Assignment 4: List and Strings

Question 1: Create a list of 5 fruits and print the first, last, and third fruit.

Solution:

Python
```
fruits = ["apple", "banana", "cherry", "date", "elderberry"]

print(fruits[0])
print(fruits[-1])
print(fruits[2])
```

Question 2: Create a list of numbers from 1 to 10. Slice the list to get the first 3 and the last 3 numbers.

Solution:

Python
```
num = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

print(num[0:3])
print(num[-1:-4:-1])
```

Question 3: Create an empty list and add the numbers 1, 2, and 3 to it. Then, add the numbers 4, 5, and 6 to the list.

Solution:

Python
```
num = []
num.append(1)
num.append(2)
num.append(3)
num.extend([4, 5, 6])
print(num)
```

Question 4: Create a list of numbers from 10 to 50 with a step of 10. Remove the number 30 and the last number from the list.

Solution:

Python
```
num = [10, 20, 30, 40, 50]

num.remove(30)
num.pop(-1)
print(num)
```

Question 5: Create a list of numbers and check if a given number is present in the list.

Solution:

Python
```
num = [10, 20, 30, 40, 50]

check = int(input("Enter a number to check: "))
if check in num:
    print("Found")
else:
    print("Not Found")
```

Question 6: Create a list of numbers and find the sum, average, maximum, and minimum of the numbers in the list.

Solution:

Python
```
num = [10, 20,30,40,50]

sum = sum(num)
average = sum / len(num)
max_num = max(num)
min_num = min(num)

print(f"Sum: {sum}")
print(f"Average: {average}")
print(f"Max: {max_num}")
print(f"Min: {min_num}")
```

Question 7: Create a list with repeated numbers and count the number of times a given number is repeated.

Solution:

Python
```
num = [10, 10, 20, 30, 30, 40, 40, 50, 50, 50]

check_num_repeated_no = int(input("Enter a number to check for times it is repeated: "))

count = num.count(check_num_repeated_no)
print(f"The number {check_num_repeated_no} is repeated {count} times.")
```

Question 8: Write a program to remove duplicates from a list.

Solution:

Python
```
num = [1, 2, 2, 3, 4, 4, 5]

unique_list = []
for item in num:
    if item not in unique_list:
        unique_list.append(item)

print(unique_list)
```

Question 9: Reverse a list.

Solution:

Python
```
num = [1, 2, 2, 3, 4, 4, 5]

print(num[ : : -1])
```

Question 10: Concatenate two lists element-wise.

Solution:

Python
```
list1 = ["M", "na", "i", "ku"]
list2 = ["y", "me", "s", "nal"]

for i,j in zip(list1, list2):
    print(i+j)
```

Question 11: From a list of numbers from 1 to 20, create a new list containing only the even numbers.

Solution:

Python
```
num = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20]

list_even = [x for x in num if x % 2 == 0]

print(list_even)
```

Question 12: Find the second largest number in a list.

Solution:

Python
```
num = [1, 2, 3, 4, 5, 6]
```

```
sorted_num = sorted(num)

sec_largest = sorted_num[-2]
print(sec_largest)
```

Question 13: Check if a list is a palindrome.

Solution:

Python
```
num = [1, 2, 3, 2, 1]

if num[::] == num [::-1]:
    print("Palindrome")
else:
    print("Not a Palindrome")
```

Question 14: Iterate over two lists simultaneously.

Solution:

Python
```
list1 = [20,35,45,78]
list2 = [100,200,300,400]

for i,j in zip(list1, list2):
    print(f"{i} {j}")
```

Question 15: Add an element to a nested list.

Solution:

Python
```
list1 = [10, 20, [300, 400, [5000, 6000], 500], 30, 40]

list1[2][2].append(7000)

print(list1)
```

Question 16: Find the first occurrence of a value in a list and replace it.

