FAQ

Module-2	
Question 1.	**What distinguishes static from dynamic allocation in particular?**
Answer	Static allocation refers to the process of reserving a certain amount of memory that stays unchanged during its existence. This approach is often used for the storage of arrays. On the other hand, dynamic allocation pertainsto the process of allocating and deallocating memory during runtime, a feature often seen in data structures such as linked lists.
Question 2.	**In certain situations, what factors would lead an individual to preferusing a stack data structure rather than a queue data structure?**
Answer	The stack data structure follows the LIFO (Last In, First Out) principle, whichmakes it well-suited for situations where the most recently added data must be processed first. This characteristic is particularly advantageous in scenarios like function call stacks or expression evaluation. A queue is a data structure that adheres to

	the FIFO (First In, First Out) principle, which means that data is processed in the same order in which it was added.
Question 3.	**Could you perhaps provide an example of a list that is limited in terms ofaccess?**
Answer	Indeed, both stacks and queues may be classified as types of access-restrictedlists. In the context of data structures, it is important to note that stacks enforce a restriction on access, allowing only the top element to be accessed.On the other hand, queues limit operations to the front and rear ends only.
Question 4.	**What are the advantages of using a double-ended queue (DEQueue) incomparison to conventional queues or stacks?**
Answer	Double-ended queues, sometimes referred to as DEQueues, integrate the characteristics of both stacks and queues. The capability to add or remove components from either end is permitted. The adaptability of this characteristic might prove to be beneficial in scenarios where the use of both LIFO and FIFO methods may be necessary.

Question 5.	**Is the use of arrays considered outmoded in light of the more flexibilityoffered by linked lists?**
Answer	It is not always the case. Although linked lists provide advantages such as dynamic allocation and flexibility in size, arrays possess some benefits likeconstant-time access to any element, contiguity in memory, and potentially reduced memory overhead. The selection between these options is contingentupon the particular specifications of the given application.