CSCI 111, Bonus Lab 2
Add list processing to the RPN Calculator

Individual work: All work must be your own. Do not share code with anyone other than the instructor and teaching assistants. This includes looking over shoulders at screens with the code open. You may discuss ideas, algorithms, approaches, *etc.* with other students but NEVER actual code.

RPN with lists: Add lists to the RPN calculator. You will need a working RPN calculator for this!

List creator keywords: You will need to add two keywords to the interpreter:

- `[`
- `]`

These will be tokenized like regular keywords.

Processing list creator keywords: When processing items from the tokenized list, the `[` operator is simply placed on top of the stack.

When the `]` operator is found, items are taken one by one off the stack and placed in a new list. When the `[` operator is found on the stack, we stop adding items to the list and place the new list on the stack. We can see this in operation here:

```
RPN>>> [ 1 2 3 ]
Stack: [] Tokens: ['[', 1, 2, 3, ']']
Stack: ['['] Tokens: [1, 2, 3, ']']
Stack: ['[', 1] Tokens: [2, 3, ']']
Stack: ['[', 1, 2] Tokens: [3, ']']
Stack: ['[', 1, 2, 3] Tokens: [']']
[[1, 2, 3]]
RPN>>> clr
Stack: [[1, 2, 3]] Tokens: ['clr']
[]
RPN>>> 99 [ 3 9 12 ] 22 [ 4 3 1 ]
Stack: [] Tokens: [99, '[', 3, 9, 12, ']', 22, '[', 4, 3, 1, ']']
Stack: [99] Tokens: ['[', 3, 9, 12, ']', 22, '[', 4, 3, 1, ']']
Stack: [99, '['] Tokens: [3, 9, 12, ']', 22, '[', 4, 3, 1, ']']
Stack: [99, '[', 3] Tokens: [9, 12, ']', 22, '[', 4, 3, 1, ']']
Stack: [99, '[', 3, 9] Tokens: [12, ']', 22, '[', 4, 3, 1, ']']
Stack: [99, '[', 3, 9, 12] Tokens: [']', 22, '[', 4, 3, 1, ']']
Stack: [99, [3, 9, 12]] Tokens: [22, '[', 4, 3, 1, ']']
Stack: [99, [3, 9, 12], 22] Tokens: ['[', 4, 3, 1, ']']
Stack: [99, [3, 9, 12], 22, '['] Tokens: [4, 3, 1, ']']
Stack: [99, [3, 9, 12], 22, '[', 4] Tokens: [3, 1, ']']
Stack: [99, [3, 9, 12], 22, '[', 4, 3] Tokens: [1, ']']
Stack: [99, [3, 9, 12], 22, '[', 4, 3, 1] Tokens: [']']
[99, [3, 9, 12], 22, [4, 3, 1]]
RPN>>>
```

List processing commands: Add the following keywords to the RPN calculator:

- `ref`: Taking a list and an integer on top of the stack, pushes the indexed item onto the list.
- `del` Taking a list and an integer off the stack, deletes the indexed item and pushes the new list on the stack.
- `append` Taking a list and anything else on the list, pushes the list with the new item onto the stack.
- `len` Taking a list off the stack, pushes the list and its length back on the stack.

Note that since `+` already works with lists, we get that for free.

Examples of this at work are seen here:

```
RPN>>> [ 1 2 3 4 ] len 1 - del
Stack: [] Tokens: ['[', 1, 2, 3, 4, ']', 'len', 1, '-', 'del']
Stack: ['['] Tokens: [1, 2, 3, 4, ']', 'len', 1, '-', 'del']
Stack: ['[', 1] Tokens: [2, 3, 4, ']', 'len', 1, '-', 'del']
Stack: ['[', 1, 2] Tokens: [3, 4, ']', 'len', 1, '-', 'del']
Stack: ['[', 1, 2, 3] Tokens: [4, ']', 'len', 1, '-', 'del']
Stack: ['[', 1, 2, 3, 4] Tokens: [']', 'len', 1, '-', 'del']
Stack: [[1, 2, 3, 4]] Tokens: ['len', 1, '-', 'del']
Stack: [[1, 2, 3, 4], 4] Tokens: [1, '-', 'del']
Stack: [[1, 2, 3, 4], 4, 1] Tokens: ['-', 'del']
Stack: [[1, 2, 3, 4], 3] Tokens: ['del']
[[1, 2, 3]]
RPN>>> [ 4 5 6 ] +
Stack: [[1, 2, 3]] Tokens: ['[', 4, 5, 6, ']', '+']
Stack: [[1, 2, 3], '['] Tokens: [4, 5, 6, ']', '+']
Stack: [[1, 2, 3], '[', 4] Tokens: [5, 6, ']', '+']
Stack: [[1, 2, 3], '[', 4, 5] Tokens: [6, ']', '+']
Stack: [[1, 2, 3], '[', 4, 5, 6] Tokens: [']', '+']
Stack: [[1, 2, 3], [4, 5, 6]] Tokens: ['+']
[[1, 2, 3, 4, 5, 6]]
RPN>>> 0 del
Stack: [[1, 2, 3, 4, 5, 6]] Tokens: [0, 'del']
Stack: [[1, 2, 3, 4, 5, 6], 0] Tokens: ['del']
[[2, 3, 4, 5, 6]]
RPN>>>
```