

**Вежби колоквиум прва**

Да се напише python скрипта која како аргумент добива име на влезна датотека. Скриптата треба да креира излезна датотека со исто име како и влезната, само со наставка “\_out”. Скриптата треба да ја измине влезната датотека линија по линија, и во излезната датотека да ја запише линијата, така што доколку во линијата текст се наоѓа датум, датумот да дојде прв во линијата, па потоа, без менување на редоследот, да дојде и останатиот текст од линијата. Форматот на датумот е “DD-MM-YYYY”, притоа, доколку има повеќе од еден датум, се запишуваат во излезната датотека онолку линии колку што има различни датуми во влезната линија, притоа текстот што следи е без датуми. Доколку скриптата се повика со втор аргумент и доколку вториот аргумент е валиден датум, тогаш дополнително и на екран се печатат оние линии каде што го има тој датум. Зборовите во датотеката се одделени со празно место.

```
# GO DEKLARIRAS MNOZESTVOTO: datumi_unikatni = set()
# mislam deka ne treba escape znak vo regex-ot
# ne zaboravaj da gi ZATVORIS datotekite
# interesno: re.sub(nesto, go zamenuvam so ova, vo toj string)
# .contains NEMA VO STRING, VO STRING koristi IN i NOT IN
# ako sakas eden da vidis dali go ispolnuva uslovot togas (re.match(uslov,
zborot))[0]

import sys
import re

def najdi_datum(linija):
    datumi = re.findall(r'[1-31]\-[1-12]\-[0-9][0-9][0-9][0-9]', linija)
    return datumi[0]

def kolku_datum(linija)
    datumi_unikatni = set()
    datumi = re.findall(r'[1-31]\-[1-12]\-[0-9][0-9][0-9][0-9]', linija)
    if len(datumi) > 1:
        for i in datumi:
            datumi_unikatni.add(i)
    return len(datumi_unikatni)

#PROVERUVAJ dali ima dovolen broj na argumenti

if len(sys.argv) < 2:
    #ako vlezan datum nemame
    print("Nema dovolen broj na argumenti!")
    vlezan_datum = False
elif re.match(r'[1-31]\-[1-12]\-[0-9][0-9][0-9][0-9]', sys.argv[2]):
    vlezan_datum = sys.argv[2]

else: #ako vlezniot datum ne go ispolnuva uslovot
    vlezan_datum = False
vlezna = sys.argv[1]
```

```

izleзна = vlezna + "_out"

v = open(vlezna, "r")
i = open(izleзна, "w")

sodrzina = v.read().split("\n")

for linija in sodrzina:
    datum = najdi_datum(linija)
    kolku = kolku_datum(linija)
    linija_nova = re.sub(datum, '', linija)
    for br in kolku:
        i.write(datum + " " + linija_nova)
    if vlezen_datum != False and vlezen_datum in linija:
        print(linija)

v.close()
i.close()

```

Да се напише python скрипта која како аргументи добива бројка и листа од имиња на датотеки. Првиот аргумент (бројката) означува колку од датотеките се влезни, останатите се излезни датотеки. Скриптата треба да провери дали има барем една излезна датотека според бројката. Потоа, треба да направи статистика за секоја излезна датотека, дали името на излезната датотека (без екстензија) се појавува како збор во содржината на влезните датотеки. Во излезните датотеки, во секоја редица одделно, се запишува колку пати е пронајдено името на излезната датотека, за секоја влезна датотека, притоа доколку името на излезната датотека не се наоѓа во содржината на некоја влезна датотека, се запишува вредност 0.

```

import sys
import re

# mozi i so re.match od kike.txt vrakja samo kike
ime_bez_ekstenzija = (re.match(r'[a-zA-Z]', izleзна_datoteka))[0]

# UBO E, vidi so re.compile
def ime_izleзна(ime):
    pattern = re.compile(r'[a-zA-Z]+')
    result = pattern.findall(ime)

    if len(result) > 0:
        print(result[0])

if len(sys.argv) < 1:
    print("Nedovolen broj na argumenti")
    sys.exit(1) # kako return;
broj_argumenti = len(sys.argv)
broj_vlezni = int(sys.argv[1])

