```
#Q1. Write a Python program that takes two lists as input and returns
a new list containing the common elements between the two input lists?


def find_common_elements(list1, list2):

    # Using list comprehension

    common_elements = [element for element in list1 if element in
list2]
    return common_elements

# Example usage:

list1 = [1, 2, 3, 4, 5]
list2 = [4, 5, 6, 7, 8]
print(find_common_elements(list1, list2))
```

```
[4, 5]
```

```
#Q2. Write a Python program that takes two strings as input and
returns 'yes' if the first string is a substring of the second string
and 'no' otherwise using comparison operaters?


def is_substring(string1, string2):
    if string1 in string2:
        return 'yes'
    else:
        return 'no'

# Example usage:

string1 = input("Enter the first string: ")
string2 = input("Enter the second string: ")
print(is_substring(string1, string2))
```

```
Enter the first string: "Hello world"
Enter the second string: "Hello world"
yes
```

```
#Q3. Write a Python program that takes two lists as input and returns
a new listcontaining only the unique elements from both input lists
using comparisonoperators?


def unique_elements(list1, list2):
    # Convert lists to sets to remove duplicates
    set1 = set(list1)
    set2 = set(list2)
    # Combine sets to get unique elements from both lists
    unique_set = set1 | set2

    # Convert the set back to a list
    unique_list = list(unique_set)

    return unique_list

# Example usage
list1 = [1, 2, 3, 4, 5]
list2 = [3, 4, 5, 6, 7]
result = unique_elements(list1, list2)
print("Unique elements from both lists:", result)
```

```
Unique elements from both lists: [1, 2, 3, 4, 5, 6, 7]
```

```
#Q4. Write a Python program that takes a list of strings as input and
returns thelongest string in the list using comparison operators?


def longest_string(strings):
    if not strings:
        return None

    longest = strings[0]
    for string in strings[1:]:
        if len(string) > len(longest):
            longest = string

    return longest

# Example usage

string_list = ["apple", "banana", "orange", "strawberry", "kiwi"]
longest = longest_string(string_list)
print("Longest string in the list:", longest)
```

```
Longest string in the list: strawberry
```

```
#Q5. Write a Python program that takes a set of strings as input and
returns a newset containing only the strings with length less than or
equal to 5?


def filter_strings(input_set):
    filtered_set = set()
    for string in input_set:
        if len(string) <= 5:
            filtered_set.add(string)
    return filtered_set
```

```
# Example usage

input_set = {"apple", "banana", "orange", "strawberry", "kiwi",
"grape"}
filtered_set = filter_strings(input_set)
print("Strings with length less than or equal to 5:", filtered_set)
```

```
Strings with length less than or equal to 5: {'apple', 'grape',
'kiwi'}
```

```
#Q6. Write a Python program that takes a set of numbers as input and
returns anew set containing only the odd numbers in the original set?


def filter_odd_numbers(input_set):
    odd_set = set()
    for number in input_set:
        if number % 2 != 0:
            odd_set.add(number)
    return odd_set


input_set = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
odd_set = filter_odd_numbers(input_set)
print("Odd numbers in the set:", odd_set)
```

```
Odd numbers in the set: {1, 3, 5, 7, 9}
```

```
#Q7. Write a Python program that takes a dictionary as input and
returns the keywith the highest value?


def key_with_highest_value(input_dict):
    if not input_dict:
        return None

    max_key = max(input_dict, key=input_dict.get)
    return max_key


input_dict = {'a': 10, 'b': 20, 'c': 15, 'd': 5}
max_key = key_with_highest_value(input_dict)
print("Key with the highest value:", max_key)
```

```
Key with the highest value: b
```

```
#Q8. Write a Python program that takes two sets as input and returns a
new setcontaining the elements that are only in one of the input sets
using comparison operators?


def elements_in_one_set_only(set1, set2):
    return set1 ^ set2

set1 = {1, 2, 3, 4, 5}
set2 = {4, 5, 6, 7, 8}
result = elements_in_one_set_only(set1, set2)
print("Elements that are only in one of the input sets:", result)
```

```
Elements that are only in one of the input sets: {1, 2, 3, 6, 7, 8}
```

```
#Q9. Write a Python program that takes a list of dictionaries as input
and returns anew list containing only the dictionaries where the value
of a specific key is equalto a certain value using comparison
operators?



def filter_dicts_by_key_value(input_list, key, value):
    filtered_list = []
    for dictionary in input_list:
        if key in dictionary and dictionary[key] == value:
            filtered_list.append(dictionary)
    return filtered_list


list_of_dicts = [
    {'name': 'Alice', 'age': 25},
    {'name': 'Bob', 'age': 30},
    {'name': 'Charlie', 'age': 25},
    {'name': 'David', 'age': 35}
]

key_to_check = 'age'
value_to_match = 25

result = filter_dicts_by_key_value(list_of_dicts, key_to_check,
value_to_match)
print("Dictionaries where the value of '{}' is {}:
{}".format(key_to_check, value_to_match, result))
```

```
Dictionaries where the value of 'age' is 25: [{'name': 'Alice', 'age':
25}, {'name': 'Charlie', 'age': 25}]
```

```
#Q10. Write a Python program that takes a dictionary of strings as
input and returns a new dictionary containing only the key-value pairs
where the value is astring with length greater than 5?



def filter_dict_by_string_length(input_dict):
    filtered_dict = {}
    for key, value in input_dict.items():
        if isinstance(value, str) and len(value) > 5:
            filtered_dict[key] = value
    return filtered_dict


input_dict = {
    'name1': 'abhi',
    'name2': 'umair',
    'name3': 'Christopher',
    'name4': 'Sarah',
    'name5': 'Michael',
    'name6': 'narsingh'
}

result = filter_dict_by_string_length(input_dict)
print("Key-value pairs where the value is a string with length greater
than 5:", result)
```

```
Key-value pairs where the value is a string with length greater than
5: {'name3': 'Christopher', 'name5': 'Michael', 'name6': 'narsingh'}
```

```
#Q11. Write a Python program that takes an integer as input and
outputs the sum of the first n natural numbers using a for loop?


def sum_of_first_n_natural_numbers(n):
    total = 0
    for i in range(1, n + 1):
        total += i
    return total


n = int(input("Enter a positive integer: "))
result = sum_of_first_n_natural_numbers(n)
print("Sum of the first {} natural numbers: {}".format(n, result))
```

```
Enter a positive integer: 5
Sum of the first 5 natural numbers: 15
```

```
#Q12. Write a Python program that takes a list of integers as input
and outputs the second smallest integer in the list using a for loop?


def second_smallest_integer(input_list):
    if len(input_list) < 2:
        return None  # Not enough elements in the list

    smallest = float('inf')
    second_smallest = float('inf')

    for num in input_list:
        if num < smallest:
            second_smallest = smallest
            smallest = num
        elif num < second_smallest and num != smallest:
            second_smallest = num

    return second_smallest


input_list = [5, 3, 1, 7, 9, 2, 8]
result = second_smallest_integer(input_list)
print("Second smallest integer in the list:", result)
```

```
Second smallest integer in the list: 2
```

```
#Q13. Write a Python program that takes a list of strings as input and
outputs the string with the most vowels using a for loop?

def count_vowels(string):
    vowels = "aeiouAEIOU"
    vowel_count = 0
    for char in string:
        if char in vowels:
            vowel_count += 1
    return vowel_count

def string_with_most_vowels(input_list):
    if not input_list:
        return None  # List is empty

    max_vowel_count = 0
    max_vowel_string = ""

    for string in input_list:
        vowel_count = count_vowels(string)
        if vowel_count > max_vowel_count:
            max_vowel_count = vowel_count
            max_vowel_string = string

    return max_vowel_string


input_list = ["hello", "world", "python", "programming", "is",
"awesome"]
result = string_with_most_vowels(input_list)
print("String with the most vowels:", result)
```

```
String with the most vowels: awesome
```

```
#Q14. Write a Python program that takes an integer as input and
determines whether the number is an Armstrong number or not. An
```

```
Armstrong number is anumber that is equal to the sum of its own digits
raised to the power of thenumber of digits example, 153 is an
Armstrong number because 1^3 + 5^3 +3^3 = 153?



def is_armstrong_number(number):
    # Convert the number to a string to count the digits
    num_str = str(number)
    num_digits = len(num_str)

    # Calculate the sum of each digit raised to the power of the total
number of digits
    armstrong_sum = sum(int(digit) ** num_digits for digit in num_str)

    # Check if the sum is equal to the original number
    return armstrong_sum == number


number = int(input("Enter a number: "))
if is_armstrong_number(number):
    print(number, "is an Armstrong number.")
else:
    print(number, "is not an Armstrong number.")
```

```
Enter a number: 150
150 is not an Armstrong number.
```

```
#Q15. Write a program to print first n prime numbers till a given
limit by user. Ex - If the user has given the input as 50 so you need
to show prime numbers till 50?


def is_prime(number):
    if number <= 1:
        return False
    for i in range(2, int(number**0.5) + 1):
        if number % i == 0:
            return False
    return True

def print_primes_up_to_limit(limit, n):
    count = 0
    number = 2
    while count < n:
        if is_prime(number):
            print(number, end=" ")
            count += 1
        number += 1
        if number > limit:
            break



limit = int(input("Enter the limit: "))
n = int(input("Enter the value of n: "))
print("First", n, "prime numbers up to", limit, "are:")
print_primes_up_to_limit(limit, n)
```

```
Enter the limit: 10
Enter the value of n: 5
First 5 prime numbers up to 10 are:
2 3 5 7
```