Solution:

Python
```
list1 = [5, 10, 15, 20, 25, 50, 20]
```

```
index = list1.index(20)
list1[index] = 200

print(list1)
```

Question 17: Split a sentence into words and then join them with a hyphen.

Solution:

```
Python
sentence = "Python is fun"

words = sentence.split()

joined_string = "-".join(words)

print("Original sentence:", sentence)
print("List of words:", words)
print("String after joining with '-':", joined_string)
```

Question 18: Reverse each string in a list of strings.

Solution:

```
Python
string_list = ["cat", "dog", "bird"]

reversed_list = [s[::-1] for s in string_list]

print("Original list:", string_list)
print("Reversed list:", reversed_list)
```

Question 19: Flatten a nested list.

Solution:

```
Python
nested_list = [[1, 2], [3, 4], [5, 6]]

flattened_list = [item for sublist in nested_list for item in sublist]

print("Original nested list:", nested_list)
print("Flattened list:", flattened_list)
```

Question 20: Find the equilibrium index of a list.

Solution:

Python

```
input_list = [1, 7, 3, 6, 5, 6]
found_index = -1

for i in range(len(input_list)):
    left_sum = sum(input_list[:i])

    right_sum = sum(input_list[i+1:])

    if left_sum == right_sum:
        found_index = i
        break

print("Input list:", input_list)

if found_index != -1:
    print("Equilibrium index found at:", found_index)
else:
    print("No equilibrium index found.")
```

Name:

Section:

SIC:

Roll No:

Assignment 5: Tuple and String

Question 1: Create a tuple of numbers from 1 to 10 and print the first, last, and middle element.

Solution:

Python
```
numbers = tuple(range(1, 11))

print("First element:", numbers[0])
print("Last element:", numbers[-1])
print("Middle element:", numbers[len(numbers) // 2])
```

Question 2: Repeat a tuple 5 times.

Solution:

Python
```
print("Repeated tuple:", ("Hi",) * 5)
```

Question 3: Check if an element exists in a tuple.

Solution:

Python
```
numbers = (10, 20, 30, 40, 50, 60)
if 50 in numbers:
    print("Found")
else:
    print("Not Found")
```

Question 4: Slice a tuple to get the first 3 elements, the last 3 elements, and every alternate element.

Solution:

Python
```
numbers = (10, 20, 30, 40, 50, 60, 70)

print("The first 3 elements:")
print(numbers[:3])

print("The last 3 elements:")
print(numbers[-3:])

print("Every alternate element:")
print(numbers[::2])
```

Question 5: Count the occurrences of a value in a tuple.

Solution:

Python
```
list_data = [1, 2, 3, 2, 4, 2, 5]
tuple_data = tuple(list_data)
count_2 = tuple_data.count(2)
print("The value 2 appears", count_2, "times in the tuple.")
```

Question 6: Find the index of an element in a tuple.

Solution:

Python
```
t = (100, 200, 300, 400, 500)

if 300 in t:
        idx = t.index(300)
        print(f"Index of 300 in {t} is: {idx}")
else:
        print("300 is not in the tuple")
```

Question 7: Concatenate two tuples and find the length, maximum, and minimum of the resulting tuple.

Solution:

Python
```
t1 = (10, 20, 30)
t2 = (40, 50, 60)

t = t1 + t2
print("Concatenated tuple:", t)
print("Length:", len(t))
print("Maximum:", max(t))
print("Minimum:", min(t))
```

Question 8: Modify an element in a tuple.

Solution:

Python
```
t = (1, 2, 3, 4)
print("Original tuple:", t)

l = list(t)
l[1] = 4
t = tuple(l)
```

```
print("Modified tuple:", t)
```

Question 9: Calculate the bill for a given furniture item and quantity.

