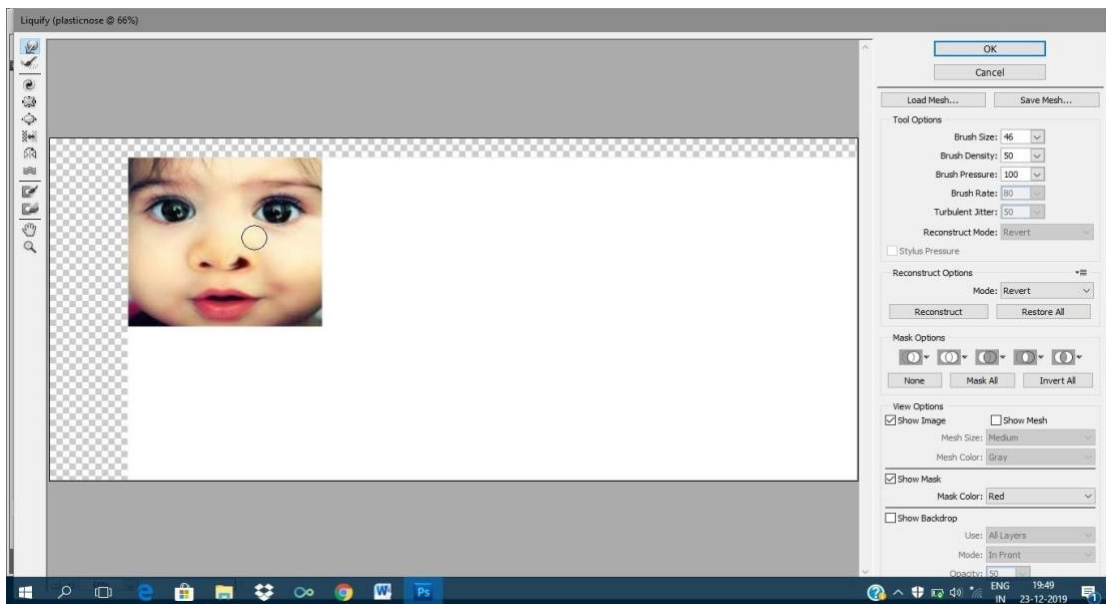
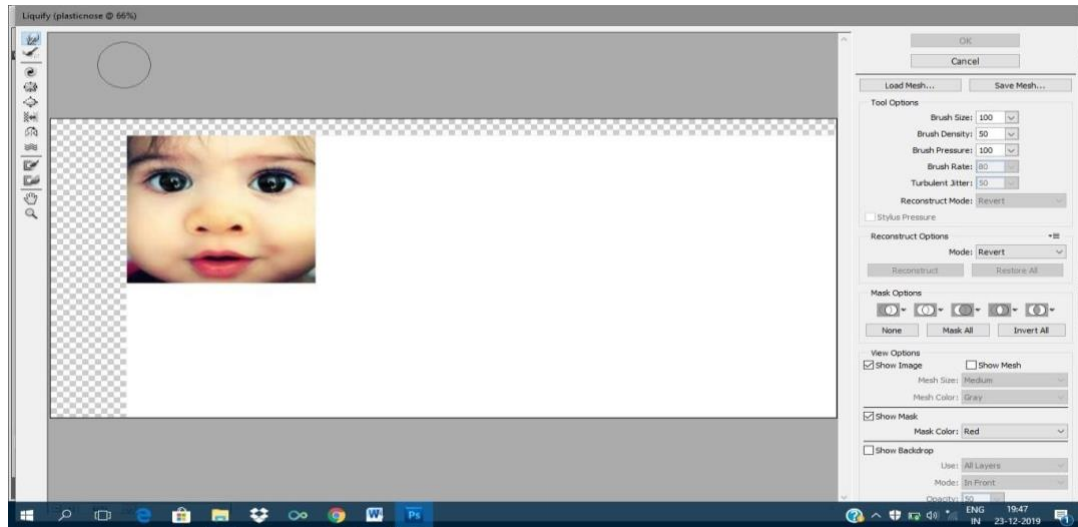
Step 8: Use all the tools in the liquefy option.

Step 9: Instead of step5-step8, using Ctrl+J or copy the selection to new layer.

Step 10: Drag the nose by using Ctrl+J.

Step 11: The image can be saved as .jpg or .bmp format.

OUTPUT:



RESULT:

The above program has been verified and executed successfully.

EX:NO:10

CREATE SEE THROUGH TEXT USING PHOTOSHOP

AIM

To create see through text using Photoshop.

ALGORITHM:

Step 1: Open your image

Step 2: Add a new layer. In the Layers panel, we see the image on the Background layer. Add a new blank layer above the image by clicking the **New Layer** icon at the bottom of the layers panel.

Step 3: Fill the layer with white by going up to the **Edit** menu in the Menu Bar and choosing **Fill**:

Step 4: Lower the opacity of "Layer 1" to **75%**. The Opacity option is found in the upper right of the Layers panel:

Step 5: Select the text Tool from the Toolbar

Step 6: Choose your font in the **Options Bar**. Choose your font in the **Options Bar**.

Step 7: Make sure your type color is set to **black**

Step 8: Click inside the document and add your text. I'll type the words "NEW YORK CITY":

Step 9: Click the **checkmark** in the Options Bar to accept it

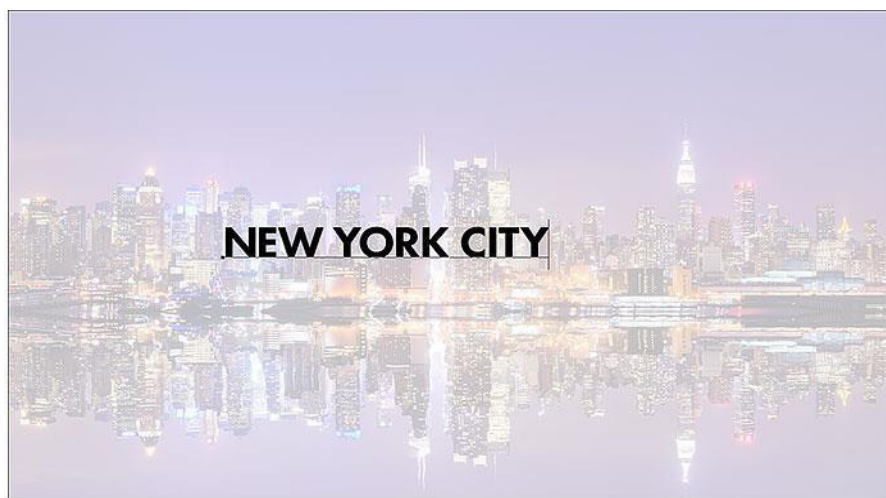
Step 10: Go up to the **Edit** menu in the Menu Bar and choose Free Transform to resize the text

Step 11: With the Type layer selected, click on the **Layer Styles** icon (the **fx** icon) at the bottom of the Layers panel. Choose **Blending Options** from the top of the list

Step 12: In the Advanced Blending section, change **Knockout** from None to **Shallow**. To see the effect, drag the **Fill Opacity** slider all the way down to **0%**.

