**Challenge #3 :- Nested object function**

def get\_value\_from\_nested\_object(obj, key):

if isinstance(obj, dict):

if key in obj:

return obj[key]

for k, v in obj.items():

result = get\_value\_from\_nested\_object(v, key)

if result is not None:

return result

elif isinstance(obj, list):

for item in obj:

result = get\_value\_from\_nested\_object(item, key)

if result is not None:

return result

return None

# Example usage

nested\_object = {

'a': 1,

'b': {

'c': {

'd': 2

}

},

'e': [

{

'f': 3

},

{

'g': {

'h': 4

}

}

]

}

key = 'h'

value = get\_value\_from\_nested\_object(nested\_object, key)

print(value)

# Output: 4

In this code, the **get\_value\_from\_nested\_object** function recursively traverses the nested object. If the object is a dictionary, it checks if the given key exists and returns its corresponding value. If the key is not found, it recursively calls the function on each value in the dictionary.