Question 6 Part A
Show that if tests can be of any size, decision lists can represent any Boolean function.

First, we know that decision lists are simply a conjunction of literals. This means that we need to be able to express each of our Boolean equations in the following form:

$(P \wedge Q \wedge \neg S) \vee (\neg P \wedge Q \wedge R) \vee (\neg P \wedge \neg R \wedge \neg S) \vee (P \wedge \neg S \wedge \neg T)$

Essentially, as shown above, the point is that since we can write any Boolean function as a disjunction of clauses, where each clause represents a conjunction of literals (as explained in class notes), then we can easily transform the above Boolean equation into a decision list that looks like the following:

$(P \wedge Q \wedge \neg S)$→ F →$(\neg P \wedge Q \wedge R)$→F→$(\neg P \wedge \neg R \wedge \neg S)$→F→$(P \wedge \neg S \wedge \neg T)$ → FALSE
     | | | |
    True   True   True   True

So as illustrated above, all we need to do is take each clause of the equation and make it into a test of the decision list. Then depending on how each test is evaluated, we can make a decision at that point of how the equation should be evaluated or to continue with the next test.

Part B
Show that if the tests can contain at most k literals each, then decision lists can represent any function that can be represented by a decision tree of depth k.

The main point is that given a tree which has a depth of *k*, each node along a path in the tree from root to leaf represents a different literal in a test of our decision list. Since the depth is k, no path can contain more than k nodes. So, each path is then converted into a test of the decision list. So, what this means is that each path from the root to leaf represents a conjunction of literals which becomes a test in the decision list. Therefore, since depth is k, no path/test can have more than k nodes/literals. The result of each path is shown at the bottom of the path and this is then the result of each test in the decision list.