TRIBHUVAN UNIVERSITY
INSTITUTE OF ENGINEERING
PULCHOWK CAMPUS

Lab 1

By:
Sinjal Dahal (081/BEL/080)

DEPARTMENT OF ELECTRICAL ENGINEERING
LALITPUR, NEPAL

A. Data Structures

1. Write a Python program to remove all duplicates from a list and print the unique elements.

```
initial_list = ["sinjal", "sinjal", "sinjal", "Dahal", "Dahal", "Dahal", "Dahal"]
initial_list_1 = ['a', 'b', 'a', 'b', 'a', 'b', 'c', 'd', 'c', 'd', 'c', 'd']
print(set(initial_list))
print(set(initial_list_1))
```

Output

```
{'sinjal', 'Dahal'}
{'d', 'a', 'b', 'c'}

=== Code Execution Successful ===
```

2. Create a tuple of 10 integers. Write a program to display the maximum and minimum numbers from the tuple.

```
tuple_1 = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
max_num = max(tuple_1)
min_num = min(tuple_1)
print("Maximum number in the tuple:", max_num)
print("Minimum number in the tuple:", min_num)
```

Output

```
Maximum number in the tuple: 10
Minimum number in the tuple: 1

=== Code Execution Successful ===
```

3. Write a Python function that accepts a list and returns a new list with only the even numbers from the original list.

```
def filter_even_numbers(numbers):
return [num for num in numbers if num % 2 == 0]
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
print(filter_even_numbers(numbers))
```

Output

```
[2, 4, 6, 8, 10]

=== Code Execution Successful ===
```

4. Write a program to count the number of each character in a given string using a dictionary.

```
def print_character_positions(input_string):
for index, char in enumerate(input_string, start=1):
print(f"{char}:{index}", end=", " if index < len(input_string) else "")
user_input = input("Enter a string: ")
print_character_positions(user_input)
```

Output

```
Enter a string: Sinjal Dahal
S:1, i:2, n:3, j:4, a:5, l:6,  :7, D:8, a:9, h:10, a:11, l:12
=== Code Execution Successful ===
```

5. Create a set of prime numbers less than 50. Write a program to check whether a given number exists in the set or not.

```
primes = {2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47}
num = int(input("Enter a number: "))
if num in primes:
print(f"{num} is a prime number less than 50.")
else:
print(f"{num} is not a prime number less than 50.")
```

Output

```
Enter a number: 12
12 is not a prime number less than 50.

=== Code Execution Successful ===
```

Output

```
Enter a number: 17
17 is a prime number less than 50.

=== Code Execution Successful ===
```

6. Given two lists, write a program to find their intersection using sets.

```
list1 = [1, 2, 3, 4, 5]
list2 = [4, 5, 6, 7, 8]
intersection = list(set(list1) & set(list2))
print("Intersection:", intersection)
```

```
Output

Intersection: [4, 5]

=== Code Execution Successful ===
```

7. Write a Python program to merge two dictionaries and sum the values of common keys.

```
def merge_and_sum(dict1, dict2):
merged = dict1.copy()
for key, value in dict2.items():
if key in merged:
merged[key] += value
else:
merged[key] = value
return merged
dict1 = {'a': 10, 'b': 20, 'c': 30}
dict2 = {'b': 5, 'c': 15, 'd': 25}
result = merge_and_sum(dict1, dict2)
print(result)
```

```
Output

{'a': 10, 'b': 25, 'c': 45, 'd': 25}

=== Code Execution Successful ===
```

8. Given a list of names, write a program to count how many times each name appears using a dictionary.

```
names = ['Sinjal', 'Sher', 'Sher', 'Sher', 'Pushpa', 'Sinjal']
name_counts = {}
for name in names:
if name in name_counts:
name_counts[name] += 1
else:
name_counts[name] = 1
print(name_counts)
```

Output

```
{'Sinjal': 2, 'Sher': 3, 'Pushpa': 1}

=== Code Execution Successful ===
```

9. Write a Python program to remove elements from a list that are also present in another list.

```
list1 = [1, 2, 3, 4, 5, 6]
list2 = [2, 4, 6]
result = [item for item in list1 if item not in list2]
print("Updated list:", result)
```

Output

```
Updated list: [1, 3, 5]

=== Code Execution Successful ===
```

10. Write a program to input key-value pairs from the user and store them in a dictionary. Allow the user to search for a key and display its value.

```
data = {}
n = int(input("How many key-value pairs do you want to enter? "))
for _ in range(n):
key = input("Enter key: ")
value = input("Enter value: ")data[key] = value
search_key = input("Enter key to search: ")
if search_key in data:
print(f"Value: {data[search_key]}")
else:
print("Key not found.")
```

Output

```
How many key-value pairs do you want to enter? 3
Enter key: a
Enter value: sinjal
Enter key: b
Enter value: dahal
Enter key: c
Enter value: king
Enter key to search: c
Value: king

=== Code Execution Successful ===
```

B. Conditions and Loops

1. Write a program to check whether a given number is prime or not.

```
num = int(input("Enter a number: "))
if num <= 1:
print(num, "is not a prime number")
else:
i = 2
while i * i <= num:
if num % i == 0:
print(num, "is not a prime number")
break
i += 1
else:
print(num, "is a prime number")
```

