This test is out of 30 points with 5 points extra credit.

In general, you use helper functions that you create, and any built-in Python functions and libraries not explicitly prohibited for each question.

1. (10 points) Create the function *StrStrip(s)* which removes any leading and trailing spaces from the string *s*. You may not use any built-in string functions or methods. Example:
```
StrStrip("  fred    harry   ") -> "fred    harry"
StrStrip("    ") -> ''
```

2. (10 points) Create the function *StrSplit(s,c)* which will be given a non-empty string *s* and a single character *c*, and will split the string the same way that *s.split(c)* would have. I strongly recommend against using the *.split()* method for this question. Examples:
```
StrSplit("ab.c.de", ".") -> ["ab", "c", "de"]
StrSplit("ab&&c&de&", "&") -> ["ab", "", "c", "de", ""]
StrSplit("fred", "a") -> ["fred"]
```

3. (10 points) Create the function *IsInt(s)* that will return *True* if the string s represents a non-negative integer, possibly surrounded by spaces, otherwise *False*. Unfortunately, you may not use the *int()* or *float()* functions nor the *.isdigit()* method. Examples:
```
IsInt("425") -> True
IsInt("-4") -> False
IsInt("  +10  ") -> True
IsInt("  + 10 ") -> False
IsInt("42.5") -> False
IsInt("42e4") -> False
IsInt("42.") -> False
IsInt("+-5") -> False
IsInt("+") -> False
```

4. **Extra credit (5p)** Create the function *Mode(L)* which will be given a non-empty list of numbers already in sorted order (from low to high, but with possible duplicates). The function should return the mode of the list, namely the number with the highest occurrence of duplicates. If there's more than one number with the same highest number of duplicates (e.g. there are 5 duplicates of the number 3 and also 5 of 18), then return the smallest (3 in this case). If all the numbers in the list are different, return the smallest. Examples:
```
Mode([1, 2, 2.3, 2.3, 4, 7, 7, 7, 9])  # returns 7
Mode([1, 2, 6, 9])  # returns 1
Mode([1, 2, 2.3, 2.3, 4, 7, 7, 9])  # returns 2.3
```