#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);

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;

Line(midx, midy,

[sec], secy[sec]);

Min = data->tm\_min % 60;

Line(midx, midy, minx[min], miny[min]);

Hr = data->tm\_hour % 12;

Line(midx, midy, hrx[hr], hry[hr]);

Delay(1000);

Cleardevice();

}

Getch();

Closegraph();

Return 0;

}