```
(defclass queue ()
  ((contents :initarg :contents :accessor queue-contents)))

(defun make-queue ()
  (make-instance 'queue :contents '()))

(defmethod empty ((q queue))
  (null (queue-contents q)))

(defmethod enq ((q queue) element)
  (make-instance
   'queue :contents (append (queue-contents q) (list element))))

(defmethod deq ((q queue))
  (assert (not (empty q)))
  (values (car (queue-contents q))
          (make-instance 'queue :contents (cdr (queue-contents q)))))
```



```
(defun make-queue ()
  '())

(defun empty (q)
  (null q))

(defun enq (q elem)
  (append q (list elem)))

(defun deq (q)
  (values (car q) (cdr q)))
```

```
Interface:      2p
   Types et nombre des paramètres
   Vraiment fonctionnel

Implémentation: 2p
   Utilisation d'une liste Lisp
   Respect de l'interface
   Valeurs multiple ou liste de valeurs

Génie Logiciel: 1p
   Indentation
   Code Bref
```