 GALGOTIAS UNIVERSITY

**Plot No.2, Sector -17 A, Yamuna Expressway,**

**Greater Noida, Gautam Buddha Nagar, U.P., India**

**SCHOOL OF COMPUTING SCIENCE & ENGINEERING**

**“LAB PRACTICAL FILE”**

**Course Name: Computer Graphics**

**Course Code: E2UC402B**

**School: SCSE**

**Program: B. Tech**

**Year: 2nd Semester: 4th**

**Session: 2023-2024**

|  |  |
| --- | --- |
| **Submitted By:** | **Submitted To:** |
| **NEERAJ SINGH**  **21SCSE1011675**  **SEC-6- P1** | **Miss Alisha Banga** |

**Index**

|  |  |  |  |
| --- | --- | --- | --- |
| Sr. No. | Title of Lab Experiments | DATE | SIGN |
| 1 | Using different graphics functions available for text formatting, write a program for displaying text in different sizes, different colors, font styles | 1/2/2023 |  |
| 2 | Write a program to divide screen into four region and draw circle, rectangle, arc and ellipse. | 8/2/2023 |  |
| 3 | Implement the DDA algorithm for drawing lines. | 15/2/2023 |  |
| 4 | Write a program to input the line coordinates from the user to generate a line using Bresenham’s algorithm | 19/2/2023 |  |
| 5 | WAP to make HUT. | 19/2/2023 |  |
| **6** | Write a program to draw a circle using Midpoint circle drawing algorithm and Bresenham’s circle drawing algorithm. |  |  |
| 7 | Write a program to draw any 2-D object and perform the transformations on it according to the input parameters from the user, namely: Translation, Rotation or Scaling. |  |  |
| 8 | Write a program to draw any 2-D object and perform the transformations on it according to the input parameters from the user, namely: Shearing and Reflection. |  |  |
| 9 | Write a program to draw a concave polygon and fill it with desired color using scan fill algorithm. |  |  |
| 10 | Write a program to implement Cohen Southerland line clipping algorithm |  |  |
| 11 | Write a program to generate Hilbert Curve using concept of fractals. |  |  |
| 12 | Write a program to draw Sunrise and Sunset. |  |  |
| 13 | Write a program for drawing a following pattern(diamond in rectangle). |  |  |
| 14 |  |  |  |
| 15 |  |  |  |
| 16 |  |  |  |
| 17 |  |  |  |
| 18 |  |  |  |

**Experiment-1**

**Aim** : Using different graphics functions available for text formatting, write a program for displaying text in different sizes, different colors, font styles

|  |
| --- |
| #include <graphics.h>  #include <conio.h>  int main(){  int gd=DETECT,gm,x=25,y=25,font=10;  initgraph(&gd,&gm,"C:\\turboC3\\BGI");  for(font=0;font<=4;font++){  settextstyle(font,HORIZ\_DIR,font+1);// sets font type, font direction, size  setcolor(font+1); // sets color for text.  outtextxy(x,y,"text with different fonts"); // prints message on screen at (x,y)  y=y+25;  }  for(font=0;font<=2;font++){  settextstyle(font,VERT\_DIR,font+2);  setcolor(font+1);  x=250;  y=100;  outtextxy(x,y,"text in vertical direction");  y=y+25;  }  getch();  closegraph();  return 0;} |

**output**

|  |
| --- |
|  |

**Experiment 2**

Aim : Write a program to divide screen into four region and draw circle, rectangle, arc and ellipse.

**input**

|  |
| --- |
| #include<conio.h>  #include<graphics.h>  #include<stdio.h>  int main()  {  int gdriver = DETECT, gmode;  int xmax,ymax;  initgraph(&gdriver, &gmode,"");  xmax = getmaxx();  ymax = getmaxy();  line(xmax/2,0,xmax/2,ymax);  line(0,ymax/2,xmax,ymax/2);  circle(170,125,100);  rectangle(58,251,304,392);  arc(500,150,45,135,100);  ellipse(500,300,0,360,75,25);  getch();  closegraph();  return 0;  } |

**output**

|  |
| --- |
|  |

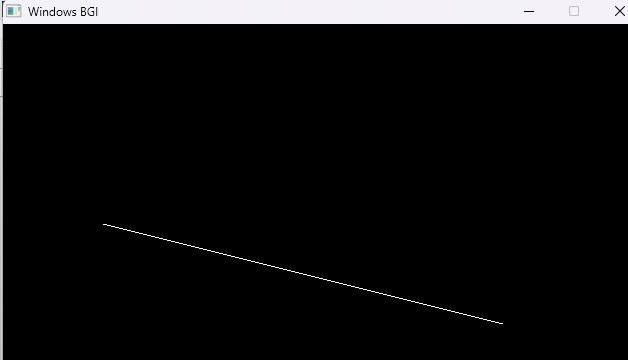
**Experiment 3**

**Aim :** Implement the DDA algorithm for drawing lines.

**input**

|  |
| --- |
| #include<graphics.h>  #include<conio.h>  #include<stdio.h>  int main() {  int gd = DETECT ,gm, i;  float x, y,dx,dy,steps;  int x0, x1, y0, y1;  initgraph(&gd, &gm, "C:\\TC\\BGI");  setbkcolor(WHITE);  x0 = 100 , y0 = 200, x1 = 500, y1 = 300;  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, WHITE);  x += dx;  y += dy;  i=i+1; }  getch();  closegraph(); } |

**output**

****

**Experiment 4**

**Aim :** Write a program to input the line coordinates from the user to generate 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;  } }  int main() {  int gdriver=DETECT, gmode, error, x0, y0, x1, y1;  initgraph(&gdriver, &gmode, "");  printf("Enter co-ordinates 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);  getch();  return 0;  } |

**output**

|  |
| --- |
|  |

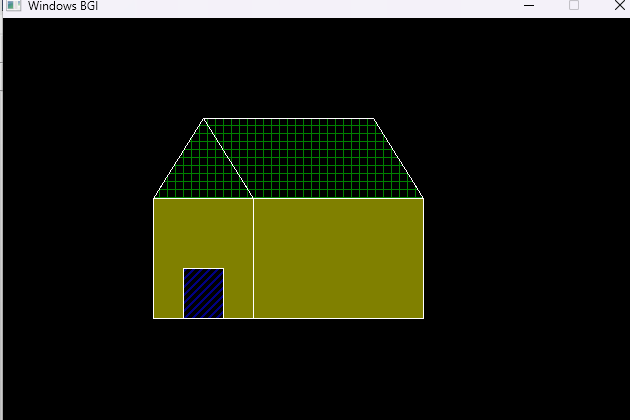
**Experiment 5**

**Aim:** WAP to make HUT.

input

|  |
| --- |
| #include<graphics.h>  int main(){  int gd = DETECT,gm;  initgraph(&gd, &gm, "X:\\TC\\BGI");  setcolor(WHITE);  rectangle(150,180,250,300);  rectangle(250,180,420,300);  rectangle(180,250,220,300);  line(200,100,150,180);  line(200,100,250,180);  line(200,100,370,100);  line(370,100,420,180);  setfillstyle(SOLID\_FILL, BROWN);  floodfill(152, 182, WHITE);  floodfill(252, 182, WHITE);  setfillstyle(SLASH\_FILL, BLUE);  floodfill(182, 252, WHITE);  setfillstyle(HATCH\_FILL, GREEN);  floodfill(200, 105, WHITE);  floodfill(210, 105, WHITE);  getch();  closegraph();  return 0;  } |

**output**

****