Sheet 1

1. Write a short Python function, is even(k), that takes an integer value and returns True if k is even, and False otherwise.

Solution

```
def is_even(k):
    return k % 2 == 0

number = int(input("Enter an integer: "))

if is_even(number):
    print(f"{number} is even.")
else:
    print(f"{number} is odd.")
```

Sheet 1

2. Write a short Python function, minmax(data), that takes a sequence of one or more numbers, and returns the smallest and largest numbers, in the form of a tuple of length two. Do not use the built-in functions min or max in implementing your solution.

Solution

```
def minmax(data):
    smallest = largest = data[0]

    for num in data:
        if num < smallest:
            smallest = num
        if num > largest:
            largest = num

    return (smallest, largest)


# Test with a list of numbers
numbers = [3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5]
result = minmax(numbers)
print(f"Smallest: {result[0]}, Largest: {result[1]}")

# Test with a tuple of numbers
numbers_tuple = (10, -2, 8, 15, 7)
result = minmax(numbers_tuple)
print(f"Smallest: {result[0]}, Largest: {result[1]}")

# Test with a single number
single_number = [42]
result = minmax(single_number)
print(f"Smallest: {result[0]}, Largest: {result[1]}")
```

Sheet 1

3. Write a short Python function that takes a positive integer n and returns the sum of the squares of all the positive integers smaller than n.

Solution

```
def sum_of_squares(n):
    total = 0

    for i in range(1, n):
        total += i * i

    return total


n = int(input("Please Enter any number : "))
result = sum_of_squares(n)
print(f"The sum of squares of all positive integers smaller than {n}
is: {result}")
```

4. Python allows negative integers to be used as indices into a sequence, such as a string. If string s has length n, and expression s[k] is used for index $-n \leq k < 0$, what is the equivalent index $j \geq 0$ such that s[j] references the same element?

Solution

```
s = "Python"
negativeIndex = -2

# Calculate the equivalent non-negative index
length = len(s)
PositiveIndex = negativeIndex + length

# Access the element using the non-negative index
print(f"Negative index {negativeIndex} corresponds to non-negative
index {PositiveIndex}, which is the element '{s[negativeIndex]}'")
```

Sheet 1

5. What parameters should be sent to the range constructor, to produce a range with values 50, 60, 70, 80?

Solution

```
# Create a range object
#        (stard,end,step)
r = range(50, 81, 10)

# Convert the range to a list to see the values
values = list(r)
print(values)
```

6. Demonstrate how to use Python's list comprehension syntax to produce the list [1, 2, 4, 8, 16, 32, 64, 128, 256].

Solution

```
# Using list comprehension to generate the list
result = [2 ** n for n in range(9)]

# Print the result
print(result)
```

Sheet 1

7. Write a Python function that takes a sequence of numbers and determines if all the numbers are different from each other (that is, they are distinct).

Solution

```
def all_distinct(sequence):
    return len(sequence) == len(set(sequence))


test1 = [1, 2, 3, 4, 5]  # All distinct
test2 = [1, 2, 3, 2, 4]  # Contains duplicates
test3 = (10, 20, 30, 40)  # All distinct (tuple)
test4 = []  # Empty sequence (considered distinct)

print(all_distinct(test1))
print(all_distinct(test2))
print(all_distinct(test3))
print(all_distinct(test4))
```

8. Write a short Python function that counts the number of vowels in a given character string.

Solution

```
def count_vowels(s):
    vowels = "aeiouAEIOU"
    count = 0

    for char in s:
        if char in vowels:
            count += 1

    return count


input_string = input("Enter any Message : ")
result = count_vowels(input_string)
   print(f"The number of vowels in '{input_string}' is: {result}")
```