```

```

broj_izlezni = broj_argumenti - broj_vlezni - 2 # ova -2 e za imeto na fajlot
i brojot na vlezni arg

if broj_izlezni >= 1:
    print("Ima: " + broj_izlezni + " izlezni datoteki")

    do_kade_v = 2 + broj_vlezni
    niza_vlezni = sys.argv[2:do_kade_v]
    niza_izlezni = sys.argv[do_kade_v:]
    for i in niza_izlezni:
        izlezna = open(i, "w")
        ime_bez = ime_izlezna(i) # imeto bez ekstenzija
        for v in niza_vlezni:
            vlezna = open(v, "r")
            sodrzina_vlezna = vlezna.read()
            go_ima = re.findall(ime_bez, sodrzina_vlezna)
            kolku = len(go_ima)

            # arno ama vika da se zapisi ovaj broj vo sekoja linija od
            izleznata datoteka
            izlezna_linii = izlezna.read().split('\n')
            for linija in izlezna_linii:
                linija.write(kolku)

            vlezna.close()
            izlezna.close()

else:
    print("Nema izlezni datoteki")
    sys.exit(1)

```

#### **14.04.2015 прва**

Да се напише python скрипта која како аргументи добива имиња на две датотеки. Скриптата треба да ја измине втората датотека, да ги детектира сите наредби кои што се наоѓаат во втората датотека, и секоја наредба, заедно со аргументите, да ја испише во посебна линија на екран. Секоја наредба започнува со името на наредбата, после кое следат низа аргументи (или нема ниту еден аргумент). Имињата на наредбите се сместени во првата датотека т.е. датотеката што се испраќа како прв аргумент. Секоја наредба е сместена во нов ред во првата датотека.

```

import sys
import re

# vo prvata imam iminja na naredbi vo posebna redica
# a vo vtorata imeto na naredбата i argumentite

if len(sys.argv) < 3:
    print("Nedovolno broj na argumenti!")

```

```

        sys.exit(1)

prva = sys.argv[1]
vtora = sys.argv[2]

dat1 = open(prva,"r")
dat2 = open(vtora, "r")

# NE MOZI VAKA : naredbi = dat1.readline() !!!!

naredbi = dat1.read().split("\n")
sodrzina_vtora = dat2.read()

for i in range(0, len(naredbi):
    komplet = ""
    for j in range(0, len(sodrzina_vtora):
        if naredbi[i] == sozdrzina_vtora[j]:
            print("Ja najdovme naredbata, gi zemame argumentite sega")
            while(sodrzina_vtora[j] not in naredbi):
                komplet+=sodrzina_vtora(j)
    print(komplet)

dat1.close()
dat2.close()

```

Да се напише python скрипта која како аргументи добива имиња на две датотеки и број. Скриптата треба да прочита онолку линии текст од тастатура колку што е бројката (пратена како трет аргумент од командна линија). На почетокот од секоја линија е сместено името на командата, а потоа во продолжение се наведени аргументите на командата. Скриптата треба во првата датотека да ги смести имињата на командите, секоја во нов ред, додека пак во втората датотека, секоја една команда да ја смести во еден ред со прилепување (заедно со аргументите на командата). На крај, скриптата треба да отпечати на екран секоја команда заедно со аргументите, секоја во посебен ред.

```

# VNESUVANJE OD TASTATURA
import sys
import re

if len(sys.argv) < 4:
    print("Nedovolen broj na argumenti!")
    sys.exit(1)

broj_linii = sys.argv[3]

file1 = sys.argv[1]
file2 = sys.argv[2]
niza_linii = []

```

```

print("Vnesuvaj komandi dodeka mozes\n")
while broj_linii > 0:
    linija = input("Vnesi komanda + argumenti\n")
    niza_linii.append(linija)

komandi_za_prva = []

for i in range(0, len(niza_linii)):
    komandi_za_prva.append(niza_linii[i].split(" ")[0])

dat1 = open(file1, "w")

for komanda in komandi_za_prva:
    dat1.write(komanda)
    dat1.write("\n")

dat1.close()

dat2 = open(file2, "w")
for linija in niza_linii:
    dat2.write(linija + "\n")

dat2.close()

