CS 430 – Computer Graphics
Fall 2016
Assignment 3

From your previous assignment(s) in this assignment you should be able to:

1. Read in data from a file and organize it into structures for drawing
2. Set up a software frame buffer to render to
3. Clip vertices of lines to a clipping window
4. Implement a line drawing algorithm to render the lines into the software frame buffer.
5. Output the software frame buffer as an XPM file.

In addition, in this assignment you should be able to:

6. Read in data from a file pertaining to creating polygons, and organize that data into structures for drawing polygons.
7. Clip polygons to a clipping window.
8. Draw clipped polygons.

Make sure you give yourself adequate time. The programming components in particular can be quite time consuming.

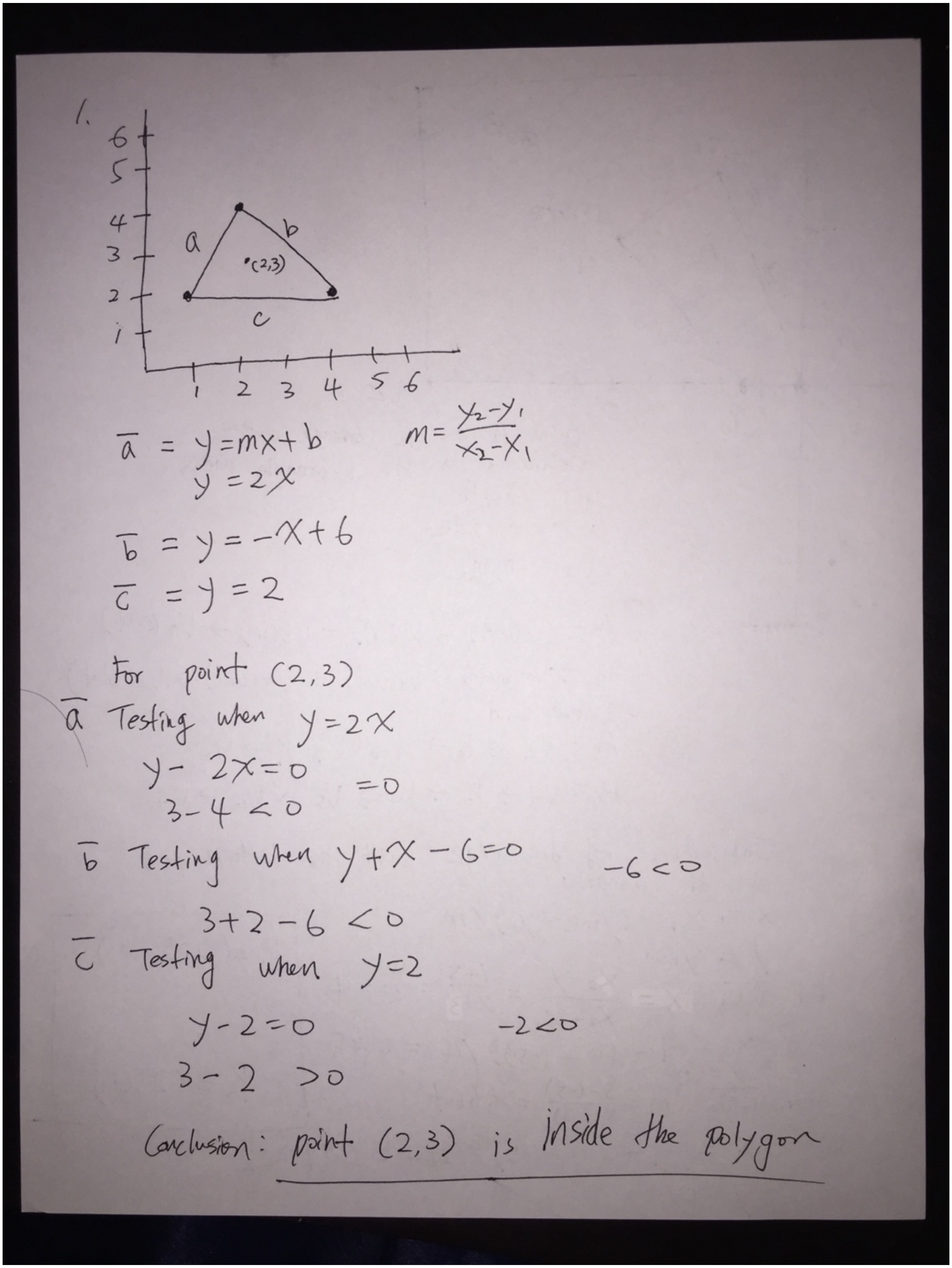
As a reminder you may use the programming language of your choice, though I recommend C/C++ and also make sure that your program can run on the Drexel tux cluster to insure its system independence.

