

Replace the following in section 4.14.2

4.14.2 Queue methods

In addition to the array operators, queues provide several built-in methods. [Assume these declarations for the examples that follow:](#)

```
typedef mytype element_t; // mytype is any legal type for a queue  
typedef element_t queue_t[$];  
element_t e;  
queue_t Q;  
int i;
```

4.14.2.1 size()

The prototype for the size() method is:

```
function int size();
```

The size() method returns the number of items in the queue. If the queue is empty, it returns 0.

```
for ( int j = 0; j < Q.size; j++ ) $display( Q[j] );
```

4.14.2.2 insert()

The prototype of the insert() method is:

```
function void insert(input int index, ref queue_type queue, input element_t item);
```

The insert() method inserts the given item at the specified index position.

— Q.insert(i, e) is equivalent to: Q = {Q[0:i-1], e, Q[i,\$]}

4.14.2.6 push_front()

The prototype of the push_front() method is:

```
function void push_front(ref queue_type queue, input element_t item);
```

The push_front() method inserts the given element at the front of the queue.

— Q.push_front(e) is equivalent to: Q = {e, Q}

4.14.2.7 push_back()

The prototype of the push_back() method is:

```
function void push_back(ref queue_type queue, input element_t item);
```

The push_back() method inserts the given element at the end of the queue.

— Q.push_back(e) is equivalent to: Q = {Q, e}