Q1. What is the Collection framework in Java?

Ans. Collection framework is a combination of classes and interfaces, which is used to store and manipulate the data in the form of objects. It provides various classes such as ArrayList, LinkedList, and HashSet etc. and interfaces such as List, Queue, Set etc.

Q2. What is the difference between ArrayList and LinkedList?

Ans.

ArrayList	**LinkedList**
ArrayList uses a dynamic array	LinkedList uses a doubly linked list.
ArrayList is not efficient for manipulation because too much is required.	Linked list is efficient for manipulation
It provides random access	It do not provide random access
It is better to store and fetch data	It is better to manipulate the data
It takes less memory overhead as it stores only objects.	It takes more memory overhead, as it stores the object as well as the address of that object.

Q3. What is the difference between Iterator and ListIterator?

Ans.

Iterator	**ListIterator**
It traverse the elements in the forward direction only	It traverses the elements in the forward direction and backward direction both.
The Iterator can be used in List, Set, and Queue.	It can only be used in the List.
It can perform remove operation while traversing the collection	It can perform add, remove, and set while traversing the collection

Q4. What is the difference between Iterator and Enumeration?

Ans.

Iterator	**Enumeration**
It can traverse legacy and non-legacy elements.	It can traverse only legacy elements.
It is fail-fast	It is not fail-fast

Slower than Enumeration	Faster than Iterator
The Iterator can perform remove operation while traversing the collection	This can perform only traverse operation on the collection.

Q5. What is the difference between List and Set?
Ans. The List and Set both both extends the collection interface. However, there are some differences:
->The list can contain duplicate elements whereas Set includes unique items.
->List is an ordered collection which maintains the order of insertion whereas Set is an unordered collection which does not maintain the insertion order.
->The list interface can allow a number of null values whereas the Set interface only allows a single unique null value.

Q6. What is the difference between HashSet and TreeSet?
Ans.
Both HashSet and TreeSet are implementations of the Set Interface in java, but they have some differences:
>**Ordering:** HashSet is an unordered collection of elements, while TreeSet is a sorted set of elements.
>**Duplication:** Both of them do not allow duplicate elements.
>**Implementation:** HashSet is implemented using Hash Table, while TreeSet is implemented using a self balancing binary search tree.
>**Performance:** Hashed Set has constant time complexity O(1), while TreeSet has logarithmic time complexity O(log n).
>**Memory usage:** HashSet uses less memory than TreeSet because it only stores the elements, while TreeSet stores additional information for maintaining the order.
>**Iteration:** HashSet provides no guarantees regarding the order of iteration, while TreeSet guarantees the elements are iterated in sorted order.

Q6. What is the difference between Array and ArrayList?
Ans.

Basis of Comparison	Array	Array List

Definition	A straightforward data structure with a continuous memory location, an array stores its contents with the same name but distinct index numbers for each element of the array it contains. It is imperative that all of the data stored in an array be of the same type. After an array has been declared, its size cannot be changed.	The Java collection framework contains a data structure known as an ArrayList, which is dynamic in nature. Additionally, it has components that are of the same type. In this case, it is not necessary for us to specify the length of the list.
Static/Dynamic	Arrays are static	ArrayList is dynamic
Resizable	Fixed Length	Can be Resizable
Initialization	When performing initialization for an array, it is required to specify the size of the array.	It is not necessary to mention the size of an ArrayList.
Performance	Arrays are faster	ArrayList is slower
Generic Type	An array can store primitive data as well as objects, but it cannot store generics.	ArrayList is able to store generics as well as objects, but it cannot

		store data of primitive types.
Iteration	Only loops are permitted in this area.	It is acceptable to use loops and iterators.
Type Safety	It is not type-safe.	It is type-safe.