Name - Himanshu Baghari
U. Roll No - 1121060

Subject Name. Computer Graphics and Animation
Subject Code - PBC-602

Ans 9. Midpoint Circle Drawing Algorithm -

→ Algorithm.

Step 1. Put $x = 0$, $y = r$ in equation 2

we have $p = 1 - r$

Step 2. Repeat steps while $x \leq y$.

plot (x, y)

if $(p < 0)$

Then set $p = p + 2x + 3$

Else

$p = p + 2(x - y) + 5$

$y = y - 1$ (end if)

$x = x + 1$ (end loop)

Step 3. End

→ Code :-

```
# include <graphics.h>
# include <stdlib.h>
# include <math.h>
# include <stdio.h>
# include <conio.h>
# include <iostream.h>
class bresen
{
```

```
    float x, y, a, b, r, p;
    public:
    void get();
    void cal();
};
    void main()
    {
      bresen b;
      b.get();
      b.cal();
      getch();
    }
    void bresen::get()
{
    cout<<" Enter center and Radius";
    cout<<" Enter (a, b)";
    cin>>a>>b;
    cout<<"Enter r";
    cin>>r;
}
    void bresen::cal()
{
    int gdriver = DETECT, gmode, errorcode;
    int midx, midy, i;
    initgraph (&gdriver, &gmode, " ");
    errorcode = graphresult();
    if (errorcode != grOK)
```

```
{
printf ("Graphics error: %s\n", grapherrormsg (errorcode));
printf ("Press any key to halt:");
getch ();
exit (1);
}
x = 0;
y = r;
putpixel (a, b+r, RED);
putpixel (a, b-r, RED);
putpixel (a-r, b, RED);
putpixel (a+r, b, RED);
P = 5/4) - r;
while (x <= y)
{
  if (P<0)
  P += (4 * x) + 6;
  else
  {
    P += (2 * (x-y)) + 5;
    y--;
  }
  x++;
putpixel (a+x, b+y, RED);
putpixel (a-x, b+y, RED);
putpixel (a+x, b-y, RED);
```

```
        putpixel (a+x, b-y, RED);
        putpixel (a+x, b+y, RED);
        putpixel (a+x, b-y, RED);
        putpixel (a-x, b+y, RED);
        putpixel (a-x, b-y, RED);
    }
}
```

NeuTroN DOS-C++ 0.77, Cpu speed: max 100% cycles, Frameskip 0, Program:

```
Enter radius of circle: 100
Enter co-ordinates of center(x and y): 150
150
```