```
--algorithm BBuf {
   variables in = Input, out = ⟨⟩,
             buf ∈ [0 .. (N − 1) → Msg], p = 0, c = 0 ;

   process ( Producer = "P" )
    { p1: while ( TRUE )
            { await p ⊖ c ≠ N ;
              buf[p%N] := IHead(in) ;
              in := ITail(in) ;
              p := p ⊕ 1
            }
    }

   fair process ( Consumer = "C" )
    { c1: while ( TRUE )
            { await p ≠ c ;
              out := Append(out, buf[c%N]) ;
              c := c ⊕ 1
            }
    }
 }
```

CLOSE