'''

author =

'''

TEXTS = ['''

Situated about 10 miles west of Kemmerer,

Fossil Butte is a ruggedly impressive

topographic feature that rises sharply

some 1000 feet above Twin Creek Valley

to an elevation of more than 7500 feet

above sea level. The butte is located just

north of US 30N and the Union Pacific Railroad,

which traverse the valley. ''' ,

'''At the base of Fossil Butte are the bright

red, purple, yellow and gray beds of the Wasatch

Formation. Eroded portions of these horizontal

beds slope gradually upward from the valley floor

and steepen abruptly. Overlying them and extending

to the top of the butte are the much steeper

buff-to-white beds of the Green River Formation,

which are about 300 feet thick.''',

'''The monument contains 8198 acres and protects

a portion of the largest deposit of freshwater fish

fossils in the world. The richest fossil fish deposits

are found in multiple limestone layers, which lie some

100 feet below the top of the butte. The fossils

represent several varieties of perch, as well as

other freshwater genera and herring similar to those

in modern oceans. Other fish such as paddlefish,

garpike and stingray are also present.'''

]

hlavicka = '''

Textovy analyzátor: první projekt do Engeto Online Python Akademie

author: Milan Kročil

email: milan.krocil@email.cz

'''

oddelovac = ('-' \* 79)

print(hlavicka, oddelovac)

jmeno = input('Zadej prihlasovaci jmeno: ')

heslo = input('Zadej heslo: ')

registrovani\_uzivatele = {

    'bob': '123',

    'ann':'pass123',

    'mike':'password123',

    'liz':'pass123'

    }

print('username:', jmeno)

print('password:', heslo)

#print(oddelovac)

if not jmeno in registrovani\_uzivatele or registrovani\_uzivatele [jmeno] != heslo:

    #print('username:', jmeno)

    #print('password:', heslo)

    print('unregistred user, terminating the program..')

    exit()

else:

    print(oddelovac)

    print('Welcome to the app, ', jmeno)

    print('We have 3 texts to be analyzed')

    print(oddelovac)

#print('username:', jmeno)

#print('password:', heslo)

#print(oddelovac)

#text\_1 = TEXTS [0]

#text\_2 = TEXTS [1]

#text\_3 = TEXTS [2]

#print(text\_1)

#print(text\_2)

#print(text\_3)

########################################################################

# VYBER TEXTU a jeho osetreni

########################################################################

vyber\_textu = input('Enter a number btw. 1 and 3 to select: ')

print(oddelovac)

limit=3

if not vyber\_textu.isnumeric():                                                  # kontrola, ze se jedna o cislo

  print('Nebylo zadano cislo textu. Program ukoncuji...')

  exit()

if not int(vyber\_textu) in range(1,limit+1):                                     # kontrola, ze je cislo ve spravnem rozsahu

  print('Nebyl vybran spravne cislo textu. Program ukoncuji...')

  exit()

######################################################################

# VYBRAN SPRAVNY TEXT k vybrane volbe

######################################################################

texty = {

   '1' : TEXTS [0],

   '2' : TEXTS [1],

   '3' : TEXTS [2]

}

vybrany\_text = texty [vyber\_textu]

#print(vybrany\_text)

#####################################################################

# UPRAVA/OCISTENI TEXT = vymazat pecialni znaky jako tecky, carky, dvojtecky..

#####################################################################

osetreny\_text = []

for ocisteni in vybrany\_text.split():

   ocistene\_slovo = ocisteni.strip('.,!-:')

   osetreny\_text.append(ocistene\_slovo)

#print(osetreny\_text)

###################################################################

# VYPOCET POCET SLOV

###################################################################

pocet\_slov = len(osetreny\_text)

#print('There are ', pocet\_slov, 'words in he selected text.')

list\_pocet\_cisel = []

for cisla in osetreny\_text:

  if cisla.isnumeric():

    list\_pocet\_cisel.append(cisla)

#################################################################

# VYPOCTY CETNOSTI SLOV (malymi, velkymi, zacinajici velkym) , CISEL a JEJICH SOUCTU

pocet\_cisel = len(list\_pocet\_cisel)

#print('Celkovy pocet cisel v textu je: ', pocet\_cisel)

list\_pocet\_cisel = []

list\_pocet\_slov\_zacinajicich = []

list\_pocet\_slov\_velkymi\_pismeny = []

list\_pocet\_slov\_malymi\_pismeny = []

sum\_cisel = 0

for cisla in osetreny\_text:

  if cisla.isnumeric():

    list\_pocet\_cisel.append(cisla)

    sum\_cisel = sum\_cisel + int(cisla)

  elif cisla.isupper() and cisla.isalpha():

    list\_pocet\_slov\_velkymi\_pismeny.append(cisla)

  elif cisla.islower():

    list\_pocet\_slov\_malymi\_pismeny.append(cisla)

  elif cisla.capitalize():

    list\_pocet\_slov\_zacinajicich.append(cisla)

pocet\_cisel = len(list\_pocet\_cisel)

pocet\_slov\_zacinajici\_velkymi\_pismeny = len(list\_pocet\_slov\_velkymi\_pismeny)

pocet\_slov\_zacinajici\_malymi\_pismeny = len(list\_pocet\_slov\_malymi\_pismeny)

pocet\_slov\_zacinajicich\_velkyma = len(list\_pocet\_slov\_zacinajicich)

celkovy\_pocet\_slov = len(osetreny\_text)

#print(osetreny\_text)

print('There are ', celkovy\_pocet\_slov, 'words in the selected text.')

print('There are ', pocet\_slov\_zacinajicich\_velkyma, 'titlecase words.')

print('There are ', pocet\_slov\_zacinajici\_velkymi\_pismeny, 'uppercase words.')

print('There are ', pocet\_slov\_zacinajici\_malymi\_pismeny, 'lowercase words.')

#print('There are ', pocet\_slov\_zacinajicich\_velkyma, 'titlecase words.')

#print('Celkovy pocet slov je: ', celkovy\_pocet\_slov)

print('There are ', pocet\_cisel, 'numeric strings.')

print('The sum of all the numbers ', sum\_cisel)

print(oddelovac)

########################################################################

# SLOUPCOVY GRAF

########################################################################

from collections import Counter

serazeny\_seznam = sorted(osetreny\_text, key = len)

#print(serazeny\_seznam)

#delka = {}.fromkeys(serazeny\_seznam,0)

pridana\_delka = []

#print(delka)

for delka\_hodnoty in serazeny\_seznam:

  pridana\_delka.append(len(delka\_hodnoty))

#print(pridana\_delka)

cetnost = (Counter(pridana\_delka))

#print(cetnost.most\_common)

list\_cetnosti = cetnost.most\_common()

print(f"{'LEN':3}|{'OCCURENCES': ^18}|{'NR.'}")

print(oddelovac)

for key, value in cetnost.items():

   print(f"{key:3}|{'\*' \* int(value):18}|{value}", sep="\n")

#print(type(cetnost))