1. Write C++ program for line drawing using Bresenhams algorithm with patterns such as solid, dotted, dashed, dash dot and thick

**Title of Assignment:** Draw a line with styles.

**Problem Definition:** Write C++ program for line drawing using DDA or Bresenhams algorithm with patterns such as solid, dotted, dashed, dash dot and thick.

**Prerequisite:** Basic primitives of computer graphics and concepts of OOP.

**Learning Objective:** Understand the functions available in Graphics library.

### Relevant Theory / Literature Survey:

**LINE:** It is the path between two end points. We can draw line by two methods

### Using inbuilt function

1. **Using pixel filling (by logic)**

### Using inbuilt function

**Purpose: -** draws a line between two specified points

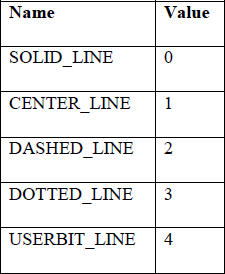
### Syntax: -void far line(int x1, int y1, int x2, int y2);

**Comments: -**Line() draws a line in the current color using current line style and thickness.

**Linestyle :** This is a function used to sets the linestyle and width or pattern . Sets the style for all lines drawn by line, lineto, rectangle,drawpoly, etc.

### Syntax : void far setlinestyle(int style, unsigned pattern, int thickness)

Total 5 styles are available in C Compiler



**upattern** is a 16-bit pattern that applies only if linestyle is USERBIT\_LINE (4). In that case, whenever a bit in the pattern word is 1, the corresponding pixel in the line is drawn in the current drawing color.

For example, a solid line corresponds to a upattern of 0xFFFF (all pixels drawn), while a dashed line can correspond to a upattern of 0x3333(short dashes) or 0x0F0F(long dashes) or 0x3F3F (longer dashes).

**Thickness:** Sets the width of the line. Two available widths are given in the table.

|  |  |  |
| --- | --- | --- |
| **Name** | **Value** | **Meaning** |
| NORM\_WIDTH | 1 | 1 pixel |
| THICK\_WIDTH | 3 | 3 pixel |

### Using pixel filling (by logic)

* **Solid Line**

We can use general line drawing algorithm to display solid line.Such as DDA, Bresenhams etc.

### Dotted Line

We can easily modify the general line drawing algorithm to display dotted line. By plotting the alternate pixels in line we can display the dotted line

### Dashed Line

We can easily modify the general line drawing algorithm to display dashed line. We have to plot alternate group of pixels along the line to get dashed line.

### Thick Line

To produce a thick line we have to run two line drawing algorithms in parallel to find the pixel along the line edges.

### Properties of good line drawing algorithm

* 1. Straight lines should appear as straight lines.
  2. They should start and end accurately.
  3. Displayed lines should have constant brightness along their length, independent of the line length and orientation
  4. Line should be drawn rapidly

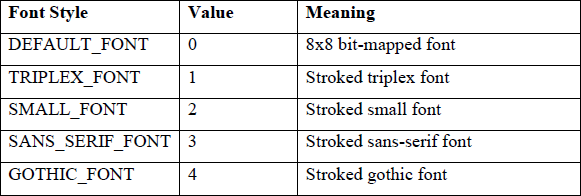
### Setting font/text Style and Size

There are various functions in graphics library for text operations.

**settextstyle() :** Sets the current text characteristics i:e text font style, the text direction and its font size

### void far settextstyle(int Style,int Horizontal/Vertical,int Size);

Available font syles are :



Two available directions are:

|  |  |  |
| --- | --- | --- |
| **Name** | **Value** | **Direction** |
| HORIZ\_DIR | 0 | Left to Right |
| VERT\_DIR | 1 | Bottom to Top |

**settextjustuify() :** It is used for text alignment i:e left, right and center

### void far settextjustify(int hor, int vert)

Three available alignments are

|  |  |  |
| --- | --- | --- |
| **Name** | **Value** | **Direction** |
| LEFT\_TEXT | 0 | Left |
| CENTER\_TEXT | 1 | Center |
| RIGHT\_TEXT | 2 | Right |

### Questions:-

Q 1: How to draw line with different styles without using inbuilt function? Q 2: What are the properties of line drawing algorithm?

Q 3: Which arguments should pass for line style? Q 4: how to draw thick and thin line?

### Conclusion:-

In This way we have studied that how to draw a line with line style.

