1) word count. (single function pass file.read file and count each words count, make the output in the same location in csv format).

A screen shot of a computer program

Description automatically generated

A screenshot of a google sheet

Description automatically generated A close up of a text

Description automatically generated

import os

def fileread():

#create an file called practise

file=open("practise.csv","w")

file.write('uwckwuckuh')

file.close()

#append the file

file=open("practise.csv","a")

file.write("iksdhncwi\n cat and dog \n mouse and cat\n")

file.close()

# read the file

#count the no of words in the file

file=open("practise.csv","r")

content=file.read()

word=content.split()

num\_of\_words=len(word)

print("num\_of\_words",num\_of\_words)

file.close()

fileread()

2) ip validation( ipv4 and ipv6 have different function and validate).

Ipv4 validation

A screen shot of a computer code

Description automatically generated

A screenshot of a computer

Description automatically generated

def validate\_ipv4(ipv4):

parts = ipv4.split('.')

if len(parts) != 4:

return False

for part in parts:

if not part.isdigit():

return False

num = int(part)

if num < 0 or num > 255:

return False

return True

ipv4 = input("Enter the IPv4 address: ")

print(validate\_ipv4(ipv4))

ipv6 validation:

A computer code with colorful text

Description automatically generated

A screenshot of a computer code

Description automatically generated

def validate\_ipv6(ipv6):

parts = ipv6.split(':')

if len(parts) != 8:

return False

for part in parts:

if not (1 <= len(part) >=4):

return False

try:

int(part,16)

except ValueError:

return False

return True

ipv6=input("Enter the ipv6 :")

print(validate\_ipv6(ipv6))

3) String validation= c6\*Dwu35ae2024 ( String should have last 4 digit as number,

3rd character should be \*, 4th character should be uppercase, from last 5th character should be alphabet, total char should be <=14). if failure each validation should be printed which dont obey. if success return success.

A screenshot of a computer code

Description automatically generated

A screen shot of a computer

Description automatically generated

def validate\_string(strr):

if len(strr) > 14:

return "Failure: The length should be less than or equal to 14"

if strr[2] != '\*':

return "Failure :The 3rd character should be \*"

if not strr[3].isupper():

return "Failure: 4th character should be uppercase"

if not strr[-5].isalpha():

return "Failure: last 5th character should be alphabet"

if not strr[-4:].isdigit():

return "Failure: The last four digit should be numbers"

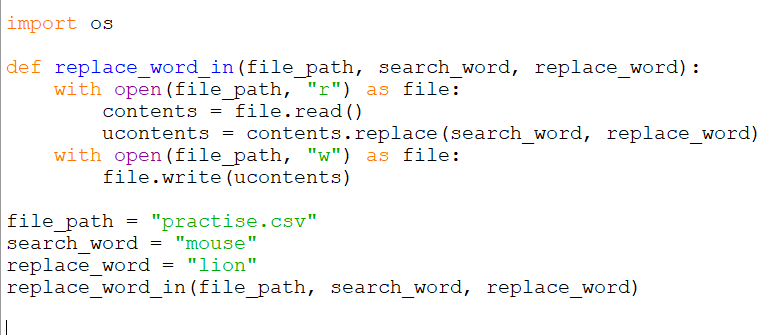
else:

return "Success"

strr=input("Enter the string: ")

print(validate\_string(strr))

4)find and replace. take file path as input, what word , which word is used to replace.



A screenshot of a computer

Description automatically generated

import os

def replace\_word\_in(file\_path, search\_word, replace\_word):

with open(file\_path, "r") as file:

contents = file.read()

ucontents = contents.replace(search\_word, replace\_word)

with open(file\_path, "w") as file:

file.write(ucontents)

file\_path = "practise.csv"

search\_word = "mouse"

replace\_word = "lion"

replace\_word\_in(file\_path, search\_word, replace\_word)

5)Find the next highest number by swapping the digit of the input given. ex 364179 -> 364197 swap two numbers to get next greater number.

A computer code with numbers and letters

Description automatically generated

A screenshot of a computer

Description automatically generated

def validate\_num(num):

lst=list(str(num))

for i in range(len(lst)-1, 0, -1):

if lst[i]>lst[i-1]:

lst[i-1],lst[i]=lst[i],lst[i-1]

return int("".join(lst))

return -1

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

result=validate\_num(num)

if result == -1 :

print("In the given number we cant find the highest number")

else:

print("The next highest number is",result)

core python poc use case.