Output

Date

Expt. No. 2 Page No. 10

Experiment -2

Aim: (a) To draw smiley face using functions.

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

void main(){
    int gdriver = DETECT, gmode;
    int graph1(&gdriver, &gmode, "C:\\TURBOC3\\BGI");
    circle(100,100,50);
    circle(80,90,10);
    circle(120,90,10);
    circle(100,107,3);
    arc(100,120,200,340,10);
    getch();
    closegraph();
}
```

Ajaya

Teacher's Signature :

Output

ELLIPSE Using Graphics in C

Expt. No. _____ Date _____

Page No. _____

Experiment (2h)

Aim:- To draw a ellipse

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

int main (){
    int gd = DETECT, gm;
    int x, y;
    initgraph (&gd, &gm, "X:\\TC\\BGI");
    x = getmaxx ()/2
    y = getmaxy ()/2

    Outtextxy (x-100, 50, "ELLIPSE Using Graphics in C");
    ellipse (x, y, 0, 360, 120, 60);
    getch();
    closegraph ();
    return 0;
}
```

AJAYA

Teacher's Signature : _____

Output

Date

Expt. No. Page No.

Experiment (2c)

Aim:- To draw a heart

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

void main()
{
    int gd = DETECT, gm;
    initgraph(&gd, &gm, "C:\\tc\\bgi");
    cleardevice();

    for(int i=1; i<=50; i++)
    {
        int b=1;
        while(!kbhit())
        {
            for(int i=1; i<=20; i++)
            settextstyle(3,0,5);
            outtextxy(270,230,"Param");
            arc(280,250,400,2000,50);
            arc(355,250,700,500,50);
            line(320,350,235,270);
            line(320,350,400,270);
            b++;
            delay(8);
            cleardevice();
```

Ajaya

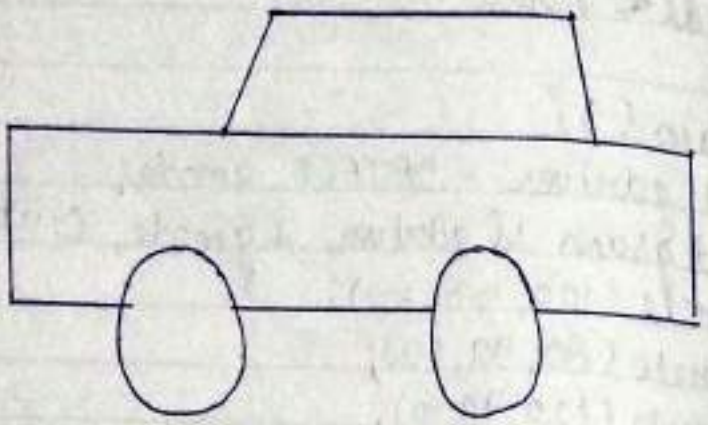
Teacher's Signature :

Date

Expt. No. Page No.

```
        if (b == 70)
        {
            break;
        }
    }
}
```


Teacher's Signature :



Date

Expt. No. 3 Page No. 11

Experiment - 3

Aim:- To draw a car using functions

```
#include <stdio.h>
#include <conio.h>
#include <graphics.h>
#include <dos.h>
void main ()
{
   int gdriver = DETECT, gmode;
   int i;
   initgraph (&gdriver, &gmode, "C:\\TURBOC3\\BGI");
   for (i=0; i<600; i+=10)
   {
      cleardevice ();
      line (100+i, 170, 100+i, 100);
      line (100+i, 100, 170+i, 100);
      line (170+i, 100, 220+i, 50);
      line (220+i, 50, 310+i, 50);
      line (310+i, 50, 370+i, 100);
      line (340+i, 100, 440+i, 170);
      line (440+i, 100, 440+i, 170);
      circle (200+i, 170, 30);
      circle (330+i, 170, 30);
      line (100+i, 170, 170+i, 170);
      line (230+i, 170, 440+i, 170);
      line (360+i, 170, 440+i, 170);
      line (225+i, 55, 260+i, 55);
```

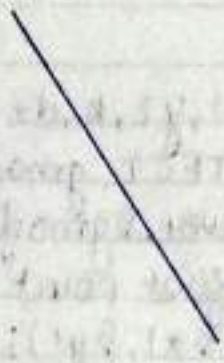
AJANTA

Teacher's Signature :

