

```
1: signature QUEUE =
2:   sig
3:     type 'a queue ;
4:     val makeQueue : 'a list -> 'a queue ;
5:     val toList : 'a queue -> 'a list ;
6:     val enqueue : ('a * 'a queue) -> 'a queue ;
7:     val dequeue : 'a queue -> ('a * 'a queue) option ;
8:     val isEmpty : 'a queue -> bool ;
9:     val rev      : 'a queue -> 'a queue ;
10:   end ;
11:
12:
13: structure Queue : QUEUE =
14: struct
15:   type 'a pair = ('a * 'a ) ;
16:   (* Invariant: non-empty queues have a non-empty head. *)
17:   datatype 'a queue = Q of 'a list pair ;
18:
19:   fun makeQueue (x:'a list) = Q(x,nil) ;
20:   fun toList (Q(a,b)) = a @ (rev b) ;
21:
22:   fun enqueue (a,Q([],_)) = Q([a], nil)
23:     | enqueue (a,Q(head,rTail)) = Q(head,a::rTail)
24:
25:   fun dequeue (Q(nil,_)) = NONE
26:     | dequeue (Q([a],b)) = SOME(a, Q(rev b,[]))
27:     | dequeue (Q(c::cs,b)) = SOME(c,Q(cs,b))
28:
29:   fun isEmpty (Q ([],_)) = true
30:     | isEmpty _ = false
31:
32:   fun rev (Q (b::bs,[])) =Q(helper(b::bs,[]))
33:     | rev (Q(a,b)) = Q(b,a)
34:   and helper([a],bs) = ([a],bs)
35:     | helper (b::bs,cs) = helper(bs,b::cs)
36: end
```