```

Да се напише python скрипта која како аргументи од командна линија добива имиња на датотеки. Скриптата треба да проверува две по две датотеки (секоја датотека со секоја останата датотека) и да споредува колку исти Хаштагови има (збор составен со букви и бројки кој започнува со специјалниот знак #). Истите Хаштагови и вкупниот број на Хаштагови се запишува во излезна датотека со име креирано од спојување на имињата на двете датотеки кои што се споредуваат. На крај, на екран се печати кој е најчесто користениот Хаштаг во сите датотеки.

```

import sys
import re

def najdi_hastag(ime_datoteka):
    dat = open(ime_datoteka, "r")
    sodrzina_dat = dat.read()
    niza1 = re.findall(r'#\w+\d+', sodrzina_dat)
    dat.close()
    return niza1

def funkcija(niza1, niza2, nova_ime):
    nova = open(nova_ime, "w")
    for i in range(0, len(niza1)):

```

```

        count = 0
        for j in range(0, len(niza2)):
            if niza1[i] == niza2[j]:
                count++
        nova.write(niza1[i], count)
        nova.close()

def max_koristen_hastag(niza_site_hastag):
    max = 0
    hastags=""
    najkoristen=""

    # za da ja napravime nizata vo string
    for i in niza_site_hastag:
        hastags += i

    for i in range(0, len(niza_site_hastag)):
        hastag = re.findall(niza_site_hastag[i], hastags)
        if max < len(hastag):
            max = len(hastag)
            najkoristen = hastag[0]

    print("Najkoristen: " + najkoristen)
    print("Broj pati iskoristen: " + max)

broj_argumenti = len(sys.argv[1:])

niza_datoteki = sys.argv[1:]

niza_site_hastag = []

for i in range(0, broj_argumenti)):
    niza1 = najdi_hastag(niza_datoteki[i])
    niza_site_hastag.extend(niza1)
    for j in range(0, broj_argumenti)):
        if niza_datoteki[i] == niza_datoteki[j]:
            continue
        else:
            niza2 = najdi_hastag(niza_datoteki[j])
            niza_site_hastag.extend(niza2)
            nova_ime = niza_datoteki[i] + niza_datoteki[j]
            funkcija(niza1, niza2, nova_ime)

```

```

# VIDI GO OVA-----
from collections import Counter

niza = [1, 1, 1, 1, 1, 2, 3]

# Use Counter to count occurrences of each element
counter = Counter(niza)

# Find the most common element
most_common_element = counter.most_common(1)[0][0]

print(most_common_element)

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Решение: import re
import sys
from collections import Counter

def extract_hashtags(text):
    return re.findall(r'\#\w+', text)

def compare_files(file1, file2, output_file):
    with open(file1 'r') as f1 open(file2 'r') as f2:
        content1 = f1.read()
        content2 = f2.read()

    hashtags1 = extract_hashtags(content1)
    hashtags2 = extract_hashtags(content2)

    common_hashtags = set(hashtags1) & set(hashtags2)
    total_hashtags = len(common_hashtags)

    with open(output_file, 'w') as of:
        of.write(f"Common Hashtags: {total_hashtags}\n")
        of.write(f"Hashtags from {f1}: {len(hashtags1)}\n")
        of.write(f"Hashtags from {f2}: {len(hashtags2)}\n")

        for hashtag in common_hashtags:
            of.write(f"{hashtag}\n")

    return common_hashtags

def find_most_common_hashtag(hashtags):
    counter = Counter(hashtags)
    most_common_hashtag, most_common_count=counter.most_common(1)[0]

```

```

        return most_common_hashtag, most_common_count

if len(sys.argv)<4:
    print("Error")
    sys.exit(1)

file1 = sys.argv[1]
file2 = sys.argv[2]

output_file = sys.argv[3]

common_hashtags = compare_files(file1, file2, output_file)
find_most_common_hashtag, most_common_count =
find_most_common_hashtag(common_hashtags)

```

### 31.01.2019

Да се напише PERL скрипта која како аргумент добива име на датотека во која се запишани

