Assigenment-02

1. What are data structures, and why are they important
=> **Data structures** are ways of organizing and storing data so that operations like access, modification, and deletion can be performed efficiently. They are essential because they help optimize performance in terms of time and space, especially when handling large datasets.

2. Explain the difference between mutable and immutable data types with examples
=> Mutable can be change after creation eg list ,dict, set
Immutable can't be change after creation eg int , float ,tuple,str

3. What are the main differences between lists and tuples in Python3
=> Main diff between lists and tuples
= list is mutable while tuple is immutable
= list is represented like [] this and tuple like this ()
= list is slower then the tuple

4. Describe how dictionaries store data
=> Dictionaries store the data in a key value pairs in a hash table structure.keys are hashed to determine where the value should be stored in memory for fast access

5. Why might you use a set instead of a list in Python
=> To store the unique items only.
=> To remove the duplicates from the list
=> Faster membership test

6. What is a string in Python, and how is it different from a list
=> A string is a sequence of unicode characters
=> a list can holds multiple types ,but a string can hold only a char type.
=> string is immutable

7. How do tuples ensure data integrity in Python
=>Tuples are immutable the data cannot be modified accidently making them ideal for constants and fixed collection

8.What is a hash table, and how does it relate to dictionaries:
=>A hash table maps keys to values using a hash function, dict use the hash table to allow for access , insert and delete operation in O(1) time complexity

9. Explain why strings are immutable in Python
=> strings are interned for memory efficiency.
=>immutables are thread safe
=>allows string to be used as dict.

10. What advantages do dictionaries offer over lists for certain tasks
=> firster lookup via keys
=> More meaningful association
=> flexible key types if hashable

11. Describe a scenario where using a tuple would be preferable over a list
=> when to use tuple instead of list:-
=>when the data should not change
=> to use dictionaries as keys
=> for faster performance

12. How do sets handle duplicate values in Python
=> set automatically discards the duplicates values
set([1,22,22,44,5])->{1,22,44,5}

13. How does the “in” keyword work differently for lists and dictionaries
=> ‘In’ keyword in list checks that the element is present in list or not in o(n) time .
=> ‘In’ keyword in dict checks for key in o(1) using hash table

14. Can you modify the elements of a tuple? Explain why or why not
=> no we can’t modify the elements in tuple because of immutability

15. What is a nested dictionary, and give an example of its use case
=>a dict where the values are also dict

```
students = {
  "Afaque": {"age": 22, "course": "CSE"},
  "Sara": {"age": 21, "course": "ECE"}
}
```

16.Describe the time complexity of accessing elements in a dictionary
=> best time complexity is O(1) time
=> worst time complexity is O(n) time

17. In what situations are lists preferred over dictionaries
=> when is the order of the element is preferred
=> when the key value pair is not needed

18 Why are dictionaries considered unordered, and how does that affect data retrieval
=>Dict uses the hashtable for insertion , without preserving the insertion order

19 Explain the difference between a list and a dictionary in terms of data retrieval.
=>list uses the indexing for retrieving of data list[0]-> accessing the first item in the list
=> Dict retrieve using the key dict[‘name’]->’Afaque’