ASSIGNMENT 05

```
# 1. Write a Python program to create a tuple containing numbers from 1 to 10. Print
# the first element, the last element, and the middle element.
numbers = tuple(range(1, 11))
print("First element:", numbers[0])
print("Last element:", numbers[-1])
print("Middle element:", numbers[len(numbers) // 2])
```

```
First element: 1
Last element: 10
Middle element: 6
```

```
# 2. Write a program to create a tuple (”Hi”,) and repeat it 5 times. Print the repeated
# tuple.
greeting = ("Hi",) * 5
print("Repeated tuple:", greeting)
```

```
Repeated tuple: ('Hi', 'Hi', 'Hi', 'Hi', 'Hi')
```

```
# 3. Write a program that checks if the number 50 exists in the tuple (10, 20, 30, 40,
# 50, 60). Print ”Found” if it exists, else print ”Not Found”.
tup = (10, 20, 30, 40, 50, 60)
if 50 in tup:
    print("Found")
else:
    print("Not Found")
```

```
Found
```

```
# 4. Given a tuple (10, 20, 30, 40, 50, 60, 70), write a Python program to print:
# The first 3 elements
# The last 3 elements
# Every alternate element
data = (10, 20, 30, 40, 50, 60, 70)
print("First 3 elements:", data[:3])
print("Last 3 elements:", data[-3:])
print("Every alternate element:", data[::2])
```

```
First 3 elements: (10, 20, 30)
Last 3 elements: (50, 60, 70)
Every alternate element: (10, 30, 50, 70)
```

```
# 5. Write a program that takes a list [1, 2, 3, 2, 4, 2, 5] and converts it into a tuple and
# count how many times the value 2 appears in the tuple.
lst = [1, 2, 3, 2, 4, 2, 5]
tup = tuple(lst)
count_2 = tup.count(2)
print("Tuple:", tup)
print("No of 2 in tuple:", count_2)
```

```
Tuple: (1, 2, 3, 2, 4, 2, 5)
No of 2 in tuple: 3
```

```
# 6. Given a tuple (100, 200, 300, 400, 500), write a program to find the index of the
# element 300.
data = (100, 200, 300, 400, 500)
ind = data.index(300)
print("Index of 300:", ind)
```

```
Index of 300: 2
```

```
# 7. Write a Python program to concatenate two tuples and display the result and find
# the length, maximum, and minimum value of that concatenated tuple.
tup1 = (10, 2, 73)
tup2 = (-4, 35, 5)
concatenated = tup1 + tup2
print("Concatenated tuple:", concatenated)
print("Length:", len(concatenated))
print("Maximum value:", max(concatenated))
```

```
print("Minimum value:", min(concatenated))
```

```
Concatenated tuple: (10, 2, 73, -4, 35, 5)
Length: 6
Maximum value: 73
Minimum value: -4
```

```
# 8. Write a Python program to convert a tuple into a list, modify one element, and
# convert it back to a tuple.
tup = (1, 2, 3, 4, 5)
lst = list(tup)
lst[2] = 99
tup_modified = tuple(lst)
print("Modified tuple:", tup_modified)
```

```
Modified tuple: (1, 2, 99, 4, 5)
```

```
# 9. The ”Variety Retail Store” sells different varieties of Furniture to the customers.
# Assume varuous furniture available with their respective cost. The furniture and its
# corresponding cost should be stored as a list. A customer can order any furniture
# in any quantity (the name and quantity of the furniture will be provided). If the
# required furniture is available in the furniture list and quantity to be purchased
# is greater than zero, then bill amount should be calculated. In case of invalid
# values for furniture required by the customer and quantity to be purchased, display
# appropriate error message and consider bill amount to be 0. Initialize required
# furniture and quantity with different values and test the results. Write a program
# to calculate and displaythe bill amount to be paid by the customer based on the
# furniture bought and quantity purchased.
# Hint – Create two different lists for ’Furniture’ and ’Cost’. Indices of two lists
# should be matched to retrieve the cost of a particular furniture.

furniture = ["Chair", "Table", "Sofa", "Bed"]
cost = [1500, 7000, 12000, 25000]
furniture_required = "Sofa"
quantity = 2
if furniture_required in furniture and quantity > 0:
    index = furniture.index(furniture_required)
    bill_amount = cost[index] * quantity
    print("Bill amount to be paid:", bill_amount)
else:
    print("Invalid furniture or quantity. Bill amount is 0.")
