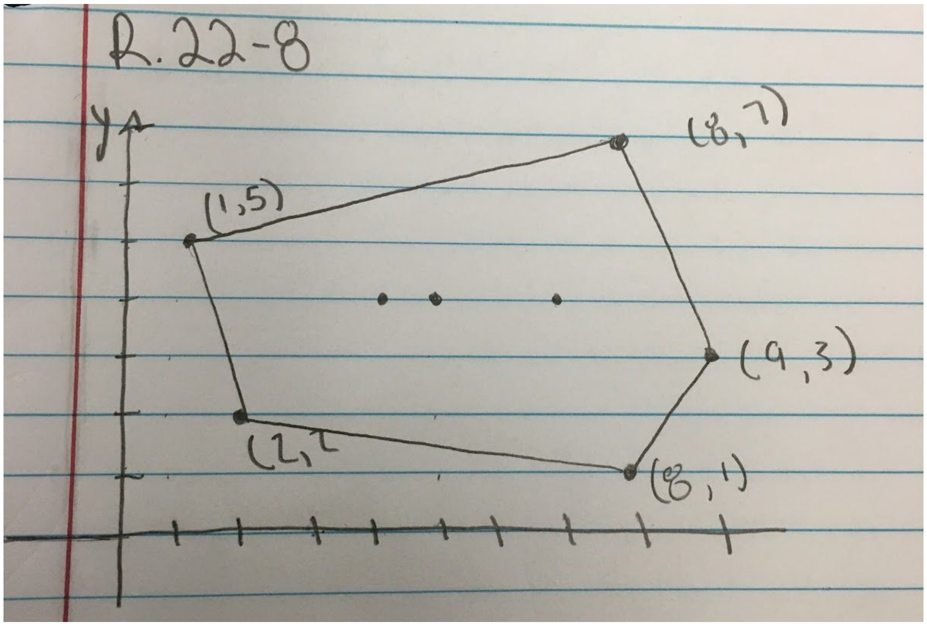
Corey Kipp (Student ID: 57723335)

kippc@uci.edu

CS 161

Homework 7

R-22.8



C-22.1

```
def inTriangle(p,q,r,s):
    //LeftTurn is a function that returns true if the three points form a left turn
    if LeftTurn(q,r,p):
        if LeftTurn(r,s,p):
            if LeftTurn(s,q,p):
                return true
    return false
```

The above pseudo code will check if all the cases are indeed left hand turns, if any of them are not then point P is not in the triangle.

Additionally you could check the following:

1. Δ q, r, p and Δ q, r, s have the same orientation
2. Δ r, s, p and Δ r, s, q have the same orientation

3. Δ s, q, p and Δ s, q, r have the same orientation

If these are all true then the point P is inside the triangle.

C-22.5

In order to determine if a polygon is convex or not you must check that no interior angle is greater than 180 degrees and not self-intersecting. So you would first go through and determine whether every interior angle in the polygon is less than 180 degrees. Then in order to determine whether it is self-intersecting or not, you could first select the y-minimum value in the convex hull, and iterate through the other points in the convex hull. All y-values should increase up until a certain point (the y-maximum) and then start decreasing (toward the y-minimum) if the polygon is convex, if it is not then the y-values will be increasing and decreasing more often. If every interior angle is less than 180 degrees and the polygon is not self-intersecting are both true it is convex, but if either one of these is false then it is not.

C-22.6

First divide the polygon into multiple triangles and calculate the area of each of those triangles using the determinant and then dividing by 2. The sum of all the triangles areas' is the area for the polygon.

In the following pseudo code, L[i] becomes the common vertex for all other triangles which is why "i" is never incremented, L[j] and L[k] are incremented on every iteration so that eventually you'll have calculated the area for every triangle with the common vertex L[i].

```
L = [ $(x_0, y_0)$ , $(x_1, y_1)$ , ... $(x_{n-1}, y_{n-1})$ ] // List of points in convex hull
def areaOfPolygon(list L):
        i = 0
        j= 1
        k = 2
        sum_of_areas = 0
        n = len(L)
```

```
if n <= 3: //meaning there are 3 or less vertices
    // ↓ calculates the determinant and then divide by 2 for area
    sum_of_areas =  (L[i][0] * L[j][1] - L[i][1] * L[j][0]
                        + L[j][0] * L[k][1] - L[j][1] * L[k][0]
                        + L[k][0] * L[i][1] - L[k][1] * L[i][0]) / 2
    return sum_of_areas
for( int w = 0; w < n-2; w++) // number of triangles formed in a polygon
    sum_of_areas += (L[i][0] * L[j][1] - L[i][1] * L[j][0]
                        + L[j][0] * L[k][1] - L[j][1] * L[k][0]
                        + L[k][0] * L[i][1] - L[k][1] * L[i][0]) / 2
    j++
    k++
return sum_of_areas
```