```
line (260+i, 55, 260+i, 95);
line (270+i, 55, 305+i, 55);
line (285+i, 95, 225+i, 55);
line (185+i, 95, 260+i, 95);
line (353+i, 95, 305+i, 55);
line (353+i, 95, 270+i, 95);
delay (50);
}
getch();
closegraph();
}
```

Output

Enter first point 100
200
Enter second point 200
100



Experiment - 4

Aim:- To draw a line using DDA Algorithm

```
#include <stdio.h>
#include <conio.h>
#include <graphics.h>
void main()
{
    int x, y, x1, x2, y1, y2, k, dx, dy, s, xi, yi;
    int gdriver = DETECT, gmode;
    initgraph (&gdriver, &gmode, "C:\\tc\\bgi: ");
    printf ("Enter first point");
    scanf ("%d %d", &x1, &y1);
    printf ("Enter second point");
    scanf ("%d %d", &x2, &y2);
    y = y1;
    putpixel (x, y, 7);
    dx = x2 - x1;
    dy = y2 - y1;
    if (abs(dx) > abs(dy))
        s = abs(dx);
    else
        s = abs(dy);
    xi = dx/s;
    yi = dy/s;
    x = x1;
    y = y1;
    putpixel (x, y, 7);
```

```
for (k=0; k<s; k++)
{
    x = x + xi;
    y = y + yi;
    putpixel (x, y, 7);
}
getch ();
closegraph ();
}
```

Output

Enter co-ordinates of first point: 100
100
Enter co-ordinates of second point: 200
200

Date

Expt. No. 5 Page No. 15

Experiment - 5

Aim:- To draw a line using Bresenham's Algorithm.

```
#include <stdio.h>
#include <graphics.h>
void drawline(int x0, int y0, int x1, int y1);
{
    int dx, dy, p, x, y;
    dx = x1 - x0;
    dy = y1 - y0;
    x = x0;
    y = y0;
    p = 2 * dy - dx;
    while (x < x1)
    {
        if (p >= 0)
        {
            putpixel (x, y, 7);
            y = y + 1;
            p = p + 2 * dy - 2 * dx;
        }
        else
        {
            putpixel (x, y, 7);
            p = p + 2 * dy; }
            x = x + 1;
        }
```

Ajay

Teacher's Signature :

Date _____

Expt. No. _____ Page No. _____

```
int main()
{
    int gdriver = DETECT, gmode, error, x0, y0, x1, y1;
    initgraph(&gdriver, &gmode, "C:\\turboc3\\bgi");
    printf("Enter co-ordinate of first point: ");
    scanf("%d %d", &x0, &y0);
    printf("Enter co-ordinates of second point: ");
    scanf("%d%d", &x1, &y1);
    drawline(x0, y0, x1, y1);
    return 0;
}
```

Ajaya

Teacher's Signature : _____

Date

Expt. No. 6 Page No. 20

Experiment - 6

Aim:- WAP to implement midpoint circle generation algorithm of given centre (x,y) & radius r.