контакти од даден именик. Во посебен ред од датотеката се наоѓа секој еден од контактите, притоа во редот најпрво е редниот број на контактот, потоа одделен со празно место е телефонскиот број, а потоа исто така одделен со празно место се сместени името и презимето. Контактите треба да се сместат во Хеш структура, каде што клуч ќе е телефонскиот број, а како вредност ќе се чува името и презимето на контактот. Потоа, скриптата треба да дозволи да се внесуваат непознат број на команди од тастатура. Доколку од тастатура се внесе 1, тогаш низ именикот се пребарува по телефонски број, што значи дека после тоа се внесува телефонскиот број кој се пребарува а на екран се печати името и презимето на тој телефонски број (или дека контактот не постои). Доколку се внесе 2, тогаш се пребарува по име и презиме и на екран се печати телефонскиот број за дадено име и презиме внесени од тастатура. Доколку се внесе 3, се печати целиот именик со броевите придружени со име и презиме. Доколку се внесе вредност различна од 1, 2 или 3, тогаш се прекинува со командите и скриптата завршува.

1. 076781325 Kristina Kalamukoska
2. 071111111 Mia Darkoska
3. 07888888 Gordana Kalamukoska

```

redenbroj  broj  ime prezime
           kluc  vrednost

```

```

import sys
import re

```

```

if len(sys.argv) < 1:

```



```

        print("Nedovolen broj na argumenti!")
        exit(1)

datoteka = sys.argv[1]
f = open(datoteka, "r")

kontakti = {}

sodrzina_linii = f.read().split('\n')
for linija in range(0, len(sodrzina_linii)):
    zborovi = linija.split(" ")
    kontakti[zborovi[1]] = zborovi[2] + zborovi[3]

print("Komandi na raspolaganje: 1, 2, 3")

while true:
    komanda = import("Vnesete komanda.\n")
    if komanda == 1 || komanda == 2 || komanda == 3:
        if komanda == 1:
            telbr = import("Vnesete go telefonskiot broj sto sakate da
go prebarate!")
            site_telbr = kontakti.keys()
            if telbr in site_br:
                print("Telefonskiot broj postoi i negoviot
sopstvenik e: \n")
                print(kontakti.get(telbr))

            elif komanda == 2:
                ime_prezime = import("Vnesete go imeto i prezimeto na
kontaktot\n")
                site_vrednosti = kontakti.values()
                if ime_prezime in site_vrednosti:
                    print("Brojot na kontaktot sto go baravte e:
najdi_broj(ime_prezime, kontakti)")
                elif komanda == 3:
                    for x in kontakti:
                        print("Broj: " x "\n")
                        print("Kontakt: " kontakti[x])
            else:
                print("Vnesovte nepravilna komanda!")
                break;

def najdi_broj(ime_prezime, kontakti):
    for x in kontakti:
        if(kontakti[x] == ime_prezime):
            return x;

```

## 2022 kolokvium

Да се напише PERL скрипта со име „prva.plx“ која како аргументи добива најпрво збор кој ќе го пребарува, опција (знаменце) и име на датотека во која ќе се прави пребарувањето. Скриптата треба да го пребара зборот (што се праќа како прв аргумент) во зависност од опцијата (што се праќа како втор аргумент) во датотеката (чие име се праќа како трет аргумент). Опцијата може да биде „l“, „w“ или „f“, притоа опцијата „l“ значи дека треба да се вратат броевите од линиите во кои е пронајден зборот, „w“ како резултат враќа колку зборови биле пронајдени пред зборот што се бара, и опцијата „f“ враќа колку пати е пронајден зборот во датотеката. Доколку зборот го нема во датотеката, опцијата „l“ враќа -1, додека пак опциите „w“ и „f“ враќаат соодветно вредност 0. Зборовите се одделени еден од друг со празно место. Доколку добие невалидна опција, скриптата прекинува со извршување.

Пример:

dat.txt:

Also when you use plain Lorem ipsum text, your design will look like a million other designs out there.