```

```
Bill amount to be paid: 24000
```

```
# 10. Accept a string as an input from the user. Check if the accepted string is palindrome
# or not. If the string is palindrome, print ”String is palindrome”, otherwise print
# ”String is not palindrome”. Also print the actual and the reversed strings.
# Note – Ignore the case of characters.

input_str = input("Enter a string: ")
reversed_str = input_str[::-1]
if input_str.lower() == reversed_str.lower():
    print(input_str, "is palindrome")
else:
    print(input_str, "is not palindrome")
```

```
Enter a string: tomato
tomato is not palindrome
```

```
# 11. Accept two strings ’string1’ and ’string2’ as an input from the user. Generate a
# resultant string, such that it is a concatenated string of all upper case alphabets
# from both the strings in the order they appear. Print the actual and the resultant
# strings.
# Note: Each character should be checked if it is a upper case alphabet and then it
# should be concatenated to the resultant string.
# Sample Input:
# string1: I Like C
# string2: Mary Likes JavaScript
# Output: ILCMLP

string1 = input("Enter string1: ")
string2 = input("Enter string2: ")
resultant = ""
for char in string1 + string2:
    if char.isupper():
```

```
        resultant += char
print("Actual strings:", string1, ",", string2)
print("Resultant string:", resultant)
```

```
Enter string1: I Like C
Enter string2: Sita Likes DataScience
Actual strings: I Like C , Sita Likes DataScience
Resultant string: ILCSLDS
```

```
# 12. Given a string containing both upper and lower case alphabets. Write a program
# to count the number of occurrences of each alphabet(case insensitive) and display
# the same.
input_str = input("Enter a string: ")
input_str = input_str.lower()
char_count = {}
for char in input_str:
    if char.isalpha():
        if char in char_count:
            char_count[char] += 1
        else:
            char_count[char] = 1
for char, count in char_count.items():
    print(f"{char}: {count}")
```

```
Enter a string: ArpiTa PatnAik
a: 4
r: 1
p: 2
i: 2
t: 2
n: 1
k: 1
```

```
# 13. Write a program to replace all spaces in a string with - (dash).
input_str = input("Enter a string: ")
modified_str = input_str.replace(" ", "-")
print("Modified string:", modified_str)
```

```
Enter a string: I like VS code
Modified string: I-like-VS-code
```

```
# 14. Write a program that counts how many times the word ”the” appears in a given sentence.

input_str = input("Enter a string: ")
words = input_str.lower().split()
count_the = words.count("the")
print("The word 'the' appears", count_the, "times.")
```

```
Enter a string: The student read the book on the table.
The word 'the' appears 3 times.
```

```
# 15. Write a Python program to split a sentence ”Python is fun” into a list of words,
# and then join them back into a single string separated by -.
input_str = "Python is fun"
words = input_str.split()
modified_str = "-".join(words)
print("Modified string:", modified_str)
```

```
Modified string: Python-is-fun
```

```
# 16. Write a program that accepts a sentence and replaces all vowels with *.
input_str = input("Enter a string: ")
vowels = "aeiouAEIOU"
modified_str = ''.join(['*' if char in vowels else char for char in input_str])
print("Modified string:", modified_str)
```

```
Enter a string: My name is Arpita Patnaik
Modified string: My n*m* *s *rp*t* P*tn**k
```

```
# 17. Write a program that takes a sentence ”Python is a powerful language” and splits
# it into words. Then print only the words that have more than 5 characters.
```

```
input_str = input("Enter a string: ")
words = input_str.split()
for word in words:
    if len(word) > 5:
        print(word)
```

```
Enter a string: python is a powerful language
python
powerful
language
```

```
# 18. Write a program to find the index of ”@” in an email address. If the ”@” symbol
# is missing, display ”Invalid email”.

input_str = input("Enter an email address: ")
if "@" in input_str:
    index_at = input_str.index("@")
    print("Index of '@':", index_at)
else:
    print("Invalid email")
```

```
Enter an email address: santoshsahoo78@gmail.com
Index of '@': 14
```

```
# 19. Write a program that takes two strings from the user, one for password and another
# for mobile number and check if a user’s password contains only letters and digits.
# If not, display ”Password should be alphanumeric only”. Also checks whether the
# entered mobile number is a valid mobile number (10 digits).

input_str = input("Enter password: ")
mobile_number = input("Enter mobile number: ")
if not input_str.isalnum():
    print("Password should be alphanumeric only")
elif not (mobile_number.isdigit() and len(mobile_number) == 10):
    print("Invalid mobile number")
else:
    print("Valid password and mobile number")
```

```
Enter password: sabgj65
Enter mobile number: 5463251054
Valid password and mobile number
```

```
# 20. Write a program to check the entered password’s strength with the following condition:
# It should contain at least one lowercase letter ,
# At least one uppercase letter ,
# At least one digit
# If all conditions are satisfied, print ”Strong password” else ”Weak password”.

input_str = input("Enter password: ")
if (any(char.islower() for char in input_str) and
    any(char.isupper() for char in input_str) and
    any(char.isdigit() for char in input_str)):
    print("Strong password")
else:
    print("Weak password")
```

```
Enter password: t65@nJK
Strong password
```