```python
# Write a function to count the number of elements of each data type in the list.

def count_data_types(lst):
    type_counts = {}

    for element in lst:
        element_type = type(element)
        if element_type in type_counts:
            type_counts[element_type] += 1
        else:
            type_counts[element_type] = 1

    return type_counts



heterogeneous_list = [42,46, "Hello", 3.14, True, None, [1, 2, 3], {"key": "value"}, (4,
5), {9, 8, 7}]
result = count_data_types(heterogeneous_list)
print(result)
```

```python
# Create a function that takes a data type as an argument and returns a new list
containing only elements of that type from the original list.

def filter_by_type(lst, data_type):
    return [element for element in lst if isinstance(element, data_type)]

heterogeneous_list = [42, "Hello", 3.14, True, None, [1, 2, 3], {"key": "value"}, (4, 5),
{9, 8, 7}]
filtered_list = filter_by_type(heterogeneous_list, int)
print(filtered_list)  # Output: [42]

filtered_strings = filter_by_type(heterogeneous_list, str)
print(filtered_strings)  # Output: ["Hello"]

filtered_floats = filter_by_type(heterogeneous_list, float)
print(filtered_floats)  # Output: [3.14]

filtered_lists = filter_by_type(heterogeneous_list, list)
```

```
print(filtered_lists)  # Output: [[1, 2, 3]]

filtered_lists = filter_by_type(heterogeneous_list, tuple)

print(filtered_lists)
```

```
# Write a function to sum all numerical values (integers and floats) in the list.
def sum_numerical_values(input_list):
    total = 0
    for item in input_list:
        # Check if the type of the item is either int or float
        if type(item) == int or type(item) == float:
            total += item
    return total

# Test the function with the given list
list1 = [2, 4, 5.4, 6.5, 7, 3, "str1"]
LIST1=[1,1,1.1]
heterogeneous_list = [42, "Hello", 3.14, True, None, [1, 2, 3], {"key": "value"}, (4, 5),
{9, 8, 7}]
print(sum_numerical_values(list1))
print(sum_numerical_values(heterogeneous_list))
```

```
# Write a function to concatenate all string elements in the list into a single string.

heterogeneous_list = [42, "Hello", 3.14, True, None, [1, 2, 3], {"key": "value"}, (4, 5),
{9, 8, 7}]

def concatenate_strings(input_list):
    result = ""
    for item in input_list:
        # Check if the item is a string
```

```
        # we can also use isinstanceof()
        if type(item) == str:
            result += item
    return result

# Test the function with the given list
heterogeneous_list = [42, "Hello", "world",3.14, True, None, [1, 2, 3], {"key": "value"},
(4, 5), {9, 8, 7}]
print(concatenate_strings(heterogeneous_list))
```

```
# Write a function to flatten the list, where nested lists, tuples, and sets are expanded
into individual elements in a single list.
def flatten_list(heterogeneous_list):
    flattened_list = []
    for item in heterogeneous_list:
        if isinstance(item, (list, tuple, set)):
            flattened_list.extend(flatten_list(item))
        else:
            flattened_list.append(item)
    return flattened_list

# Test the function with the given heterogeneous list
heterogeneous_list = [42, "Hello", "world", 3.14, True, None, [1, 2, 3], {"key":
"value"}, (4, 5), {9, 8, 7}]
print(flatten_list(heterogeneous_list))
```

```
# Write a function to check if a specific key exists in any dictionary elements within
the list.

def key_exits(heterogeneous_list,key):
    for item in heterogeneous_list:
        if type(item)==dict:
            for KEY in item.keys():
                if KEY ==key:
                    return True
    return False

heterogeneous_list = [42, "Hello", "world", 3.14, True, None, [1, 2, 3], {"key":
"value"}, (4, 5), {9, 8, 7}]
```

```
print(key_exits(heterogeneous_list,"key"))
# Write a function to count the number of True and False boolean values in the list

def boolen_values(heterogeneous_list):
    tc=0
    fc=0
    for item in heterogeneous_list:
        if item==True:
            tc+=item
        elif item ==False:

            fc += item
    return tc,fc
heterogeneous_list = [42, "Hello", "world", 3.14, True, None, [1, 2, 3], {"key":
"value"}, (4, 5), {9, 8, 7}]
tc1,fc1=boolen_values(heterogeneous_list)
print("Number of True count",tc1)
print("number of false count",fc1)
```

```
# Write a function to convert all elements in the list to their string representations
and return a new list of these strings.

def convert_to_strings(input_list):
    return [str(item) for item in input_list]

# Test the function with a list containing various types of elements
heterogeneous_list = [42, "Hello", "world", 3.14, True, None, [1, 2, 3], {"key":
"value"}, (4, 5), {9, 8, 7}]
string_list = convert_to_strings(heterogeneous_list)
print(string_list)
```

```
# Write a function to find the maximum value among all numeric elements in the list.
def find_max_numeric(lst):

    # numeric_elements = [x for x in lst if isinstance(x, (int, float))]
    numeric_elements = []
    for x in lst:
        if isinstance(x, (int, float)):
            numeric_elements.append(x)
```

```
    return max(numeric_elements, default=None)

heterogeneous_list = [42, "Hello", "world", 3.14, True, None, [1, 2, 3], {"key":
"value"}, (4, 5), {9, 8, 7}]
result = find_max_numeric(heterogeneous_list)
print(result)  # Output: 4
```

```
# Write a function to replace all None values in the list with a default value provided
as an argument.
def replace_none_with_default(lst, default_value):
    for i in range(len(lst)):
        if lst[i] is None:
            lst[i] = default_value
    return lst


heterogeneous_list = [42, "Hello", "world", 3.14, True, None, [1, 2, 3], {"key":
"value"}, (4, 5), {9, 8, 7}]
default_value = 6
result = replace_none_with_default(heterogeneous_list, default_value)
print(result)
```

```
# Find the second highest value in a list.
def findLargest(arr):
    secondLargest = 0
    largest = min(arr)

    for i in range(len(arr)):
        if arr[i] > largest:
            secondLargest = largest
            largest = arr[i]
        else:
            secondLargest = max(secondLargest, arr[i])

    # Returning second largest element
    return secondLargest

print(findLargest([10, 20, 4, 45, 43]))
```

```python
# Program to check if a number is prime or not



# To take input from the user
num = int(input("Enter a number: "))

# define a flag variable
flag = False

if num == 1:
    print(num, "is not a prime number")
elif num > 1:
    # check for factors
    for i in range(2, num):
        if (num % i) == 0:
            # if factor is found, set flag to True
            flag = True
            # break out of loop
            break

    # check if flag is True
    if flag:
        print(num, "is not a prime number")
    else:
        print(num, "is a prime number")
```

```python
# checking evenor not
# num = int(input("Enter a number: "))
num=[1,2,3,4,5]
for i in num:
    if i%2==0:
        print(i ," is even number")
    else:
        print(i," odd number")
```

