Program 1

Aim: Write a program to draw/create a hut using DDA line algorithm.

Theory:

A digital differential analyzer (DDA) is hardware or software used
for interpolation of variables over an interval between start and end point. DDAs are used for rasterization of lines, triangles and polygons. They can be extended to nonlinear functions, such as perspective correct texture mapping, quadratic curves, and traversing voxels.

Algorithm:

A linear DDA starts by calculating the smaller of dy or dx for a unit increment of the other. A line is then sampled at unit intervals in one coordinate and corresponding integer values nearest the line path are determined for the other coordinate.

Considering a line with positive slope, if the slope is less than or equal to 1, we sample at unit x intervals (dx=1) and compute successive y values as

$$y_{k+1} = y_k + m$$

$$x_{k+1} = x_k + 1$$

Subscript k takes integer values starting from 0, for the 1st point and increases by 1 until endpoint is reached. y value is rounded off to nearest integer to correspond to a screen pixel. For lines with slope greater than 1, we reverse the role of x and y i.e. we sample at dy=1 and calculate consecutive x values as

$$x_{k+1} = x_k + \frac{1}{m}$$

$$y_{k+1} = y_k + 1$$

Code:

```
#include<graphics.h>

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

int abs(int n){
   return ((n>0) ? n : ( n *(-1)));
}

void DDA(int X0, int Y0, int X1, int Y1){

   int dx = X1 - X0;
   int dy = Y1 - Y0;

   int steps = abs(dx) > abs(dy) ? abs(dx) : abs(dy);

   float x_inc = dx / (float) steps;
   float y_inc = dy / (float) steps;

   float X = X0;
   float Y = Y0;
   for(int i = 0; i <= steps; i++){
      putpixel (X,Y,WHITE);
      X += x_inc;
      Y += y_inc;
   }
}

int main(){

   int gd=DETECT,gm;
   initgraph(&gd,&gm,"C:\\TURBOC3\\BGI");

      DDA(150,300,250,300);
      DDA(150,180,250,180);
      DDA(150,300,150,180);
      DDA(250,300,250,180);

      DDA(180,300,220,300);
      DDA(180,250,220,250);
      DDA(180,300,180,250);
      DDA(220,300,220,250);
```

```
        DDA(160,230,190,230);
        DDA(160,200,190,200);
        DDA(160,230,160,200);
        DDA(190,230,190,200);

        DDA(210,230,240,230);
        DDA(210,200,240,200);
        DDA(210,230,210,200);
        DDA(240,230,240,200);

        DDA(200,100,150,180);
        DDA(200,100,250,180);

    getch();
    closegraph();
    return 0;
}
```

Output: