

# CG Experiment No. 9.

Aim: Character Generation: Bit Map  
Stroke Method.

Theory: BITMAP Method

This method for character generation is the bitmap method. It is also called dot matrix because in this method character are represented by an array of dots in the matrix form. It is a two dimensional array having ~~array~~ columns & rows. A  $5 \times 7$  array is commonly used to represent character.

However  $7 \times 9$  &  $9 \times 13$  arrays are also used. Higher resolution devices such as inkjet printer or laser printer may use character arrays that are over  $100 \times 100$ .



Each dot in the matrix is a pixel.

The character is placed on the screen by copying pixel values from the character array into some position of the screen's frame buffer. The value of the pixel controls the intensity.

Character Generation by using ~~stroke~~ stroke method:-

This method uses small line segments to generate a character. The small series of line segments are drawn like a stroke of a pen. We build our own stroke method character generator by calls to the line drawing algorithm.

# **Program:**

## **1. Bitmap Method:**

```
#include <stdio.h>
#include <conio.h>
#include <graphics.h>
void main()
{
    int i,j,k,x,y;
    int gd=DETECT,gm;//DETECT is macro defined in graphics.h
    /* ch1 ch2 ch3 ch4 are character arrays that display alphabets */
    int ch1[][10]={ {1,1,1,1,1,1,1,1,1,1},
                    {1,1,1,1,1,1,1,1,1,1},
                    {0,0,0,0,1,1,0,0,0,0},
                    {0,0,0,0,1,1,0,0,0,0},
                    {0,0,0,0,1,1,0,0,0,0},
                    {0,0,0,0,1,1,0,0,0,0},
                    {0,0,0,0,1,1,0,0,0,0},
                    {0,1,1,0,1,1,0,0,0,0},
                    {0,1,1,0,1,1,0,0,0,0},
                    {0,0,1,1,1,0,0,0,0,0}};
    int ch2[][10]={ {0,0,0,1,1,1,1,0,0,0},
                    {0,0,1,1,1,1,1,0,0},
                    {1,1,0,0,0,0,0,0,1,1},
                    {1,1,0,0,0,0,0,0,1,1},
                    {1,1,0,0,0,0,0,0,1,1},
                    {1,1,0,0,0,0,0,0,1,1},
                    {1,1,0,0,0,0,0,0,1,1},
                    {1,1,0,0,0,0,0,0,1,1},
                    {0,0,1,1,1,1,1,0,0},
                    {0,0,0,1,1,1,1,0,0}};
    int ch3[][10]={ {1,1,0,0,0,0,0,0,1,1},
                    {1,1,0,0,0,0,0,0,1,1},
                    {1,1,0,0,0,0,0,0,1,1},
```

```

        {1,1,0,0,0,0,0,0,1,1},
        {1,1,1,1,1,1,1,1,1,1},
        {1,1,1,1,1,1,1,1,1,1},
        {1,1,0,0,0,0,0,0,1,1},
        {1,1,0,0,0,0,0,0,1,1},
        {1,1,0,0,0,0,0,0,1,1},
        {1,1,0,0,0,0,0,0,1,1}};

int ch4[][10]={ {1,1,0,0,0,0,0,0,1,1},
        {1,1,1,1,0,0,0,0,1,1},
        {1,1,0,1,1,0,0,0,1,1},
        {1,1,0,1,1,0,0,0,1,1},
        {1,1,0,0,1,1,0,0,1,1},
        {1,1,0,0,1,1,0,0,1,1},
        {1,1,0,0,0,1,1,0,1,1},
        {1,1,0,0,0,1,1,0,1,1},
        {1,1,0,0,0,0,1,1,1,1},
        {1,1,0,0,0,0,0,0,1,1}};

initgraph(&gd,&gm,"D:\\TC\\BGI");//initialize graphic mode
setbkcolor(LIGHTGRAY);//set color of background to darkgray
for(k=0;k<4;k++)
{
    for(i=0;i<10;i++)
    {
        for(j=0;j<10;j++)
        {
            if(k==0)
            {
                if(ch1[i][j]==1)
                    putpixel(j+250,i+230,RED);
            }
            if(k==1)
            {
                if(ch2[i][j]==1)
                    putpixel(j+300,i+230,RED);
            }
            if(k==2)

```

```

    {
        if(ch3[i][j]==1)
            putpixel(j+350,i+230,RED);
    }
    if(k==3)
    {
        if(ch4[i][j]==1)
            putpixel(j+400,i+230,RED);
    }
}
delay(200);
}
}
getch();
closegraph();
}

```

## 2. Stroke method:

```

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

void main() {

    float dx, dy, step;
    int j, i, x[3], y[3], gd = DETECT, gm;
    int p, q;

    initgraph( & gd, & gm, "c:\\turbo3\\bgi");

    for (j = 0; j < 3; j++) {
        printf("Enter the value of x(j) and y(j) : ");
        scanf("%d%d", & x[j], & y[j]);
        printf("Enter the value of x(j+1) and y(j+1) : ");
    }
}

```

```
scanf("%d%d", & x[j + 1], & y[j + 1]);
```

```
dx = abs(x[j + 1] - x[j]);
```

```
dy = abs(y[j + 1] - y[j]);
```

```
if (dx >= dy)
```

```
    step = dx;
```

```
else
```

```
    step = dy;
```

```
dx = dx / step;
```

```
dy = dy / step;
```

```
p = x[j];
```

```
q = y[j];
```

```
i = 1;
```

```
while (i <= step) {
```

```
    putpixel(p, q, 15);
```

```
    p = p + dx;
```

```
    q = q + dy;
```

```
    i = i + 1;
```

```
    delay(100);
```

```
}
```

```
getch();
```

```
}
```

```
closegraph();
```

```
}
```

-----

## Output:-

J O H N

```
Enter the value of x(j) and y(j) : 100 100
Enter the value of x(j+1) and y(j+1) : 100 200
Enter the value of x(j) and y(j) : 100 100
Enter the value of x(j+1) and y(j+1) : 150 100
Enter the value of x(j) and y(j) : 100 150
Enter the value of x(j+1) and y(j+1) : 125 150
```

