Loops and Recursion exercises

Do not enter dashes? Really? Couldn't they accept the number with or without dashes? We can. Let's write (and test!) a function that removes punctuation.

Nested lists

Adding the numbers in [2, 3, 8, 19, -2] is easy

How about adding the numbers in

[ [2, 3, [4, 5], 6], [ ], [[7, 8]]

Ooh, I think we're going to need some recursion for this ...

Lists are references to sequences of cells ...
So we can represent a matrix (like our cavern in Project 4) as a list of lists ...

\[
\text{lst} = \begin{bmatrix}
[1, 2, 3], \\
[4, 5, 6], \\
[7, 8, 9]
\end{bmatrix}
\]

And since Python is dynamically typed, we can mix and match types of data in a list, and ask about the type of each element ...

```python
lst = [ [ 1, 2, 3 ], [4, 5, 6], [7, 8, 9] ]
```

```
type(lst[0][0]) == int
```  

```
type(lst) == list
```  

```
type(lst[0]) == list
```  

```
type(lst[0][2][1]) == int
```  

We’ll need a combination of looping and recursion to traverse the list.

How about punctuating an integer or numeric string, e.g.,

punctuate(999999999, “###-###-###”) = “999-999-999”

Strategy?  
Clarification of requirements?