1

```
# Write a program to demonstrate various tuple operations in python.
# Creating tuples
t1 = (1, 2, 3)
t2 = ('apple', 'banana', 'cherry')
t3 = (1.23, 'hello', True)
print(t1[0])
print(t2[-1])
print(t1[1:3])
print(t2[:2])
print(t3[1:])
t4 = t1 + t2
print(t4)
t5 = t3 * 2
print(t5)
print(1 in t1)
print('orange' not in t2)
print(len(t3))
```

```
# Write a program to find the size of a tuple
my_tuple = (1, 2, 3, 'hello', 'world')
size = len(my_tuple)
print(f"The size of the tuple is {size}")
```

```
# Write a program to find the maximum and minimum K elements in a tuple
my_tuple = (3, 7, 1, 9, 2, 6, 5)
K = 3
max_elements = sorted(my_tuple, reverse=True)[:K]
min_elements = sorted(my_tuple)[:K]
print(f"The maximum {K} elements in the tuple are: {max_elements}")
print(f"The minimum {K} elements in the tuple are: {min_elements}")
```

```
# Write a program to create a list of tuples from given list having number and its cube in each tuple
numbers = [2, 3, 4, 5, 6]
cubes = [(x, x ** 3) for x in numbers]
print(cubes)
```

2

```
# Write a program to demonstrate working with dictionaries in python
# DICTIONARY
print("******")
print("DICTIONARY")
dict1 = {1: "Abc", 2: "Def", 4: "xyz"}
print(dict1)
print(dict1[4])
# print(dict[3])   ----------> gives error
print(dict1.get(4))
print(dict1.get(3))
print(dict1.get(3, "NotFound"))
print(dict1)

# Important
keys = ['abb', 'bcc', 'cdd']
values = ['Python', 'Java', 'C++']
data = dict(zip(keys, values))
print(data)
del data['abb']
print(data)

prog = {'JS': 'Atom', 'CS': 'V', 'Python': ['Pycharm', 'Sublime'], 'Java': {'JSE': 'NetBeans', 'JEE':
'Eclipse'}}
print(prog)
a = {'C++': 'VSCode'}
prog.update(a)
print(prog)
prog.update({'C': 'NetBeans'})
print(prog)




# Dictionary Student:
print("*****Student dictionary*****")
student_dict = {
    "name": "Alice",
    "age": 21,
    "major": "Computer Science",
    "grades": [95, 85, 90, 92],
}
print(f"Name: {student_dict['name']}")
print(f"Age: {student_dict['age']}")
print(f"Major: {student_dict['major']}")
print(f"Grades: {student_dict['grades']}")
student_dict["gender"] = "Female"
student_dict["major"] = "Data Science"
del student_dict["grades"]
for key, value in student_dict.items():
    print(f"{key}: {value}")


# Write a Program to create a dictionary from a sequence
seq = [1, 2, 3, 4, 5]
seq_dict = {x: x**2 for x in seq}
print(seq_dict)
```

3

```
# Write a Program to generate dictionary of numbers and their squares (i, i*i) from 1 to N
N = int(input("Enter a positive integer N: "))
squares_dict = {i: i*i for i in range(1, N+1)}
print(squares_dict)
```

```
# Write a Program that determines and displays the number of unique characters in a string
# entered by the user. For example, “Hello, World!” has 10 unique characters while “zzz”
# has only one unique character. Use a dictionary to solve this problem.
# Prompt the user to enter a string
string = input("Enter a string: ")
char_dict = {}
for char in string:
    if char not in char_dict:
        char_dict[char] = 1
num_unique_chars = len(char_dict)
print("The string", string, "has", num_unique_chars, "unique characters.")
```

```
# Two words are anagrams if they contain all of the same letters, but in a different order.
# For example, “evil” and “live” are anagrams because each contains one ‘e’, one ‘I’, one ‘l’, and
one ‘v’.
# Create a program that reads two strings from the user, determines
# whether or not they are anagrams, and reports the result.
string1 = input("Enter the first string: ")
string2 = input("Enter the second string: ")

string1 = string1.lower().replace(" ", "")
string2 = string2.lower().replace(" ", "")

if sorted(string1) == sorted(string2):
    print("The strings", string1, "and", string2, "are anagrams.")
else:
    print("The strings", string1, "and", string2, "are not anagrams.")
```

```
# Write a program to replace list’s item with new value if found
my_list = [1, 2, 3, 4, 5, 6]
old_value = 3
new_value = 7

for i in range(len(my_list)):
    if my_list[i] == old_value:
        my_list[i] = new_value

print(my_list)
```

```
#Write a program that reads integers from the user and stores them in a list.
