```
Deque (E, C): trait
  % Double ended queue operators
  includes Integer
  introduces
    empty: → C
    __ ⊣ __: E, C → C
    __ ⊢ __: C, E → C
    count: E, C → Int
    __ ∈ __: E, C → Bool
    head, last: C → E
    tail, init: C → C
    len: C → Int
    isEmpty: C → Bool
  asserts
    C generated by empty, ⊢
    ∀ e, e1, e2: E, d: C
      count(e, empty) == 0;
      count(e, e1 ⊣ d) ==
        count(e, d) + (if e = e1 then 1 else 0);
      e ∈ d == count(e, d) > 0;
      e ⊣ empty == empty ⊢ e;
      (e1 ⊣ d) ⊢ e2 == e1 ⊣ (d ⊢ e2);
      head(e ⊣ d) == e;
      last(d ⊢ e) == e;
      tail(e ⊣ d) == d;
      init(d ⊢ e) == d;
      len(empty) == 0;
      len(d ⊢ e) == len(d) + 1;
      isEmpty(d) == d = empty
  implies
    Stack (head for top, tail for pop,
           ⊣ for push, len for size),
    Queue (⊣ for append, last for head,
           init for tail)
    C generated by empty, ⊣
    C partitioned by len, head, tail
    C partitioned by len, last, init
    ∀ d: C
      d ≠ empty
        ⇒ (head(d) ⊣ tail(d) = d
             ∧ init(d) ⊢ last(d) = d)
   converts head, last, tail, init, len
    exempting head(empty), last(empty),
      tail(empty), init(empty)
```