17/10/23

CHAPTER- 4

DATA FILE HANDLING

Q WAP to read a file ABC.txt and count the number of characters.

Sol.
```
f = open("ABC.txt", 'r')
data = f.read()
c=0
for i in data:
    c+=1
print("No. of characters :", c)
f.close()
```

Q WAP to read para.txt and count

(i) upper case
(ii) lower case
(iii) spaces
(iv) digits
(v) vowels
(vi) ~~m or n~~
(vii) m or n

```
f = open("para.txt", 'r')
data = f.read()
c_upper = c_lower = c_spaces = c_digits = c_vowels = c_m_n = 0
for char in data:
    if char.isupper():
        c_upper += 1
    elif char.islower():
        c_lower += 1
    elif char.isspace():
        c_spaces += 1
```

```
elif char.isdigit():
    c_digits += 1

if char in "aeiouAEIOU":
    c_vowels += 1
elif char in "mnMN":
    c_m_n += 1

print("No. of uppercase letters:", c_upper)
print("No. of lowercase letters:", c_lower)
print("No. of spaces:", c_spaces)
print("No. of digits:", c_digits)
print("No. of vowels:", c_vowels)
print("No. of m or n:", c_m_n)

f.close()
```

Q WAF copy() to read data from abc.txt and write all the vowels to xyz.txt.

Sol.
```
def copy():
    f_abc=open("abc.txt")
    f_xyz = open("xyz.txt", 'w')
    data = f_abc.read()
    for char in data:
        if char in "aeiouAEIOU":
            f_xyz.write(char)
    f_abc.close()
    f_xyz.close()
```

Q WAF to read a file para.txt and count word 'hello' / 'hello'

Sol.
```
def count_hello():
    f = open("para.txt")
```

```
data = f.read()
words = data.split()
c = 0
for word in words:
    if words in ['hello', 'Hello']:
        c += 1
print("Hello:", c)
f.close()
```

Q WAF to count the number of words in a file abc.txt

Sol.
```
def count_words():
    f = open("abc.txt")
    words = f.read().split()
    c = 0
    for word in words:
        c += 1
    print("No. of words:", c)
    f.close()
```

Q WAF count_four() to count those words in a file para.txt having length less than five.

Sol.
```
def count_four():
    f = open("para.txt")
    words = f.read().split()
    c = 0
    for word in words:
        if len(word) <= 4:
            c += 1
    print("No. of words having length < 5:", c)
    F.close()
```

Q WAF to read a file para.txt and count 'this' and 'that'

Sol.
```
def count_this_that():
    f = open("para.txt")
    words= f.read().split()
    for word
    c_this = c_that =0
    for word in words:
        if word.lower() == 'this':
            c_this +=1
        elif word.lower() == 'that':
            c_that +=1
    print("No. of this:", c_this)
    print("No. of that:", c_that)
    f.close()
```

Q WAF to read a file para.txt and count 'this' or 'that'.

Sol.
```
def count_this_or_that():
    f = open("para.txt")
    words = f.read().split()
    c =0
    for word in words:
        if word.lower() in ('this', 'that'):
            c +=1
    print("No. of this or that:", c)
    f.close()
```

Q WAF to read a file para.txt and count words ending with 'a'.

Sol.
```
def count_ending_a():
    f = open("para.txt")
    words = f.read().split()
    c=0
    for word in words:
        if word[-1] == 'a':
```

c += 1
print("No. of words ending with a:", c)
f.close()

Q WAF in python to read the data from para.txt and count number of lines starting with vowel.

Sol.
```
def count_lines_vowel():
    f = open("para.txt")
    lines = f.readlines()
    c = 0
    for line in lines:
        if line[0] in "aeiouAEIOU":
            c += 1
    print("Number of lines starting with vowels:", c)
    f.close()
```

Q WAF in Python to read a file abc.txt and copy all the lines having first letter h to xyz.txt

Sol.
```
def copy_abc_to_xyz():
    f_abc = open("para.txt")
    f_xyz = open("xyz.txt", 'w')
    lines = f_abc.readlines()
    for word is words:
        if word.lower()
```

```
def copy_abc_to_xyz():
    f_abc = open("para.txt")
    f_xyz = open("xyz.txt", 'w')
    lines = f_abc.readlines()
    for line in lines:
        if line[0] in 'hH':
            f_xyz.writeline(line)
f_abc.close()
f_xyz.close()
```

Q WAP in Python to add five names in a file emp.txt, (already exists)

Sol.
```
f = open("emp.txt", 'a')
for _ in range(5):
    name = input("Enter name:")
    f.write(name + "\n")
f.close()
```

