Multiple Choice Questions (MCQs):

Which of the following is a correct data type in Python?

a) Int

b) Real

c) Character

d) List

What will be the output of the following code snippet?

python

Copy code

x = 5

y = 2

print(x \*\* y)

a) 10

b) 25

c) 32

d) 2.5

Which operator is used for floor division in Python?

a) /

b) //

c) %

d) \*\*

Which of the following is not a valid variable name in Python?

a) my\_var

b) \_var

c) 1var

d) Var1

What is the output of the following code snippet?

s = "Hello, World!"

print(s[3:8])

a) Hello

b) lo, W

c) lo, Wo

d) lo, Wor

Which of the following is an immutable data type in Python?

a) List

b) Tuple

c) Dictionary

d) Set

What is the purpose of the enumerate() function in Python?

a) To generate a sequence of numbers.

b) To iterate over the elements of a sequence while keeping track of the index.

c) To combine multiple sequences into a single sequence.

d) To create a dictionary from two sequences.

Which operator is used to check if an element is present in a sequence in Python?

a) in

b) is

c) ==

d) !=

What is the output of the following code snippet?

x = [1, 2, 3]

y = x

x[0] = 0

print(y)

a) [1, 2, 3]

b) [0, 2, 3]

c) [0, 2, 3, 4]

d) [0, 0, 0]

Which of the following statements about Python sets is true?

a) Sets preserve the order of elements.

b) Sets can contain duplicate elements.

c) Sets are mutable.

d) Sets are implemented using square brackets [].

Exercises:

**Write a Python program to calculate the area of a rectangle using user input for length and width.**

length = float(input("enter the length of the rectangle"))

width = float(input("enter the width of the rectangle"))

area = length \* width

print(area)

output:

enter the length of the rectangle20

enter the width of the rectangle15

300.0

**Write a Python program to find the maximum of three numbers using conditional statements.**

num1 = input("Enter first number: ")

num2 = input("Enter second number: ")

num3 = input("Enter third number: ")

if num1 >= num2 and num1 >= num3:

max\_num = num1

elif num2 >= num1 and num2 >= num3:

max\_num = num2

else:

max\_num = num3

print("The maximum of the three numbers is:", max\_num)

output:

Enter first number: 40

Enter second number: 50

Enter third number: 45

The maximum of the three numbers is: 50

**Write a Python program to swap the values of two variables without using a temporary variable**.

a = 30

b = 60

print("Before swapping:")

print("a =", a)

print("b =", b)

a, b = b, a

print("After swapping:")

print("a =", a)

print("b =", b)

output:

Before swapping:

a = 30

b = 60

After swapping:

a = 60

b = 30

**Write a Python program to convert temperature from Celsius to Fahrenheit and vice versa using functions.**

def celsius\_to\_fahrenheit(celsius):

"""

Convert temperature from Celsius to Fahrenheit.

Formula: (Celsius \* 9/5) + 32

"""

fahrenheit = (celsius \* 9/5) + 32

return fahrenheit

def fahrenheit\_to\_celsius(fahrenheit):

"""

Convert temperature from Fahrenheit to Celsius.

Formula: (Fahrenheit - 32) \* 5/9

"""

celsius = (fahrenheit - 32) \* 5/9

return celsius

celsius\_temp = 30

fahrenheit\_temp = 86

print("Celsius to Fahrenheit:")

print(f"{celsius\_temp}°C = {celsius\_to\_fahrenheit(celsius\_temp)}°F")

print("\nFahrenheit to Celsius:")

print(f"{fahrenheit\_temp}°F = {fahrenheit\_to\_celsius(fahrenheit\_temp)}°C")

output:

Celsius to Fahrenheit:

30°C = 86.0°F

Fahrenheit to Celsius:

86°F = 30.0°C

**Write a Python program to count the number of vowels in a given string**

def count\_vowels(string):

"""

Count the number of vowels in a given string.

"""

vowels = "aeiouAEIOU" # Define the set of vowels

vowel\_count = 0

for char in string:

if char in vowels:

vowel\_count += 1

return vowel\_count

input\_string = input("Enter a string: ")

print("Number of vowels in the string:", count\_vowels(input\_string))

output:

Enter a string: sasikanth reddy

Number of vowels in the string: 4

**Write a Python program to check if a given number is prime or not.**

def is\_prime(number):

"""

Check if a given number is prime.

"""

if number <= 1:

return False

elif number <= 3:

return True

elif number % 2 == 0 or number % 3 == 0:

return False

i = 5

while i \* i <= number:

if number % i == 0 or number % (i + 2) == 0:

return False

i += 6

return True

num = int(input("Enter a number: "))

if is\_prime(num):

print(num, "is a prime number")

else:

print(num, "is not a prime number")

output:

Enter a number: 7

7 is a prime number

**Write a Python program to find the factorial of a given number using recursion.**

def factorial(n):

"""

Calculate the factorial of a given number using recursion.

"""

if n == 0:

return 1

else:

return n \* factorial(n - 1)

num = int(input("Enter a number: "))

if num < 0:

print("Factorial is not defined for negative numbers.")

else:

print("Factorial of", num, "is", factorial(num))

output:

Enter a number: 5

Factorial of 5 is 120

**Write a Python program to generate the Fibonacci sequence up to a certain number of terms.**

def generate\_fibonacci(n):

"""

Generate the Fibonacci sequence up to n terms.

"""

fibonacci\_sequence = [0, 1]

while len(fibonacci\_sequence) < n:

next\_fibonacci = fibonacci\_sequence[-1] + fibonacci\_sequence[-2]

fibonacci\_sequence.append(next\_fibonacci)

return fibonacci\_sequence

num\_terms = int(input("Enter the number of terms for Fibonacci sequence: "))

if num\_terms <= 0:

print("Please enter a positive integer.")

else:

fibonacci\_sequence = generate\_fibonacci(num\_terms)

print("Fibonacci sequence up to", num\_terms, "terms:")

print(fibonacci\_sequence)

output:

Enter the number of terms for Fibonacci sequence: 8

Fibonacci sequence up to 8 terms:

[0, 1, 1, 2, 3, 5, 8, 13]

**Write a Python program to remove duplicates from a list.**

def remove\_duplicates(input\_list):

"""

Remove duplicates from a list.

"""

return list(set(input\_list))

input\_list = [1, 2, 3, 4, 5, 2, 1, 5, 6, 5, 7]

print("Original list:", input\_list)

print("List after removing duplicates:", remove\_duplicates(input\_list))

output:

Original list: [1, 2, 3, 4, 5, 2, 1, 5, 6, 5, 7]

List after removing duplicates: [1, 2, 3, 4, 5, 6, 7]

**Write a Python program to find the intersection of two lists.**

def find\_intersection(list1, list2):

"""

Find the intersection of two lists.

"""

set1 = set(list1)

set2 = set(list2)

return list(set1.intersection(set2))

list1 = [1, 2, 3, 4, 5]

list2 = [3, 4, 5, 6, 7]

print("List 1:", list1)

print("List 2:", list2)

print("Intersection:", find\_intersection(list1, list2))

output:

List 1: [1, 2, 3, 4, 5]

List 2: [3, 4, 5, 6, 7]

Intersection: [3, 4, 5]

**Write a Python program to find the longest word in a given list of words.**

def find\_longest\_word(words):

longest\_word = ""

for word in words:

if len(word) > len(longest\_word):

longest\_word = word

return longest\_word

word\_list = ["apple", "banana", "orange", "strawberry", "kiwi"]

longest = find\_longest\_word(word\_list)

print("The longest word is:", longest)

output:

The longest word is: strawberry

**Write a Python program to count the occurrences of each word in a given string.**

def count\_word\_occurrences(text):

text = text.lower()

punctuation\_marks = '''!()-[]{};:'"\,<>./?@#$%^&\*\_~'''

for char in punctuation\_marks:

text = text.replace(char, " ")

words = text.split()

word\_count = {}

for word in words:

if word in word\_count:

word\_count[word] += 1

else:

word\_count[word] = 1

return word\_count

input\_text = "This is a sample text. This text contains some words, and some words may repeat."

