**Module – 2 (Fundamentals of python)**

**1) Write a Python program to check if a number is positive, negative or**

**zero.**

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

if num > 0:

print("Positive number")

elif num == 0:

print("Zero")

else:

print("Negative number")

**2) Write a Python program to get the Factorial number of given number.**

a=int(input("Enter a Number:"))

fact=1

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

fact=fact\*i

print("Factorial: ",fact)

**3) Write a Python program to get the Fibonacci series of given range.**

 a=int(input("Enter the times of loop should execute:”))

b=0

c=1

for i in range(a):

d=b+c

b=c

c=d

print(b)

**4) How memory is managed in Python?**

Memory management in Python involves a private heap containing all Python objects and data structures. The management of this private heap is ensured internally by the Python memory manager.

**5) What is the purpose continue statement in python?**

The continue statement is used to skip the remaining code inside a loop for the current iteration only. For instance, let's use continue instead of a break statement in the previous example.

**6) Write python program that swap two number with temp variable and without temp variable.**

x=int(input("Enter 1st Number"))

y=int(input("Enter 2nd Number"))

print("Before Swapping",x,y)

print("-"\*10)

x,y = y,x

print("After Swapping",x,y)

**7) Write a Python program to find whether a given number is even or odd, print out an appropriate message to the user.**

num = int (input (“Enter any number to test whether it is odd or even: “)

if (num % 2) == 0:

print (“The number is even”)

else:

print (“The provided number is odd”)

**8) Write a Python program to test whether a passed letter is a vowel or not.**

l = input("Input a letter of the alphabet: ")

if l in ('a', 'e', 'i', 'o', 'u'):

print("%s is a vowel." % l)

elif l == 'y':

print("Sometimes the letter y stands for a vowel, sometimes for a consonant.")

else:

print("%s is a consonant." % l)

**9) Write a Python program to sum of three given integers. However, if two values are equal sum will be zero.**

def sum\_three(x, y, z):

if x == y or y == z or x == z:

sum = 0

else:

sum = x + y + z

return sum

print(sum\_three(2, 1, 2))

print(sum\_three(3, 2, 2))

print(sum\_three(2, 2, 2))

print(sum\_three(1, 2, 3))

**10) Write a Python program that will return true if the two given integer values are equal or their sum or difference is 5.**

def test\_number5(x, y):

if x == y or abs(x - y) == 5 or (x + y) == 5:

return True

else:

return False

print(test\_number5(7, 2))

print(test\_number5(3, 2))

print(test\_number5(2, 2))

print(test\_number5(7, 3))

print(test\_number5(27, 53))

**11) Write a python program to sum of the first n positive integers.**

n = int(input("Input a number: "))

sum\_num = (n \* (n + 1)) / 2

print("Sum of the first", n, "positive integers:", sum\_num)

**12) Write a Python program to calculate the length of a string.**

a = "Adi"

print(len(a))

**13) Write a Python program to count the number of characters (character frequency) in a string.**

s = "AdityaGajjar"

freq = {}

for c in s:

if c in freq:

freq[c] += 1

else:

freq[c] = 1

print(freq)

**14) What are negative indexes and why are they used?**

Negative indexes are used to access elements in a list, array, or string from the end, rather than the beginning. They are useful when working with large lists or when you need to access the last few elements without knowing their exact position.

**15) Write a Python program to count occurrences of a substring in a string.**

str1 = 'The quick brown fox jumps over the lazy dog.'

print()

print(str1.count("fox"))

print()

**16) Write a Python program to count the occurrences of each word in a given sentence.**

def word\_count(str):

counts = dict()

words = str.split()

for word in words:

if word in counts:

counts[word] += 1

else:

counts[word] = 1

return counts

print( word\_count('the quick brown fox jumps over the lazy dog.'))

**17) Write a Python program to get a single string from two given strings, separated by a space and swap the first two characters of each string.**

def chars\_mix\_up(a, b):

new\_a = b[:2] + a[2:]

new\_b = a[:2] + b[2:]

return new\_a + ' ' + new\_b

print(chars\_mix\_up('abc', 'xyz'))

**18) Write a Python program to add 'ing' at the end of a given string (length should be at least 3). If the given string already ends with 'ing' then add 'ly' instead if the string length of the given string is less than 3, leave it unchanged.**

def modify\_string(word):

if len(word) < 3:

return word

elif word.endswith('ing'):

return word + 'ly'

else:

return word + 'ing'

input\_string = input("Enter a string: ")

result = modify\_string(input\_string)

print("Modified string:", result)

**20) Write a Python program to find the first appearance of the substring 'not' and 'poor' from a given string, if 'not' follows the 'poor', replace the whole 'not'...'poor' substring with 'good'. Return the resulting string.**

def replace\_not\_poor\_with\_good(sentence):

not\_index = sentence.find('not')

poor\_index = sentence.find('poor')

if not\_index != -1 and poor\_index != -1 and not\_index < poor\_index:

sentence = sentence[:not\_index] + 'good' + sentence[poor\_index + 4:]

return sentence

input\_string = input("Enter a sentence: ")

result = replace\_not\_poor\_with\_good(input\_string)

print("Modified sentence:", result)

**21) Write a Python function that takes a list of words and returns the length of the longest one.**

def find\_longest\_word(words):

if not words:

return 0

longest\_word = max(words, key=len)

return len(longest\_word)

word\_list = ["apple", "banana", "strawberry", "kiwi"]

result = find\_longest\_word(word\_list)

print("The length of the longest word is:", result)

**22) Write a Python function to reverses a string if its length is a multiple of 4.**

def reverse\_if\_multiple\_of\_4(s):

if len(s) % 4 == 0:

return s[::-1]

else:

return s

input\_string = input("Enter a string: ")

result = reverse\_if\_multiple\_of\_4(input\_string)

print("Result:", result)

**23) Write a Python program to get a string made of the first 2 and the last 2 chars from a given a string. If the string length is less than 2, return instead of the empty string.**

def first\_and\_last\_two\_chars(s):

if len(s) < 2:

return ''

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

input\_string = input("Enter a string: ")

result = first\_and\_last\_two\_chars(input\_string)

print("Result:", result)

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

def insert\_string\_middle(main\_string, insert\_string):

middle\_index = len(main\_string) // 2

return main\_string[:middle\_index] + insert\_string + main\_string[middle\_index:]

main = input("Enter the main string: ")

insert = input("Enter the string to insert: ")

result = insert\_string\_middle(main, insert)

print("Resulting string:", result)