

```
In [ ]: def count_word_frequency(paragraph, word):
        words = paragraph.lower().split()
        return words.count(word.lower())

paragraph = "Hello, my name is Manju Thampan. I am passionate about the domain of Pet Shop Management System. " \
            "I am a student with register number 2347236, currently pursuing my studies in the year 2023. " \
            "My interest is in Pet Shop Management System.. " \
            "Pet shop Management System is a platform that allows the different pet shop owners or managers to post a

word_to_find = input("Enter the word to find its frequency: ")

frequency = count_word_frequency(paragraph, word_to_find)
print(f"The word '{word_to_find}' appears {frequency} times in the paragraph.")
```

The word 'pet' appears 8 times in the paragraph.

```
In [ ]: def count_symbols(paragraph):
        num_alpha = 0
        num_numerals = 0
        num_special = 0

        for char in paragraph:
            if char.isalpha():
                num_alpha += 1
            elif char.isdigit():
                num_numerals += 1
            else:
                num_special += 1

        return num_alpha, num_numerals, num_special

def main():
    paragraph = """
    My name is Manju Thampan.My register number is 2347236.My domain is Pet Shop Management System
    """

    num_alpha, num_numerals, num_special = count_symbols(paragraph)
```

```

print(f"Number of alphabets: {num_alpha}")
print(f"Number of numerals: {num_numerals}")
print(f"Number of special symbols: {num_special}")

main()

```

Number of alphabets: 71
Number of numerals: 7
Number of special symbols: 26

```

In [ ]: print("djkhfsj")
def classify_element(element):
    try:
        if "." in element:
            float(element)
            return "float"
        else:
            int(element)
            return "int"
    except ValueError:
        return "string"

def main():
    print("halkjdhfl")
    paragraph = """
    My name is "Manju Thampan" and my registration number is 2347236.
    I have studied bca degree. My percentage is 89.3

    Pet Shop Management System is my Domain .Pet shop Management System is a platform that allows the different pet s

    """

    elements = paragraph.split()

    for element in elements:
        data_type = classify_element(element.strip('\\".,'))
        print(f"{element} - {data_type}")

if __name__ == "__main__":
    main()

```

djkfhsj
halkjdhfl
My - string
name - string
is - string
"Manju - string
Thampan" - string
and - string
my - string
registration - string
number - string
is - string
2347236. - int
I - string
have - string
studied - string
bca - string
degree. - string
My - string
percentage - string
is - string
89.3 - float
Pet - string
Shop - string
Management - string
System - string
is - string
my - string
Domain - string
.Pet - string
shop - string
Management - string
System - string
is - string
a - string
platform - string
that - string
allows - string
the - string
different - string
pet - string
shop - string

owners - string
or - string
managers - string
to - string
post - string
and - string
publish - string
their - string
pet - string
and - string
other - string
products - string
on - string
their - string
pet - string
shop. - string
The - string
platform - string
is - string
an - string
ecommerce - string
website - string
specifically - string
intended - string
for - string
pets, - string
pet - string
shop - string
services, - string
pet - string
shop - string
accessories - string
and - string
other - string
products - string

In []: `def set_operations_example():`

```
    mixed_set = {42, 3.14, "Hello", True, (1, 2, 3)}
```

```
    print("Original set:", mixed_set)
```

```

# Using pop() to remove and return an arbitrary element
popped_element = mixed_set.pop()
print("Popped element:", popped_element)
print("Set after pop:", mixed_set)

# Using clear() to remove all elements
mixed_set.clear()
print("Set after clear:", mixed_set)

# Repopulating the set
mixed_set = {42, 3.14, "Hello", True, (1, 2, 3)}

# Using discard() to remove a specific element
mixed_set.discard(3.14)
print("Set after discard:", mixed_set)

# Using del to completely delete the set
del mixed_set
# Uncommenting the following line would result in an error since the set is deleted
# print("Set after del:", mixed_set)

# Call the function to demonstrate set operations
set_operations_example()

```

Original set: {'Hello', True, 3.14, 42, (1, 2, 3)}

Popped element: Hello

Set after pop: {True, 3.14, 42, (1, 2, 3)}

Set after clear: set()

Set after discard: {'Hello', True, 42, (1, 2, 3)}

```

In [ ]: def sort_set_descending(input_set):

    return sorted(input_set, reverse=True)

# Example set with string attributes from my domain
domain_strings = { "Artificial Intelligence", "Machine Learning", "Chatbots", "Knowledge Representation"}

# Sorting the set in descending order
sorted_set = sort_set_descending(domain_strings)

print("Original set:", domain_strings)
print("Sorted set (descending):", sorted_set)

```

Original set: {'Machine Learning', 'Chatbots', 'Artificial Intelligence', 'Knowledge Representation'}
Sorted set (descending): ['Machine Learning', 'Knowledge Representation', 'Chatbots', 'Artificial Intelligence']

In []: **def** tuple_packing_unpacking():

```
# Tuple packing - combining attributes into a tuple
language_tech_tuple = ("Natural Language Processing", "Chatbots", "Sentiment Analysis")

# Tuple unpacking - assigning tuple elements to individual variables
nlp, chatbots, sentiment_analysis = language_tech_tuple

# Display the original tuple and the unpacked variables
print("Original tuple:", language_tech_tuple)
print("Unpacked variables:")
print("NLP:", nlp)
print("Chatbots:", chatbots)
print("Sentiment Analysis:", sentiment_analysis)

# Call the function to demonstrate tuple packing and unpacking
tuple_packing_unpacking()
```

Original tuple: ('Natural Language Processing', 'Chatbots', 'Sentiment Analysis')

Unpacked variables:

NLP: Natural Language Processing

Chatbots: Chatbots

Sentiment Analysis: Sentiment Analysis

In []: **def** count_character(domain_name, target_character):

```
    char_count = domain_name.count(target_character)
    return char_count

# Get the domain name from the user
domain_name = input("Enter your domain name as characters: ")

# Get the target character to count
target_character = input("Enter the character to count: ")

# Call the function to count and print the result
character_count = count_character(domain_name, target_character)
print(f"Count of '{target_character}' = {character_count}")
```

Count of 'e' = 3

```
In [ ]: def slicing_and_negative_indexing(input_string):

    # Original string
    print("Original string:", input_string)

    # Positive slicing
    print("Positive slicing:")
    print("1st to 5th characters:", input_string[0:5])
    print("2nd to last characters:", input_string[1:])
    print("Every second character:", input_string[::2])

    # Negative indexing
    print("Negative indexing:")
    print("Last character:", input_string[-1])
    print("Last 3 characters:", input_string[-3:])
    print("3rd to last character:", input_string[-3])

    # Slicing with negative indexing
    print("Slicing with negative indexing:")
    print("From 2nd to 4th characters from the end:", input_string[-4:-1])
    print("From 2nd to last character from the end:", input_string[-2:])

    # Example input string
    domain_name = "petshopmanagement"

    # Call the function to demonstrate slicing and negative indexing
    slicing_and_negative_indexing(domain_name)
```

```
Original string: petshopmanagement
Positive slicing:
1st to 5th characters: petsh
2nd to last characters: etshopmanagement
Every second character: pthpaaet
Negative indexing:
Last character: t
Last 3 characters: ent
3rd to last character: e
Slicing with negative indexing:
From 2nd to 4th characters from the end: men
From 2nd to last character from the end: nt
```

