```
/*
      lab5.c
      Nathan Christian Copyright (C) (nchristi@iu.edi) 2017
      ALL FUNCTIONS ARE LOCATED IN FUNCS.C
*/
#include <stdio.h>
#include <stdlib.h>
#include "fig.h"
#include "shape.h"
#define MAXBUF 1024
#define MAX         64

int main(int argc, char *argv[], char *env[])
{
      Polygon  p;
      Polyline l;
      Circle      c;
      Arc         a;

      Point temp_low, temp_high, phigh, plow, lhigh, llow;
      Point chigh, clow, ahigh, alow, low, high, center;
      high.x=high.y=ahigh.x=ahigh.y=chigh.x=chigh.y=-1000000;
      lhigh.x=lhigh.y=phigh.x= phigh.y=-1000000;
      low.x=low.y=alow.x=alow.y=clow.x=clow.y= 10000000;
      llow.x=llow.y=plow.x=plow.y=1000000;
                        //Initializes Points used to find high/low
boundaries

      int i;
      double dist, angle=90;
                                    //Initialize angle to default of 90
degrees

      char buffer[MAXBUF], command[MAX], filename[MAX], language[MAX];

      for(i=1;i<argc;i++){
                  if (argv[i][0]=='-'){
                              switch (argv[i][1]){
                                    case 'h': case 'H':
                                                print_help();
                                                      //Help function
located in funcs.c
                                    break;
                                    case 'A': case 'a':
                                                if (argv[i+1] != NULL)
                                                            sscanf(argv[i+1],
"%lf", &angle);                                 //change angle when entered
                                    break;
                                    case 'l': case 'L':
                                                strcpy(language,argv[i+1]);


                                    break;
                                    case 'v': case 'V':
```

```
                                                printf("The version is
5.0");
                                    break;
                                    case 'O': case 'o':
                                                strcpy(filename,argv[i+1]);
                                                //User selects file to open
                                    break;
                              }
                  }
      }
      angle=(angle*M_PI/180);
                                    //convert degree to radians

      FILE *fin, *fout;
    fin = fopen(filename, "r");
      if (fin == NULL) {
        fprintf(stderr, "Can't open file. Enter lab5 -h for help.");

        exit(1);
                              //basic error statement for NULL file
      }

      if (!strcmp(language,"fig") ||!strcmp(language, "FIG")){
                  fout=fopen("output.fig", "w");
                  fig_open(fout);
                  fig_userinfo(fout, 1, "ECE263L5P1", "Nathan Christian",
"6/27/2017");
      }
                                          //fig data prints on fig files
only

      else if (!strcmp(language,"shape") ||!strcmp(language, "SHAPE"))
                  fout=fopen("output.txt", "w");


      else {
                  printf("Error! Choose 'fig or 'shape'. Enter lab5 -h for
help.");
                  exit(1);
                                          //Error statement for files not
shape or fig
      }

      while (fget_line(buffer, MAXBUF, fin)) {
                        //Reads one line at a time

                  if (!strcmp(language,"shape") ||!strcmp(language,
"SHAPE"))
                  {
                              for (i=0; buffer[i]!='\0';i++){
                                    if (buffer[i]=='#')
                                                buffer[i]='\0';
                              }
```

```
                        fprintf(fout, "%s\n", buffer);
                                //For shape files, prints the
clean output to output.txt
                }

                else{
                        sscanf(buffer, "%s", command);
                                //Scans the first line to
determine shape type

                        if (!strcmp(command, "polygon")) {
                              polygon_sscanf(buffer, &p);
                              polygon_draw(fout, &p, NULL);
                              polygon_boundary(&p, &temp_low,
&temp_high);                        //Finds high/low of all shapes on
each line.
                              high_low(&plow, &phigh, &temp_low,
&temp_high);                        //Finds high/low of all shapes in
file. Located in funcs.c
                        }

                        else if (!strcmp(command, "polyline"))  {
                              polyline_sscanf(buffer, &l);
                              polyline_draw(fout, &l, NULL);
                              polyline_boundary(&l, &temp_low,
&temp_high);                        //Finds high/low of all shapes on
each line.
                              high_low(&llow, &lhigh, &temp_low,
&temp_high);                        //Finds high/low of all shapes in
file. Located in funcs.c

                        }
                        else if (!strcmp(command, "circle")) {
                              circle_sscanf(buffer, &c);
                              circle_draw(fout, &c, NULL);
                              circle_boundary(&c, &temp_low,
&temp_high);                              //Finds high/low of all
shapes on each line.
                              high_low(&clow, &chigh, &temp_low,
&temp_high);                        //Finds high/low of all shapes in
file. Located in funcs.c

                        }
                        else if (!strcmp(command, "arc")) {


                              arc_sscanf(buffer, &a);
                              arc_draw(fout, &a, NULL);
                              arc_boundary(&a, &temp_low,
&temp_high);                              //Finds high/low of all
shapes on each line.
                              high_low(&alow, &ahigh, &temp_low,
&temp_high);                        //Finds high/low of all shapes in
file. Located in funcs.c
```

```
                              }

                              else
                                    fprintf(stderr, "Unknown shape:
[%s]\n", buffer);                         //Error for unknown shape
                  }
      }
      if (!strcmp(language,"fig") ||!strcmp(language, "FIG")){
                  high_low(&low, &high, &alow, &ahigh);
                  high_low(&low, &high, &clow, &chigh);
                  high_low(&low, &high, &llow, &lhigh);
                  high_low(&low, &high, &plow, &phigh);
                  center.x=low.x+((high.x-low.x)/2);
                  center.y=low.y+((high.y-low.y)/2);
                              //Finds high/low of all points and center of
entire shape.

                  rewind(fin);
                                          //Rewinds file so we can read it
again.
      }

      else
                  exit(1);

      while (fget_line(buffer, MAXBUF, fin)) {
                        //Now that centerpoint is known, file is read again
to rotate.

                  if (!strcmp(language,"fig") ||!strcmp(language, "FIG")){

                              sscanf(buffer, "%s", command);


                              if (!strcmp(command, "polygon"))  {
                                    polygon_sscanf(buffer, &p);

                                    for (i=0; i<p.npoints;i++){
                                                polygon_move(&p, angle,
center);
                                                p.vertex++;
                                                      //Move to the next
coordinate and rotate until n.points are reached
                                    }

                                    for (i=0; i<p.npoints;i++)
                                                p.vertex--;
                                                      //Rewind shape for
drawing/printing.

                                    polygon_draw(fout, &p,
"color=red;line_style=1");                  //draw the rotated shape
                              }
```

```
                              else if (!strcmp(command, "polyline"))  {
                                    polyline_sscanf(buffer, &l);

                                    for (i=0; i<l.npoints;i++){
                                                polyline_move(&l, angle,
center);                                        //Rotates shape. function
located in func.c
                                                l.vertex++;
                                                      //Move to the next
coordinate and rotate until n.points are reached
                                    }
                                    for (i=0; i<l.npoints;i++)
                                                l.vertex--;
                                                      //Rewind shape for
drawing/printing.

                                    polyline_draw(fout, &l,
"color=red;line_style=1");
                              }

                              else if (!strcmp(command, "circle")) {
                                    circle_sscanf(buffer, &c);
                                    dist=distance(c.center, center);
                                    //dist function using pythagorean
theorum
                                    circle_move(&c, angle, dist, center);
                                    //Rotates shape. function located in
func.c
                                    circle_draw(fout, &c,
"color=red;line_style=1");
                              }
                              else if (!strcmp(command, "arc")) {

                                    arc_sscanf(buffer, &a);
                                    dist=distance(a.center, center);
                                    arc_move(&a, angle, dist, center);
                                    //Rotates shape. function located in
func.c
                                    arc_draw(fout, &a,
"color=red;line_style=1");
                              }
                        }
            }
      fclose(fin);
      fclose(fout);
                                    //  after use, close the files
}
```