More List Problems

1. Write a function `list_em(x)` that takes as an argument a string `x` of uppercase letters and returns a sorted list containing the letters in `x`, each letter of only appearing once in the returned list (no matter how many times they appeared in `x`).

Here is an example:

   `a="ADDEAADEE"

   list_em(a) will return ['A', 'D', 'E']

2. Write a function `valid(x)` that takes as an argument a string `x` of uppercase characters and returns True if all the distinct characters in `x` appear the same number of times. If they don’t, `valid(x)` return False.

Here is an example:

   `a="ADEAADEE"

   valid(a) will return True because A,D and E each appear three times. but
   a="ADEAADEE"

   valid(a) will return False because A and E each appear three times. but D only twice.

3. Write a function `position(x,a)` that takes as an argument a string `x` of uppercase characters and a single character `a`, and returns a list giving the positions that `a` appears in `x`. If `a` is not in `x`, return an empty list.

4. Do the problems on page 104 in the class notes.

5. Do the problems on page 110 in the class notes. (Hint: read page 111 first)

6. Do the problems on page 112-114 in the class notes.

7. Write a function `dot(x,y)` where `x` and `y` are lists of numbers and both `x` and `y` are the same length. The function returns the “dot product” of the two lists.

8. Write a function `get_max_and_positions(x)` that takes as an argument a two dimensional list `x` (i.e. a “table”, not necessarily square) of integers and returns three values: the largest value in the table, and the row and column indices where the maximum value was found. If the maximum appears in multiple entries in the table, return the value, and the indices of the first place it was encountered.