Generic counting function:

```
def some_counting_function(s):
  total = 0
  for pos in range(0, len(s)):
    if <test s[pos] for something>:
      total = total + 1
  return total
```

Generic filtering function:

```
def some_filtering_function(s):
  answer = ""
  for pos in range(0, len(s)):
    if <test s[pos] for something>:
      answer = answer + s[pos]
  return answer
```

Practice:

1. Write a function called count_digits that returns the number of digits in a string.
   Example: count_digits("abc123def5") returns 4

2. Write a function called filter_digits that returns only the digits from a string.
   Example: filter_digits("abc123def5") returns "1235"

3. Write a function called sum_digits that returns the sum of all the digits in a string.
   Example: sum_digits("abc123def5") returns 30

4. Write a function called reverse that returns (not prints) the reverse of string s.
   Example: reverse("abc") returns "cba"

5. Write a function called remove_capitals that returns the string s with capital letters removed.
   Example: remove_capitals("AbCDeFGhi9") returns "behi9"

6. Write a function called count_first that counts the number of characters in a string that are identical to the first character.
   Example: count_first("purple") returns 2

7. Write a function called count_dups that counts the number of back-to-back duplicated characters in a string.
   Example: count_dups("balloon") returns 2.

8. Write a function called count_distinct that counts the number of distinct characters in a string. In other words, count the total number of different characters that make up the string.
   Example: count_unique("abracadabra") returns 5.

   Hint: Think about how you would solve this on paper. If I give you a string, and you look at each character in the string left to right, how can you count the total number of different character types?