Q WAF in Python to read a file abc.txt and count number of digits and number of spaces

Sol.
```
def count_digits_and_spaces():
    f = open("abc.txt")
    c_digits = c_spaces = 0
    data = f.read()
    for char in data:
        if char.isdigit():
            c_digits += 1
        elif char.isspace():
            c_spaces += 1
    print("Number of digits:", c_digits)
    print("Number of spaces:", c_spaces)
    f.close()
```

☆ BINARY FILE OPERATIONS

①
```
def create():
    import pickle
    f = open("emp.dat", 'wb')
    L = []
    L1 = ['empno', 'name', 'salary']
    pickle.dump(L1, f)
```

pickle.

```
for i in range (10):
    en = input (" Enter empno: ")
    name = input(" Enter name : ")
    sal = int ( input (" Enter salary : "))
    ls = [en, name, sal]
    l.append (ls)
pickle. dump (l, f)
f.close()
```

② def create():

```
    import pickle
    f = open ('emp. dat', 'wb')
    l = [ ]
    li = ['empno', 'name', 'salary']
    pickle. dump (li, f)

    while True:
        en = input (" Enter emp no: ")
        name = input (" Enter emp name : ")
        sal = int (input (" Enter emp salary: "))
        ls = [en, name, sal]
        l. append (ls)
        ch = input (" Do you want to continue (y, n) : "). lower()
        if ch == 'n' :
            break
    pickle. dump (l, f)
    f. close()
```

③ def create():

```
    import pickle
    f = open (" school. dat ", 'wb')
```

```
l = []
li = ['rollno', 'name', 'class', 'age']
pickle.dump(li, f)

While True:
    rollno = int(input("Enter stu rollno: "))
    name = input("Enter stu name: ")
    class = int(input("Enter stu class: "))
    age = int(input("Enter stu age: "))
    ls = [rollno, name, class, age]
    l.append(ls)
    ch = input("Do you want to continue (y/n): ").lower()
    if ch == 'n':
        break
pickle.dump(l, f)
f.close()
```

④
```
import pickle
f = open("emp.dat", 'rb')
records = pickle.load(f)
for record in records:
    eno = record[0]
    eno, ename, esal = record
    print(eno, ename, esal)
f.close()
```

Q WAP to input emp no and check whether the emp no is available in emp.dat or not.

Sol.
```
import pickle
f = open("emp.dat", 'rb')
records = pickle.load(f)
```

```
eno = input("Enter eno for searching : ")
for record in records:
    if record[0] == eno :
        print ("Record found")
        print(record[0], record[1], record[2])
        break
else:
    print("Record not found")
f.close()
```

Q WAF to read a file emp.dat and prints all the records.

Sol.
```
def read_emp():
    import pickle
    f = open("emp.dat", 'rb')
    records = pickle.load(f)
    for record in records:
        print (record)
    f.close()
```

Q WAF to search a record according to emp no in a file emp.dat. (emp no passed as an argument)

Sol.
```
def search(emp_no):
    import pickle
    f = open("emp.dat", 'rb')
    records = pickle.load(f)
    for record in records:
        if record[0] == emp_no:
            print("Record found")
            print(record)
    else:
        print("Record not found")
f.close()
```

Q WAP to read a file emp.dat and count the number of records having salary greater than 60000.
Record structure is 'emp_no, name, salary'

```
from pickle import load
def count_records():
    f = open("emp.dat", "rb")
    records = load(f)
    c = 0
    for record in records:
        if record[2] > 60000:
            c += 1
    print("No. of records having salary > 60000:", c)
    f.close()
```

★ CSV FILE HANDLING

Q Write a function in Python to read a file emp.csv and count the number of employees having salary greater than 50000.

Structure of record: Emp no, Emp name, Emp age, Salary

Sol.

```
import csv

def count_records():
    f = open("emp.csv", "r")
    csv_reader = csv.reader(f)
    record_num = 0
    for record in csv_reader:
        if csv_reader.line_num > 0 and int(record[3]) > 50000:
            record_num += 1
    print("Number of employees having salary > 50000 :", record_num)
    f.close()
```

- WRITING TO CSV FILE

```
import csv
f = open("emp.csv", "w", newline = '')
L = []
writer = csv.writer(f)
writer.writerow(['Emp No', 'Name', 'Salary'])

while True:
    emp-no = int(input("Enter emp no : "))
    name = input(" Enter employee name: ")
    Salary = float(input(" Enter salary : "))
    L.append([emp-no, name, salary])

    choice = input(" Do you want to continue (y/n): ")
    if choice == in ['nN']:
        break

writer.writerows(L)
f.close()
```