Submission Guidelines

1. Assignments must be submitted via Bd Learn
2. Submit a single compressed file (zip, tar, etc..) containing:
    a. A PDF file with your solutions to the theory question(s).
    b. A README text file (**not** Word or PDF) that explains
        i. Features of your program
        ii. Language and OS used
        iii. Compiler or interpreter used
        iv. Name of file containing main()
        v. How to compile/link your program
    c. Your source files and any necessary makefiles, scripts files, etc… to compile and run your program

Theory Question(s):

1. Given polygon with defined by vertices ((1,2),(4,2),(2,4)) determine if the point (2,3) is within (or on) the polygon using the even-odd algorithm. Show your work which should include the computation of several intersections. (10pts)


6
5
4
3
2
1
1
2
3
4
5
6
a
b
c
(2,3)

1.

$\bar{a} = y = mx + b$ $\quad m = \frac{y_2 - y_1}{x_2 - x_1}$

$y = 2x$

$\bar{b} = y = -x + 6$

$\bar{c} = y = 2$

For point (2,3)

$\bar{a}$ Testing when $y = 2x$

$y - 2x = 0$ $\quad = 0$

$3 - 4 < 0$

$\bar{b}$ Testing when $y + x - 6 = 0$ $\quad -6 < 0$

$3 + 2 - 6 < 0$

$\bar{c}$ Testing when $y = 2$

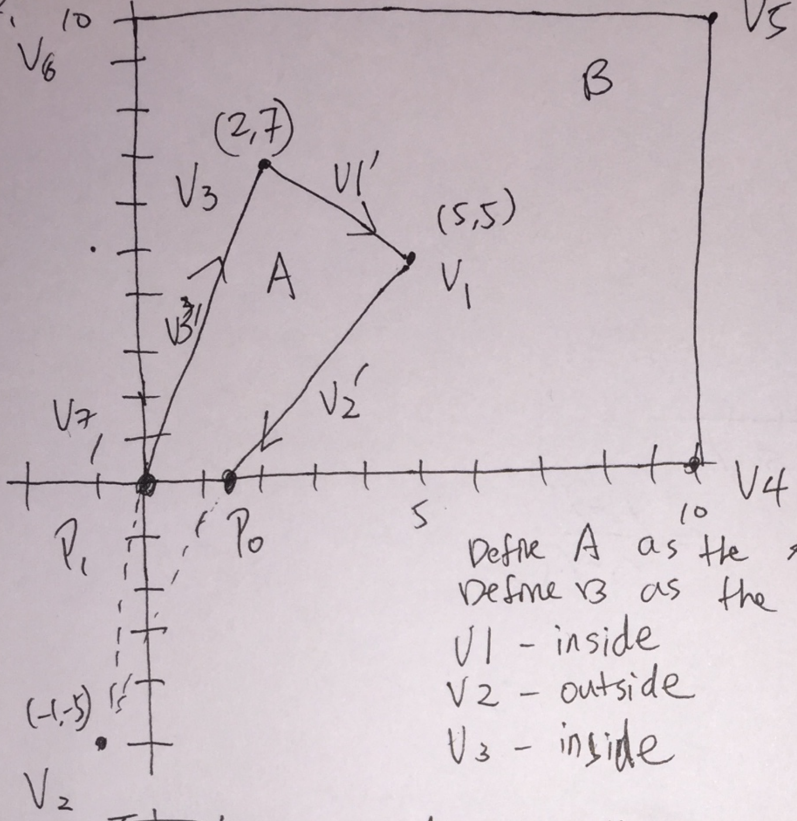
$y - 2 = 0$ $\quad -2 < 0$

$3 - 2 > 0$

Conclusion: point (2,3) is inside the polygon

Given a clipping rectangle defined by vertices ((0,0), (10,0), (10,10), (0,10)) show how the Sutherland-Hodgman Polygon Clipping algorithm can be used to clip the triangle defined by vertices ((-1,-5), (5,5),(2,7)). You **DO NOT** need to show your work for determining inside-out status but you **DO** need to show your computations for finding the intersection point between the segments and infinite clip edges. Your final answer should include the new vertices for the edges of the clipped polygon. (20 points)

2.



