EXPERIMENT 4 BEYOND SYLLABUS

COMPUTER GRAPICS AND MULTIMEDIA

Aim

To Write a program in C to display a digital and analog clock displaying current time.

Syeda Reeha Quasar
14114802719
3C7

EXPERIMENT - 4

AIM:

To Write a program in C to display a digital and analog clock displaying current time.

Digital Clock

Source Code:

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

int main(){
	initwindow(1000, 500);

	time_t rawTime;
	struct tm * currentTime;
	char a[100];

	while(1) {
		rawTime = time(NULL);
		currentTime = localtime(&rawTime);
		strftime(a, 100, "%I:%M:%S", currentTime);

		setcolor(11);
		settextstyle(3, HORIZ_DIR, 10);
		outtextxy(200, 100, a);

		strftime(a, 100, "%p", currentTime);
		settextstyle(3, HORIZ_DIR, 2);
		outtextxy(600, 8, a);
```

```
            strftime(a, 100, "%a, %d %b, %Y", currentTime);
            settextstyle(3, HORIZ_DIR, 5);
            outtextxy(130, 310, a);

            delay(1000);
        }

        getch();
        return 0;
}
```

Output:

Analog Clock

Source Code:

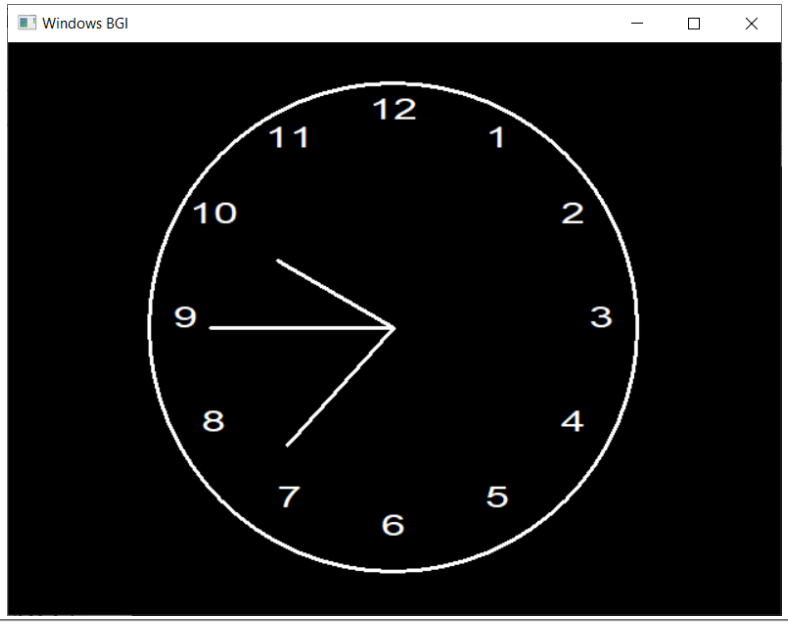
```
/*Program for analog CLock*/
 #include <stdio.h>
 #include <conio.h>
 #include <math.h>
 #include <string.h>
 #include <graphics.h>
 #include <time.h>
 #include <dos.h>

void minSecPos(int xrad, int midx, int midy, int x[60], int y[60])
{
 int i, j=45;
 for (i=360; i>=0; i=i-6)
 {
  x[j] = midx-(xrad*cos((i*3.14)/180));
  y[j--] = midy-(xrad*sin((i*3.14)/180));
  j = (j==-1)?59:j;
 }
 return;
}