Solution:

```
Python
furniture = ['Chair', 'Table', 'Bed', 'Sofa']
cost = [100, 200, 300, 400]

required_furniture = 'Bed'
quantity = 2

bill_amount = 0

if required_furniture in furniture and quantity > 0:
    furniture_index = furniture.index(required_furniture)
    price_per_item = cost[furniture_index]
    bill_amount = price_per_item * quantity
else:
    if required_furniture not in furniture:
        print(f"Error: The furniture '{required_furniture}' is not available.")
    if quantity <= 0:
        print("Error: The quantity to be purchased must be greater than zero.")

print(f"Item: {required_furniture}")
print(f"Quantity: {quantity}")
print(f"Bill Amount: ${bill_amount}")
```

Question 10: Check if a string is a palindrome.

Solution:

```
Python
s = input("Enter a string: ").strip()
s_clean = s.lower()
rev = s_clean[::-1]
if s_clean == rev:
        print("String is palindrome")
else:
        print("String is not palindrome")

print("Actual:", s)
print("Reversed:", rev)
```

Question 11: Concatenate two strings and keep only the uppercase letters.

Solution:

Python
```
string1 = input("Enter string1: ")
string2 = input("Enter string2: ")

result = ''.join([c for c in (string1 + string2) if c.isupper() and c.isalpha()])

print("Actual string1:", string1)
print("Actual string2:", string2)
print("Resultant uppercase concatenation:", result)
```

Question 12: Count the frequency of each character in a string.

Solution:

Python
```
s = input("Enter a string: ")
s_letters = [c.lower() for c in s if c.isalpha()]
counts = [(ch, s_letters.count(ch)) for ch in set(s_letters)]

for ch, cnt in sorted(counts):
        print(f"{ch}: {cnt}")
```

Question 13: Replace spaces with hyphens in a string.

Solution:

Python
```
s = input("Enter a string: ")
print(s.replace(' ', '-'))
```

Question 14: Count the number of occurrences of the word 'the' in a sentence.

Solution:

Python
```
sentence = input("Enter a sentence: ")
words = [w.strip(".,!?;:") for w in sentence.lower().split()]
count_the = sum(1 for w in words if w == 'the')
print(f"Occurrences of 'the' = {count_the}")
```

Question 15: Split a sentence into words and then join them with a hyphen.

Solution:

Python

```
sentence = "Python is fun"
words = sentence.split()
print("Split words:", words)
joined = '-'.join(words)
print("Joined with -:", joined)
```

Question 16: Replace all vowels in a sentence with an asterisk.

Solution:

Python
```
vowels = set('aeiouAEIOU')
sentence = input("Enter a sentence: ")
result = ''.join('*' if c in vowels else c for c in sentence)
print(result)
```

Question 17: Find all words in a sentence that have more than 5 characters.

Solution:

Python
```
sentence = "Python is a powerful language"
words = sentence.split()
long_words = [w for w in words if len(w) > 5]
print("Words with more than 5 characters:", long_words)
```

Question 18: Find the index of the '@' symbol in an email address.

Solution:

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

Question 19: Validate a password and mobile number.

Solution:

Python
```
password = input("Enter password: ")
mobile = input("Enter mobile number: ")

if not password.isalnum():
	print("Password should be alphanumeric only")
```

```
else:
        print("Password is alphanumeric")

if mobile.isdigit() and len(mobile) == 10:
        print("Mobile number is valid")
else:
        print("Invalid mobile number")
```

Question 20: Check the strength of a password.

Solution:

Python

```
pwd = input("Enter password to check strength: ")

has_lower = any(c.islower() for c in pwd)
has_upper = any(c.isupper() for c in pwd)
has_digit = any(c.isdigit() for c in pwd)

if has_lower and has_upper and has_digit:
        print("Strong password")
else:
        print("Weak password")
```

Name:

Section:

SIC:

Roll No:

Assignment 6: Dictionary

Question 1: Iterate over a dictionary of countries and their capitals and print them.

Solution:

Python
```
countries = {
    "USA": "Washington, D.C.",
    "France": "Paris",
    "Japan": "Tokyo"
}
for country, capital in countries.items():
    print(f"The capital of {country} is {capital}.")
```

Question 2: Access and print the name and city from a dictionary.

