Question			Marks
1	1	**All marks for AO1 (understanding)** **Max 2 for advantages of dynamic data structures** No wasted memory; Can grow as more data is added to the data structure // no limit on number of items that can be added (except due to hardware limitations); Resources only allocated as they are needed (with static data structures they are allocated at creation even if not needed until later); **Max 2 for disadvantages of dynamic data structures** Additional memory needed for pointers; Can result in memory leak (if memory that is no longer needed is not returned to the heap); Can take longer to access an item directly (for data structures that allow this); **A.** can take longer to add a new item to the data structure (as memory needs to be allocated)	4
1	2	**All marks for AO1 (understanding)** 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; **Max 2** if any errors **Max 1** if circular queue has been described	3

1	3	**All marks for AO1 (understanding)** 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; **NE.** same priority **NE.** higher priority Add the new item in the position before that item; **A.** answers which have the front of the queue as the last item in the array, start at the front and move each item forward one until the correct insertion point is found. **A.** answers that start from the front of the queue until position to insert item is found and then start at the back and move each item back one until position to insert item is found.	3

Question			Marks
02	1	**All marks for AO1 (understanding)** Static data structures have storage size determined at compile-time / before program is run / when program code is translated / before the data structure is first used // dynamic data structures can grow / shrink during execution / at run-time // static data structures have fixed (maximum) size // size of dynamic data structures can change; Static data structures can waste storage space / memory if the number of data items stored is small relative to the size of the structure // dynamic data structures only take up the amount of storage space required for the actual data; Dynamic data structures require (memory to store) pointers to the next item(s) // static data structures (typically) do not need (memory to store) pointers; Static data structures (typically) store data in consecutive memory locations // dynamic data structures (typically) do not store data in consecutive memory locations; **Max 3**	**3**