```
#include <stdio.h>
#include <stdlib.h>
#include "fig.h"
#include "shape4.h"
#define MAX 1024



void polygon_free(struct Polygon *polygon)
{
    if (polygon && polygon->vertex) {
        free(polygon->vertex);
        polygon->vertex = NULL;
        polygon->npoints = 0;
    }
}

int polygon_sprintf(char *stream, struct Polygon polygon)
{
      int i;
      char *buffer;
      buffer = stream;
      sprintf(buffer, "polygon null %d", polygon.npoints); // \ is for
continuation
      buffer += strlen(buffer);
      for (i=0; i<polygon.npoints; i++) {
                  sprintf(buffer, " %.2f %.2f", polygon.vertex[i].x,
polygon.vertex[i].y);
                  buffer += strlen(buffer);
      }
      return 1;
}

/*
      scan a struct Polygon from string stream
*/
int polygon_sscanf(char *stream, struct Polygon *p)
{
    fprintf(stderr, "# %s\n", stream);
      p->npoints = 7;
      p->vertex = (struct Point *) malloc(sizeof(struct Point) *
p->npoints);   // keep this line

    p->vertex[0].x=5.50; p->vertex[0].y = 4.00;
    p->vertex[1].x=4.94; p->vertex[1].y = 5.17;
    p->vertex[2].x=3.67; p->vertex[2].y = 5.46;
    p->vertex[3].x=2.65; p->vertex[3].y = 4.65;
    p->vertex[4].x=2.65; p->vertex[4].y = 3.35;
    p->vertex[5].x=3.67; p->vertex[5].y = 2.54;
    p->vertex[6].x=4.94; p->vertex[6].y = 2.83;
      return 1;
}

char* fgetshape()
```

```
      {
                char shape2[MAX];
                char temp[MAX];
                int n, j=0, del_newline=1;

                while(del_newline){
                          n=0;
                          del_newline=0; //  templine is used to
determine whether a backslash has been read

                          fgets(shape2, MAX, stdin); //read till end of
line

                          if (j>0) {
                               temp[j]=' ';
                               j++;
                          }                         //adds a space between
newlines when a backslash was read (old line\' 'new line)

                          while (isspace(shape2[n]))
                               n++;       //left trim

                          while(shape2[n] != 0 && shape2[n] != '#'){
//scan until # or end of line

                               if (shape2[n] != 92){
                                         temp[j]=shape2[n];
                                         j++;
                                         n++;
                                         } // if the character is not
a \, \n, or a # copy it to temp array

                               while (isspace(shape2[n]) &&
isspace(shape2[n+1]))
                                         n++; // while two spaces are
side by side, ignore them until only onespace is found.

                               if (shape2[n]==92) {
                                         n++;
                                         del_newline=1;
                                         } // if a bslash is found,
set delete templine to true (1)

                               if (temp[j-1]=='\t')
                                         temp[j-1]=' '; //reads tabs
and converts them to spaces
                          }
                          if (shape2[n] == '#')
                               temp[j]='\0';

                          while (isspace(temp[j-1]))
                          {
                               temp[j-1]='\0';
                               j--;
```

```
                  }   //right trim

                  temp[j]='\0';
         }
         strcpy(shape2,temp);

return shape2;
}
```