Defne A as the triangle area

Define B as the Rectangle area

V1 - inside

V2 - outside

V3 - inside

~~Intersection~~

$A = V_2(out) \rightarrow V1(in) \rightarrow V3(in) \rightarrow V_2(out)$

$B = V_7(out) \rightarrow V_4(out) \rightarrow V_5(out) \rightarrow V_6(out)$

Linked List

$A = V_2 \rightarrow P_0 \rightarrow V_1 \rightarrow V_3 \rightarrow P_1 \rightarrow V_2$

$B = V7 \rightarrow P_0 \rightarrow V4 \rightarrow V_5 \rightarrow V_{10} \rightarrow V7$

Intersection of the edge with the bottom side of the window

$x = x_1 + (y_{min} - y_1)/m$

$y = y_{min}$

For $\overline{V_2V_3}$ $m = \frac{y_2 - y_1}{x_2 - x_1} = \frac{7+5}{2+1} = \frac{12}{3} = 4$

$x = -1 + (0-(-5))/4 = 0.25$

$y = 0$

For $\overline{V_1V_2}$ $m = \frac{5-(-5)}{5-(-1)} = 1.667$

$X = -1 + (0-(-5)/1.667 = 1.99 \approx 2$

$y = 0$

$P_1 = (0.25, 0)$

$P_0 = (2, 0)$

Introduction

In this assignment you now should support Postscript commands that include: `x y moveto`

```
x y lineto

stroke
```

- The *moveto* command simply moves the “pen” to a particular location without drawing anything
- The *lineto* command draws a line from the current position of the pen to the point indicated

- The *stroke* command can be ignored (but it’s actually used by the Postscript interpreters to say to draw)

For example the following commands will draw two boxes:

```
%%%BEGIN
100 100 moveto
200 100 lineto
200 200 lineto
100 200 lineto
100 100
lineto stroke
300 300 moveto
400 300 lineto
400 400 lineto
300 400 lineto
300 300
lineto stroke
%%%END
```

You can assume that the *moveto stroke* pair defines a single closed polygon with no holes and with vertices given in counter-clockwise order. Note that the last vertex should be the same as the first vertex.

Assignment Details

Write a program that accepts the following command arguments. Defaults are in parenthesis.

a. [-f] The next argument is the input “Postscript” file (hw3_1.ps)

For now we will hard-code the size of your frame buffer to be 500x500 (that is 500 pixels wide by 500 pixels high).

Your program should print output images in the XPM file format to stdout (`cout`) such that it can be piped to a file. All pixels should be initialized to white. Draw your objects in black.

Your general program flow should be:

1. Read in polygon specifying data (PS)
2. Clip the polygons to the frame buffer clipping window using the **Sutherland-Hodgman** algorithm.
3. Draw the polygons by rastering the line segments that defined the clipped polygons into software frame buffer using the algorithm of your choice.
4. Output your image (the frame buffer and header information necessary to make an XPM file) via `cout`

Grading Scheme:

In order to get any credit at all you must be able to generate at least read in a PS file and write out a valid XPM file

1. Theory question(s) (30pts)
2. Successfully handles basic single line test cases (6 @ 7pts = 42pts)
3. Successfully handles a multi-line test cases (18pts)
4. Program easy to compile and run based on README (10pts)

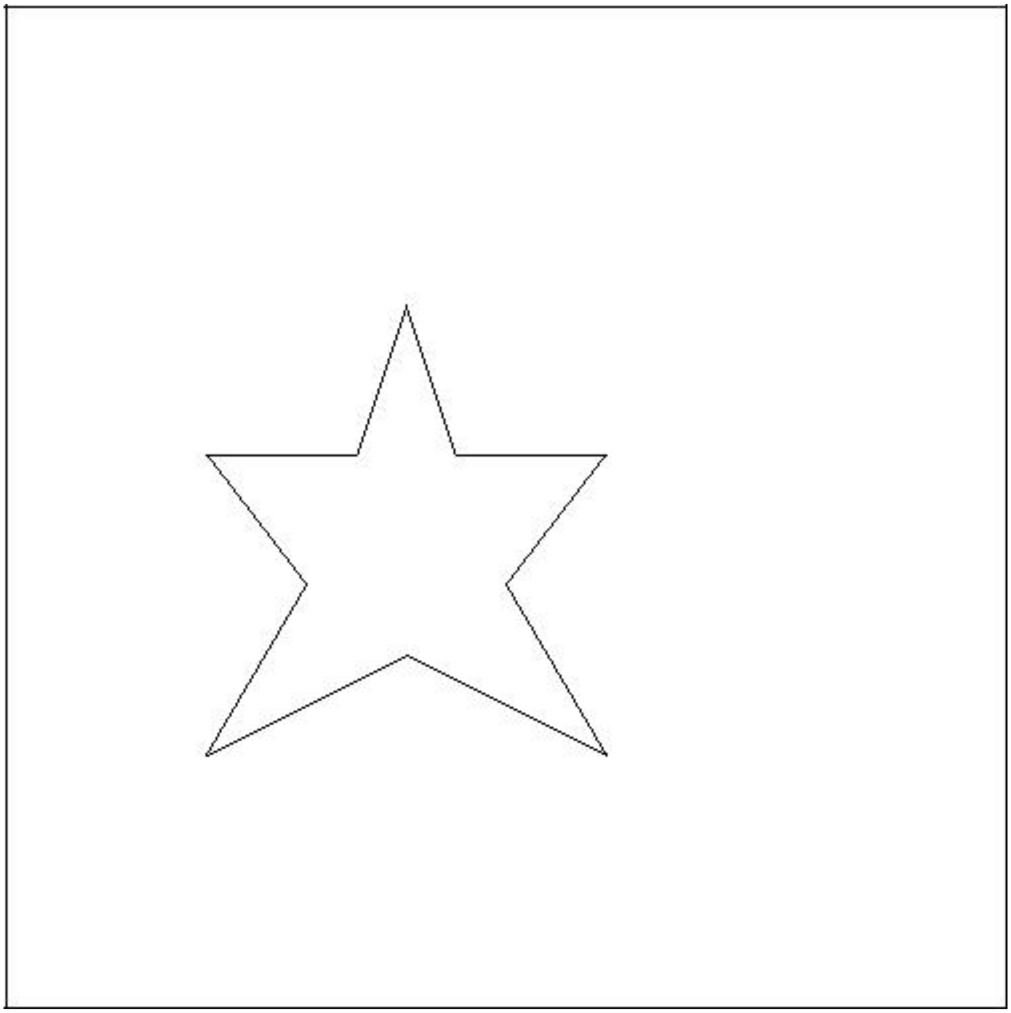
Common deductions:

1. Nothing drawn (-100pts)
2. Missing files (including readme) (-5pts each)
3. Cannot compile/run out-of-the-box on TUX (-10pts)

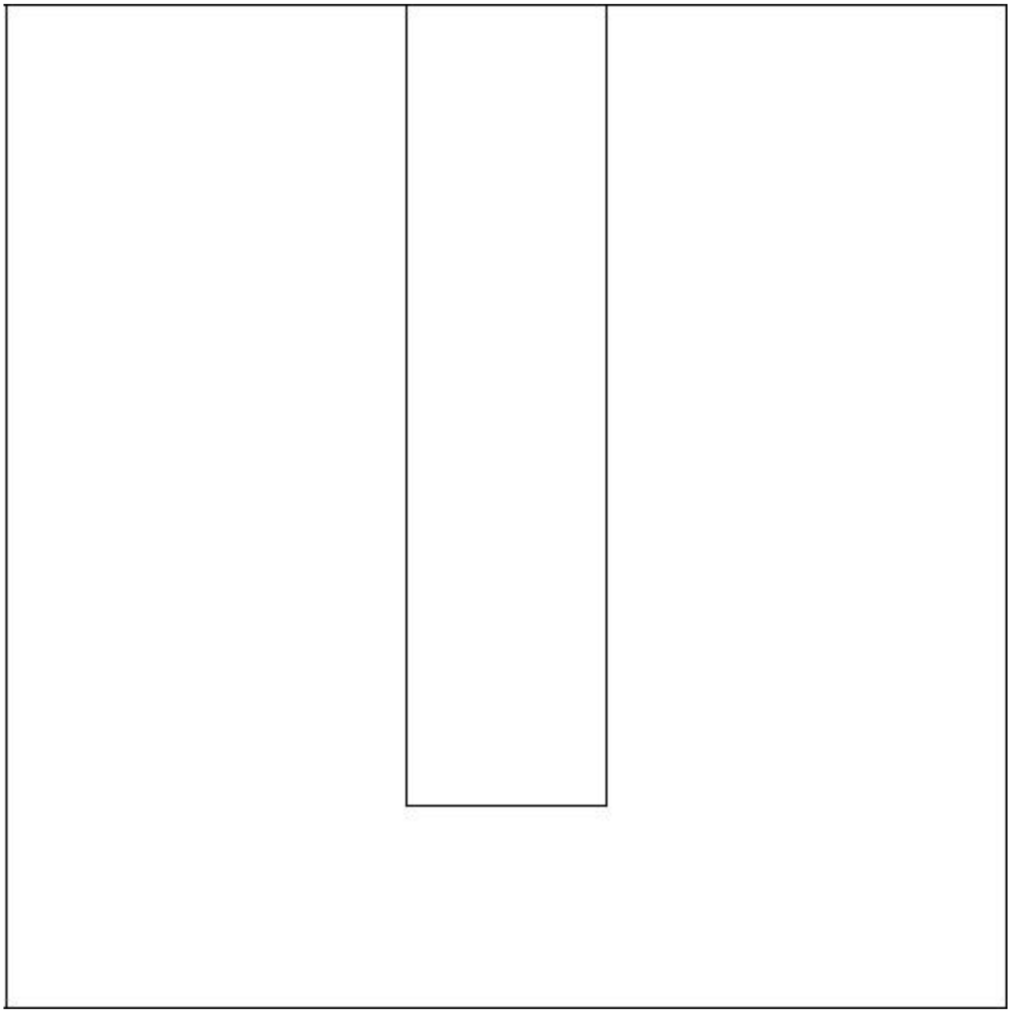
Provided Tests

Here are the output of the tests provided for you:

