 Determine whether the following grammar is an LL grammar by performing the pairwise disjointness test, showing the FIRST sets of each RHS of each nonterminal:

A -> aAb | bBB

Fails pairwise Disjoint test because A -> aAb

B -> aB | bA | Cc

Fails pairwise Disjoint test because B -> aB

C -> aaB | c | caB

Passes Disjointness test

In the above productions upper case letters are nonterminals and lower case letters are terminal symbols.

 Given the following grammar:

S -> aB | b | cC

B -> aB | bS

C -> aaS | b | caB

1) Perform the pairwise disjointness test for each rule to show that this grammar allows top-down parsing. Submit this part as a text file h2-1.txt.

2) Write a recursive descent parser for this grammar in your favourite language: C, C++ or Java. Submit the source code of your parser (including the very simple lexer for this grammar). The main method will ask the user for a string and will attempt to parse it, displaying the result of this attempt and all the steps followed (just like the example from the textbook - pages 182 - 185). Report all syntax errors.

3) Verify that the following string is recognized: caaabccabb. Submit the output of your program as a text file h2-3.txt.