

```

-----MODULE BoundedBuffer-----
EXTENDS Integers, Sequences, ISequences

CONSTANT N, Msg, Input
ASSUME /\ N_ in Nat \ {0}
        /\ Input_ in ISeq(Msg)

a_ (+) b_ == (a_ + b_) % 2 * N
a_ (-) b_ == (a_ - b_) % 2 * N

--algorithm BBuf{
  variables in_ = Input, out_ = << >>,
             buf_ \ in_ [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
    }
  }
}

*****
=====

```