1. In your opinion, out of the given test cases, list the easiest test case(s). Why? [1 pt]  
test1.micro – It only tests that a symbol table is generated without having any code blocks  
test5.micro – Only one symbol table  
test14.micro – it tests comments, so there isn’t a lot being added to the symbol table  
test16.micro – it has two empty symbol tables

2. In your opinion, out of the given test cases, list the hardest test cases(s). Why? [1 pt]  
test9.micro – error handling and symbol table dropping  
test20.micro – global string declaration with 7 symbol tables, one with a custom name  
test21.micro – error handling in a separate method block

3. List all enterX and exitX functions that you think need to be implemented. (hint: check the auto-generated LittleBaseListener class) [4 pts]  
enter\_Program, exit\_Program, enterFunc\_decl, exitFunc\_decl, enterPgm\_body, exitPgm\_body, enterIf\_stmt, exitIf\_stmt, enterElse\_part, exitElse\_part, enterFor\_stmt, exitFor\_stmt, enterWhile\_stmt, exitWhile\_stmt, enterString\_decl, exitString\_decl, enterVar\_decl, exitVar\_decl

4. In your opinion, which Java data structure(s) should be used to implement the symbol table(s)? Why? [2 pts]

Hash tables provide a consistently high-performance solution due to their ability to insert and retrieve symbols in constant time on average. This means symbol tables can grow or shrink with each scope and performance will not be impacted.

5. You also have the option of using a Visitor (instead of the Listener). What is the main difference between a Listener and a Visitor? (hint: read the above excerpt) [2 pts]  
The main difference between the two is how the methods are called. In a visitor, methods have to give their children specific calls or they won’t be visited; however in a Listener, the methods are called by an Antlr-provided walker object. Visitors are useful in situations where you may need to return data from nodes, or if you need to visit several nodes before performing an operation to change the parse tree.