1. What exactly is []?

Ans – Tuple , square bracket

1. In a list of values stored in a variable called spam, how would you assign the value 'hello' as the third value? (Assume [2, 4, 6, 8, 10] are in spam.)

Ans - spam = [2,4,6,8,10]

spam[2] = ' hello '

print(spam)

Let's pretend the spam includes the list ['a', 'b', 'c', 'd'] for the next three queries.

1. What is the value of spam[int(int('3' \* 2) / 11)]?

Ans – 8

1. What is the value of spam[-1]?

ANS - 10

1. What is the value of spam[:2]?

Ans = 8

Let's pretend bacon has the list [3, 14, 'cat,' 11, 'cat,' True] for the next three questions.

1. What is the value of bacon.index('cat')?

Ans - 2

1. How does bacon.append(99) change the look of the list value in bacon?

[3,14, “cat” , 11 , “cat” , “true” , 99]

1. How does bacon.remove('cat') change the look of the list in bacon?

Ans - [3, 14, 11, 'cat', True, 99]

1. What are the list concatenation and list replication operators?

Ans - so concatenation combines the tow list and give the result, however replication is repeat a list multiple times .

1. What are the two methods for removing items from a list?

Ans - pop() and remove()

1. Describe how list values and string values are identical.

Ana - because these are assign values by variable so they are identical. And they support indexing and silicing

1. What's the difference between tuples and lists?

Ans - tuples are immutable and list are mutable

1. How do you type a tuple value that only contains the integer 42?

A = (42,)

1. How do you get a list value's tuple form? How do you get a tuple value's list form?

Ans - tuple\_value = tuple(list\_value) , list\_value = list(tuple\_value)

1. Variables that "contain" list values are not necessarily lists themselves. Instead, what do they contain?

Ans - I don’t know the ans can u explain me pls 8218056131

1. How do you distinguish between copy.copy() and copy.deepcopy()?

Ans – Idk this ans

But I did a chatgpt for this so I am writing the ans for future refrence and for reading , don’t consider my ans , you can give me zero for this ans.

original ──► [ 1, 2, ─┐

▲ ▼

shallow [3, 4] ← Shared nested list

original aur shallow dono outer list alag-alag hain.

Lekin [3, 4] wali nested list same memory location point karti hai.

Isiliye agar tu shallow[2][0] = 99 karega, original[2][0] bhi change ho jaayega.

Deep = original ──► [ 1, 2, ─┐ deep ──► [ 1, 2, ─┐

▼ ▼

[3, 4] [3, 4] ← Completely new nested list

Yahan deep aur original dono ki outer lists bhi alag hain aur

[3, 4] wali inner list bhi completely new copy hai.

Toh agar deep[2][0] = 88 karega, original bilkul safe rahega.