

Name - Ayush Dhasmana
Course - BCA 6th A
University roll no. 1121032

Class roll no. 29
Subject - Com. graphics
and animation (practical)

Ans 1

```
#include <stdio.h>
#include <graphics.h>

int main()
{
    int x0(float num)
    {
        return num < 0 ? num - 0.5 : num + 0.5;
    }

    int x1 = 100, x2 = 250, y1 = 100, y2 = 250, step;
    int gd = DETECT, gm;
    float x, y, m;
    int dx = x2 - x1;
    int dy = y2 - y1;
    m = dy / dx;
    if (dx > dy)
        step = dx;
    else
        step = dy;
    initgraph(&gd, &gm, "");
    outtextxy(x1, y1, "A");
    outtextxy(x2, y2, "B");
```

```

    putpixel(x1, y1, RED);
    x = x1, y = y1;
    while (step > 0)
    {
        if (m < 1)
        {
            x = x + 1;
            y = y + m;
        }
        if (m >= 1)
        {
            x = x + 1 / m;
            y = y + 1;
        }
        putpixel(x, y, RED);
        step--;
    }
    getch();
    return 0;
}

```

Algorithm

step 1: Start Algorithm

step 2: Declare $x_1, y_1, x_2, y_2, dx, dy, x, y$ as integer variables

step 3: initialize the value of x_1, y_1, x_2, y_2

step 4: calculate $dx = x_2 - x_1$

step 5: Calculate $dy = y_2 - y_1$

● step 6: if $dx > dy$

then $step = dx$; else $step = dy$;

step 7: assign $x = x_1, y = y_1$

step 8: set pixel (x, y)

step 9: starting while loop ($step > 0$)

if ($m < 1$) then $x = x + 1$;
 $y = y + m$;

● and if ($m >= 1$) then $x = x + 1/m$;
 $y = y + 1$;

step 10: set pixel ($x_{ov}(x), x_{ov}(y)$)

step 11: repeat step 9 until $x = x_2$

step 12: End

Name - Ayush Dhasmana
Course - BcA 6th A
Subject - Com. graphics and animation
(practical and term)

Class roll no. 29
University roll no. 1121032

Ans 3

```
#include <stdio.h>
#include <graphics.h>

int main()
{
    int gd = DETECT, gm;
    initgraph(&gd, &gm, "")
    line(0, 200, getmaxx(), 200);
    line(0, 360, getmaxx(), 360);
    setcolor(WHITE);

    xrectangle(150, 210, 260, 230);
    floodfill(152, 220, WHITE);

    xrectangle(152, 240, 260, 260);
    floodfill(152, 241, WHITE);

    xrectangle(150, 270, 260, 290);
    floodfill(152, 271, WHITE);

    xrectangle(150, 300, 260, 320);
    floodfill(152, 301, WHITE);

    xrectangle(140, 200, 145, 130);
```

```
rectangle (130, 130, 155, 70);
```

```
setcolor (RED);
```

```
circle (142, 82, 6);
```

```
fill (142, 82, RED);
```

```
setcolor (YELLOW);
```

```
circle (142, 100, 6);
```

```
fill (142, 100, YELLOW);
```

```
setcolor (GREEN);
```

```
circle (142, 118, 6);
```

```
fill (142, 118, GREEN);
```

```
setcolor (WHITE)
```

```
getch();
```

```
closegraph();
```

```
return 0;
```

```
}
```

SDL-ibgraph - Graphics on GNU/Linux

2:10 PM

```
adn@adn-lab6-pc:~$ touch dda.c
adn@adn-lab6-pc:~$ gedit dda.c
adn@adn-lab6-pc:~$ gcc dda.c -lgraph -o dda
dda.c: In function 'main':
dda.c:136:16: error: request for member 'row' in something not a structure
on
    putpixel(row(x).row(y),RED);
               ^
dda.c:136:11: error: too few arguments to function 'putpixel'
    putpixel(row(x).row(y),RED);
            ^
In file included from dda.c:2:0:
/usr/local/include/graphics.h:72:6: note: declared here
void putpixel(int x, int y, int color);
      ^
adn@adn-lab6-pc:~$ gcc dda.c -lgraph -o dda
adn@adn-lab6-pc:~$ ./dda
[xcb] Unknown sequence number while processing queue
[xcb] Most likely this is a multi-threaded client and XinitThreads has not
called
[xcb] Aborting, sorry about that.
dda: .././src/xcb_to.c:274: poll_for_event: Assertion 'xcb_xlib_threads_
ce_last' failed.
```

SDL-ibgraph - Graphics on GNU/Linux



