1. [10 pts.] Trace though the store states of the following PDA, on the input ‘a b e c d c’.

![PDA Diagram]

2. [10 pts.] Write a CFG equivalent to the above PDA. Try to make your answer as concise as possible.

3. [10 pts.] Write a CFG equivalent to the following PDA (noting that CFGs are not defined to allow regular expressions on the right hand side). Try to make your answer as concise as possible.

![PDA Diagram]

4. [10 pts.] PROGRAMMING: Write a recursive Python program to read any tree (using the Tree class from the lecture notes) and print the total number of non-lexical constituents in that tree. You may not use string functions to solve this problem. Try to make your program as concise as possible. For example, on the following input:

\[(V \ (N \ (N \ a \ big \ cat) \ with \ (N \ a \ hat) \ on \ it) \ (V-aN \ sat \ down \ (R-aN \ on \ (N \ the \ bed))))\]

Your program should print:

7

5. [10 pts.] PROGRAMMING: Write a recursive Python program to read any number \(n\), followed by any tree specified as above, and print the \(n\)th word from the end of the tree. You may not use string functions to solve this problem. Try to make your program as concise as possible. For example, on the following input:

\[(V \ (N \ (N \ a \ big \ cat) \ with \ (N \ a \ hat) \ on \ it) \ (V-aN \ sat \ down \ (R-aN \ on \ (N \ the \ bed))))\]

your program should print:

down