CSC236 – Tutorial 7: Recursive Program Correctness

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

$$|u| = |v| \land \forall 1 \leq i \leq |u| - 1, \; u_i = v_{|u|+1-i}$$

where $|u|$ denotes the **length** of $u$, and $u_i$ is the $i$-th character of $u$. We also assume that strings are indexed from 1 to the length of the string.

For example, “abcde” = (“edcba”)$^R$. Consider the algorithm below that reverses a string $u$:

```python
1 rev(u):
2   l = |u|
3   if l < 2:
4       return u
5   else:
6       m = l div 2
7       v = rev(u[1..m])
8       w = rev(u[m+1..|u|])
9       return wv
```

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 state a precise statement for correctness of `rev`. Then, show that `rev` is correct according to your statement.
2. Consider the recursive selection sort algorithm of Tutorial 4:

```python
def recSS(A, i):
    if i < len(A) - 1:
        # Find the minimum element in A[i:]
        small = i
        for j in range(i + 1, len(A)):
                small = j
        # Swap A[i] and A[small]
        temp = A[i]
        A[small] = temp
        # Sort the remainder of the list
        recSS(A, i + 1)
```

Write pre- and postconditions for `recSS`. Can you argue informally why this program is correct, assuming the loop does indeed find the minimum element in A[i..len(A)]?