

Solution 1

```
def count_vowels_consonants(input_string):
    vowels = "aeiouAEIOU"
    vowel_count = 0
    consonant_count = 0

    for char in input_string:
        if char.isalpha():
            if char in vowels:
                vowel_count += 1
            else:
                consonant_count += 1

    return vowel_count, consonant_count

# Take an input from the user
input_string = input("Enter a string (a sentence): ")

# Count vowels and consonants
vowels, consonants = count_vowels_consonants(input_string)

# Print the result
print(f"The given sentence(string) has Vowels: {vowels} and Consonants: {consonants}")
```

Enter a string (a sentence): It's a bright day
The given sentence(string) has Vowels: 4 and Consonants: 9

Solution 2

```
def min_max_tuple(input_tuple):
    min_element = min(input_tuple)
    max_element = max(input_tuple)
    return min_element, max_element

# Run the logic on a given tuple.
input_tuple = (5, 12, 10, 7, 4)
min_element, max_element = min_max_tuple(input_tuple)
print(f"Minimum element: {min_element}, Maximum element: {max_element}")
```

Minimum element: 4, Maximum element: 12

Solution 2 with user input

```
def min_max_tuple(input_tuple):
    min_element = min(input_tuple)
    max_element = max(input_tuple)
    return min_element, max_element
```

```

# Take the input from the user and convert it to a tuple of integers
user_input = input("Enter a few numbers separated by spaces: ")
input_tuple = tuple(map(int, user_input.split()))

# Get the minimum and maximum values from the given numbers
min_element, max_element = min_max_tuple(input_tuple)

# Print the result
print(f"Minimum element: {min_element}, Maximum element: {max_element}")

```

```

Enter a few numbers separated by spaces: 100 1000 10 1 10000
Minimum element: 1, Maximum element: 10000

```

Solution 3

```

# Defining the function with logic
def unpack_tuple(input_tuple):
    # Unpack the first two elements and the rest
    first, second, *rest = input_tuple
    # This will return the result in the form of a tuple.
    return first, second, rest

# Implementing the logic
input_tuple = (101, 78, 50, 62, 90)
first, second, rest = unpack_tuple(input_tuple)

# Print the results
print(f"First element: {first}, Second element: {second}, Rest of the elements: {rest}")

```

```

First element: 101, Second element: 78, Rest of the elements: [50, 62, 90]

```

Solution 4

```

def word_frequency(input_string):
    # Split the string into words
    sentence = input_string.split()

    # Create a dictionary to store word counts
    frequency = {}

    for word in sentence:
        word = word.lower()
        if word in frequency:
            frequency[word] += 1
        else:
            frequency[word] = 1

```

```

    return frequency

# Take an input sentence from the user.
user_input = input("Enter a sentence: ")

# Get the word frequency
freq = word_frequency(user_input)

# Print the frequency
print("Word frequencies: ")
for word, count in freq.items():
    print(f"{word}: {count}")

Enter a sentence: Hello User Are you a new User
Word frequencies:
hello: 1
user: 2
are: 1
you: 1
a: 1
new: 1

```

Solution 5

```

def find_duplicates(user_list):
    catch = set()
    duplicates = set()

    for item in user_list:
        if item in catch:
            duplicates.add(item)
        else:
            catch.add(item)

    return list(duplicates)

# Take the input from the user
user_input = input("Enter elements(any words) separated by spaces: ")
user_list = user_input.split()

# Find the duplicates
duplicates = find_duplicates(user_list)

# Print the result
if duplicates:
    print("Duplicate elements:", duplicates)
else:
    print("No duplicates found.")

```

```
Enter elements separated by spaces: raw mango raw jackfruit raw guava
Duplicate elements: ['raw']
```

Incase you want the above to run for intergers or float values - (Use this logic : `user_list = list(map(float, user_input.split()))`) It will handle intergers as well as float values.

Solution 6

```
# Ask the user to enter only integers
user_input = input("Enter INTEGERS ONLY, separated by spaces: ")

# Split the input and convert each to int using a list.
user_list = []
for x in user_input.split():
    user_list.append(int(x))

# Use filter with lambda to get even numbers
even_numbers = list(filter(lambda x: x % 2 == 0, user_list))

# Print the result
print("Even numbers in the given list are: ", even_numbers)

Enter INTEGERS ONLY, separated by spaces: 5 7 12 14 10 20 15 30
Even numbers in the given list are: [12, 14, 10, 20, 30]
```

Solution 7

```
%%writefile mymath.py

def add(a, b):
    return a + b

def subtract(a, b):
    return a - b

def multiply(a, b):
    return a * b

Writing mymath.py

import mymath

# Using the module
a = int(input("Enter the first number: "))
b = int(input("Enter the second number: "))

print("Addition:", mymath.add(a, b))
print("Subtraction:", mymath.subtract(a, b))
print("Multiplication:", mymath.multiply(a, b))
```

```
Enter the first number: 5
Enter the second number: 7
Addition: 12
Subtraction: -2
Multiplication: 35
```

Solution 8

```
# A list of tuples
my_list = [(5, 2), (7, 1), (4, 3), (8, 4)]

''' Sort by the second element of each tuple using a lambda function
sorted() is a built-in Python function that returns a new sorted
list.
It does not change the original list, it returns a new one.
key is a special argument that tells Python how to sort the items.
For each element x in the list (x is a tuple), use the second element
(x[1]) as the value to sort by.'''
sorted_list = sorted(my_list, key=lambda x: x[1])

# Print the result
print("Sorted list based on second element in each tuple:",
sorted_list)

Sorted list based on second element in each tuple: [(7, 1), (5, 2),
(4, 3), (8, 4)]
```

Solution 9

```
#The symmetric difference between two sets means the elements that are
in either one of the sets but not in both.
def symmetric_difference(set1, set2):
    # ^ operator in Python is a set operator that computes the
    symmetric difference between two sets.
    return set1 ^ set2

# Take inputs from the user.
input1 = input("Enter elements of first set with space in between: ")
input2 = input("Enter elements of second set with space in between: ")

# Convert inputs to sets
set1 = set(input1.split())
set2 = set(input2.split())

# Find symmetric difference
result = symmetric_difference(set1, set2)

# Print the final result
print("Symmetric difference:", result)
```

```
Enter elements of first set with space in between: 2 5 7 9 10 11 15
Enter elements of second set with space in between: 5 11 25 66 80 12
60
Symmetric difference: {'60', '10', '12', '80', '2', '15', '7', '9',
'25', '66'}
```

Solution 10

```
def getprimes_upto_n(n):
    primenums = []

    for num in range(2, n + 1):
        is_prime = True

        # Check if num is divisible by any number from 2 to num-1
        for i in range(2, num):
            if num % i == 0:
                is_prime = False
                break

        if is_prime:
            primenums.append(num)

    return primenums

n = int(input("Enter a number of your choice: "))
print("Prime numbers up to", n, "are: ", getprimes_upto_n(n))

Enter a number of your choice: 12
Prime numbers up to 12 are: [2, 3, 5, 7, 11]
```