1. **Write C++ program to draw circle using Bresenham‘s algorithm**

**Title of Assignment:** Draw a Circle.

**Problem Definition:** Write C++ program to draw circle using Bresenham‘s algorithm.

* 1. **Prerequisite:** Line Generation algorithms.

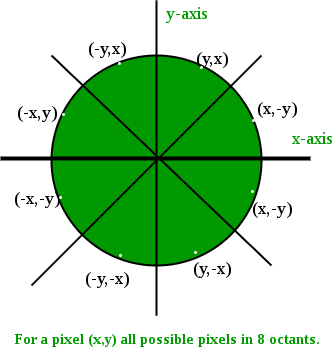
### Learning Objective:

Understand the concept of circle generation algorithms.

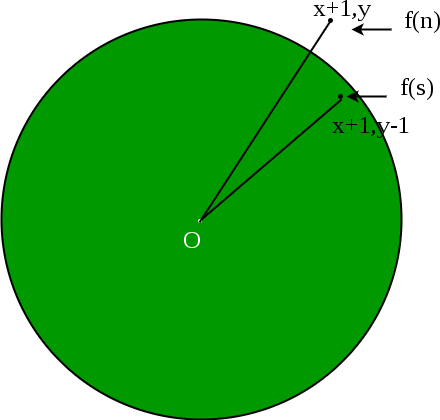
### Relevant Theory / Literature Survey:

Drawing a circle on the screen is a little complex than drawing a line. There are two popular algorithms for generating a circle − **Bresenham’s Algorithm** and **Midpoint Circle Algorithm**. These algorithms are based on the idea of determining the subsequent points required to draw the circle. Let us discuss the algorithms in detail −

The equation of circle is *X*2+*Y*2=*r*2, where r is radius.

For every pixel (x, y) it calculates, we draw a pixel in each of the 8 octants of the circle as shown below,

### Bresenham’s Algorithm

Now, we will see how to calculate the next pixel location from a previously known pixel location (x, y). In Bresenham’s algorithm at any point (x, y) we have two option either to choose the next pixel in the east i.e. (x+1, y) or in the south east i.e. (x+1, y-1).

And this can be decided by using the decision parameter d as:

* If d > 0, then (x+1, y-1) is to be chosen as the next pixel as it will be closer to the arc.
* else (x+1, y) is to be chosen as next pixel.

Now to draw the circle for a given radius ‘r’ and centre (xc, yc) We will start from (0, r) and move in first quadrant till x=y (i.e. 45 degree). We should start from listed initial condition:

d = 3 - (2 \* r) x = 0

y = r

Now for each pixel, we will do the following operations:

* + 1. Set initial values of (xc, yc) and (x, y)
    2. Set decision parameter d to d = 3 – (2 \* r).
    3. Repeat steps 4 to 8 until x < = y
    4. call drawCircle(int xc, int yc, int x, int y) function.
    5. Increment value of x.

6. If d < 0, set d = d + (4\*x) + 6

1. Else, set d = d + 4 \* (x – y) + 10 and decrement y by 1.
2. call drawCircle(int xc, int yc, int x, int y) function

### drawCircle() function:

// function to draw all other 7 pixels

// present at symmetric position drawCircle(int xc, int yc, int x, int y)

{

putpixel(xc+x, yc+y, RED); putpixel(xc-x, yc+y, RED); putpixel(xc+x, yc-y, RED); putpixel(xc-x, yc-y, RED); putpixel(xc+y, yc+x, RED); putpixel(xc-y, yc+x, RED); putpixel(xc+y, yc-x, RED); putpixel(xc-y, yc-x, RED);

}

### Questions:-

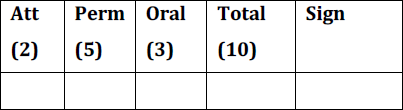
Q 1: Explain the Bresenhams circle drawing algorithm? Q 2: Explain the circle equation.

Q 3: Difference between Bresenham’s and Mid point Circle drawing algorithm?

### Conclusion:-

In This way we have studied that how to draw a circle on the screen.

**10 (a) Write C++ program to simulate following scene- Clock with pendulum.**



**Title of Assignment:** Clock with pendulum.

**Problem Definition:** Write C++ program to simulate following scene- Clock with pendulum

**Prerequisite:** Line & Circle Generation algorithms.

**Learning Objective:** To study about different animation creation using Line & Circle Generation algorithms with moving the line in non-linear path.

### Relevant Theory / Literature Survey:

A **pendulum** is a weight suspended from a pivot so that it can swing freely. When a pendulum is displaced sideways from its resting, equilibrium position, it is subject to a restoring force due to gravity that will accelerate it back toward the equilibrium position. When released, the restoring force combined with the pendulum's mass causes it to oscillate about the equilibrium position, swinging back and forth. The time for one complete cycle, a left swing and a right swing, is called the period. The period depends on the length of the pendulum and also to a slight degree on the amplitude, the width of the pendulum's swing.

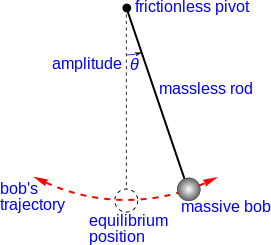


Figure: Simple gravity pendulum" model assumes no friction or air resistance.

### Questions:-

Q 1: Explain different Line Generation algorithms? Q 2: Explain different Circle Generation algorithms?

Q 3: Discuss the advantages and disadvantages of different Line Generation algorithms? Q 4: Discuss the advantages and disadvantages of different Circle Generation algorithms? Q.5: Explain the technique used for moving the pendulum?

### Conclusion:-

Thus we had created a simple clock with moving pendulum using C++.

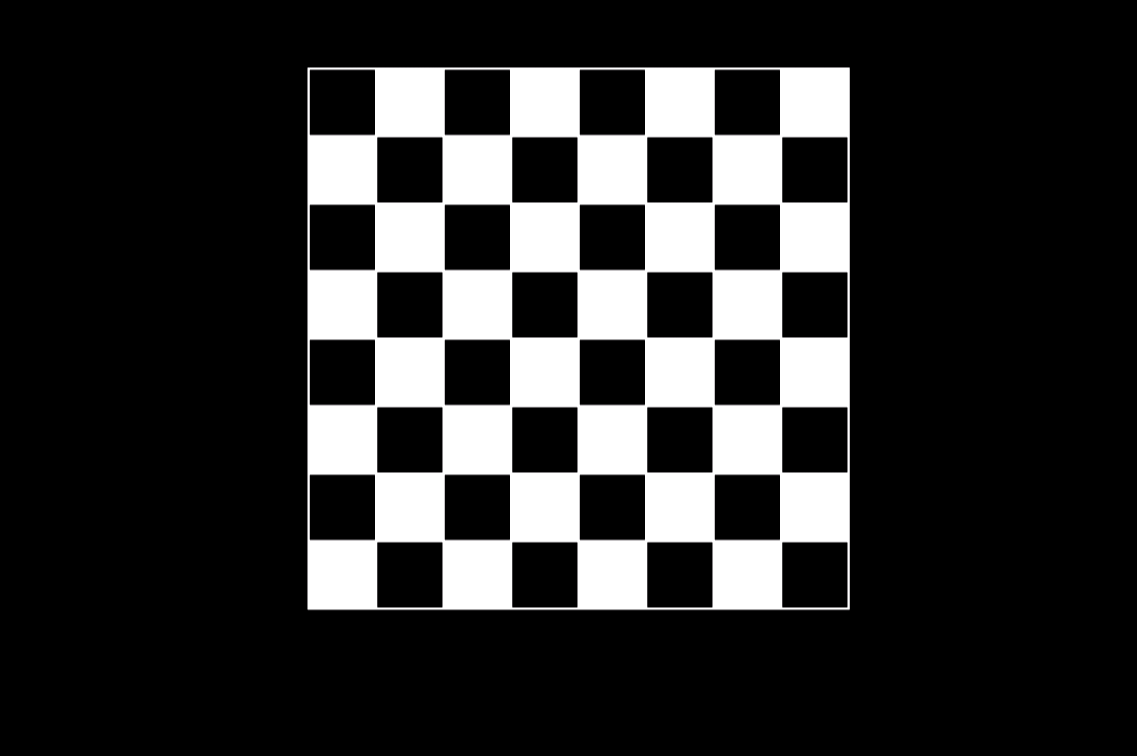
**4.write c++ program to draw a 4\*4 chessboard use dda and bresenham's drawing algorithm to draw lines use seed fill algorithm to fill black squares of the board**

[Computer Graphics](https://www.geeksforgeeks.org/computer-graphics-2/), we use [graphics.h](https://www.geeksforgeeks.org/add-graphics-h-c-library-gcc-compiler-linux/) which provide direct functions to draw different coordinate shapes(like circle, rectangle etc). By using these functions we can draw different objects like car, hut, trees etc. In this program, the task is to draw a **Chess Board** using the functions in [graphics](https://www.geeksforgeeks.org/basic-graphic-programming-in-c/).

To run the program we have the include the below header file

#include <graphic.h>

**Approach:** We will create a Chess Board with the help below functions:



1. [rectangle(left, top, right, bottom)](https://www.geeksforgeeks.org/intersecting-rectangle-when-bottom-left-and-top-right-corners-of-two-rectangles-are-given/)**:** A function from **graphics.h** header file which is used to draw a rectangle. Coordinates of the left top and right bottom corners are required to draw the rectangle. left specifies the X-coordinate of the top left corner, top specifies the Y-coordinate of the top left corner, right specifies the X-coordinate of the right bottom corner, bottom specifies the Y-coordinate of the right bottom corner.
2. **delay():** This function is present in library **“dos.h”** is used for holding the program output for a small period of time since processing is very fast so use it to see the result.
3. [setcolor()](https://www.geeksforgeeks.org/setcolor-function-c/)**:** A function from graphics.h header file which sets the color of the pointer (cursor). There are some predefined colors in computer graphics. Here n is the color number.
4. [setfillstyle():](https://www.geeksforgeeks.org/setfillstyle-floodfill-c/) A function from **graphics.h**header file which sets the current fill pattern and fill color.
5. [floodfill()](https://www.geeksforgeeks.org/setfillstyle-floodfill-c/)**:** A function from **graphics.h** header file which is used to fill an enclosed area.

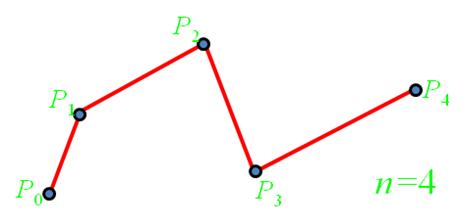
## **8.(a)write a program to draw bezier curve using basic concept of object oriented programming algorithm**

## **Introduction**

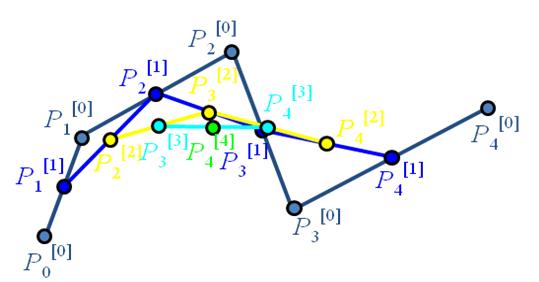
Bezier curves are the most fundamental curves, used generally in computer graphics and image processing. These curves are mainly used in interpolation, approximation, curve fitting, and object representation. In this article, I will demonstrate, in a very simple and straightforward way, how one can construct these curves and make use of them.

##### Constructive Bezier Curve Algorithm

Consider the n+1 points P0,…,Pn and connect the points into a polyline we will denote hereafter as the control polygon.



Given points Pi, i = 0,...,n, our goal is to determine a curve g (t), for all values t Î [0,1]. The idea is demonstrated below:



##### **8.(a) write a program to draw Bezier curve using basic concept of object-oriented programming algorithm**

##### Basic Algorithm

The objective here is to find points in the middle of two nearby points and iterate this until we have no more iterations. The new values of points will give us the curve. The famous Bezier equation is the exact formulation of this idea. Here is the algorithm:

**Step 1**: Select a value t Î [0,1]. This value remains constant for the rest of the steps.

**Step 2**: Set Pi[0] (t) = Pi, for i = 0,...,n.

**Step 3**: For j= 0,...,n, set  for i = j,...,n.

**Step 4**: g (t) = Pn[n] (t)

##### Special & General Cases

Now, I will give formulas for common, special cases that can be helpful in certain applications. The code of the article does not demonstrate any of them, but it uses the generalized formula. So, let me start with the generalized formula:



For the sake of simplicity and convention used in this article and code, it is better to represent this formula as:



What this equation tells us is nothing but the formulation of the above algorithm (the mid-point iterations). It is very important in the sense that a whole algorithm could be summarized into a formula and a straightforward implementation would yield correct results. Here, n denotes the number of points and P denotes the points themselves. The factorial coefficients of the points are simply called the Bernstein basis functions, because of the name of the founder.

Here are the special cases:

Linear Bezier:



Quadratic Bezier:



Cubic Bezier:

