```
#include <stdio.h>
#include <stdlib.h>
#include "fig.h"
#include "shape.h"
#include <math.h>
#define PI M_PI
#define MAXBUFFER 1024
#define MAX         64

/*
      Nathan Christian, IUPUI, (nchristi@iu.edu) Copyright (c) 2017
*/
char *name1, *name2;
void select_circles(int argc, char *argv[]);
      //Allows user to select which circles to connect.

int main(int argc, char *argv[], char *env[])
{
      Shape s1, s2, *line;
      char buffer[MAXBUFFER], temp_name[MAXBUFFER];
    FILE *fin, *fout;

      select_circles(argc, argv);
            //Call circle selection

      fin = fopen("lab06-testcases.txt", "r");
      if (fin==NULL){
                  printf("Error reading fin.");
                  exit(0);
      }
      strcpy(temp_name, name1);
      strcat(temp_name, "-");
      strcat(temp_name, name2);
      strcat(temp_name, ".fig");
                  //assign temp_name the value of the filename (circle1-
circle2.fig)

      fout = fopen(temp_name, "w");
      if (fout==NULL){
                  printf("Error reading fout.");
                  exit(0);
      }

      fig_open(fout);
      fig_userinfo(fout, 1, "ECE263L6", "Nathan Christian", "7/14/2017");

      while (fget_line(buffer, MAXBUFFER, fin)) {

                  sscanf(buffer, "%s %s", temp_name, temp_name);
            //Scan for the name of the circles

                  if (strcmp(name1, temp_name)==0){
                              shape_sscanf(buffer, &s1);
```

```
                        shape_draw(fout, &s1, NULL);
                  //If selected circle is read, print it.
            }
            if (strcmp(name2, temp_name)==0){
                        shape_sscanf(buffer, &s2);
                        shape_draw(fout, &s2, NULL);
                  //If second selected circle is read, print it.
            }

      }


      line=malloc(sizeof(Shape));
            //allocate memory for one shape.
      line->object.polyline.vertex=malloc(sizeof(Point)*2);
      //allocate memory for two Points.
      line=connect_line(&s1, &s2);
      //calculates location of points and returns pointer to shape.
      line->object.polyline.npoints=2;
      //npoints will always be 2 for this function.
      shape_draw(fout, line, NULL);


      strcat(name2, "-");
      strcat(name2, name1);
      fig_text(fout, 10, 25, name2, name2);
      //combines names of circles and prints to fig.

      free(line);
      free(line->object.polyline.vertex);
      fclose(fin);
      fclose(fout);
            //Free memory and close files.
}

void select_circles(int argc, char *argv[])
{
            char *a;
      argv++;
      a = *argv;

      if (a==NULL){
            printf("Error, '-' not read, type draw -h for help");
            exit(0);
                  //prints error message if user types draw instead
of draw-c
      }

      if (*a++ == '-') {
            argv++;
            switch (*a) {
                        case 'h' :
                              print_help();
```

```
                              exit(0);
                        //prints help message and exits
                        break;
                        case 'c' :  {
                              name1 = *argv;
                              argv++;
                              name2=*argv;
                              argv++;
                              break;
                        default:
                              printf("Error: No h or c read. type
draw -h for help");
                              exit(0);
                        //prints error message if user forgets to
type h or c
                              break;
                        }
            }
      }
      else {
            printf("'-' was not read, enter draw -h for help.");
            exit(0);
                  //prints error message of user forgets to type -
      }
}
```