# # Python🐍 Programming

# ## 1. Recursively find the factorial of a natural number

# In[1]:

def recursive\_factorial(n):

if n < 1:

return print(f"Sorry {n}, is not a natural number. Please enter a Natural Number", end = "\n")

elif n == 1:

return n

else:

return n\*recursive\_factorial(n-1)

print(recursive\_factorial(6), end="\n")

print(recursive\_factorial(1), end="\n")

recursive\_factorial(0)

# ## 2. Write a Recursive code to find the sum of all elements in a list

# In[2]:

def recursive\_list\_sum(arr: list):

n = len(arr)

if len(arr)== 1:

return arr[0]

else:

return arr[0]+ recursive\_list\_sum(arr[1:])

recursive\_list\_sum([1,2,3])

# ## 3. Write a Recursive code to compute the $n^{th}$ fibonacci number

# In[3]:

def recursive\_fibonacci(n):

if n <= 1:

return n

else:

return(recursive\_fibonacci(n-1) + recursive\_fibonacci(n-2))

recursive\_fibonacci(4), recursive\_fibonacci(5), recursive\_fibonacci(6), recursive\_fibonacci(7)

# ## 4. Read a text file by line and display every word seperated by a #

#

# Here we read the `sample.txt` which contains the following text

#

# ```

# For instance, on the planet Earth, man had always assumed that he was more intelligent than dolphins because he had achieved so much—the wheel, New York, wars and so on—whilst all the dolphins had ever done was muck about in the water having a good time. But conversely, the dolphins had always believed that they were far more intelligent than man—for precisely the same reasons.

# ```

# In[4]:

with open("sample.txt", "r") as file:

data = file.readlines()

for i in data:

print(i.replace(" ", "#"))

# ## 5. Read a text file and display the number of Vowels, Consonants, Uppercase Characters and Lowercase Characters

# In[5]:

vowels = ["a" ,"e" ,"i" ,"o" ,"u"]

consonants = ["b" ,"c" ,"d" ,"f" ,"g" ,"h" ,"j",

"k", "l", "m", "n", "p", "q", "r",

"s", "t", "v", "w", "x", "y", "z"]

with open("sample.txt", "r") as file:

# Read File

data = file.read()

# Initialize counter

count\_c = 0

count\_v = 0

count\_upper = 0

count\_lower = 0

# Loop through Text

for c in data:

if (c.islower()):

count\_lower +=1

elif(c.isupper()):

count\_upper +=1

c = c.lower()

if c in vowels:

count\_v += 1

elif c in consonants:

count\_c += 1

print(f"""Number of Consonants is {count\_c}, Number of Vowels is {count\_v}, Number of Uppercase

characters is {count\_upper}, Number of Lowercase characters is {count\_lower}""")

# ## 6. Create a Binary File, with name and roll number. Search for a given roll number and display the name, if not found display appropriate error message

# In[6]:

name = "Saurav"

roll = 37

with open("sample.bin", "wb") as file:

file.write(f"{name},{roll}".encode('utf8'))

with open("sample.bin", "rb") as file:

data = file.read().decode('utf8')

if (name in data) and (str(roll) in data):

print("Found!")

else:

print("Not Found, run again")

# ## 7. Create a Binary File with roll number, name and remarks. Input a roll number and update the marks

# In[7]:

name = "Saurav"

roll = 37

marks = 100

with open("record.bin", "wb") as file:

file.write(f"{name},{roll},{marks}".encode('utf8'))

with open("record.bin", "r+b") as file:

data = file.read().decode('utf8')

if (str(roll) in data):

marks += 100

file.write(f"{name},{roll},{marks}".encode('utf8'))

print("Record Updated")

else:

print("invalid roll number")

# ## 8. Remove all the lines that contain the character 'a' in a file and write it to another file

# In[8]:

with open("sample.txt","r") as file:

arr = file.readlines()

dump = open('dump.txt','w+')

for i in arr:

if 'a' in i:

i = i.replace('a','')

dump.write(i)

dump.close()

# ## 9. Write a random number generator that generates random number between 1 and 6

# In[9]:

import random

def roll\_dice():

return random.randint(1, 6)

roll\_dice()

# ## 10. Write a Python Program to implement Stack and Queue using list data-structure

# In[10]:

def push(arr,item):

arr.append(item)

top=len(arr)-1

def pop(arr):

if (len(arr) == []):

print("Underflow of items")

else:

item=arr.pop()

if len(arr)==0:

top=None

else:

top=len(arr)

print(f"Popped item is {str(item)}")

stack=[]

top=None

push(stack,1)

push(stack,2)

push(stack,3)

print(stack, end = "\n")

pop(stack)

print(stack)

# ## 11. Take a sample of ten phishing e-mails (or any text-file) and find most commonly occuring word(s)

# In[11]:

def enqueue(data):

queue.insert(0,data)

def dequeue():

if len(queue)>0:

return queue.pop()

return ("Queue Empty!")

queue=[]

enqueue(5)

enqueue(6)

enqueue(9)

print(queue, end="\n")

print(f"Upon Popping, popped element is {dequeue()}")

# In[12]:

from collections import Counter

with open('sample.txt', 'r') as file:

corpus = file.read().split()

count = Counter(corpus)

print(count.most\_common(1))