CENTRAL CALCUTTA POLYTECHNIC

21, Convent Road, Philips, Sealdah, Kolkata, West Bengal 700014

DEPT. : COMPUTER SCIENCE AND TECHNOLOGY

- NAME : SUMAN MONDAL
- ROLL : DCCPCSTS4
- NUMBER : 10005537
- REG NUMBER : D192005242
- SUBJECT : COMPUTER GRAPHICS
- SESSION : 2021
- EMAIL: SUMAN.MONDAL@OUTLOOK.IN

Contents

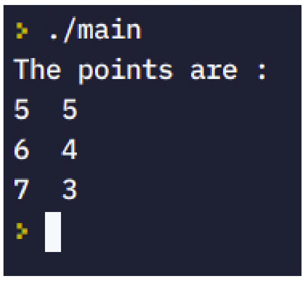
1 Computer Graphics Assignment

1.1 Write a C program to draw a line using DDA algorithm

Source Code :

```c
//01. Write a C program to draw a line using DDA algorithm

#include <stdio.h>
#include <math.h>

void dda_line(int xs, int ys, int xe, int ye)
{
  int steps, sign_dx, sign_dy, i = 1;
  float x, y, delta_x, delta_y;
  // Calculate steps
  if (fabs(xe - xs) >= fabs(ye - ys))
    steps = fabs(xe - xs);
  else
    steps = fabs(ye - ys);

  delta_x = (float)(xe - xs) / steps;
  delta_y = (float)(ye - ys) / steps;
  if (delta_x > 0)
    sign_dx = 1;
  else if (delta_x == 0)
    sign_dx = 0;
  else
    sign_dx = -1;
  if (delta_y > 0)
    sign_dy = 1;
  else if (delta_y == 0)
    sign_dy = 0;
  else
    sign_dy = -1;
  x = (float)(xs + 0.5 * sign_dx);
  y = (float)(ys + 0.5 * sign_dy);

  printf("The points are :\n");
  while (i <= steps)
  {
    printf("%d  %d\n", (int)floor(x), (int)floor(y));
    x += delta_x;
    y += delta_y;
    i += 1;
  }
}

int main()
{
  dda_line(5, 6, 8, 3);
  return 0;
}
```

Program Output :



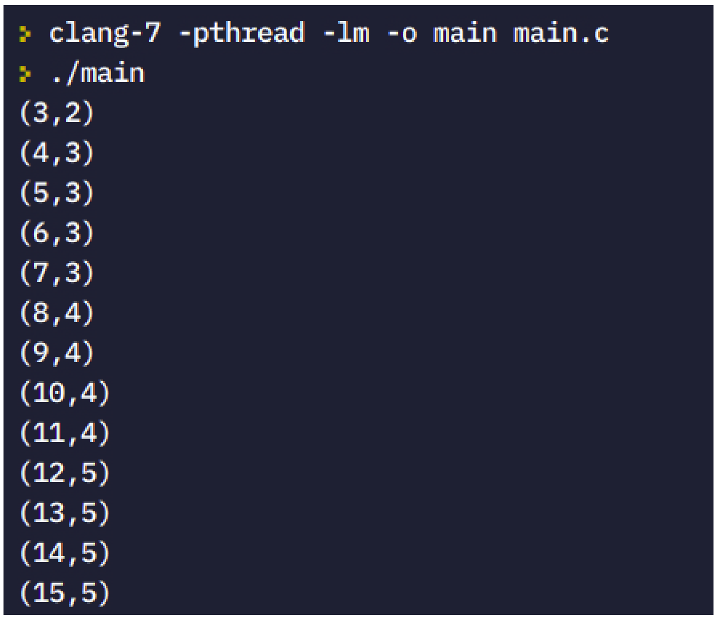
```
› ./main
The points are :
5  5
6  4
7  3
›
```

