Assignment 3 FAQ

Can we assume that a LISP tree has only two children or a single node tree?

It depends on how you built the tree in ANTLR. The main idea behind using ANTLR is to create a valid expression tree that can be used to evaluate an arithmetic expression. We recommend that you parse the lisp tree without any assumption on its structure; all you need to assume is that it has a valid representation of a general tree.
For example, it would be useful to be able to parse trees of the form:

- (A) or (A(B)(C))

But when you build your grammar file in ANTLR you can make sure that these trees are not valid as an output. This can be useful when you need to extend your code to support new features in the project. For example we may want to support the function max() that returns the maximum value between its numbers, thus the expression max(5,3,2) may be valid later.
For the above example, one possible representation of a lisp tree would be as follow:

- (max(5)(3)(2)) – Which clearly has more than two children

Does SPCalculator support negative numbers?

Yes, for example the following expressions are valid:

- 1 - - 5
- + 1 - 5 - - - - 7
- 09 + - 19
- 5/-1

Please note however, you control the structure of the tree in ANTLR. Think how you can represent a tree for the above expressions in ANTLR.

Does SPCalculator support floats?

No, in assignment 3 you are not needed to support fractions. However you may need to support real numbers in the project.
Clarification: The result itself can be a real number however operands will only be integers.

Why do we need to support more than 2 children in the tree data structure?

This could be useful when working on the project. We will add new features to the calculator in the project, so you may need to support functions that support more than two arguments.

Can we use different algorithm to evaluate the expression?

It is not recommended, your algorithm may not work later when you need to extend it to support new features in the project

When an unexpected error occurs, do I need to print "Exiting…" before terminating the program?

No, you need only to print "Unexpected error occurred!" followed by a new line and exit.

Can we assume that there are no spaced between numbers? And there are no spaces between "<" and ">" in termination command?

Yes, you may assume inputs like the following are not possible:

1- 1 + 2 3
2- < >
3- 3 + 4 <>

Can we assume that the maximum line length in the C program is given?

Yes, you may assume that MAX_LINE_LENGTH is 1024

Can we assume implicit operations are invalid?
Yes, you may assume implicit operations are impossible inputs. For example expressions of this kind will not be an input:

1- 3(5+2)
2- (5$10)2

However these are valid:

1- 3*(5+2)
2- (5$10)*2

Do we need to check if a dollar operation has integers as it's operands?

No, you may assume operands for the dollar sign are integer values. The following inputs are impossible:

1- (5/3)$7
2- (7/3)$(10/2)

However, the following inputs are possible:

1- (6/2)$(10/2)
2- (10/2)$(6/2) – **NOTE** This is possible input but invalid.

Do we need to support arithmetic expression with parenthesis ?
Yes, These inputs are valid: "(1+2)*6", "(1+(4/2)$5)"