With Random Text Generator your designs will look more unique while still containing text which truly means nothing.

perl prvа.plx will l dat.txt

Резултат: 1 2

perl prvа.plx will w dat.txt

Резултат: 10 24

perl prvа.plx will f dat.txt

Резултат: 2

```
zbor zname(opcija) ime_datoteka
```

l -> se vrakjaat broevite od liniite vo koi e pronajden zborot

w -> kolku zborovi bile pronajdeni pred zborot

f -> kolku pati e pronajden zborot vo datotekata

```
import sys
```

```
import re
```

```
if len(sys.argv) < 3:
```

```
    print("Nmeate dovolno argumenti!")
```

```
    exit(1)
```

```
zbor = sys.argv[1]
```

```
zname = sys.argv[2]
```

```
ime_datoteka = sys.argv[3]
```

```
f = open(ime_datoteka, "r")
```

```
sodrzinaf = f.read()
```

```
if len(re.findall(zbor, sodrzinaf)) != 0:
```

```

    if zname == 'l' || zname == 'w' || zname == 'f':
        if zname == 'l':
            print(funkcijaL(zbor, sodrzinaf))
        elif zname == 'w':
            print(funkcijaW(zbor, sodrzinaf))
        else:
            print(funkcijaF(zbor, sodrzinaf))

    else:
        print("Vnesovte nevalidna opcija (zname)!\n")

else:
    print("Zborot ne se pojavuva nitu ednas vo datotekata!")


def funkcijaL(zbor, sodrzinaf):
    rednibr_od_linii = []
    sodrzinaf_linii = sodrzinaf.split('\n')
    for redenbr in range(0, len(sodrzinaf_linii)):
        if zbor in sodrzinaf_linii[redenbr]:
            rednibr_od_linii.append(redenbr)

    return rednibr_od_linii


def funkcijaW(zbor, sodrzinaf):
    br_zborovi = 0
    sodrzinaf_zborovi = sodrzinaf.split('\n').split(" ")
    for i in sodrzinaf_zborovi:
        if zbor != i:
            br_zborovi++
        else:
            break

    return br_zborovi


def funkcijaF(zbor, sodrzinaf):
    sodrzinaf_zborovi = sodrzinaf.split('\n').split(" ")
    pronajden = 0
    for i in sodrzinaf_zborovi:
        if zbor == i:
            pronajden++
        else:
            continue

```

```
return pronajden
```

#### 10.04.2019

Да се напише PERL скрипта која како аргументи од командна линија добива листа од кориснички имиња. Секое корисничко име започнува со специјалниот знак „@“ а потоа следи листа од букви и бројки. Невалидните имиња од командна линија да се игнорираат. Скриптата дозволува да бидат внесени непознат број на имиња на датотеки од тастатура (прекинува кога ќе се внесе зборот „крај“). Скриптата прави статистика за секоја една датотека, притоа статистиката вклучува секое едно корисничко име, во која линија од датотеката се појавува. Скриптата на екран ги печати (во секој ред одделно) во кој ред било пронајдено корисничкото име (бројот на редот) и кои други кориснички имиња биле пронајдени во истиот ред.

```
import sys
import re

if len(sys.argv) < 1:
    print("Nedovolen br na argumenti!")
    exit(1)

lista_site = sys.argv[1:]

# vo string za da mozam da ja upotrebam findall, dr ideja nemam
for i in lista_site:
    str_lista_site += i

validni = re.findall(r'@[a-zA-Z0-9]+' , str_lista_site)

print("Vnesuvajte iminja na datoteki. Vnesete kraj za kraj\n")
while True:
    ime_datoteka = input("Vnesete ime na datoteka\n")
    if ime_datoteka == "kraj":
        break
    else:
        statistika(ime_datoteka, validni)

def statistika(ime_datoteka, validni):
    dat = open(ime_datoteka, "r")
    sodrzinaPoLinii = dat.read().split("\n")
    for kIme in validni:
        for linija in range(0, len(sodrzinaPoLinii)):
            if kIme in sodrzinaPoLinii[linija]:
                #go ima korisnickoto ime vo taa linija
                print("Reden broj linija: " linija)
                print("Drugi korisnicki iminja pronajdeni vo istiot
red: ")
```

```

        for kImeRed in validni:
            for linijaDrugi in range(0,
len(sodrzinaPoLinii)):
                if kImeRed in
sodrzinaPoLinii[linijaDrugi]:
                    print(kImeRed)
                    print("\n") # zavrshuvame so linijata i za
korisnickoto ime i za drugi korisnicki iminja sto go ima vo redot