Step 13: Click on **Layer 1** in the Layers panel to select it Go up to the **Edit** menu and once again choose **Free Transform**: Finally, adjust the opacity of "Layer 1" to 85% for the visibility of the text.

OUTPUT:





RESULT:

The above program has been verified and executed successfully.

EX:NO:11

CREATE A WEBPAGE USING PHOTOSHOP

AIM:

To create a webpage using Photoshop.

ALGORITHM:

Step 1: Click the file menu and select a new Photoshop window.

Step 2: Fill the background color as you like using paint bucket tool or fill color option.

Step 3: Select the rectangular marquee tool using it to the shape of buttons.

Step 4: Select the stroke option from the menu.

Step 5: Change the size, position, mode, opacity and color.

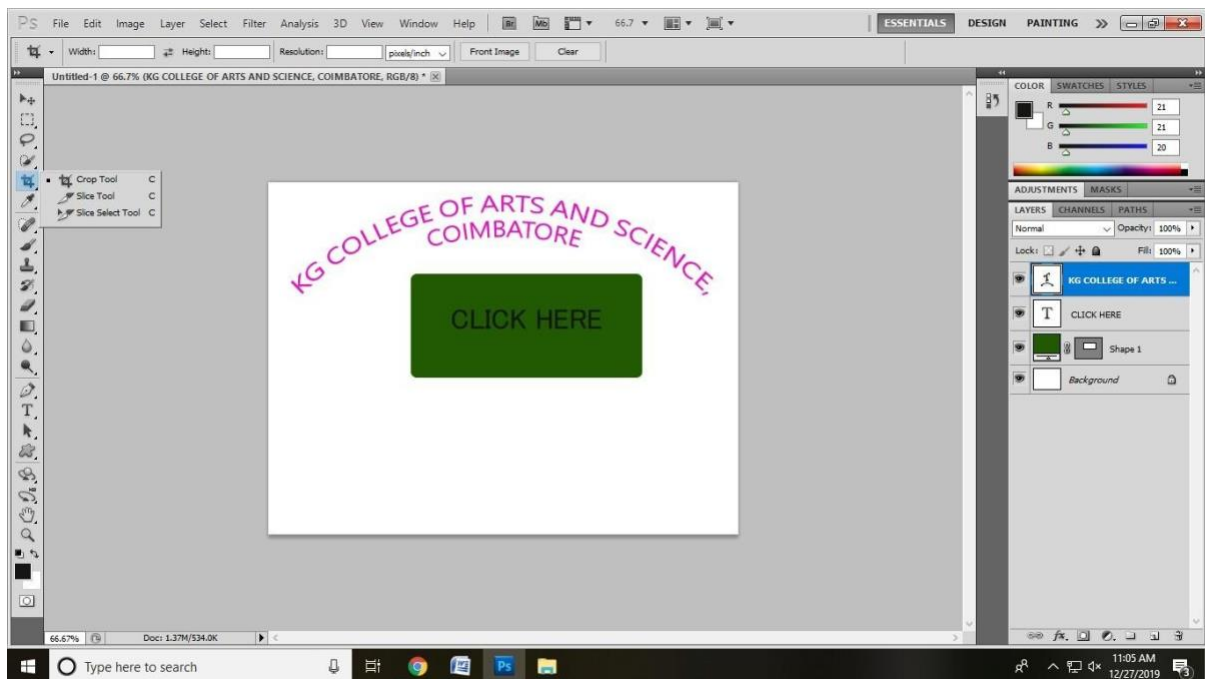
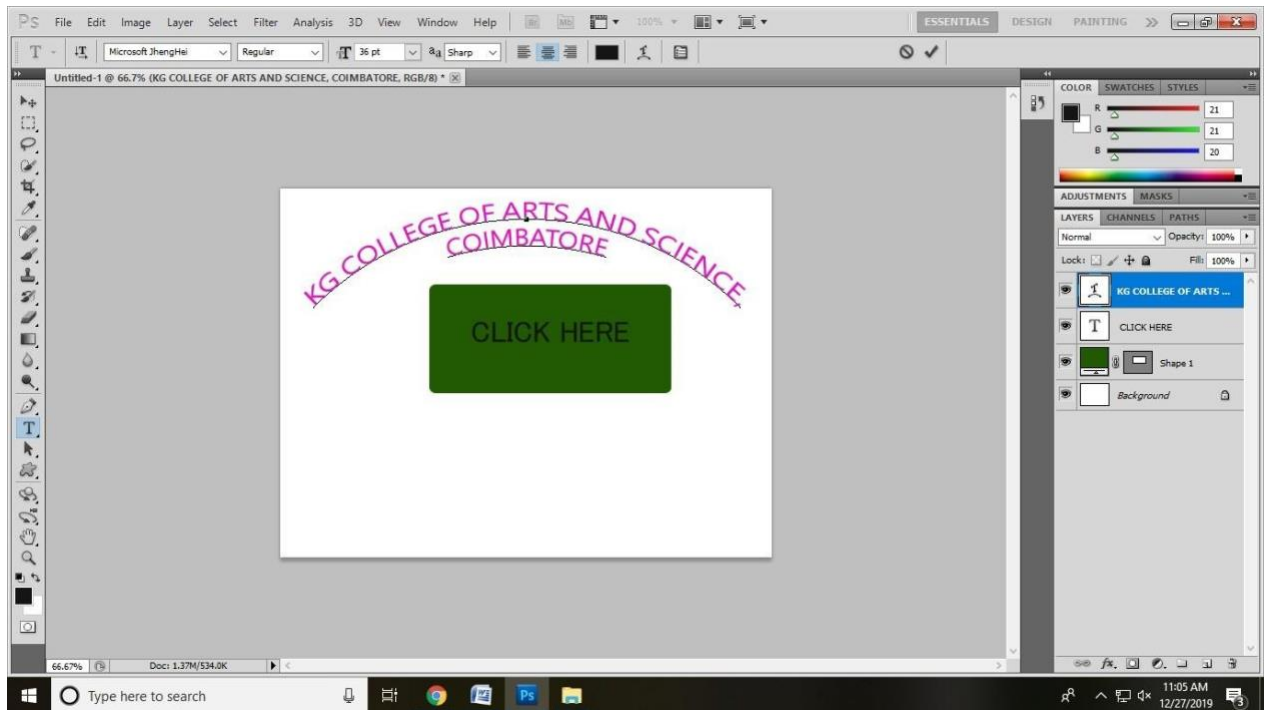
Step 6: Then select the texture option to change the pattern, scale and depth.

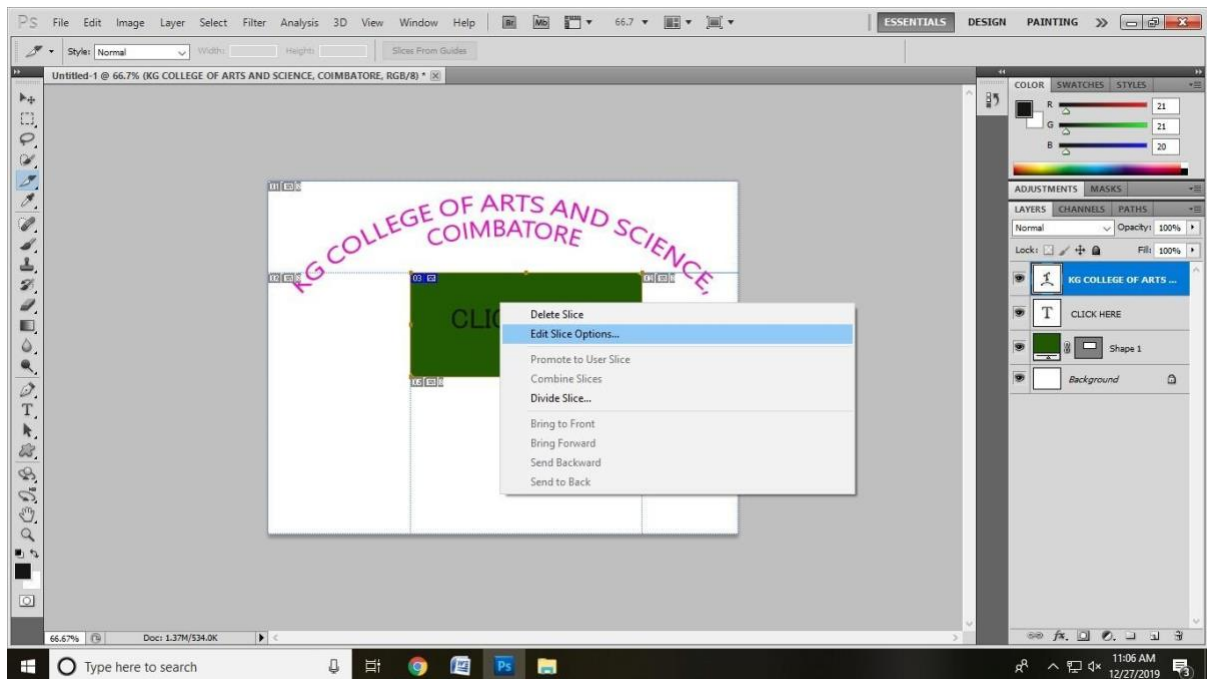
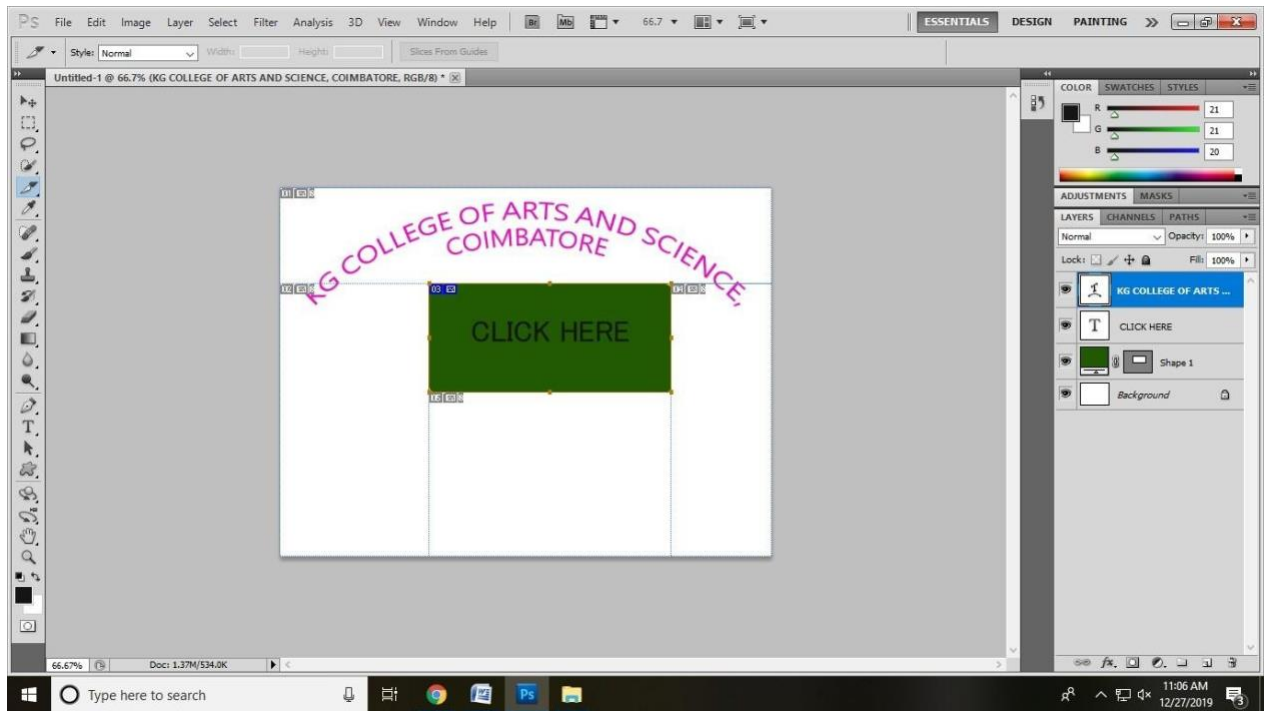
Step 7: Similarly add the styles such as Inner glow, drop shadow ,color overlay and pattern overlay etc.