Difference between list,tuple,sets,dict

list	tuple	sets	dict
mutable	immutable	mutable	mutable
Duplicates are allowed	Duplicates are allowed	Not allowed	Key are unique but values duplicates are allowed
ordered	ordered	unordered	unordered

A list is mutable i.e we can make any changes in the list.	A tuple is immutable i.e we can not make any changes in the tuple.	A set is mutable i.e we can make any changes in the set, its elements are not duplicated.	A dictionary is mutable, its Keys are not duplicated

Question

What are the primary data types in Python and their uses?

How do you handle type conversion in Python?

What are mutable and immutable data types? Give examples.

Explain the differences between lists and tuples. When would you use each?

How do dictionaries work in Python, and what are their key features?

Question

Explain the use of if-else statements in Python. Provide an example.
How do you use the elif statement, and when is it useful?
Describe the use of the try-except block in error handling.
What is the purpose of the finally clause in a try-except block?

Question

How do you use for loops in Python? Provide an example.
What is the difference between for and while loops?
How can you iterate over a dictionary using a loop?
Explain the use of break, continue, and pass statements within loops.

Question

How do you define a function in Python? Provide a syntax example.
What are *args and **kwargs in Python functions? How are they used?
Explain the concept of recursion with an example.
What is the scope of a variable inside a function?

Question

What are the common types of exceptions in Python?
How do you raise an exception in Python?
Explain the use of custom exceptions and how to create them.

Question

What is a class in Python, and how do you create one?
Explain the concept of an object in Python. How do you instantiate an object?
What is the purpose of the **init** method in a class?

Question

What is inheritance in Python, and why is it useful?

How do you implement inheritance in Python? Provide an example.
What is method overriding in the context of inheritance?

Question

Explain encapsulation and its importance in object-oriented programming.
How do you implement getter and setter methods in a Python class?
What is the property decorator, and how is it used for encapsulation in Python?
Provide an example of using private variables in a class with getter and setter methods.