```

#### 10.04.2019

1. Да се напише Python скрипта која како аргументи од командна линија добива имиња на датотеки. Скриптата треба да ги измине датотеките и да провери колку различни валути на пари ќе најде. Знаците на валутите кои што се пребаруваат се: „\$“ „€“ „Ј“ „kn“ „kr“ „Rp“ „CHF“. Секоја валута започнува со знакот на валутата, по кое што следи сумата на истата (одделена со празно место). Доколку после валутата нема бројка, тогаш валутата не се зема во предвид. Сите валути што ќе бидат пронајдени во датотеката се собираат и се добива вкупна сума по валута. За секоја валута што била пронајдена во датотеката, му се дозволува на корисникот да внесе колку денари е истата (не се внесува за оние валути кои што не биле пронајдени во тековната датотека). На крајот од изминувањето на датотеката да се отпечати вкупната сума (во денари) што била пронајдена во датотеката. Скриптата, на крај треба да отпечати кои валути биле пронајдени во сите датотеки.

```

import sys
import re

#mora da e globalna bidejkji za sekoj file povikuvame funkcija i ke se resetira
ako e vnatre vo funkcijata
sumdollars=0
sumevro=0
sumkn=0
sumkr=0
sumCHF=0
sumrp=0
valuti_unikatni = set()

def extract(file_name):
    global sumdollars
    global sumevro
    global sumkn
    global sumkr
    global sumCHF
    global sumrp
    global list

    with open(file_name, 'r') as file:

```

```

sodrzina = file.read();
currencies = re.findall(r'\$ \d+\.\d+|EVRO \d+\.\d+|CHF \d+\.\d+|kn
\d+\.\d+|kr \d+\.\d+|Rp \d+\.\d+', sodrzina);

for curr in currencies:
    if str(curr).contains("$"):
        sumdollars+=float(str(curr).split(" ")[1])#za da go zememe
iznosot
        valuti_unikatni.add(str(curr).split(" ")[0])
    if str(curr).contains("EVRO"):
        sumevro+=float(str(curr).split(" ")[1])
        valuti_unikatni.add(str(curr).split(" ")[0])
    if str(curr).contains("CHF"):
        sumCHF+=float(str(curr).split(" ")[1])
        valuti_unikatni.add(str(curr).split(" ")[0])

    if str(curr).contains("kr"):
        sumkr+=float(str(curr).split(" ")[1])
        valuti_unikatni.add(str(curr).split(" ")[0])

    if str(curr).contains("Rp"):
        sumrp+=float(str(curr).split(" ")[1])
        valuti_unikatni.add(str(curr).split(" ")[0])

    if str(curr).contains("kn"):
        sumkn+=float(str(curr).split(" ")[1])
        valuti_unikatni.add(str(curr).split(" ")[0])

print(currencies) # $12.23 @ 2344.33 CHF 3232,22
                  # EVRO 34.423
list.append(currencies)

for datoteka in sys.argv[1:]:
    extract(datoteka)

denari = 0
dolar = input("Kolku e dolarot?")
evro = input("Kolku e evroto?")
kn = input("Kolku e kunata?")
kr = input("Kolku e krunata?")
rp = input("Kolku e rp?")
CHF = input("Kolku e CHF?")

```

## Python

```
denari = sumdollars*dolar + sumevro*evro + sumCHF*CHF + sumkn*kn + sumkr*kr +  
sumrp*rp  
  
print(denari)  
  
print(len(valuti_unikatni))
```

Пример:

```
if [ -z "$dali" ]
then
dat="prva.py3"
zbor=$1
shift
else
dat=$1
zbor=$2
shift
shift
fi
kolku=0
for d in $@
do
k=$(python3 $dat $zbor f $d)
kolku=$(expr $kolku + $k)
done
echo $kolku
```