**Lab 5 – Functions and Lists**

**Questions:**

**Functions**

**1. Write a function that takes a number as a parameter and prints the numbers from 1 to that number on the screen.**

def print\_numbers(n):

for i in range(1, n+1):

print(i)

# Example usage:

print\_numbers(10) # prints the numbers from 1 to 10 on the screen

**2. Write a function that takes a number as a parameter and iterates from 0 to that number. For each iteration, it will check if the current number is even or odd, and report that to the screen (e.g. “1 is odd”, “2 is even”).**

num = int(input("Enter a number: "))

if (num % 2) == 0:

print("{0} is Even".format(num))

else:

print("{0} is Odd".format(num))

OR

def even\_odd\_numbers(n):

for i in range(n+1):

if i % 2 == 0:

print("{} is even".format(i))

else:

print("{} is odd".format(i))

# Example usage:

even\_odd\_numbers(10) # prints whether each number from 0 to 10 is even or odd

**3. Write a function that takes a number as a parameter, iterates from 0 to that number, and for each iteration of the loop, multiplies the current number by 9 and prints the result (e.g. “2 \* 9 = 18”).**

def multiply\_by\_nine(n):

for i in range(n+1):

result = i \* 9

print("{} \* 9 = {}".format(i, result))

# Example usage:

multiply\_by\_nine(10) # multiplies each number from 0 to 10 by 9 and prints the result

**4. Write a function that asks the user for a number and prints the sum of all numbers from 1 to the number they enter.**

num = int(input("Enter a number: "))

output = ""

for i in range(1, num+1):

output += "{}".format(i)

if i != num:

output += "+"

output += " = {}".format(sum(range(num+1)))

print(output)

**5. Write a function to print a factorial of a number.**

num = int(input("Enter a number: "))

factorial = 1

if num< 0:

print(" Factorial does not exist for negative numbers")

elif num == 0:

print("The factorial of 0 is 1")

else:

for i in range(1,num + 1):

factorial = factorial\*i

print("The factorial of", num,"is", factorial)

**6. Write a function that takes a string as a parameter and returns a string that is made up of the first two characters and the last two characters. If the string has a length less than 4 the program prints a message on the screen. For example: “hello there” will result in “here”**

def get\_first\_and\_last\_two\_chars(s):

if len(s) < 4:

print("String length is less than 4.")

return ""

else:

return s[:2] + s[-2:]

# Example usage:

result = get\_first\_and\_last\_two\_chars("hello there")

print(result) # prints "here"

result = get\_first\_and\_last\_two\_chars("hi")

# prints "String length is less than 4." and returns an empty string

**7. Write a Python program to remove the characters which have odd index**

**values of a given string. The function should return the new string.**

# Function to remove the odd

# indexed characters from a given string

def removeOddIndexCharacters(s):

# Stores the resultant string

new\_s = ""

i = 0

while i<len(s):

# If the current index is odd

if (i % 2 == 1):

# Skip the character

i+= 1

continue

# Otherwise, append the character

new\_s += s[i]

i+= 1

# Return the modified string

return new\_s

# Driver Code

if \_\_name\_\_ == '\_\_main\_\_':

str = "abcdef"

# Remove the characters which

# have odd index

str = removeOddIndexCharacters(str)

print(str)

**8. Write a Python function to get the first half of a specified string of even length.**

**Sample function and result:**

**string\_first\_half(“Python”)**

**should return Pyt**

def string\_first\_half(s):

if len(s) % 2 == 0:

return s[:len(s)//2]

else:

return "String length is not even."

# Example usage:

result = string\_first\_half("Python")

print(result) # prints "Pyt"

result = string\_first\_half("odd")

print(result) # prints "String length is not even."

**9. Write a Python function to insert a string in the middle of a string.**

**Sample function and result:**

**insert\_string\_middle(“{{}}”,”PHP”)**

**Should return {{PHP}}**

def insert\_string\_middle(s, insert\_str):

return s[:len(s)//2] + insert\_str + s[len(s)//2:]

# Example usage:

result = insert\_string\_middle("{{}}", "PHP")

print(result) # prints "{{PHP}}"

result = insert\_string\_middle("[]", "Python")

print(result) # prints "[Python]"

**10. Write a Python function that takes a string and two indices, and returns**

**a string with the part between the indices removed.**

**For example: remove\_substring(“Hello there”, 2, 6) should return “Hehere”**

**Lists**

def remove\_substring(s, start\_index, end\_index):

return s[:start\_index] + s[end\_index:]

# Example usage:

result = remove\_substring("Hello there", 2, 6)

print(result) # prints "Hehere"

**11. Write a Python function to sum all numbers in a list.**

**Sample list: [1, 2, 3, 4, 5, 6]**

**Expected Output: 21**

list = [1, 2, 3, 4, 5, 6]

total = 0

for x in list:

total += x

print("Expected Output: ", total)

OR

def sum\_list\_numbers(lst):

return sum(lst)

# Example usage:

my\_list = [1, 2, 3, 4, 5, 6]

result = sum\_list\_numbers(my\_list)

print(result) # prints 21

**12. Write a Python function to get the largest number from a list.**

**Sample list: [1, 2, 3, 4, 5, 6]**

**Expected Output: 6**

list = [1, 2, 3, 4, 5, 6]

list.sort()

print("Largest element is:", list[-1])