Solution:

Python
```
person = {
    "name": "Alice",
    "age": 22,
    "city": "Delhi"
}
print(person["name"])
print(person["city"])
```

Question 3: Update the age and add a new grade to a student dictionary.

Solution:

Python
```
student = {
    "name": "Alice",
    "age": 20,
    "branch": "Computer Science"
}
student["age"] = 21
student["grade"] = "A"
print(student)
```

Question 4: Remove an item from a dictionary of fruits and their prices.

Solution:

Python
```
fruits = {
```

```
    "apple": 60,
    "banana": 30,
    "orange": 50
}
fruits.pop("banana")
print(fruits)
```

Question 5: Iterate over a dictionary of book details and print each key-value pair.

Solution:

Python
```
book = {
    "title": "To Kill a Mockingbird",
    "author": "Harper Lee",
    "price": 399
}
for key, value in book.items():
    print(f"{key}: {value}")
```

Question 6: Merge two dictionaries.

Solution:

Python
```
d1 = {"a": 10, "b": 20}
d2 = {"c": 30, "d": 40}
merged_dict = {**d1, **d2}
print(merged_dict)
```

Question 7: Check if a value exists in a dictionary.

Solution:

Python
```
sampleDict = {'a': 100, 'b': 200, 'c': 300}
if 200 in sampleDict.values():
    print("Value 200 exists in the dictionary.")
else:
    print("Value 200 does not exist in the dictionary.")
```

Question 8: Find the product with the maximum price from a dictionary.

Solution:

Python
```
prices = {"laptop": 55000, "phone": 25000, "tablet": 18000}
```

```
max_price_product = max(prices, key=prices.get)
print(f"The product with the maximum price is: {max_price_product}")
```

Question 9: Count the frequency of words in a sentence.

Solution:

```
Python
text = "Python is fun and easy"
word_count = {}
for word in text.split():
    word_count[word] = word_count.get(word, 0) + 1
print(word_count)
```

Question 10: Invert a dictionary.

Solution:

```
Python
students = {101: "Amit", 102: "Riya", 103: "John"}
inverted_dict = {name: roll for roll, name in students.items()}
print(inverted_dict)
```

Question 11: Add and remove elements from a set.

Solution:

```
Python
my_set = {10, 20, 30}
my_set.add(40)
my_set.add(50)
my_set.remove(20)
print(my_set)
```

Question 12: Perform union, intersection, difference, and symmetric difference on two sets.

Solution:

```
Python
A = {1, 2, 3, 4}
B = {3, 4, 5, 6}
print("Union:", A | B)
print("Intersection:", A & B)
print("Difference A - B:", A - B)
print("Symmetric Difference:", A ^ B)
```

Question 13: Remove duplicates from a list using a set.

Solution:

Python
```
numbers = [1, 2, 2, 3, 4, 4, 5]
unique_numbers = list(set(numbers))
print(unique_numbers)
```

Question 14: Find the common elements between two lists.

Solution:

Python
```
list1 = [1, 2, 3, 4]
list2 = [3, 4, 5, 6]
common_elements = set(list1) & set(list2)
print(common_elements)
```

Question 15: Find the unique words in a sentence.

Solution:

Python
```
sentence = "Python is fun and learning Python is easy"
words = sentence.split()
unique_words = set(words)
print(unique_words)
```

Question 16: Find the common elements among three sets and the elements present in at least one of the sets.

Solution:

Python
```
A = {1, 2, 3}
B = {2, 3, 4}
C = {3, 4, 5}
print("Common elements:", A & B & C)
print("Elements in at least one set:", A | B | C)
```

Question 17: Find the intersection of two sets and update the first set.

Solution:

Python
```
set1 = {10, 20, 30, 40, 50}
```

```
set2 = {30, 40, 50, 60, 70}
set1.intersection_update(set2)
print(set1)
```

Name:

Section:

SIC:

Roll No: