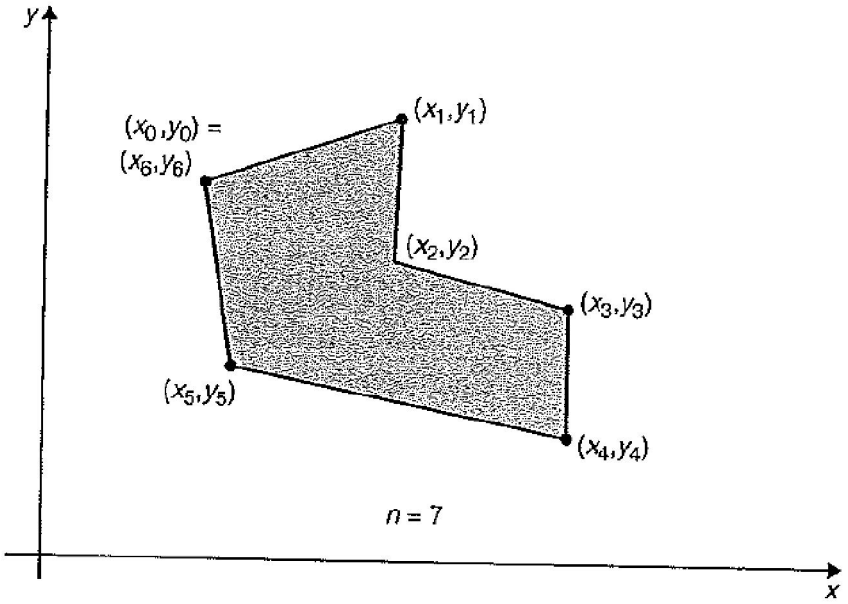
CSE102
HW04

Last Submission Date: Apr. 18, 2012 – 23:50

Part-1. If n points are connected to form a closed polygon as shown below, the area A of the polygon can be computed as

$$A = \frac{1}{2}\left|\sum_{i=0}^{n-2}(x_{i+1}+x_i)(y_{i+1}-y_i)\right|$$



Notice that although the illustrated polygon has only six distinct corners, n for this polygon is 7 because the algorrithm expects that the last point, (x_6, y_6) will be a repeat of the initial point, (x_0, y_0).

Represent the (x, y) coordinates of the connected points as one two-dimensional array of at most 20 type double values. For one of your tests, use the following dara set, which defines a polygon whose area is 25.5 square units.

x	y
4	0
4	7.5
7	7.5
7	3
9	0
7	0
4	0

Implement the following functions:

getCorners - Takes as parameters an input file, a 2D array of point coordinates (x, y), and the array's maximum size. Fills the array with data from the file (ignoring any data that woluld overflow the array) and returns as the function value the number of (x, y) coordinates stored in the array.

outputCorners - Takes as parameters an output file, a 2D array of point coordinates, and the number of points. It outputs to the file the contents of the array.

polygonArea - Takes as parameters a 2D array of point coordinates of a closed polygon and its actual size and returs as the function value the area of the closed polygon.

Part-2.

A C program can represent a real polynomial $p(x)$ of degree n as an array of the real coefficients $a_0, a_1, \ldots, a_n$ $(a_n \neq 0)$.

$$p(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n$$

Write a program that inputs a polynomial of maximum degree 8 and then evaluates the polynomial at various values of x. Include a function `get_poly` that fills the array of coefficients and sets the degree of the polynomial, and a function `eval_poly` that evaluates a polynomial at a given value of x. Use these function prototypes:

Your program should at least contain the functions below, you can write additional functions if you want.

```
/* Read coefficients from a given input file */
void getPoly(FILE * inputFile, double coeff[], int * degreep);

/* Evaluate a polynomial at a given value of x */
double evaluate(const double coeff[], int degree, double x);

/* Set all coeffients to zero */
double setToZero(double coeff[], int degree);

/* result = poly1 + poly2 */
double addPoly(const double poly1[], const double poly2[], double result[], int degreePoly1,
int degreePoly2, int degreeResult);

/* result = poly1 * poly2 */
double multiplyPoly(const double poly1[], const double poly2[], double result[], int
degreePoly1, int degreePoly2, int degreeResult);

/* Sample Input File -1 ----------   4x^2 + 3x + 4*/
4 3 4

/* Sample Input File -2 ----------   x^4 + 3x^3 + 4x^2 + 7*/
1 3 4 0 7
```