word\_occurrences = count\_word\_occurrences(input\_text)

print("Word occurrences:")

for word, count in word\_occurrences.items():

print(f"{word}: {count}")

output:

Word occurrences:

this: 2

is: 1

a: 1

sample: 1

text: 2

contains: 1

some: 2

words: 2

and: 1

may: 1

repeat: 1

**Write a Python program to reverse a given string.**

def reverse\_string(input\_string):

return input\_string[::-1]

input\_string = "Hello, world!"

reversed\_string = reverse\_string(input\_string)

print("Reversed string:", reversed\_string)

output:

Reversed string: !dlrow ,olleH

**Write a Python program to sort a list of tuples based on the second element of each tuple**

def sort\_tuples\_by\_second\_element(tuple\_list):

sorted\_tuples = sorted(tuple\_list, key=lambda x: x[1])

return sorted\_tuples

tuple\_list = [(1, 5), (2, 3), (3, 8), (4, 1), (5, 7)]

sorted\_tuples = sort\_tuples\_by\_second\_element(tuple\_list)

print("Sorted list of tuples:", sorted\_tuples)

output:

Sorted list of tuples: [(4, 1), (2, 3), (1, 5), (5, 7), (3, 8)]

**Write a Python program to find the sum of all elements in a list using a loop.**

def sum\_of\_elements(input\_list):

total = 0

for element in input\_list:

total += element

return total

my\_list = [1, 2, 3, 4, 5]

total\_sum = sum\_of\_elements(my\_list)

print("Sum of elements:", total\_sum)

output:

Sum of elements: 15

**Write a Python program to remove the last element from a list.**

def remove\_last\_element(input\_list):

if input\_list

input\_list.pop()

my\_list = [1, 2, 3, 4, 5]

remove\_last\_element(my\_list)

print("List after removing the last element:", my\_list)

output:

List after removing the last element: [1, 2, 3, 4]

**Write a Python program to check if a given string is a palindrome**

def is\_palindrome(input\_string):

processed\_string = input\_string.replace(" ", "").lower()

return processed\_string == processed\_string[::-1]

test\_string = "A man, a plan, a canal, Panama"

result = is\_palindrome(test\_string)

if result:

print("The string is a palindrome.")

else:

print("The string is not a palindrome.")

output:

The string is not a palindrome.

**Write a Python program to find the common characters between two strings.**

def common\_characters(str1, str2):

set1 = set(str1)

set2 = set(str2)

common\_chars = set1.intersection(set2)

return common\_chars

string1 = "hello"

string2 = "world"

common\_chars = common\_characters(string1, string2)

print("Common characters:", common\_chars)

output:

Common characters: {'o', 'l'}

**Write a Python program to find the length of the longest consecutive sequence of a given list of integers**

def longest\_consecutive\_sequence(nums):

num\_set = set(nums)

max\_length = 0

for num in num\_set:

if num - 1 not in num\_set:

current\_num = num

current\_length = 1

while current\_num + 1 in num\_set:

current\_num += 1

current\_length += 1

max\_length = max(max\_length, current\_length)

return max\_length

numbers = [100, 4, 200, 1, 3, 2, 5, 101, 102, 103]

longest\_sequence\_length = longest\_consecutive\_sequence(numbers)

print("Length of the longest consecutive sequence:", longest\_sequence\_length)

output:

Length of the longest consecutive sequence: 5

**Write a Python program to find the difference between two sets.**

def set\_difference(set1, set2):

return set1 - set2

set1 = {1, 2, 3, 4, 5}

set2 = {3, 4, 5, 6, 7}

difference = set\_difference(set1, set2)

print("Difference between set1 and set2:", difference)

output:

Difference between set1 and set2: {1, 2}