```
#1.Write a Python function to find maximum and minimum numbers from a given a sequence of numbers.

def maxmin(items):
  max = items[0]
  min = items[0]

  for i in items:
    if i > max:
      max = i

    if i < min:
      min = i

  return max, min
```

```
a = [1,7,2,3,21,14,15,29]
b = maxmin(a)
print(b)
```

```
(29, 1)
```

```
#2.WAP that takes a positive interger and returns the sum of the cube of all the positive integers smaller than the specifi
def sum_of_cubes(n):
  total_sum=0
  for i in range(1,n):
    total_sum += i**3

  return total_sum
n = int(input("Enter postive integer:"))
number = n
result = sum_of_cubes(n)
print(f"The sum of cubes smaller than {number} is: {result}")
```

```
Enter postive integer:4
The sum of cubes smaller than 4 is: 36
```

```
#3.Write a Python function to print 1 to n using recursion. (Note: Do not use loop)
def print_to_n(n):
    # Base case: stop when n reaches 0
    if n == 0:
        return

    # Recursive call first: this goes down to the smallest number (1)
    print_to_n(n - 1)

    # Print after the recursive call to output in ascending order
    print(n, end=' ')

# Example
n = 10
print_to_n(n)
```

```
1 2 3 4 5 6 7 8 9 10
```

```
#4.Write a recursive function to print Fibonacci series upto n terms
def fibonacci_series(n, a=0, b=1):
    if n <= 0:
        return

    # Print the current number in the sequence
    print(a, end=' ')

    # Recursive call:
    # n-1 tracks how many terms are left
    # b becomes the new 'a', and (a+b) becomes the new 'b'
    fibonacci_series(n - 1, b, a + b)

# Example: Print the first 10 terms
n_terms = 10
print(f"Fibonacci series ({n_terms} terms):")
```

```
fibonacci_series(n_terms)



Fibonacci series (10 terms):
0 1 1 2 3 5 8 13 21 34
```

```
#5.Write a lambda function to find volume of cone.
import math

# Lambda function: volume = (1/3) * pi * r^2 * h
cone_volume = lambda r, h: (1/3) * math.pi * (r**2) * h

# Example
radius = 5
height = 12

volume = cone_volume(radius, height)
print(f"The volume of the cone is: {volume:.2f}")
```

```
The volume of the cone is: 314.16
```

```
#6.Write a lambda function which gives tuple of max and min from a list.
#Sample input: [10, 6, 8, 90, 12, 56]
#Sample output: (90,6)

# Lambda function to return (max, min)
get_max_min = lambda data: (max(data), min(data))

# Sample Input
nums = [10, 6, 8, 90, 12, 56]

# Execution
result = get_max_min(nums)
print(result)
```

```
(90, 6)
```

```
#7.Write functions to explain the mentioned concepts:
#a. Keyword argument
#b. Default argument
#c. Variable-length argument

#a.keyword argument
def greet(first_name, last_name):
    print(f"Hello, {first_name} {last_name}!")

# Calling with keyword arguments
greet(last_name="KUMAR", first_name="ROUSHAN")
```

```
Hello, ROUSHAN KUMAR!
```

```
#b.Default argument
def welcome(name, message="Welcome to the team"):
    print(f"Hi {name}, {message}!")

# Uses default message
welcome("Roushan")

# Overrides default message
welcome("John", "Good Morning")
```

```
Hi Roushan, Welcome to the team!
Hi John, Good Morning!
```

```
#c.Variable-length argument
def show_details(*args, **kwargs):
    # args is a tuple of all positional arguments
    print("Positional arguments:", args)

    # kwargs is a dictionary of all named arguments
    print("Keyword arguments:", kwargs)

# Calling with multiple different arguments
show_details(1, 2, 3, city="New York", status="Active")
```

```
Positional arguments: (1, 2, 3)
Keyword arguments: {'city': 'New York', 'status': 'Active'}
```

```
#8.Write a program to check whether all the values in a dictionary are the same or not using a lambda function.
# Lambda function: Checks if the number of unique values is 1
check_same_values = lambda d: len(set(d.values())) <= 1

# Sample Inputs
dict1 = {'a': 10, 'b': 10, 'c': 10}
dict2 = {'a': 10, 'b': 20, 'c': 10}

print(f"Are all values in dict1 same? {check_same_values(dict1)}") # True
print(f"Are all values in dict2 same? {check_same_values(dict2)}") # False
```

```
Are all values in dict1 same? True
Are all values in dict2 same? False
```

```
#9.Write a program to create two lists and generate a dictionary with keys from list1 and values from list2.

# Step 1: Create two lists
list1 = ['name', 'age', 'city']      # These will be the keys
list2 = ['Roushan', 18, 'Dehradun']     # These will be the values

# Step 2: Generate the dictionary
result_dict = dict(zip(list1, list2))

# Display the output
print(result_dict)
```

```
{'name': 'Roushan', 'age': 18, 'city': 'Dehradun'}
```

NAME = VINIT RANJAN

SAP ID = 590026420

GITHUB LINK = https://github.com/vinxtluvvv/python-