1.2 Write a C program to draw a line using Bresenham's line drawing algorithm

Source Code :

```c
//02. Write a C program to draw a line using Bresenham's line drawing
// algorithm.

#include <stdio.h>

void bresenham_line(int x1, int y1, int x2, int y2)
{
  int m_new = 2 * (y2 - y1);
  int slope_error_new = m_new - (x2 - x1);
  for (int x = x1, y = y1; x <= x2; x++)
  {
    printf("(%d,%d)\n", x, y);
    slope_error_new += m_new;
    if (slope_error_new >= 0)
    {
      y++;
      slope_error_new -= 2 * (x2 - x1);
    }
  }
}
int main()
{
  int x1 = 3, y1 = 2, x2 = 15, y2 = 5;
  bresenham_line(x1, y1, x2, y2);
  return 0;
}
```

Program Output :



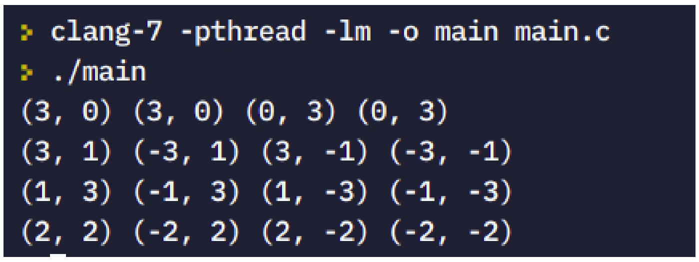
```
› clang-7 -pthread -lm -o main main.c
› ./main
(3,2)
(4,3)
(5,3)
(6,3)
(7,3)
(8,4)
(9,4)
(10,4)
(11,4)
(12,5)
(13,5)
(14,5)
(15,5)
```

1.3 Write a C program to draw a circle using Midpoint circle drawing algorithm

Source Code :

```
//03. Write a C program to draw a circle using Midpoint circle drawing algorithm.

#include <stdio.h>

void mid_point_circle(int x_centre, int y_centre, int r)
{
  int x = r, y = 0;

  // Printing the initial point on the axes
  // after translation
  printf("(%d, %d) ", x + x_centre, y + y_centre);
  if (r > 0)
  {
    printf("(%d, %d) ", x + x_centre, -y + y_centre);
    printf("(%d, %d) ", y + x_centre, x + y_centre);
    printf("(%d, %d)\n", -y + x_centre, x + y_centre);
  }
  int P = 1 - r;
  while (x > y)
  {
    y++;
    if (P <= 0)
      P = P + 2 * y + 1;
    else
    {
      x--;
      P = P + 2 * y - 2 * x + 1;
    }

    if (x < y)
      break;

    printf("(%d, %d) ", x + x_centre, y + y_centre);
    printf("(%d, %d) ", -x + x_centre, y + y_centre);
    printf("(%d, %d) ", x + x_centre, -y + y_centre);
    printf("(%d, %d)\n", -x + x_centre, -y + y_centre);
    if (x != y)
    {
      printf("(%d, %d) ", y + x_centre, x + y_centre);
      printf("(%d, %d) ", -y + x_centre, x + y_centre);
      printf("(%d, %d) ", y + x_centre, -x + y_centre);
      printf("(%d, %d)\n", -y + x_centre, -x + y_centre);
    }
  }
}

int main()
{

  mid_point_circle(0, 0, 3);
  return 0;
}
```

Program Output :



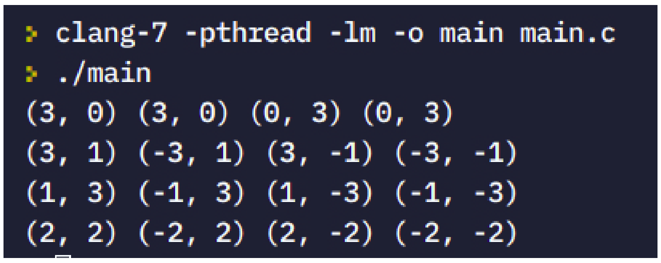
```
> clang-7 -pthread -lm -o main main.c
> ./main
(3, 0) (3, 0) (0, 3) (0, 3)
(3, 1) (-3, 1) (3, -1) (-3, -1)
(1, 3) (-1, 3) (1, -3) (-1, -3)
(2, 2) (-2, 2) (2, -2) (-2, -2)
```