 void calcPoints(int radius, int midx, int midy, int x[12], int y[12])
{
 int x1, y1;
 x[0] = midx, y[0] = midy-radius;
 x[6] = midx, y[6] = midy+radius;
 x[3] = midx+radius, y[3] = midy;
 x[9] = midx-radius, y[9] = midy;
```

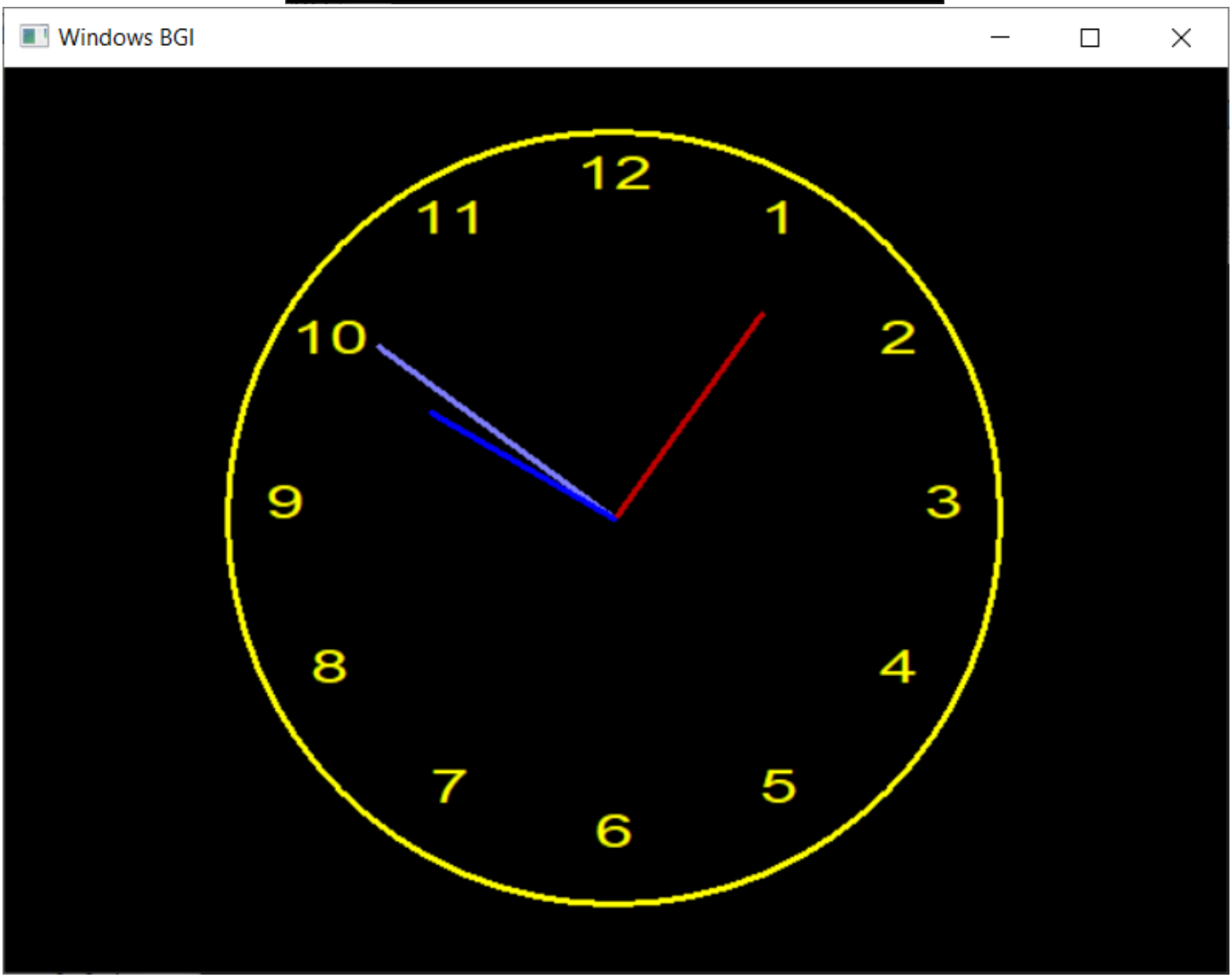
```
 x1 = (int) ((radius/2)*sqrt(3));
 y1 = (radius/2);
 x[2] = midx+x1, y[2] = midy-y1;
 x[4] = midx+x1, y[4] = midy+y1;
 x[8] = midx-x1, y[8] = midy+y1;
 x[10] = midx-x1, y[10] = midy-y1;

 x1 = radius/2;
 y1 = (int) ((radius/2)*sqrt(3));
 x[1] = midx+x1, y[1] = midy-y1;
 x[5] = midx+x1, y[5] = midy+y1;
 x[7] = midx-x1, y[7] = midy+y1;
 x[11] = midx-x1, y[11] = midy-y1;
 return;
}

int main() {
  int gd=DETECT, gm, err, tmp;
  initgraph(&gd, &gm, "C:\\tc\\bgi");

 int i, j, midx, midy, radius, hr, min, sec;
int x[12], y[12], minx[60], miny[60];
int hrx[12], hry[12], secx[60], secy[60];
int secx1, secy1;
char str[256];
time_t t1;
struct tm*data;

 err = graphresult();

 if (err != grOk)
```

```
{
 printf("Graphics Error: %s",
 grapherrormsg(err));
 return 0;
}

 midx = getmaxx()/2;
 midy = getmaxy()/2;

 radius = 200;

 calcPoints(radius-30, midx, midy, x, y);
calcPoints(radius-90, midx, midy, hrx, hry);

 minSecPos(radius-50, midx, midy, minx, miny);
minSecPos(radius-70, midx, midy, secx, secy);

 while (!kbhit())
{
 setlinestyle(SOLID_LINE, 1, 3);
 settextstyle(GOTHIC_FONT, 0, 3);

          setcolor(14);
  circle(midx, midy, radius);

  for (j=0; j<12; j++)
 {
  if (j==0)
   {
    sprintf(str, "%d", 12);
     } else {
   sprintf(str, "%d", j);
```

```
  }
 settextjustify(CENTER_TEXT, CENTER_TEXT);
 moveto(x[j], y[j]);
 outtext(str);
 }

 t1 = time(NULL);
data = localtime(&t1);

 sec = data->tm_sec % 60;

 setcolor(4);
 line(midx, midy, secx[sec], secy[sec]);

 min = data->tm_min % 60;

 setcolor(9);
line(midx, midy, minx[min], miny[min]);

 hr = data->tm_hour % 12;

 setcolor(1);
line(midx, midy, hrx[hr], hry[hr]);
delay(1000);
cleardevice();
}

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

Output:


Windows BGI
12
1
2
3
4
5
6
7
8
9
10
11


Windows BGI
12
1
2
3
4
5
6
7
8
9
10
11