Output

```
Enter a number: 7
7 is a prime number

=== Code Execution Successful ===
```

Output

```
Enter a number: 10
10 is not a prime number

=== Code Execution Successful ===
```

2. Write a program to print all the even numbers between 1 and 100 using a loop.

```
for num in range(2, 101, 2):
print(num)
```

Output

```
2
4
6
8
10
12
14
16
18
20
22
24
26
28
30
32
34
36
38
40
42
44
46
48
56
58
60
62
64
66
68
70
72
74
76
78
80
82
84
86
88
90
92
94
96
98
100
```

3. Write a program that reads a number and prints the factorial of that number using a while loop.

```
num = int(input("Enter a number: "))
factorial = 1
```

```
i = 1
while i <= num:
factorial *= i
i += 1
print("Factorial of", num, "is", factorial)
```

Output

```
Enter a number: 10
Factorial of 10 is 3628800

=== Code Execution Successful ===
```

4. Write a program to print the multiplication table of a given number using a for loop.

```
num = int(input("Enter a number: "))
for i in range(1, 11):
print(f"{num} x {i} = {num * i}")
```

Output

```
Enter a number: 5
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50

=== Code Execution Successful ===
```

5. Write a program to find the largest and smallest number in a list entered by the user.

```
numbers = input("Enter numbers : ").split()
numbers = [float(num) for num in numbers]
largest = max(numbers)
smallest = min(numbers)
print("Largest number:", largest)
print("Smallest number:", smallest)
```

Output

```
Enter numbers : 1 2 55 98526 22 37
Largest number: 98526.0
Smallest number: 1.0

=== Code Execution Successful ===
```

6. Write a program that accepts 10 integers from the user and counts how many are positive, negative, and zero.

```
positive_count = 0
negative_count = 0
zero_count = 0
for i in range(10):
num = int(input(f"Enter integer #{i+1}: "))
if num > 0:
positive_count += 1
elif num < 0:
negative_count += 1
else:
```

```
zero_count += 1
print("Positive numbers:", positive_count)
print("Negative numbers:", negative_count)
print("Zeros:", zero_count)
```

Output

```
Enter integer #1: 10
Enter integer #2: -21
Enter integer #3: 36
Enter integer #4: 42
Enter integer #5: 0
Enter integer #6: 52
Enter integer #7: -68
Enter integer #8: 001
Enter integer #9: 0
Enter integer #10: 14
Positive numbers: 6
Negative numbers: 2
Zeros: 2

=== Code Execution Successful ===
```

7. Write a program to generate the Fibonacci sequence up to n terms.

```
n = int(input("Enter the number of terms: "))
fib_sequence = []
a, b = 0, 1
for _ in range(n):
fib_sequence.append(a)
```

```
a, b = b, a + b
print("Fibonacci sequence up to", n, "terms:")
print(fib_sequence)
```

Output

```
Enter the number of terms: 10
Fibonacci sequence up to 10 terms:
[0, 1, 1, 2, 3, 5, 8, 13, 21, 34]

=== Code Execution Successful ===
```

8. Write a program that reads a number and prints whether it is a palindrome or not.

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

Output

```
Enter a number: 131
Palindrome

=== Code Execution Successful ===
```

Output

```
Enter a number: 100
Not a palindrome

=== Code Execution Successful ===
```

9. Write a program that finds all numbers between 100 and 999 where the sum of the cubes of the digits equals the number itself (Armstrong numbers).

```
for num in range(100, 1000):
digits = [int(d) for d in str(num)]
if sum(d ** 3 for d in digits) == num:print(num)
```

```
Output
153
370
371
407

=== Code Execution Successful ===
```

10. Write a menu-driven program to perform arithmetic operations (+, -, *, /) based on user choice using conditional statements.

```
def menu():
print("Select operation:")
print("1. Addition (+)")
print("2. Subtraction (-)")
print("3. Multiplication (*)")
print("4. Division (/)")
while True:
menu()
choice = input("Enter choice (1/2/3/4) or 'q' to quit: ")
if choice.lower() == 'q':
print("Exiting program.")
break
if choice not in ('1', '2', '3', '4'):
print("Invalid input. Please try again.")
```

```
continue
try:
num1 = float(input("Enter first number: "))
num2 = float(input("Enter second number: "))
except ValueError:
print("Invalid number. Please try again.")
continue
if choice == '1':
print(f"{num1} + {num2} = {num1 + num2}")
elif choice == '2':
print(f"{num1} - {num2} = {num1 - num2}")
elif choice == '3':
print(f"{num1} * {num2} = {num1 * num2}")
elif choice == '4':
if num2 == 0:
print("Error: Division by zero.")
else:
print(f"{num1} / {num2} = {num1 / num2}")
```

Output

```
Select operation:
1. Addition (+)
2. Subtraction (-)
3. Multiplication (*)
4. Division (/)
Enter choice (1/2/3/4) or 'q' to quit: 3
Enter first number: 12
Enter second number: 5
12.0 * 5.0 = 60.0
Select operation:
1. Addition (+)
2. Subtraction (-)
3. Multiplication (*)
4. Division (/)
Enter choice (1/2/3/4) or 'q' to quit:
```

SINJAL DAHAL

081BEL080

https://github.com/sinjaldahal/sinjaldahal_BEL