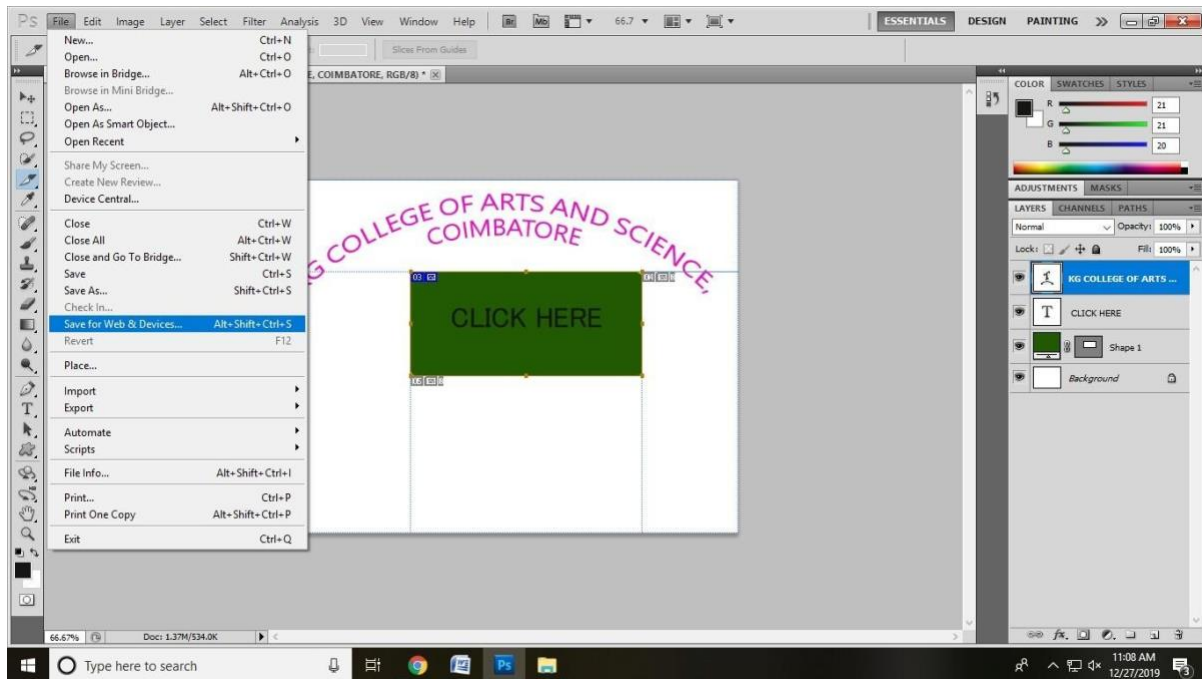
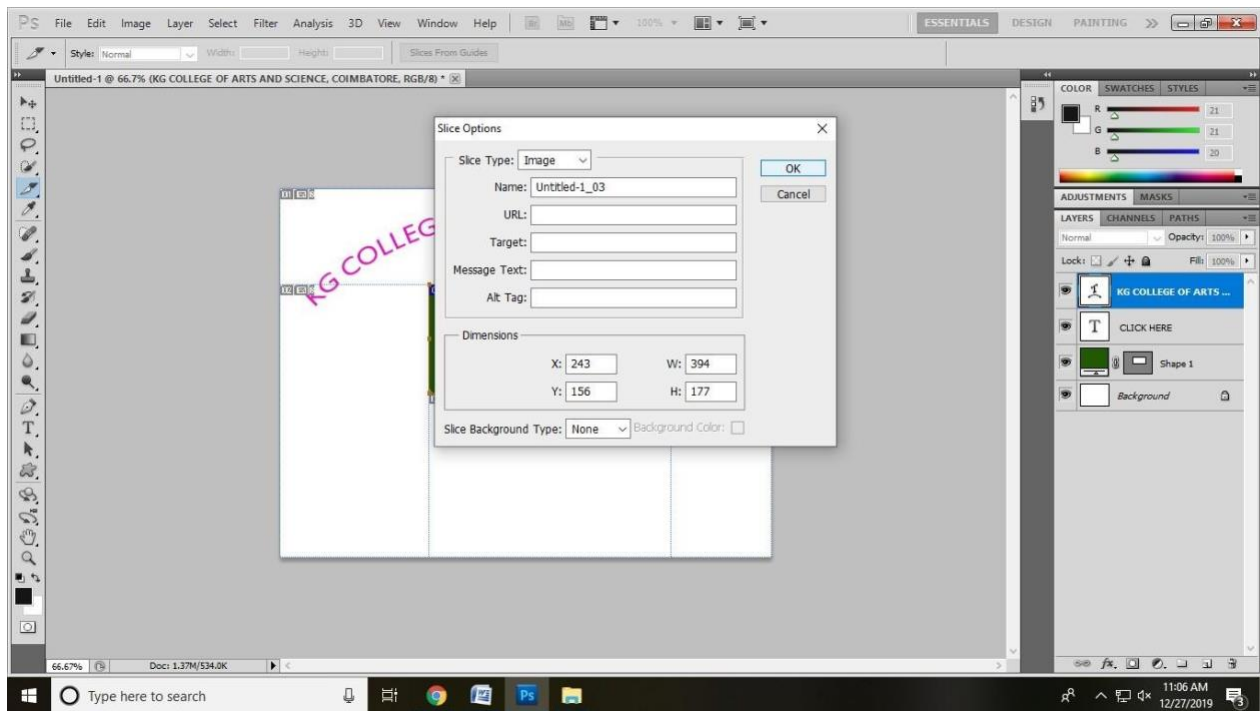
Step 8: Create a new layer and create a new text to place on the buttons. The textssuch as About Us, Departments, admission and contact us.

