10. Write a Prolog predicate, countLists(Alist, Ne, Nl), using accumulators, that is true when Nl is the number of items that are listed at the top level of Alist and Ne is the number of empty lists. Suggestion: First try to count the lists, or empty lists, then modify by adding the other counter.

```
% Entry point

countLists(Alist, Ne, Nl) :-

countLists_acc(Alist, 0, 0, Ne, Nl).

% Base case: empty input list

countLists_acc([], NeAcc, NlAcc, NeAcc, NlAcc).

% Recursive case: Head is an empty list

countLists_acc([[] | T], NeAcc, NlAcc, Ne, Nl) :-

    NeAcc1 is NeAcc + 1,

    NlAcc1 is NlAcc + 1,

    countLists_acc(T, NeAcc1, NlAcc1, Ne, Nl).

% Recursive case: Head is a non-empty list

countLists_acc([[ _ | _ ] | T], NeAcc, NlAcc, Ne, Nl) :-

    NlAcc1 is NlAcc + 1,

    countLists_acc(T, NeAcc, NlAcc1, Ne, Nl).

% Recursive case: Head is not a list

countLists_acc([_ | T], NeAcc, NlAcc, Ne, Nl) :-

    countLists_acc(T, NeAcc, NlAcc, Ne, Nl).
```