PYTH ON

# ASSIGNMENT BOOK



**TASK THREE**

**DATA STRUCTURES**

1. Create a list of 10 elements of four different data types like int, string, complex and float.

my\_list = [1, 'hello', 3+4j, 2.5, 'world', 6+2j, 3.7, 10, 'python', 5.2]

print(my\_list)

1. Create a list of size 5 and execute the slicing structure

# Create a list of size 5

my\_list = [10, 20, 30, 40, 50]

# Perform slicing

sliced\_list = my\_list[1:4] # Slicing from index 1 to 4 (exclusive)

# Print the sliced list

print("Sliced List:", sliced\_list)

#output: Sliced List: [20, 30, 40]

1. Write a program to get the sum and multiply of all the items in a given list.

# Get input from the user

input\_str = input("Enter a list of numbers, separated by spaces: ")

# Convert the input string into a list of numbers

num\_list = [float(num) for num in input\_str.split()]

# Calculate the sum and product of the numbers in the list

sum\_list = sum(num\_list)

product\_list = 1

for num in num\_list:

product\_list \*= num

# Print the sum and product

print("Sum:", sum\_list)

print("Product:", product\_list)

1. Find the largest and smallest number from a given list.

# Get input from the user

input\_str = input("Enter a list of numbers, separated by spaces: ")

# Convert the input string into a list of numbers

num\_list = [float(num) for num in input\_str.split()]

# Find the largest and smallest numbers

largest\_num = max(num\_list)

smallest\_num = min(num\_list)

# Print the largest and smallest numbers

print("Largest number:", largest\_num)

print("Smallest number:", smallest\_num)

1. Create a new list which contains the specified numbers after removing the even numbers from a

predefined list.

predefined\_list = [1, 2, 3, 4, 5, 6, 7, 8, 9]

# Create a new list without even numbers

new\_list = [num for num in predefined\_list if num % 2 != 0]

print("New List:", new\_list)

1. Create a list of elements such that it contains the squares of the first and last 5 elements between 1 and30(both included).

squares\_list = [num \*\* 2 for num in range(1, 31) if num <= 5 or num >= 26]

print("Squares List:", squares\_list)

putput: Squares List: [1, 4, 9, 16, 25, 676, 729, 784, 841, 900]

1. Write a program to replace the last element in a list with another list.

**Sample input:** [1,3,5,7,9,10], [2,4,6,8]

**Expected output:** [1,3,5,7,9,2,4,6,8]

list1 = [1, 3, 5, 7, 9, 10]

list2 = [2, 4, 6, 8]

# Replace the last element of list1 with list2

list1[-1:] = list2

print("Updated List:", list1)

1. Create a new dictionary by concatenating the following two dictionaries:

**Sample input:** a={1:10,2:20} b={3:30,4:40}

**Expected output:** {1:10,2:20,3:30,4:40}

a = {1: 10, 2: 20}

b = {3: 30, 4: 40}

# Create a new dictionary by concatenating dictionaries a and b

new\_dict = {\*\*a, \*\*b}

print("New Dictionary:", new\_dict)

1. Create a dictionary that contain numbers in the form(x:x\*x) where x takes all the values between 1 andn(both 1 and n included).

**Sample input:** n=5

**Expected output:** {1:1, 2:4, 3:9, 4:16, 5:25}

n = 5

# Create the dictionary with numbers and their squares

result\_dict = {x: x \* x for x in range(1, n+1)}

print("Resulting Dictionary:", result\_dict)

1. Write a program which accepts a sequence of comma-separated numbers from console and generates a list and a tuple which contains every number in the form of string.

**Sample input:** 34,67,55,33,12,98

**Expected output:** [‘34’,’67’,’55’,’33’,’12’,’98’] (‘34’,’67’,’55’,’33’,’12’,’98’)

numbers = input("Enter comma-separated numbers: ")

# Split the input string into a list of numbers

number\_list = numbers.split(',')

# Create a tuple by converting each number to a string

number\_tuple = tuple(number\_list)

print("List:", number\_list)

print("Tuple:", number\_tuple)

**TASK FOUR**

**TRADITIONAL FUNCTIONS,ANONYMOUS FUNCTIONS & HIGHER**

**ORDER FUNCTIONS**

1. Write a program to reverse a string.

**Sample input:** “1234abcd”

**Expected output:** “dcba4321”

input\_string = "1234abcd"

# Reverse the string using slicing

reversed\_string = input\_string[::-1]

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

1. Write a function that accepts a string and prints the number of uppercase letters and lowercase letters.

**Sample input:** “abcSdefPghijQkl”

**Expected Output:** No. of Uppercase characters : 3 No. of Lower case Characters : 12

def count\_letters(string):

uppercase\_count = 0

lowercase\_count = 0

for char in string:

if char.isupper():

uppercase\_count += 1

elif char.islower():

lowercase\_count += 1

print("Number of Uppercase Letters:", uppercase\_count)

print("Number of Lowercase Letters:", lowercase\_count)

input\_string = "Hello World"

count\_letters(input\_string)