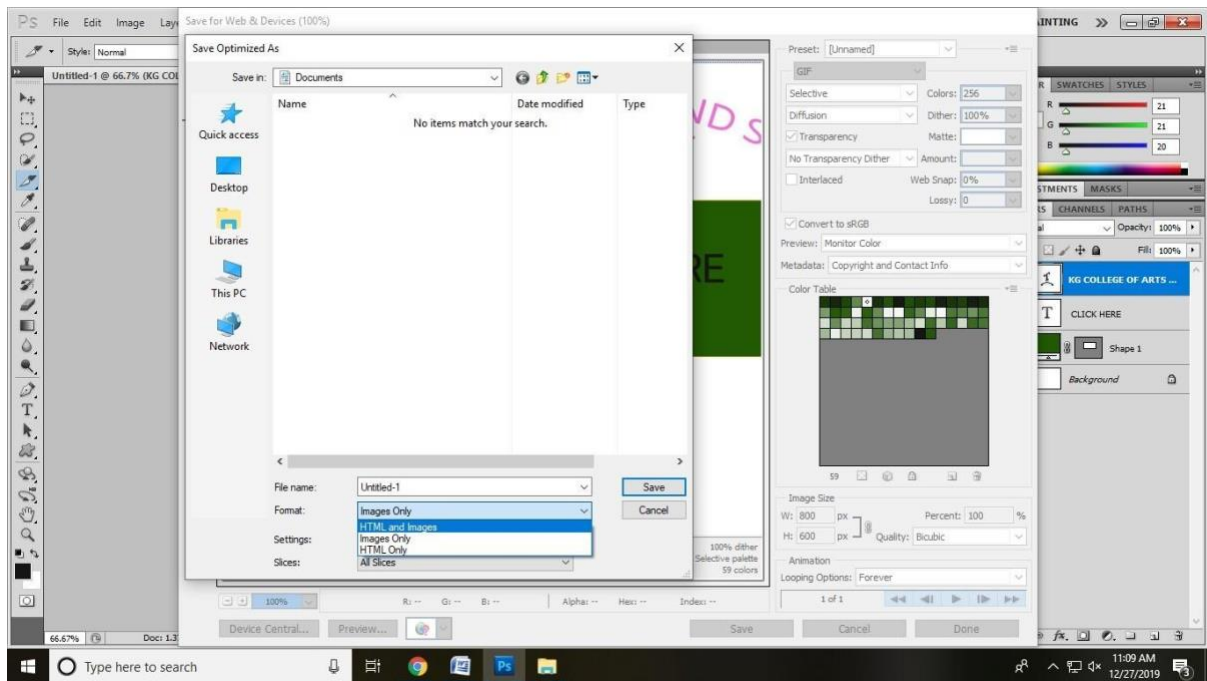
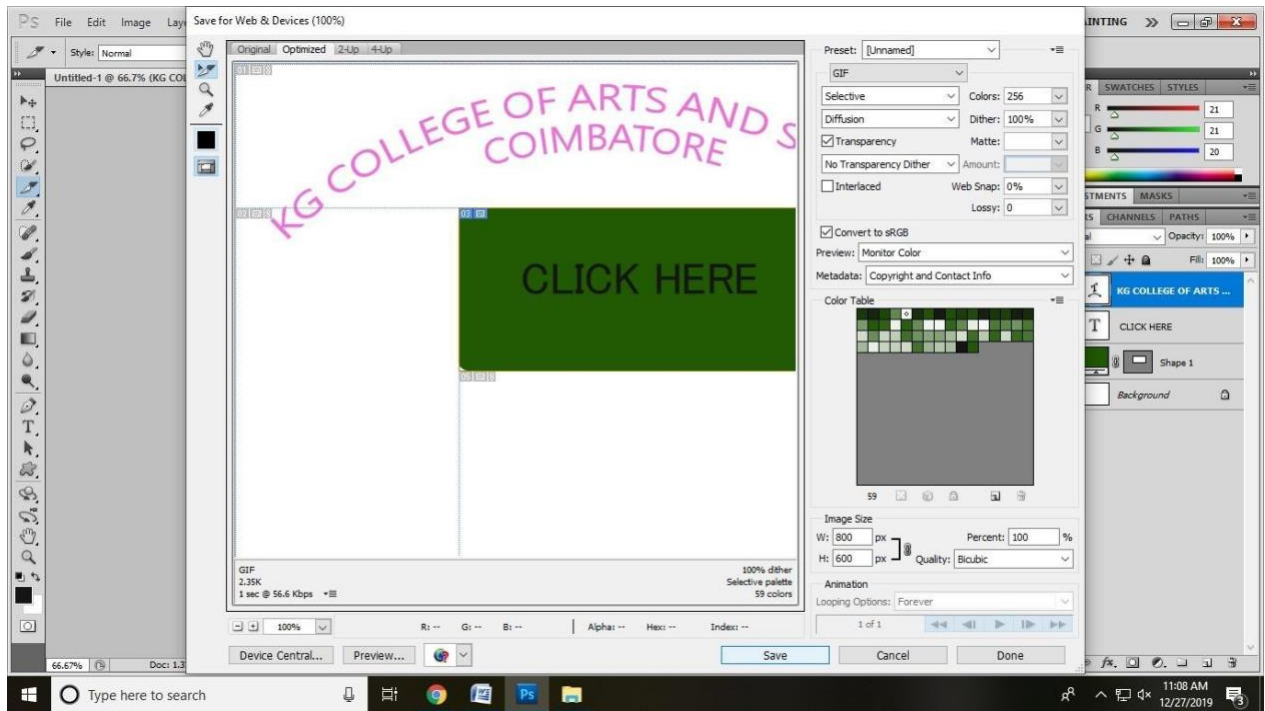
Step 9: The webpage is successfully created. Then the page can be saved as .jpg or .psd formats.

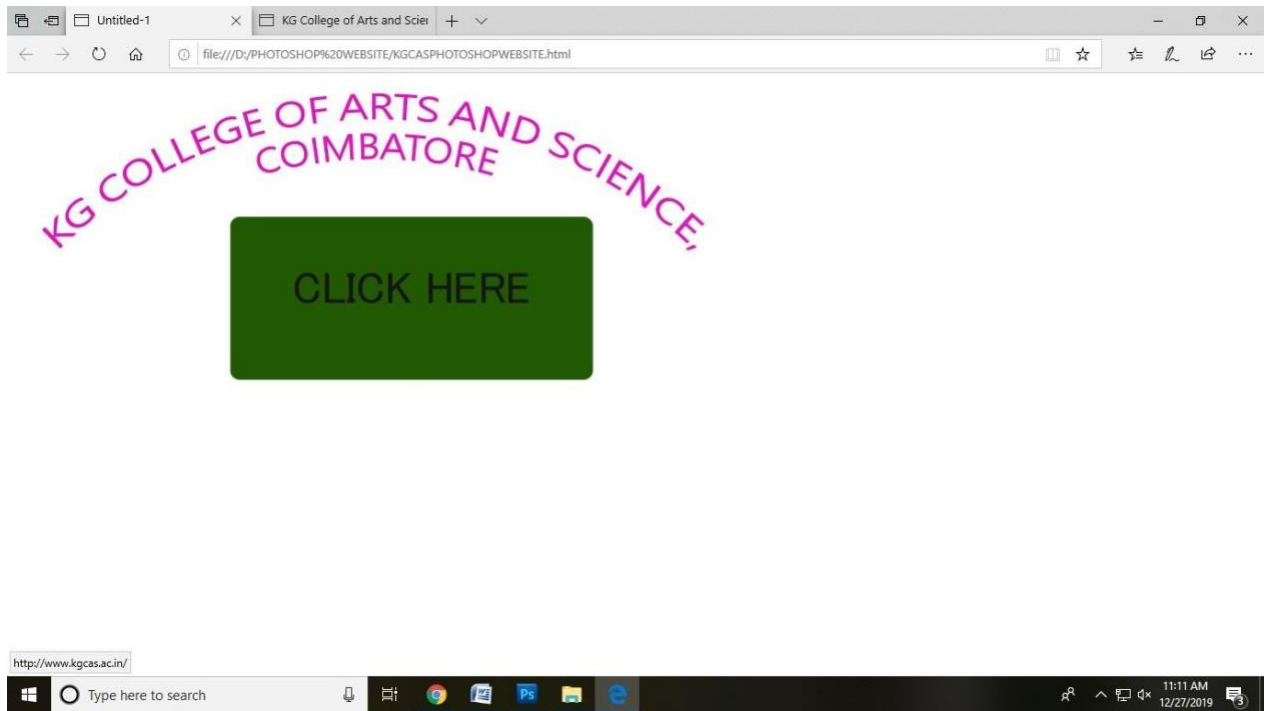
OUTPUT:











RESULT:

The above program has been verified and executed successfully.

EX:NO:12

**CONVERT BLACK AND WHITE PHOTO TO COLOR PHOTO
USING PHOTOSHOP.**

AIM:

To convert black and white photo to color photo using Photoshop.

ALGORITHM:

Step 1: Click the file menu and select the open option to open a black and white photo.

Step 2: Create a new layer.

Step 3: Under the image option choose Mode → RGB color.

Step 4: Under the image option choose, Adjustments → color balance.

Step 5: Change the cyan, red, magenta, and green, yellow, blue and select the highlights radio button and click OK button.

Step6: Under the Image option choose, Adjustments→brightness/Contrast.

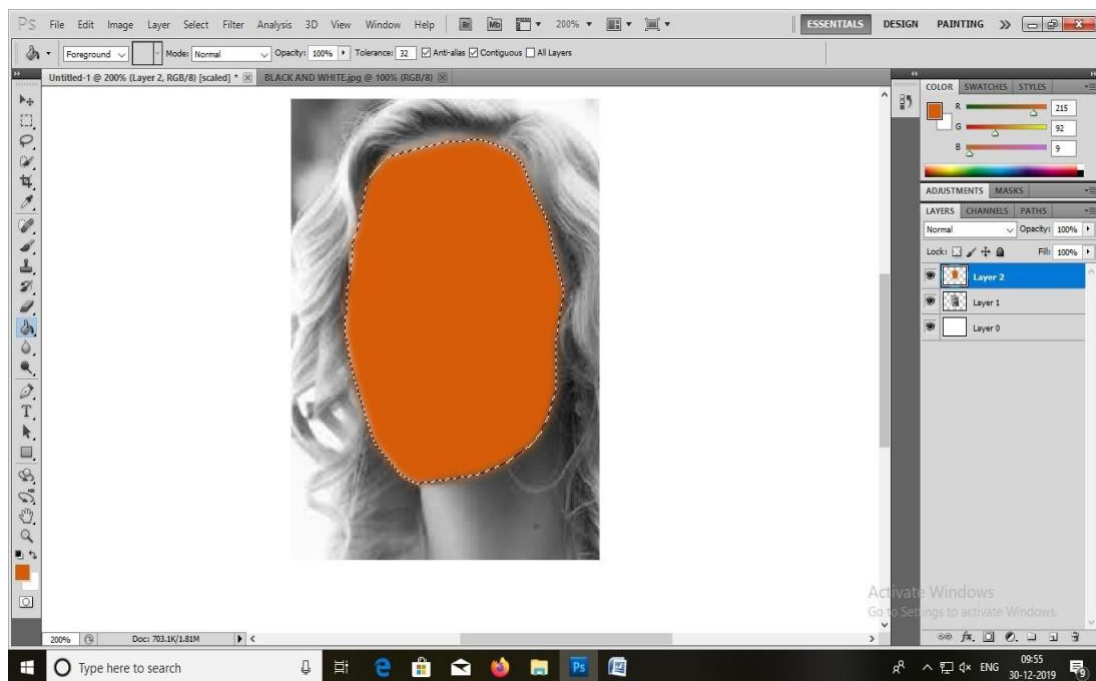
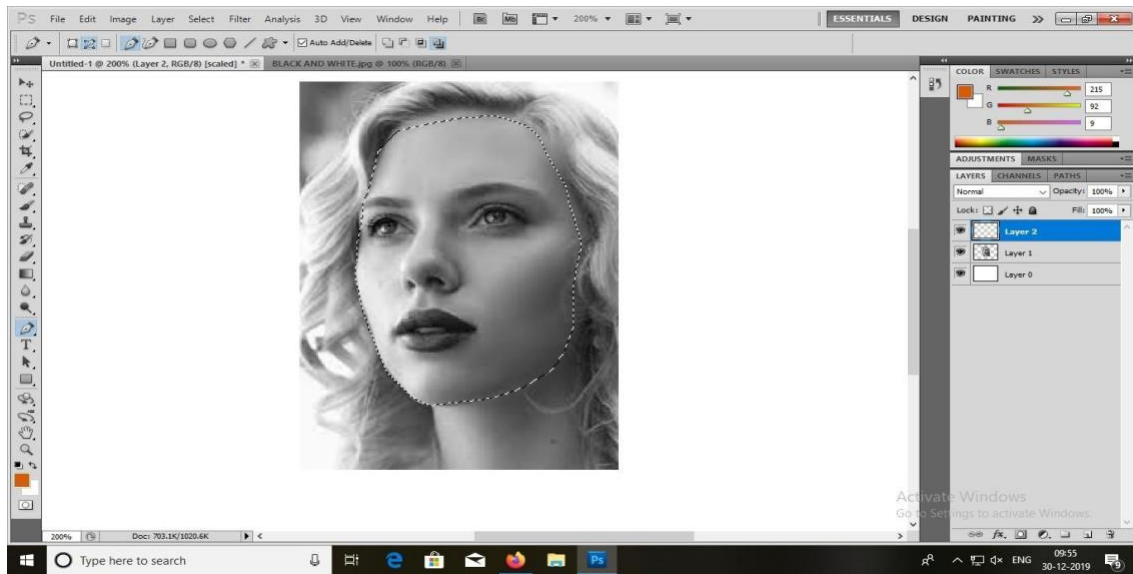
Step7: Change the brightness and contrast Levels and then click OK button.

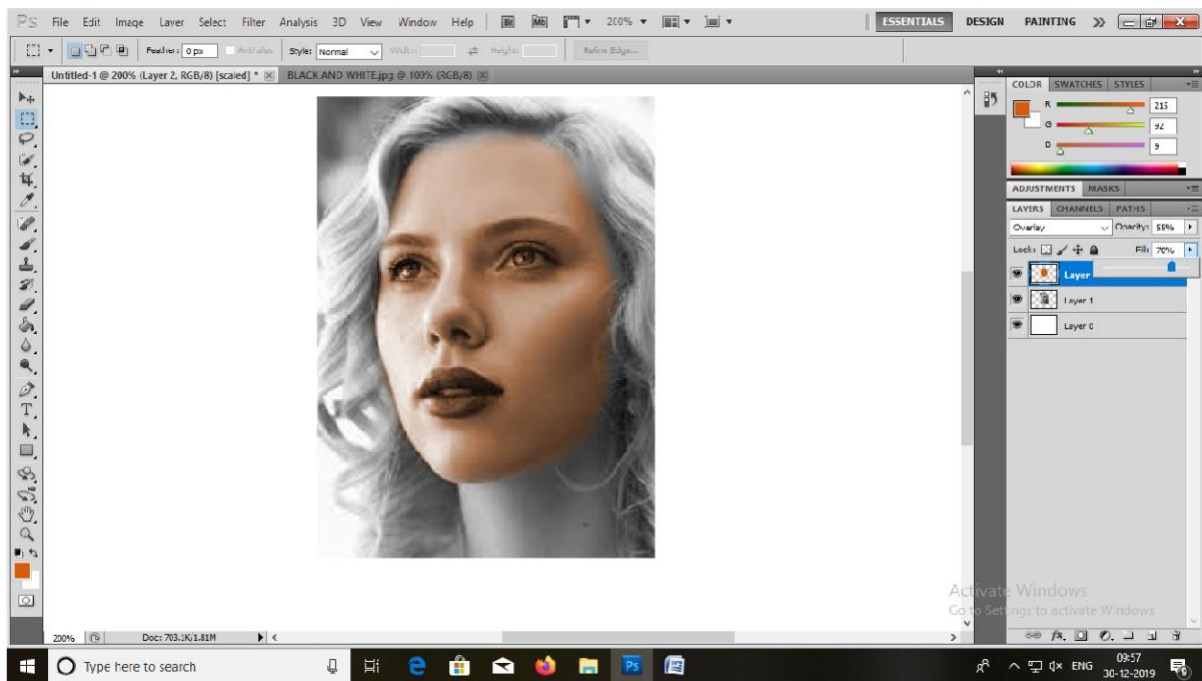
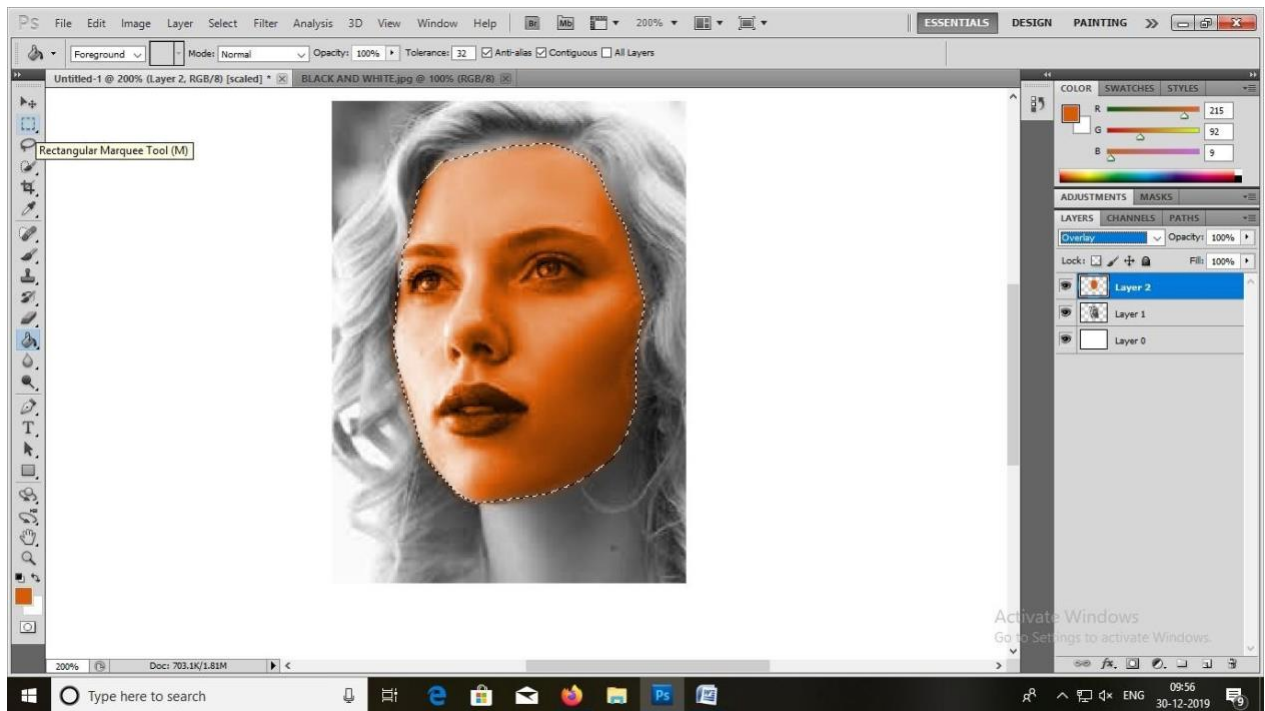
Step8: Under the image option choose, Adjustments→Hue/Saturation.

