```
String (E, C): trait
  % Index, substring
  includes List
  introduces
    __[__]: C, Int → E
    prefix: C, Int → C
    removePrefix: C, Int → C
    substring: C, Int, Int → C
  asserts ∀ e: E, s: C, i, n: Int
    tail(empty) == empty;
    init(empty) == empty;
    s[0] == head(s);
    n ≥ 0 ⇒ s[n + 1] = tail(s)[n];
    prefix(empty, n) == empty;
    prefix(s, 0) == empty;
    n ≥ 0
      ⇒ prefix(e ⊣ s, n + 1) = e ⊣ prefix(s, n);
    removePrefix(s, 0) == s;
    n ≥ 0
      ⇒ removePrefix(s, n + 1)
          = removePrefix(tail(s), n);
    substring(s, 0, n) == prefix(s, n);
    i ≥ 0
      ⇒ substring(s, i + 1, n)
          = substring(tail(s), i, n)
  implies
    IndexOp (⊣ for insert)
    C partitioned by len, __[__]
     converts tail, init
```