1. Create a function that takes a list and returns a new list with unique elements of the first list.

def get\_unique\_elements(lst):

unique\_list = list(set(lst))

return unique\_list

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

# Call the function to get the unique elements

result\_list = get\_unique\_elements(input\_list)

print("Unique Elements:", result\_list)

1. Write a program that accepts a hyphen-separated sequence of words as input and prints the words in ahyphen-separated sequence after sorting them alphabetically.

input\_sequence = input("Enter a hyphen-separated sequence of words: ")

# Split the input sequence into a list of words

words\_list = input\_sequence.split('-')

# Sort the list of words alphabetically

sorted\_words = sorted(words\_list)

# Join the sorted words with hyphens

sorted\_sequence = '-'.join(sorted\_words)

print("Sorted Sequence:", sorted\_sequence)

1. Write a program that accepts a sequence of lines as input and prints the lines after making all charactersin the sentence capitalized.

**Sample input:** Hello world Practice makes man perfect

**Expected output:** HELLO WORLD PRACTICE MAKES MAN PERFECT

# Accept a sequence of lines from the user

input\_lines = input("Enter a sequence of lines: ")

# Split the input lines into a list of lines

lines\_list = input\_lines.splitlines()

# Capitalize each line and create a new list of capitalized lines

capitalized\_lines = [line.upper() for line in lines\_list]

# Join the capitalized lines into a single string with line breaks

output\_text = '\n'.join(capitalized\_lines)

# Print the resulting lines with capitalized characters

print("Capitalized Lines:")

print(output\_text)

1. Define a function that can receive two integral numbers in string form and compute their sum and print it inthe console.

def compute\_sum(num1\_str, num2\_str):

num1 = int(num1\_str)

num2 = int(num2\_str)

sum\_result = num1 + num2

print("Sum:", sum\_result)

# Example usage of the function

num1\_str = input("Enter the first number: ")

num2\_str = input("Enter the second number: ")

compute\_sum(num1\_str, num2\_str)

1. Define a function that can accept two strings as input and print the string with the maximum length in theconsole. If two strings have the same length, then the function should print both the strings line by line.

def print\_string\_with\_max\_length(str1, str2):

if len(str1) > len(str2):

print(str1)

elif len(str1) < len(str2):

print(str2)

else:

print(str1)

print(str2)

string1 = input("Enter the first string: ")

string2 = input("Enter the second string: ")

print\_string\_with\_max\_length(string1, string2)

1. Define a function which can generate and print a tuple where the values are square of numbers between 1and 20 (both 1 and 20 included).

def generate\_square\_tuple():

square\_tuple = tuple(i\*\*2 for i in range(1, 21))

print(square\_tuple)

generate\_square\_tuple()

1. Write a function called showNumbers that takes a parameter called limit. It should print all the numbersbetween 0 and limit with a label to identify the even and odd numbers.

**Sample input:** show Numbers(3) (where limit=3)

**Expected output:**

1. EVEN
2. ODD
3. EVEN
4. ODD

def showNumbers(limit):

for num in range(limit + 1):

if num % 2 == 0:

print(num, "EVEN")

else:

print(num, "ODD")

# Example usage of the function

showNumbers(3)

1. Write a program which uses filter() to make a list whose elements are even numbers between 1 and 20(both included)

even\_numbers = list(filter(lambda x: x % 2 == 0, range(1, 21)))

print(even\_numbers)

1. Write a program which uses map() and filter() to make a list whose elements are squares of even numbersin [1,2,3,4,5,6,7,8,9,10].

Hints: Use filter() to filter even elements of the given listUse map() to generate a list of squares of the numbers in the filtered list. Use lambda() to define anonymous functions.

numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

even\_numbers = list(filter(lambda x: x % 2 == 0, numbers))

squared\_numbers = list(map(lambda x: x \*\* 2, even\_numbers))

print(squared\_numbers)

1. Write a function to compute 5/0 and use try/except to catch the exceptions.

def divide\_by\_zero():

try:

result = 5 / 0

print("Result:", result)

except ZeroDivisionError:

print("Error: Division by zero is not allowed!")

# Call the function

divide\_by\_zero()

1. Flatten the list [1,2,3,4,5,6,7] into 1234567 using reduce().

from functools import reduce

numbers = [1, 2, 3, 4, 5, 6, 7]

result = reduce(lambda x, y: 10 \* x + y, numbers)

print(result)

1. Write a program in Python to find the values which are not divisible by 3 but are a multiple of 7. Makesure to use only higher order functions.

numbers = range(1, 101) # Assume the range of numbers is from 1 to 100

result = list(filter(lambda x: x % 3 != 0 and x % 7 == 0, numbers))

print(result)

1. Write a program in Python to multiply the elements of a list by itself using a traditional function and passthe function to map() to complete the operation.

def multiply\_by\_self(num):

return num \* num

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

result = list(map(multiply\_by\_self, numbers))

print(result)

1. What is the output of the following codes:

**(i)** def foo(): try: return 1

finally:

return 2 k = foo() print(k)

(ii) def a():

try:

f(x, 4)

finally:

print('after f')

print('after f?')

a()