OR

def get\_largest\_number(lst):

return max(lst)

# Example usage:

my\_list = [1, 2, 3, 4, 5, 6]

result = get\_largest\_number(my\_list)

print(result) # prints 6

**13. Write a Python function that takes a list of words and counts how many**

**of them begin with ‘o’.**

**Sample list: ['Always', 'look', 'on', 'the', 'bright', 'side', 'of', 'life']**

**Expected Output: 2**

def count\_words\_starting\_with\_o(words\_list):

count = 0

for word in words\_list:

if word.startswith('o') or word.startswith('O'):

count += 1

return count

# Example usage:

my\_list = ['Always', 'look', 'on', 'the', 'bright', 'side', 'of', 'life']

result = count\_words\_starting\_with\_o(my\_list)

print(result) # prints 2

**14. (modify Ex13) Write a Python function that takes a list of words and a character, and counts how many of the words in the list begin with that character.**

def count\_words\_starting\_with\_char(words\_list, char):

count = 0

for word in words\_list:

if word.startswith(char) or word.startswith(char.upper()):

count += 1

return count

# Example usage:

my\_list = ['Always', 'look', 'on', 'the', 'bright', 'side', 'of', 'life']

char = 'o'

result = count\_words\_starting\_with\_char(my\_list, char)

print(result) # prints 2

**15. Write a Python function that takes a list of numbers and returns a new list containing only the even numbers from the first list.**

**Sample list: [1, 2, 3, 4, 5, 6]**

**Expected Output: [2, 4, 6]**

list = [1, 2, 3, 4, 5, 6]

for num in list:

if num % 2 == 0:

print(num, end=" ")

OR

def get\_even\_numbers(lst):

return [num for num in lst if num % 2 == 0]

# Example usage:

my\_list = [1, 2, 3, 4, 5, 6]

result = get\_even\_numbers(my\_list)

print(result) # prints [2, 4, 6]

**16. Create a list of 100 integers whose value and index are the same, e.g., L[5]=5.**

my\_list = [i for i in range(100)]

print(my\_list)

**17. Given a = [1,2,3] and b = [1,2,3] , what is the result of:**

**(a) a == b**

**(b) a is b**

(a) The result of `a == b` is `True`. This is because `a` and `b` have the same values at each index, so the comparison returns `True`.

(b) The result of `a is b` is `False`. This is because `a` and `b` are two different objects, even though they contain the same values. The `is` operator checks whether two objects are the same object in memory, and in this case, `a` and `b` are two different objects.

**18. Write a Python program to remove duplicates from a list.**

**Sample list:** **[1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6]**

**Expected Output: [1, 2, 3, 4, 5, 6]**

sample\_list = [1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6]

new\_list = []

for i in sample\_list:

if i not in new\_list:

new\_list.append(i)

sample\_list = new\_list

print(sample\_list)

OR

def remove\_duplicates(lst):

return list(set(lst))

# Example usage:

my\_list = [1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6]

result = remove\_duplicates(my\_list)

print(result) # prints [1, 2, 3, 4, 5, 6]

**19. Write a Python function that takes two lists and returns True if they have at least one common member.**

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

**Expected Output: True**

def lists(list1, list2):

result = False

# traverse in the 1st list

for x in list1:

# traverse in the 2nd list

for y in list2:

# if one common

if x == y:

result = True

return result

return result

# driver code

a = [1, 2, 3, 4, 5, 6]

b = [10, 9, 8, 7, 6]

print(lists(a, b))

OR

def has\_common\_member(lst1, lst2):

return bool(set(lst1) & set(lst2))

# Example usage:

list1 = [1, 2, 3, 4, 5, 6]

list2 = [10, 9, 8, 7, 6]

result = has\_common\_member(list1, list2)

print(result) # prints True

**20. Write a Python program to get the difference between the two lists.**

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

**Expected Output: [1, 2, 3, 4, 5]**

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

**Expected Output: [10, 9, 8, 7]**

my\_list1 = [1, 2, 3, 4, 5, 6]

my\_list2 = [10, 9, 8, 7, 6]

result = []

for elen in my\_list1:

if elen not in my\_list2:

result.append(elen)

print(result)

OR

def list\_difference(lst1, lst2):

return list(set(lst1) - set(lst2))

# Example usage:

list1 = [1, 2, 3, 4, 5, 6]

list2 = [10, 9, 8, 7, 6]

result = list\_difference(list1, list2)

print(result) # prints [1, 2, 3, 4, 5]

list1 = [10, 9, 8, 7, 6]

list2 = [1, 2, 3, 4, 5, 6]

result = list\_difference(list1, list2)

print(result) # prints [10, 9, 8, 7]

**21. Write a Python program to convert a list of multiple integers into a single integer.**

**Sample list: [11, 33, 50]**

**Expected Output: 113350**

list = [11, 33, 50]

for i in list:

print(i, end="")

OR

def convert\_list\_to\_int(lst):

return int(''.join(str(num) for num in lst))

# Example usage:

my\_list = [11, 33, 50]

result = convert\_list\_to\_int(my\_list)

print(result) # prints 113350