```
#include <graphics.h>
#include <stdlib.h>
#include <math.h>
#include <stdio.h>
#include <conio.h>
#include <iostream.h>

class bresen {
    float x,y,a,b,r,p;
    public:
        void get();
        void cal();
};
void main () {
    bresen b;
    b.get();
    b.cal();
    getch();
}
Void bresen :: get()
{
    cout << " Enter centre & radius "
    cout << " Enter (a,b)";
    cin >> a >> b;
```

Teacher's Signature :

Output

Enter Centre & Radius
Enter (a, b) 319, 239
Enter r 100

```
    cout << "Enter r ";
    cin >> r;
}
void brosen :: cal()
{
    int gdriver = DETECT, gmode, errorcode;
    int midx, midy, i;
    initgraph (&gdriver, &gmode, " ");
    errorcode = graphresult ();
    if (errorcode != grOK)
    {
        printf ("Graphics error : %s \n", grapherror
                                    msg (errorcode));
        printf ("Press any key to halt : ");
        getch();
        exit(1);
    }
    x = 0;
    y = r;
    putpixel (a, b+r, RED);
    putpixel (a, b-r, RED);
    putpixel (a-r, b, RED);
    putpixel (a+r, b, RED);
    p = (5/4) - r;
    while (x <= y)
    {
        if (p<0)
        p += (4*x) + 6;
        else
```

```
    {
        p+=(2*(x-y))+5;
        y--;
    }
    x++;
    putpixel(a+x,b+y, RED);
    putpixel(a-x,b+y, RED);
    putpixel(a+x,b-y, RED);
    putpixel(a+x,b-y, RED);
    putpixel(a+x,b+y, RED);
    putpixel(a+x,b-y, RED);
    putpixel(a-x,b+y, RED);
    putpixel(a-x,b-y, RED);
  }
}
```

Date

Expt. No. 7

Page No. 23

Experiment-7

Aim:- WAP in C to implement the ellipse generation Algorithm for drawing an ellipse of given centre (x,y) & radius rx & ry.

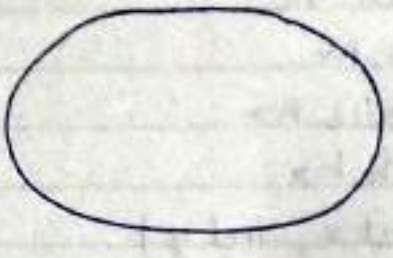
```
#include <stdio.h>
#include <conio.h>
#include <graphics.h>
#include <math.h>
void ellips (int x, int y)
void completeellipse (int r, int g, int u, int v)
{
    float s,k,e,f,x;
    double p1, p2;
    s=r;
    k=g
    e= (pow((s+.5), 2));
    f=(pow((k-1),2));
    p2=((u*e)+(v*f)-(u*v));
    ellips (s,k);
    while (k>=0)
    {
        if(p2>0)
        p2=(p2+v-(2*v*s));
        else
        {
        }
        k--;
```

Ajaya

Teacher's Signature :

Output

Enter the length of major axis : 100
Enter the length of minor axis : 50



```
        p2 = (p2 + (2*u*(s+1)) - (2*v*(k-1)) + v);
        s++;
        ellips(s,k);
    }
}
void main()
{
    int gdriver = DETECT, gmode;
    int a, b, x, y;
    long u, v, p1;
    initgraph(&gdriver, &gmode, "C:\\tc\\bgi:");
    printf("\n Enter the length of major axis: ");
    scanf("\t%d", &a);
    printf("\n Enter the length of minor axis: ");
    scanf("\t%d", &b);
    x = 0;
    y = b;
    u = pow(b,2);
    v = pow(a,2);
    p1 = (u-(v*b)+(.25*v));
    ellips(x,y);
    while (2*(u*x) <= 2*(v*y));
    {
        x++;
        if(p1<0)
        p1 = (p1+(2*u*v)+v);
        else
        {
        }
```

```
        p1=(p1+(2*u*x)-(2*v*y)+u);
        y--;
        ellips(x,y);
        }
        complete ellipse (x,y,u,v);
        getch();
        closegraph();
    }
void ellipse(int x, int y)
{
    putpixel (x+200, y+200, 8);
    putpixel (-x+200, y+200, 8);
    putpixel (x+200, -y+200, 8);
    putpixel (-x+200, -y+200, 8);
}
```

Date

Expt. No. 8 Page No. 26

Experiment - 8

Aim:- WAP in C++ to implement 2D transformations such as translation, scaling, rotation, shearing & reflection for a given 2D object.

(a) To rotate an object about origin