1.4 Write a C Program to show the two dimensional Translation of an Object

Source Code :

```c
//04. Write a C Program to show the two dimensional Translation of an Object

#include <stdio.h>
void translate(int a[][2], int b[], int n)
{
  int i = 0;
  while (i < n)
  {
    a[i][0] = a[i][0] + b[0];
    a[i][1] = a[i][1] + b[1];
    printf("(%d, %d)\n", a[i][0], a[i][1]);
    i++;
  }
}
int main()
{
  int size1 = 4;
  int points[][2] = {{7, 4}, {6, 8}, {10, 11}, {9, 8}};
  int trans[] = {3, 2};
  translate(points, trans, size1);
  return 0;
}
```

Program Output :



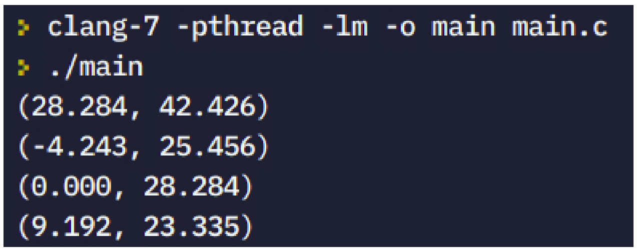
```
> clang-7 -pthread -lm -o main main.c
> ./main
(3, 0) (3, 0) (0, 3) (0, 3)
(3, 1) (-3, 1) (3, -1) (-3, -1)
(1, 3) (-1, 3) (1, -3) (-1, -3)
(2, 2) (-2, 2) (2, -2) (-2, -2)
```

1.5 Write a C Program to show the two dimensional Rotation of an Object

Source Code :

```c
//05. Write a C Program to show the two dimensional Rotation of an Object.

#include <math.h>
#include <stdio.h>
#define SIN(x) sin(x * 3.14159 / 180)
#define COS(x) cos(x * 3.14159 / 180)

void rotate(float a[][2], int n, int x_pivot, int y_pivot,
            int angle)
{
  int i = 0;
  while (i < n)
  {
    int x_shifted = a[i][0] - x_pivot;
    int y_shifted = a[i][1] - y_pivot;
    a[i][0] = x_pivot + (x_shifted * COS(angle) - y_shifted * SIN(angle));
    a[i][1] = y_pivot + (x_shifted * SIN(angle) + y_shifted * COS(angle));
    printf("(%.3f, %.3f)\n", a[i][0], a[i][1]);
    i++;
  }
}

int main()
{

  int size1 = 4;

  float points_list1[][2] = {{50, 10},
                             {15, 21},
                             {20, 20},
                             {23, 10}};
  rotate(points_list1, size1, 0, 0, 45);
  return 0;
}
```

Program Output :

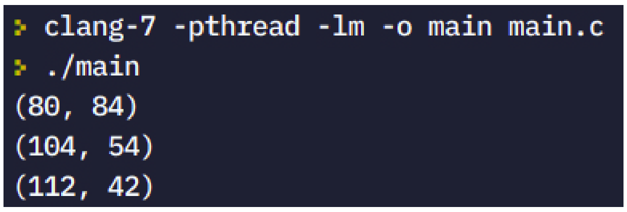

```
› clang-7 -pthread -lm -o main main.c
› ./main
(28.284, 42.426)
(-4.243, 25.456)
(0.000, 28.284)
(9.192, 23.335)
```

1.6 Write a C Program to show the two dimensional Scaling of an Object

Source Code :

```c
//06. Write a C Program to show the two dimensional Scaling of an Object.

#include <stdio.h>
void scale(int a[][2], int b[], int n)
{
  int i = 0;
  while (i < n)
  {
    a[i][0] = a[i][0] * b[0];
    a[i][1] = a[i][1] * b[1];
    printf("(%d, %d)\n", a[i][0], a[i][1]);
    i++;
  }
}
int main()
{
  int size_o = 3;
  int points[][2] = {{10, 14}, {13, 9}, {14, 7}};
  int scl[] = {8, 6};
  scale(points, scl, size_o);
  return 0;
}
```

Program Output :



```
› clang-7 -pthread -lm -o main main.c
› ./main
(80, 84)
(104, 54)
(112, 42)
```