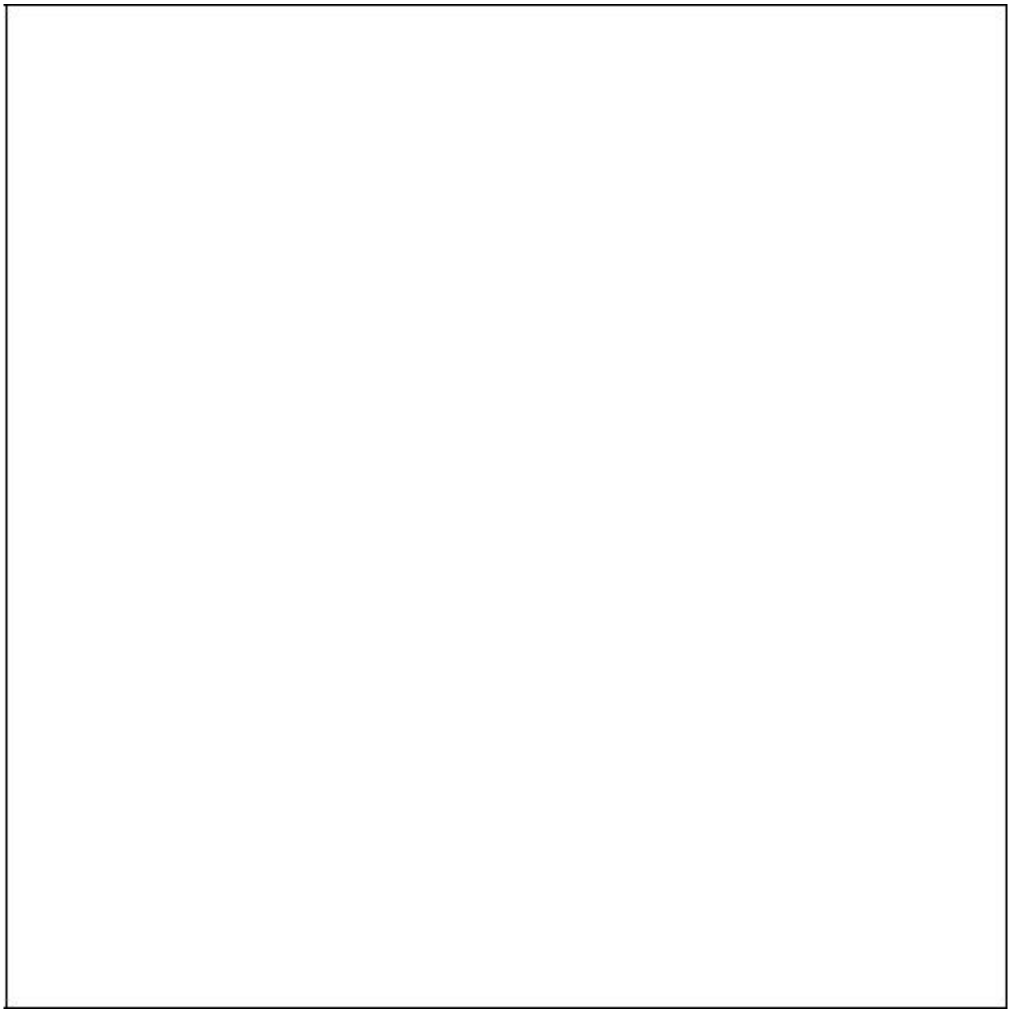
```
./A3 -f hw3_1.ps > out1.xpm
```



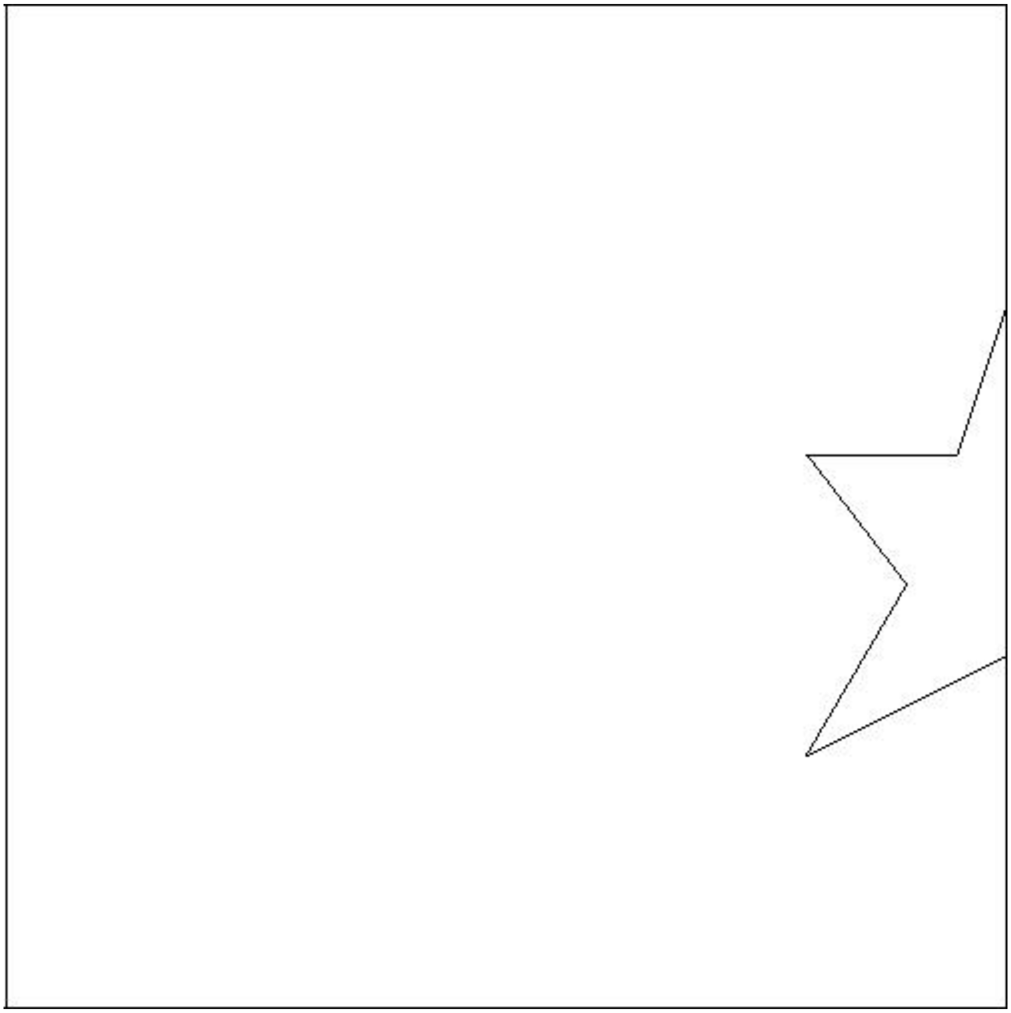
```
./A3 -f hw3_2.ps > out2.xpm