```
#include <iostream.h>
#include <conio.h>
#include <graphics.h>
#include <process.h>
#include <math.h>
void main()
{
    clrscr();
    int graphdriver = DETECT, graphmode;
    initgraph(&graphdriver, &graphmode, "\\tc\\bgi");
    int x, y, xl, a[3][3];
    double b[3][3], c[3][3];
    cout<<"\nEnter 1st coordinate of triangle: ";
    cin>> a[0][0] >> a[1][0];
    cout<<"\nEnter 2nd coordinate of triangle: ";
    cin>>a[0][1]>>a[1][1];
    cout<<"\nEnter 3rd coordinate of triangle: ";
    cin>>a[0][2]>>a[1][2];
    line(a[0][0], a[1][0], a[0][1], a[1][1]);
    line(a[0][1], a[1][1], a[0][2], a[1][2]);
    line(a[0][0], a[1][0], a[0][2], a[1][2]);
```

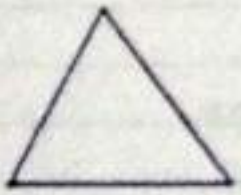
Teacher's Signature

Output

Enter 1st coordinate of triangle: 100 100

Enter 2nd coordinate of triangle: 200 100

Enter 3rd coordinate of triangle: 150 50



Enter angle of rotation
30

Triangle after rotation

Enter angle of rotation
30

Triangle after rotation

```
getch();
cleardevice();
cout<<"Enter angle of rotation:\n";
cin>>x;
b[0][0]=b[1][1]=cos((x*3.14)/180);
b[0][1]=-sin((x*3.14)/180);
b[1][0]=sin((x*3.14)/180);
b[2][2]=1;
b[2][0]=b[2][1]=b[0][2]=b[1][2]=0;
for(int i=0;i<3;i++)
{
    for(int j=0;j<3;j++)
    {
        c[i][j]=0;
        for(int k=0;k<3;k++)
        {
            c[i][j]+=a[i][k]*b[k][j];
        }
        x1=(c[i][j]+0.5);
        a[i][j]=x1;
    }
}
cout<<"\n Triangle after rotation is:\n";
line(a[0][0],a[1][0],a[0][1],a[1][1]);
line(a[0][1],a[1][1],a[0][2],a[1][2]);
line(a[0][0],a[1][0],a[0][2],a[1][2]);
getch();
closegraph();
}
```

Output

Enter 1st coordinates of triangle: 100
100
Enter 2nd coordinates of triangle: 200
100
Enter 3rd coordinates of triangle: 150
50

Date

Expt. No. Page No. 28

(B) Aim:- To translate an object with translation parameter in x & y directors.

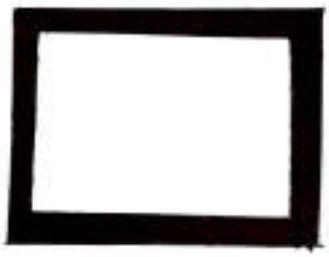
```
#include<iostream.h>
#include<conio.h>
#include<graphics.h>
#include<process.h>
#include<math.h>

void main()
clrscr();
int graphdriver=DETECT, graphmode;
initgraph(&gdriver, &graphmode, "..\\bgi");
int x, y, x1, y1, x2, y2, x3, y3;
cout<<"Enter 1st coordinates of triangle:"cin>>x1>>y1;
cout<<"Enter 2nd coordinates of triangle:"; cin>>x2>>y2;
cout<<"Enter 3rd coordinates of triangle:"; cin>>x3>>y3;
cleardevice();
line(x1, y1, x2, y2);
line(x2, y2, x3, y3);
line(x1, y1, x3, y3); getch(); cleardevice();
cout<<"\nEnter translation factors:\n"; cin>>x>>y;
x1-=x;
y1-=y;
x2-=x;
y2-=y;
x3-=x;
y3-=y;
cleardevice();
```

AJAYA

Teacher's Signature :

