Instructions: In-class exercises are meant to introduce you to a new topic and provide some practice with the new topic. **Work in a team of up to 4 people to complete this exercise.** You can work simultaneously on the problems, or work separate and then check your answers with each other. **Turn in <u>one</u> copy of the exercise per group**.

Names:

Algorithms: Introduction

Aspects of an Algorithm

Algorithms are step-by-step sequences of instructions meant to help solve a problem. An algorithm will generally have 1 or more input, and 1 or more output.

Some characteristics of algorithms are: [a]

Input	The algorithm receives input.
Output	The algorithm produces output.
Precision	The steps are precisely stated.
Determinism	The intermediate results of each step of execution are unique and determined only by the inputs and the results of the preceding steps.
Finiteness	The algorithm *terminates*; that is, it stops after finitely many instructions have been executed.
Correctness	The output produced by the algorithm is correct; that is, the algorithm correctly solves the problem.
Generality	The algorithm applies to a set of inputs.

[a] From Discrete Mathematics, Johnsonbaugh, p181-182

Question 1
Given the following algorithm:

```
1 def GetMax( a, b ):
2     if ( a > b ):    return a
3     else:            return b
```

For each function call, step through each line, writing down the variable values at each step.

a. `GetMax( 1, 2 )`

Line #	Code	a	b	result
1	GetMax(a, b)	1	2	Enter function
2	if (a > b)	1	2	False, continue
3	else	1	2	Return b (2)

Result: 2

b. `GetMax( 2, 2 )`

Line #	Code	a	b	result

c. `GetMax( 3, 2 )`

Line #	Code	a	b	result

Question 2

Given the following algorithm:

```
1 def Sum( myList ):
2     sumVal = 0
3     for num in myList:
4         sumVal += num
5     return sumVal
```

For each function call, step through each line, writing down the variable values at each step.

a. `Sum( [1, 2] )`

Line #	Code	`myList`	`num`	`sumVal`	result
1	Sum(myList)	[1, 2]	-	-	Enter function
2	sumVal = 0	[1, 2]	-	0	Set sumVal to 0
3	for num in myList:	[1, 2]	1	0	Begin loop
4	sumVal += num	[1, 2]	1	1	Add to sumVal
3	for num in myList:	[1, 2]	2	1	num++
4	sumVal += num	[1, 2]	2	3	Add to sumVal
5	return sumVal	[1, 2]	-	3	Return 3

Result: 3

```
1 def Sum( myList ):
2     sumVal = 0
3     for num in myList:
4         sumVal += num
5     return sumVal
```

b. `Sum( [2, 4, 6, 8] )`

Line	Code	`myList`	`num`	`sumVal`	result

Question 3
For this question, one student will write the algorithm, and another student will step through it to make sure it works.

a. Write an algorithm (in the form of a function called AbsVal) that takes one input n, and returns $|n|$.

b. Using the function call `AbsVal( 2 )`, step through your classmate's algorithm and check if the result is correct.

c. Using the function call `AbsVal( -2 )`, step through your classmate's algorithm and check if the result is correct.

Question 4
For this question, one student will write the algorithm, and another student will step through it to make sure it works.

a. Write an algorithm (in the form of a function called GetMax) that finds the maximum of 3 input variables a, b, and c.

b. Using the function call `GetMax( 3, 5, 2 )`, step through your classmate's algorithm and check if the result is correct.