# Your program should continue reading values until the user enters 0.
# Then it should display all of the values entered by the user
# (except for the 0) in order from smallest to largest, with one value appearing on each line.
numbers = []
while True:
    num = int(input("Enter an integer (0 to stop): "))
    if num == 0:
        break
    numbers.append(num)

numbers.sort()
for num in numbers:
    print(num)
```

```
#Write a program that reads integers from the user and stores them in a list.
# Use 0 as a sentinel value to mark the end of the input.
# Once all of the values have been read your program should display them (except for the 0) in
reverse order.
numbers = []
while True:
    num = int(input("Enter an integer (0 to stop): "))
    if num == 0:
        break
    numbers.append(num)

numbers.reverse()
for num in numbers:
    print(num)
```

```
# Write a Program to write user defined function to swap two number and
# display number before swapping and after swapping



def swap_numbers(a, b):
    print("Before swapping: a =", a, "b =", b)
    a, b = b, a
    print("After swapping: a =", a, "b =", b)
num1 = int(input("Enter the first number: "))
num2 = int(input("Enter the second number: "))
swap_numbers(num1, num2)



# Write a Program to calculate arithmetic operation on two number using user defined function
def perform_arithmetic(num1, num2, operation):
    if operation == "+":
        result = num1 + num2
    elif operation == "-":
        result = num1 - num2
    elif operation == "*":
        result = num1 * num2
    elif operation == "/":
        result = num1 / num2
    else:
        print("Invalid operation")
        return

    print("Result of", num1, operation, num2, "=", result)

num1 = float(input("Enter the first number: "))
num2 = float(input("Enter the second number: "))
operation = input("Enter an arithmetic operation (+, -, *, /): ")
perform_arithmetic(num1, num2, operation)



# Write a Program to Calculate diameter and area of circle using user defined function
import math

def calculate_circle_properties(radius):
    diameter = 2 * radius
    area = math.pi * radius ** 2
    return diameter, area

# Prompt the user to enter the radius of the circle
radius = float(input("Enter the radius of the circle: "))

# Call the calculate_circle_properties function with the radius as argument
diameter, area = calculate_circle_properties(radius)

# Display the calculated diameter and area
print("The diameter of the circle is:", diameter)
print("The area of the circle is:", area)
```

```
# Write a function that takes three numbers as parameters, and
# returns the median value of those parameters as its result. Include a main program that reads
three values from
# the user and displays their median.
def calculate_median(num1, num2, num3):
    sorted_nums = sorted([num1, num2, num3])
    return sorted_nums[1]

num1 = float(input("Enter the first number: "))
num2 = float(input("Enter the second number: "))
num3 = float(input("Enter the third number: "))

median = calculate_median(num1, num2, num3)
print("The median of the three numbers is:", median)
```

```
# Write a function that generates a random password. The password should have a random
# length of between 7 and 10 characters. Each character should be randomly selected from
# positions 33 to 126 in the ASCII table. Your function will not take any parameters.
# It will return the randomly generated password as its only result.
import random

def generate_password():
    length = random.randint(7, 10)
    password = ""
    for i in range(length):
        ascii_val = random.randint(33, 126)
        password += chr(ascii_val)
    return password
print(generate_password())
```

```
# Write a Program to filter even values from list using lambda function
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
even_numbers = list(filter(lambda x: x % 2 == 0, numbers))
print(even_numbers)
```

```
# Write a Program to find the sum of elements of a list using lambda function
from functools import reduce
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
total = reduce(lambda x, y: x + y, numbers)
print(total)

# Write a Program to find small number between two numbers using Lambda function
smaller = lambda x, y: x if x < y else y

num1 = float(input("Enter the first number: "))
num2 = float(input("Enter the second number: "))

result = smaller(num1, num2)

print("The smaller number is:", result)
```

```
# Write a Program to implement destructor and constructors using _del() and __init_()
class MyClass:
    def _init_(self):
        print("Constructor called")

    def _del_(self):
        print("Destructor called")

obj = MyClass()
del obj


# Write a Program to calculate student grade using class
class Student:
    def _init_(self, name, marks):
        self.name = name
        self.marks = marks

    def calculate_grade(self):
        total_marks = sum(self.marks)
        average_marks = total_marks / len(self.marks)

        if average_marks >= 90:
            return "A+"
        elif average_marks >= 80:
            return "A"
        elif average_marks >= 70:
            return "B"
        elif average_marks >= 60:
            return "C"
        else:
            return "F"

name = input("Enter student name: ")
marks = []

for i in range(5):
    mark = int(input(f"Enter mark {i+1}: "))
    marks.append(mark)

student = Student(name, marks)
grade = student.calculate_grade()

print(f"{student.name}'s grade is {grade}")
```