```
line(x1, y1, x2, y2);
line(x2, y2, x3, y3);
line(x1, y1, x3, y3);
getch();
closegraph();
}
```

Output



Date

Expt. No. 9 Page No. 30

Experiment - 9

Aim - WAP in C to implement flood fill algo. filling a rectangle with given color.

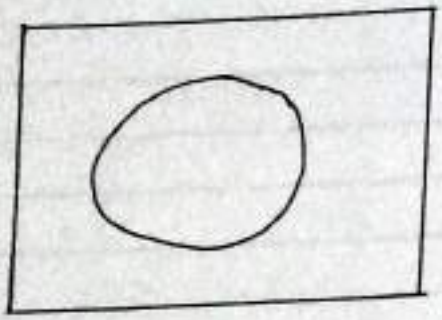
```
#include <graphics.h>
#include <stdio.h>

void flood (int x, int y, int new_col, int old_col)
{
    if(getpixel(x,y)== old_col) {
        putpixel (x, y, new_col);
        flood (x+1, y, new_col, old_col);
        flood (x-1, y, new_col, old_col);
        flood (x, y+1, new_col, old_col);
        flood (x, y-1, new_col, old_col);
    }
}
int main()
{
    int gd, gm = DETECT;
    initgraph (&gd, &gm, "");
    int top, right, bottom, left ;
    top = left = 50
    bottom = right = 300;
    rectangle, left, top, right, bottom);
    int x = 51;
    int y = 51;
    int newcolor = 12;
```

Ajaya

Teacher's Signature :

```
int oldcolor = 0;
flood(x,y, newcolor, oldcolor);
getch();
return 0;
}
```

Output



Date

Expt. No. 10 Page No. 32

Experiment -10

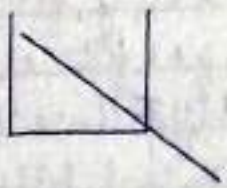
Aim:- WAP in C to implement boundary fill algo. for ~~a~~ filling rectangle with given color

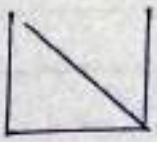
```
#include<graphics.h>
void boundaryFill4(int x, int y, int fill_color, int
                   ~~to or~~ boundary_color)
{
  if (getpixel(x,y) != boundary_color &&
   getpixel(x,y) != fill_color)
    {
      putpixel(x,y,fill_color);
      boundaryFill4(x+1, y, fill_color, boundary_color);
      boundaryFill4(x, y+1, fill_color, boundary_color);
      boundaryFill4(x-1, y, Fill_color, boundary_color);
      boundaryfill4(x, y-1, fill_color, boundary_color);
    }
}
int main()
{
    int gd = DETECT, gm;
    initgraph(&gd, &gm, "");
    int x=250, y=200, radius=50;
    circle(x, y, radius);
    boundaryFill4(x, y, 6, 15);
    delay(10000);
    getch();
    closegraph(); return 0; }
```

Teacher's Signature :

Output

Enter the value of
x1, y1, x2, y2 :> 10
10
100
100
Enter the value of
xmax, ymax, xmin, ymin 50
50
0
0





Date

Expt. No. 11

Page No. 33

EXPERIMENT=11

Aim :- WAP in C to implement cohen Sutherland line clipping & windowing algorithm.

```
#include <stdio.h>
#include<conio.h>
#include <graphics.h>
#include<dos.h>
void storepoints(int,int,int,int,int,int,int[]);
void main()
{
    int gdriver = DETECT,gmode;
    int x1,x2,y1,y2,xmax,ymax,xmin,ymin,
      a[10],b[10],xi1,x12,yi1,yi2 flag=0;
      float m;
      int i; clrscr();
      printf("output);
      printf("\n");
      printf("Enter the value of x1,y1,x2,y2: ");
      scanf("%d %d %d %d",&x1,&y1,&x2,&y2);
      printf("Enter the value of xmax,ymax,xmin,ymin);
      storepoints(x2,y2,ymin,ymax,xmax,xmin,b);
      for(i=1; i<=4;i++)
      {
          if a[i]*b[i]==0)
          flag=1;
          else
          {
```

Teacher's Signature :

```
flag = 0;
if(flag = 1)
{
    m = (y2-y1)/(x2-x1);
    xi1 = x1;
    yi1 = y1;
}
if(a[1] == 1)
{
}
else
yi1 = ymax;
xi1 = x1+((1/m)*(yi1-y1));
if(a[2] == 1)
{
    yi1 = ymin;
    x11 = x1+((1/m)*(yi1-y1));
}
}
if(a[3] == 1)
{
    xi1 = xmax;
    yi1 = y1+(m*(xi1-x1));
}
if(a[4] == 1)
{
}
else
xi1 = xmin;
```

```
yi1=y1+(m*(xi1-x1));
if(b[1]==1)
{
}
else
yi2=ymax;
xi2=x2+((1/m)*(yi2-y2));
if(b[2]==1)
{
}
else
yi2=ymin;
xi2=x2+((1/m)*(yi2-y2));
if(b[3]==1)
{
  closegr();
}
else
xi2=xmax;
yi2=y2+((1/m)*(xi2-x2));
if(b[4]==1)
{
   xi2=xmin;
   yi2=y2+(m*(xi2-x2));
}
initgraph(&gdriver,&gmode,"C://tc//bgi:");
rectangle(xmin,ymin,xmax,ymax);
line(x1,y1,x2,y2);
delay(5000);
```

Teacher's Signature :

```
closegraph();
clrscr();
initgraph(&gdriver, &gmode, "C:\\tc\\bgi: ");
line(xi1, yi1, xi2, yi2);
rectangle(xmin, ymin, xmax, ymax);
if (flag==0)
{
   printf("\n No clipping required");
}
getch();
closegraph();
}
void storepoints(int x1, int y1, int ymax, int xmax,
  int xmin, int ymin, int c[10])
if ((y1-ymax)>0)
c[1]=1;
else
c[1]=0;
if((ymin-y1)>0)
c[2]=1;
else
c[2]=0
if((x1-xmax)>0)
c[3]=1;
else
c[3]=0;
if((xmin-x1)>0)
c[4]=1;
else { c[4]=0; }
```


Teacher's Signature :

Date

Expt. No. 12

Page No. 37

Experiment = 12

Aim:- WAP in C to implement the basic transformation such as translation, scaling rotation for a given 3D object.

```
#include <stdio.h>
#include <conio.h>
#include <graphics.h>
#include <math.h>
void trans();
void scale();
Void rotate();
int maxx, maxy, midx, midy;
void main()
{
   int ch;
   int gd = DETECT, gm;
   detectgraph(&gd, &gm);
   initgraph(&gd, &gm, "c:\\tc\\bgi
   printf("\n1. Translation\n2. Scaling\n3. Rotation\n4.exit");
   printf("Enter your choice");
   scanf("%d", &ch);
   do
   {
      switch(ch)
      {
         case1: trans()
                getch()
```

Ajaya

Teacher's Signature :

```
        break;
    case 2 : scale();
             getch();
             break;
    case 3: rotate();
             getch();
             break;
    case 4: break;
    }
    printf("Enter your choice ");
    scanf("%d", &ch);
    } while(ch<4);
    }
    void trans()
    {
      int x,y,z,o,x1,x2,y1,y2;
      maxx = getmaxx();
      maxy = getmaxy();
      midx = maxx/2;
      midy = maxy/2;
      bar3d(midx+50,midy-100,midx+60,midy-90,
                                    10,1);
      printf("Enter translation factor ");
      scanf("%d %d %d",&x,&y);
      printf("After translation: ");
      bar3d(midx+50,midy-(y+100),midx+x+60,
            midy-(y+90),10,1);
    }
```

Date

Expt. No. Page No.

