CSC236 – Tutorial 7: Correctness of Recursive Functions

Exercise 1: Reversing a String

For all strings u, v, we say that v is the *reversal* of u, denoted $v = u^R$, if

$$|u| = |v| \wedge \forall 0 \leq i \leq |u| - 1,\ u_i = v_{|u|-i-1}$$

where $|u|$ denotes the *length* of u, and u_i is the i-th character of u. We assume that strings are indexed from 0 to the length of the string minus 1.
For example, "abcde" = ("edcba")^R. Consider the algorithm below that reverses a string u:

```
1 def rev(u):
2   L = len(u)
3   if L < 2:
4     return u
5   else:
6     m = L // 2
7     v = rev(u[0..m-1])
8     w = rev(u[m..L - 1])
9     return wv # string concatenation
```

where $u[i..j]$ is the substring of u from position i to position j (both inclusive). The goal is to prove that algorithm `rev` correctly reverses a string.

Write pre- and postconditions for the given function `rev`, and give a precise statement for correctness of `rev`. Then, show that `rev` is correct according to your correctness statement.

Exercise 2: Proving Correctness

Consider the following function.

```
1  def mystery(a):
2    '''Pre: ???
3    Post: ???
4    '''
5    i = 1
6    b = 1
7    while i < a:
8      b = b + 2*i + 1
9      i = i + 1
10   return b
```

1. What does this function do? State preconditions and postconditions for this function.

2. Prove that the function is correct using a loop invariant and variant.