



4.2.2 Queues

1) Circular queues and linear queues are examples of data structures that can be implemented using a fixed-length array.

a) Explain why, when implemented using a fixed-length array, a circular queue is usually considered to be a better choice of data structure than a linear queue.

[2 marks]

- When an item is removed from a linear queue;
- all other items need to be shuffled up one
// this can result in unusable space in the array;

Alternative answer

- No need to shuffle items up one
// no unusable spaces;
- After deleting an item from a circular queue;

b) Describe the steps involved in adding an item to a linear queue that has been implemented as a static data structure using an array. Your answer should include a description of how any pointers are used and changed.

[3 marks]

- Check that the queue is not already full;
- if it isn't then add 1 to the value of the rear pointer;
- then add the new item to the position indicated by the rear pointer;

Alternative answer

- Check that the queue is not already full;
- if it isn't then add the new item to the position indicated by the rear pointer;
- then add 1 to the value of the rear pointer;

c) Describe the process that should be followed to add an item to a circular queue implemented as a static data structure using an array.

Your method should deal appropriately with any issues which could arise.

[5 marks]

- Check the queue is not already full;
- Compare the value of the rear pointer with the maximum size of the array;
- If equal then rear pointer becomes zero
Accept. index of the first position in the array instead of zero;
- Otherwise, add one to the rear pointer;
- Insert new item in position indicated by rear pointer;





- d) Describe the steps that must be completed to remove (dequeue) an item from a circular queue that has been implemented using a fixed-length array.

[5 marks]

- Check that the queue is not already empty
// check to see if the current size is 0;
Reject. reference to array instead of queue;
- if it is then deal with (underflow) error
// If it is not then process/dequeue the front item in the queue
Reject. reference to array instead of queue (unless reference to array is made using value of front pointer)
Reject. if dequeue would only happen under some of the correct circumstances;
- Reduce the value of the variable used to store the current size by 1;
- Check if the front is in the last position of the array and if it is set it to the first position in the array
Accept. 0 or 1 instead of first position in the array;
- Else add 1 to the value of the front pointer;

- e) For a priority queue, the process to determine where a new item should be added is more complex than it is with a linear queue.

Describe the steps involved when adding an item to a priority queue, implemented as a static data structure using an array, that were not required when adding an item to a linear queue..

[3 marks]

- Starting with the item at the rear of the queue move each item back one place in the array;
- Until you (reach the start of the queue or) find an item with the same OR higher priority than the item to add;
- Add the new item in the position before that item;