```
void scale()
{
    int x,y,z,o,x1,y1,x2,y2;
    maxx=getmaxx();
    maxy=getmaxy();
    midx=maxx/2;
    midy=maxy/2;
    bar3d(midx+50,midy-100,midx+60,midy-90,
                5,1);
    printf("Before translation\n");
    printf("Enter scaling factors\n");
    scanf("%d %d %d",&x,&y,&z);
    printf("After scaling\n");
    bar3d(midx+(x*50),midy-(y*10),
     midx+(z*60),midy-(y*90),5*z,1);
}
void rotate()
{
  int x,y,z,o,x1,x2,y1,y2;
  maxx=getmaxx();
  maxy=getmaxy();
  midx=maxx/2;
  midy=maxy/2;
  bar3d(midx+50,midy-100,midx+60,midy-90,
              5,1);
  printf("Enter rotating angle");
  scanf("%d",&o);
  x1=50*cos(o*3.14/180)-100*sin(o*3.14/180);
  y1=50*sin(o*3.14/180)+100*cos(o*3.14/180);
```

Teacher's Signature :

Date

Expt. No. Page No.

```
x2 = 60 * cos(0 * 3.14/180) - 90 * sin(0 * 3.14/180);
y2 = 60 * sin(0 * 3.14/180) + 90 * cos(0 * 3.14/180);
printf("After rotation about x axis");
bar3d(midx+50, midy-x1, midx+60, midy-x2,
      5, 1);
printf("After Rotation about y axis");
bar3d(midx+x1, midy-100, midx+x2,
      midy-90, 5, 1);
```

Teacher's Signature :

Expt. No. 13 Date

Page No.

Experiment - 13

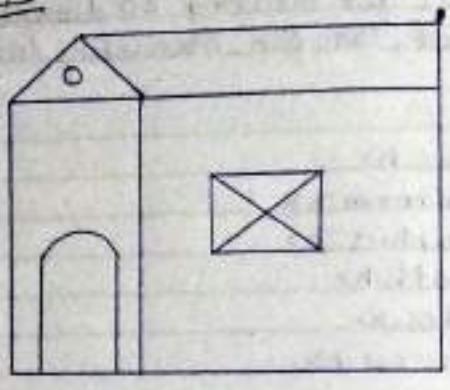
Aim:- WAP in C for creating 2D dimensional shapes of house, car, fish, man using lines, circles etc.

```
#include <conio.h>
#include <iostream.h>
#include <graphics.h>
#include<math.h>
#include <dos.h>
#include <process.h>
void main()
{
  int graphdriver = DETECT, graphmode;
  initgraph (&graphdriver, &graphdriver, &graphmode, "
            ..\\bgi");
  line(100,100,150,50);
  line(150,50,200,100);
  line(100,100,200,100);
  line(150,50,350,50);
  line(200,100,350,100);
  line(350,50,350,100);
  circle(150,75,10);
  rectangle(100,100,200,300);
  rectangle(200,100,350,300);
  rectangle(250,175,300,225);
  line(250,175,300,225);
  line(300,175,250,225);
```

AJAYA

Teacher's Signature :

Output



Expt. No. _____ Pa

```
line (125, 300, 125, 225);
line (175, 300, 175, 225);
arc (150, 225, 0, 180, 25);
getch ();
closegraph ();
}
```

Ajanta

Teacher's Signature

Date

Expt. No. Page No.

```
line (125, 300, 125, 225);
line (175, 300, 175, 225);
arc (150, 225, 0, 180, 25);
getch ();
closegraph ();
}
```

AJAY

Teacher's Signature :