CSC108H Worksheet: While Loops

1. In the boxes below, fill in the missing code that will make the function definition match its description.

```
def every_nth_character(s, n):
    """ (str, int) -> str

    Return every nth character in s starting at index 0.

    >>> every_nth_character('Computer Science', 3)
    'CpeSee'
    """

    result = ''
    i = 0

    while [                              ]:

        result = result + s[i]

        i = [                            ]

    return result
```

2. In the boxes below, fill in the missing code that will make the function definition match its description.

```
def find_letter_n_times(s, letter, n):
    """ (str, str, int) -> str

    Precondition: letter occurs at least n times in s

    Return the smallest substring of s starting from index 0 that contains
    n occurrences of letter.

    >>> find_letter_n_times('Computer Science', 'e', 2)
    'Computer Scie'
    """

    i = 0
    count = 0

    while [                              ]:
        if [                             ]:
            count = count + 1
        i = i + 1

    return [                             ]
```

CSC108H Worksheet: While Loops

3. In math, the Collatz conjecture states that starting from any number, you will eventually obtain 1 by repeatedly following these two steps:

   - if the number is even, divide by 2
   - if the number is odd, multiply by 3 and add 1

   Complete this function to count Collatz steps for a particular number.

```
def count_collatz_steps(n):
    """ (int) -> int

    Return the number of steps it takes to reach 1, by applying the two steps
    of the Collatz conjecture beginning from n.

    >>> count_collatz_steps(6)
    8
    """
```

4. The function below has an incomplete header and docstring. Based on the code in the function body, fill in the missing parts: the Type Contract, Header, Description, and Examples.

```
def                              :
    """




    """

    i = 0

    while i < len(s) and s[i] not in '0123456789':
        i = i + 1
    return i
```