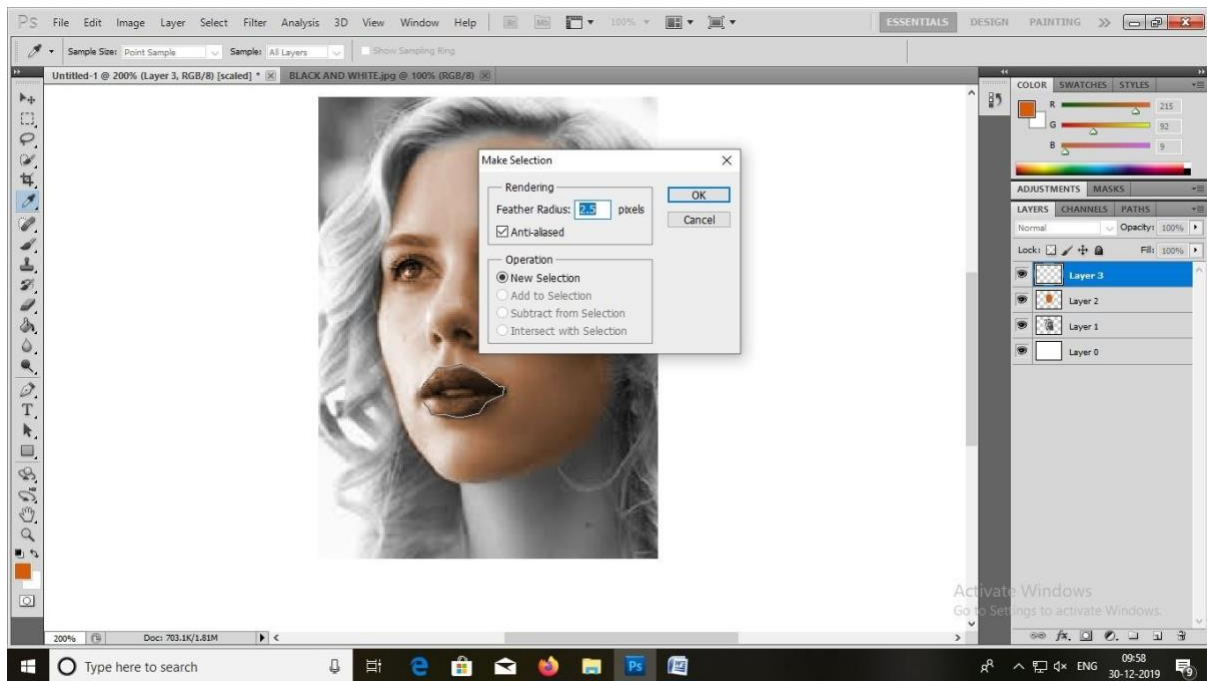
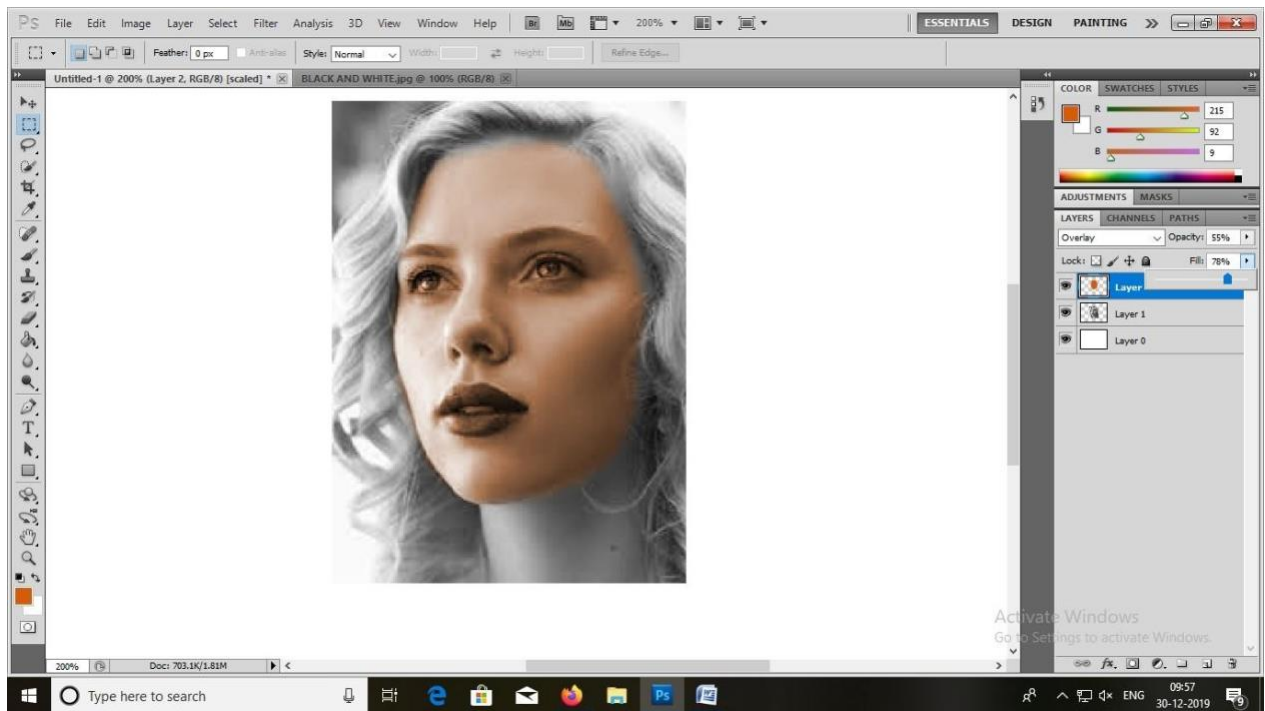
Step 9: Change the Hue, saturation and lightness levels and then click OK button.

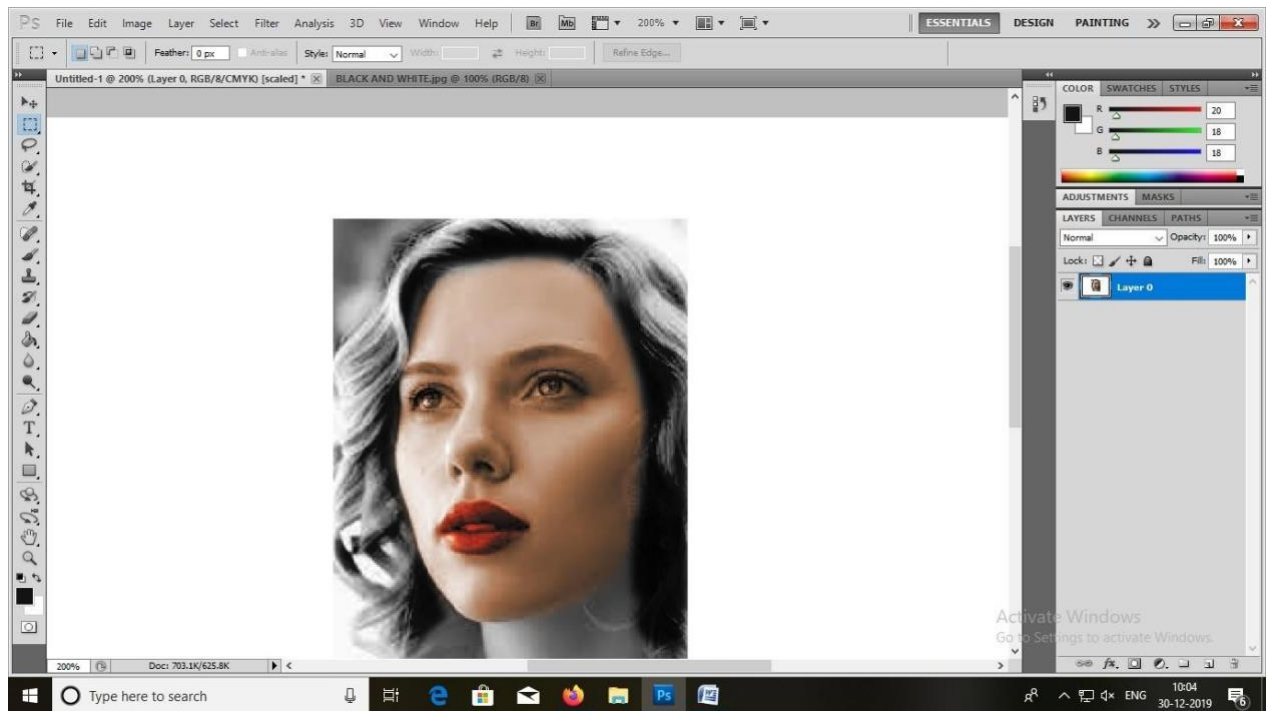
Step 10: The black & white photo is successfully converted to color photo. Then the photo can be saved as .jpg or .psd formats.

OUTPUT:









RESULT:

The above program has been verified and executed successfully.
