Ans. 1. In programming, [] is an empty list or array. A list is a collection of items, and an array is a collection of data items that are of the same type. It's a fundamental data structure that is used to store and manipulate data. Lists and arrays are commonly used in programming to store and organize data, which can be accessed and modified based on its index or position in the list or array.

Ans. 2. [2, 4, 'hello', 8, 10]

Ans. 3. d

Ans. 4. d

Ans. 5. ['a', 'b']

Ans. 6. 1

Ans. 7. [3.14, 'cat', 11, 'cat', True, 99]

Ans. 8. [3.14, 11, 'cat', True]

Ans. 9. The list concatenation operator is +, which can be used to join two lists end-to-end, resulting in a new list containing elements from both lists.

list1 = [10, 11, 12, 13, 14]

list2 = [20, 30, 42]

res = list1 + list2

print("Concatenated list:", res)

Concatenated list: [10, 11, 12, 13, 14, 20, 30, 42]

The list replication operator is \*, which can be used to repeat a list a specified number of times.

list1 = [1, 2, 3]

n = 3

result = list1 \* n

print("List replication:", result)

List replication: [1, 2, 3, 1, 2, 3, 1, 2, 3]

Ans. 10. The append() method is used to add an element to the end of a list. It takes one argument, the element to be added, and modifies the original list.

The insert() method is used to insert an element at a specified position in the list. It takes two arguments: the position where the element should be inserted, and the element itself. The original list is modified, and any elements to the right of the insertion point are shifted over to accommodate the new element.

Ans. 11. The remove() method removes the first occurrence of a specified element from a list. It takes the element to be removed as its argument and modifies the original list.

The pop() method removes and returns the element at a specified index from a list. It takes the index of the element to be removed as its argument and modifies the original list. If no index is specified, it removes and returns the last element in the list.

Ans. 12. Lists and strings share similarities in that they both have a length and can be indexed. We can access individual elements in a list or characters in a string using square-bracket indexing . Both can be used in for loops, either iterating over the elements directly or their indices.

Ans. 13. The main difference between tuples and lists in Python is that tuples are immutable, while lists are mutable. This means that elements in a tuple cannot be changed, added, or removed, whereas elements in a list can be modified. Tuples are more memory-efficient than lists and can be used as keys in dictionaries, while lists cannot.

Ans. 14. To create a tuple with the integer value 42

integer\_tuple = (42)

Ans. 15.

my\_list = [1, 2, 3]

my\_tuple = tuple(my\_list)

print(my\_tuple) # Output: (1, 2, 3)

my\_tuple = (1, 2, 3)

my\_list = list(my\_tuple)

print(my\_list) # Output: [1, 2, 3]

Ans. 16. Variables that "contain" list values are actually references to the list object itself. When you create a variable with a list value, you are storing the memory address of the list object in the variable. This means that when you modify the list through the variable, you are directly modifying the original list object

Ans. 17. copy.copy() creates a shallow copy of the list, meaning that it only creates new references to the elements within the list. Changes made to the elements in the copied list will also affect the original list since they still reference the same objects. On the other hand, copy.deepcopy() creates a deep copy of the list, which means that it recursively creates new objects for the elements within the list, including any nested objects. Changes made to the elements in the copied list will not affect the original list because they are now separate objects