CS 1622 – Homework 3

Due: Tuesday, February 20, 2018, at the start of class

Please submit a typewritten document.

1.) For the grammar from homework 1, rewrite it to encode precedence and left factor. Not (!) should be highest precedence, ^ middle precedence, and v should be lowest.

E → E ^ E

E → E v E

E → ! E

E → true

E → false

2.) Construct an LL(1) parse table for your grammar from 1. Show the First and Follow sets you generated.

3.) Show the LL(1) parsing action list for the input:

```
!false v false ^ false
```

4.) Based upon your grammar for 1), write a recursive descent parser. On the next page, I have provided a skeleton and minimal test suite for you to follow. You do not need to support whitespace and should use T for true and F for false. Fill in the skeleton with the implementation of a function per nonterminal in your grammar. Print out your code (it shouldn't be more than 2 pages) and submit it.

```
class RecursiveDescent {
	public static void main(String[] args) {
		String[] tests= {
			"T",			//Accept
			"!F",			//Accept
			"T^F",			//Accept
			"T^T^",			//Reject
			"!TvF",			//Accept
			"!vvF",			//Reject
			"!F",			//Accept
			"!T^!T"			//Accept
		};

		//Note: We are not writing an interpreter, only accepting valid strings
		//and rejecting invalid ones. That is, I don't care whether the answer
		//is true or false for the valid ones.

		for (String test: tests) {
			if(S(new StringBuffer(test))) {
				System.out.println(test + ": Accept");
			}
			else {
				System.out.println(test + ": Reject");
			}
		}
	}

	/**
		Assuming your start symbol is S
		@param str – the StringBuilder of your input (mutable to "eat" tokens)

		@return whether the parse is valid at this point or not
	*/
	private static boolean S(StringBuilder str) {

		//Use str.charAt(0) to peek at the first character
		//Use str.deleteCharAt(0) to eat the first character

		//based upon the character (or not), recursively call
		//other production functions

		//Return false if you encounter something bad or return the
		//return values of the recursive calls you make
	}

	//Add your additional functions here.

}
```