```



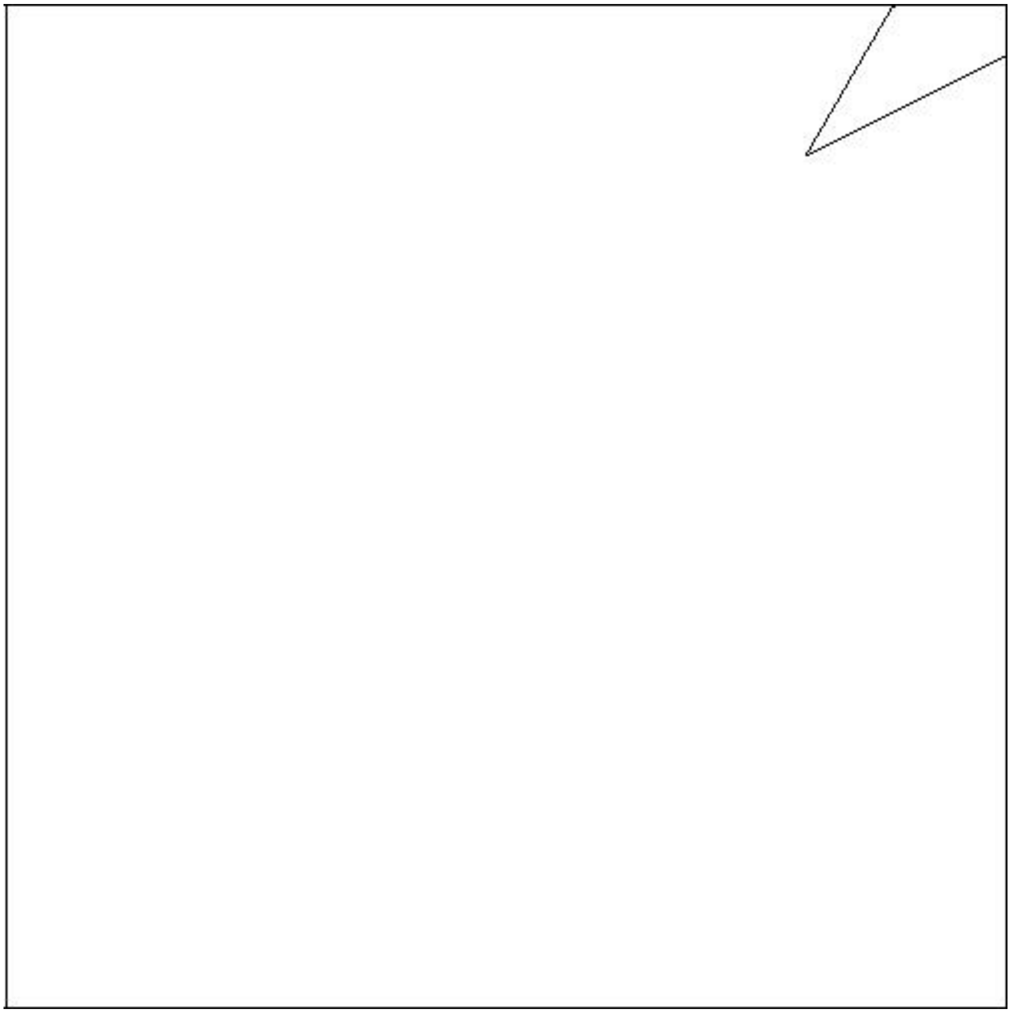
```
./A3 -f hw3_3.ps > out3.xpm
```



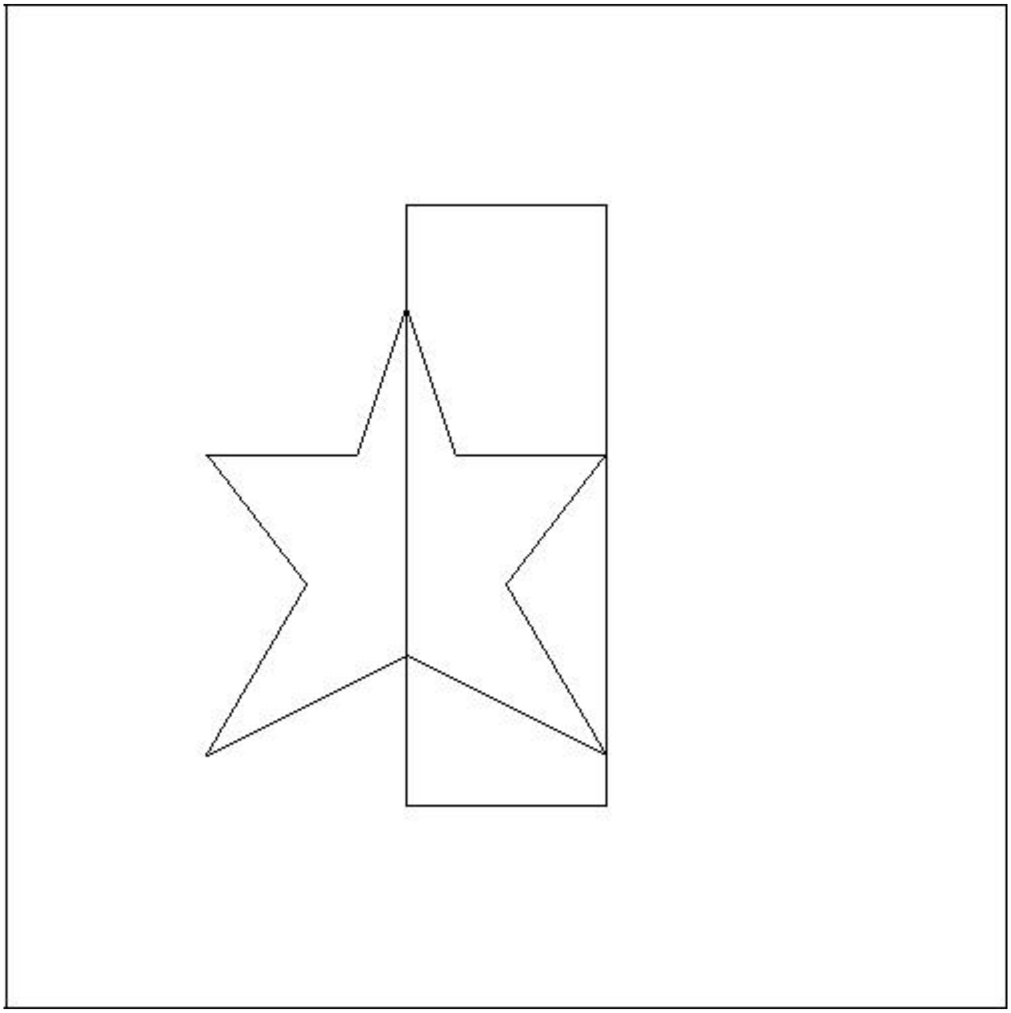
```
./A3 -f hw3_4.ps > out4.xpm
```



```
./A3 -f hw3_5.ps > out5.xpm
```



```
./A3 -f hw3_6.ps > out6.xpm
```



```
./A3 -f hw3_7.ps > out7.xpm
```