1. # Да се напише Python скрипта која како аргумент добива име на датотека во која се запишани ценовник на производи. Во посебен ред од датотеката се наоѓа секој еден од производите, притоа во редот најпрво е баркодот на производот, потоа одделен со празно место е името на производот, а потоа исто така одделен со празно место е сместена цената на производот. Производите треба да се сместат во Хеш структура, каде што клуч ќе е баркодот, а како вредност ќе се чува името на производот. Треба да се креира и посебна Хеш структура каде како клуч ќе се земе повторно баркодот на производот, а како вредност ќе се чува цената на производот. Потоа, скриптата треба да дозволи да се внесуваат непознат број на команди од тастатура. Доколку од тастатура се внесе 1, тогаш низ ценовникот се пребарува по име на производ, што значи дека после тоа се внесува името на производот кој се пребарува а на екран се печати цената на производот (или дека производот не постои). Доколку се внесе 2, се листаат производите чија што цена им е помала од таа внесена од тастатура. Доколку се внесе 3, тогаш се печатат сите производи (секој производ во посебен ред) со името и цената на производот. Доколку се внесе вредност различна од 1, 2 или 3, тогаш се прекинува со командите и скриптата завршува.

#!/usr/bin/python3

import sys

if len(sys.argv) < 2:  #skripta, datoteka

    print("Gresno vnesuvanje!")

    quit()

barkodIME = {}

barkodCena = {}

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

for linija in dat:

    elementi = linija.split(" ")

    if len(elementi) >=3:

        barkod = elementi[0]

    ime = " ".join(elementi[1: -1]) #od prviot do posledniot ako ima poveke iminja jadenjeto ke gi zeme pvoeketo

    cena = float(elementi[2:])

    barkodIME[barkod] = ime

    if cena.isdigit():

        barkodCena[barkod] = cena

dat.close()

while True:

    print("1 - prebaruvaj po ime na proizvod, treba da ovozmozime i vnes na ime= se pecati cena")

    print("2 - prebaruvaj po cena, vnesuvas i cena i barash po cenata pomali od taa")

    print("3 - pecati se ")

    komanda = input("Vnesi komanda 1, 2 ili 3: ")

    if komanda == 1:

        vleznoIme = input("Vnesi ime na hrana: ").strip()

        najden = False

        for barkod, ime in barkodIME.items():

                if ime.lower() == vleznoIme.lower():

                   print(f"{ime} so cena {barkodCena[barkod]}")

                   najden = True

        if not najden:

            print("Nema proizvod so toa ime!")

    elif komanda == 2:

        vleznaCena = float(input("Vnesi cena: ").strip())

        najden = False

        for barkod, cena in barkodCena.items():

            if cena <= vleznaCena:

                    print(f"Imeto na proizvodot e {barkodIME[barkod]} so cena {cena}")

                    najden = True

        if not najden:

                  print("Nema proizvod so cena poniska od ponudenata!")

    elif komanda == 3:

            for barkod, ime in barkodIME.items():

                 print(f"Proizvod {ime}, Cena: {barkodCena[barkod]}")

    else:

        print("Nevalidna vnesena opcija!!!")

        break

2.# 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.

#!/usr/bin/python3

import sys

if len(sys.argv)<3:

    print("Gresno vnesuvanje! ")

    quit()

datoteki = sys.argv[1:] #site datoteki ke gi zemam vo ova

global\_hastag = {} #za broenje na site hash

for i in range(len(datoteki)):

    for j in range(i+1, len(datoteki)):

        dat1 = datoteki[i]

        dat2 = datoteki[j]

        hash\_tag1 = []

        hash\_tag2 = []

        prva = open(dat1, " r ")

        for linija in prva:

            zborovi = linija.strip().split(" ") #lista od site zborovi

            for zbor in zborovi:

                if zbor.startswith("#"):

                    hash\_tag1.append(zbor)

        prva.close()

        vtora = open(dat2, "r")

        for linija in vtora:

            zborovi = linija.strip().split(" ")

            for zbor in zborovi:

                if zbor.startswith("#")

                hash\_tag2.append(zbor)

        vtora.close()

        hash\_zaednicki = []

        for tag in hash\_tag1:

            if tag in hash\_tag2 and tag not in hash\_zaednicki: #dali go ima vo dr datoteka i dali ima duplikat

                hash\_zaednicki.append(tag)

        hash\_brojac = {}

        for htag in hash\_zaednicki:

            if htag in hash\_brojac:

                hash\_brojac[htag] += 1

            else:

                hash\_brojac[htag] = 1

        ime\_dat1 = dat1.split("/")[-1] #izvlekuvanje bez pat

        ime\_dat2 = dat2.split("/")[-1] #izvlekuvanje bez pat

        samo\_ime1 = ime\_dat1.split(".")[0]

        samo\_ime2 = ime\_dat2.split(".")[0]

        ime\_izlezna\_datoteka = f"{samo\_ime1}\_{samo\_ime2}.txt"

        izlezna\_dat = open(ime\_izlezna\_datoteka, "w")

        izlezna\_dat.write(f"Zaednicki hash tags pomegu {dat1} i {dat2:}")

        for htags, brojac in hash\_brojac.items():

            izlezna\_dat.write(f"{htags}: {brojac}")

        izlezna\_dat.close()

if global\_hastag:

    najcesto\_koristen\_hash\_tag = max(global\_hastag, key=global\_hastag.get)

    print(f"Najcesto koristen tag: {najcesto\_koristen\_hash\_tag} so {global\_hastag[najcesto\_koristen\_hash\_tag]} pati")

else:

    print("Nema zaednicki!")

3.Да се напише Python скрипта со име „prva.py3“ која како аргументи добива име на влезна датотека и опционален втор аргумент. Во влезната датотека има мени, така што во секој ред одделно има по едно јадење од менито во формат reden\_broj ime\_na\_jadenje kategorija\_na\_ jadenje cena. Секој дел е одделен помеѓу себе со празно место. Доколку корисникот не внесе втор аргумент, тогаш се печати целото мени, но доколку внесе аргумент, тогаш ги печати само оние јадења чија категорија е иста со вториот аргумент (ги филтрира по категорија внесена како втор аргумент од командна линија). Потоа скриптата дозволува да се внесе од тастатура три јадења од менито преку внесување на нивниот реден број. На крај скриптата треба да ги отпечати имињата на одбраните јадења од корисникот (само оние кои ги има на менито што е претходно отпечатено) и вкупната сума за наплата (вкупната цена).

#!/usr/bin/python3

import sys

if len(sys.argv) <2 or len(sys.argv) > 3:  #skripta, datoteka , vtor argument

    print("Gresno vnesuvanje!")

    quit()

dat = open(sys.argv[1], "r") #ja otvaram datotekata i citam

pomLista = []

for line in dat:

    element = line.split(" ")

    pomLista.append(element)

dat.close()

if len(sys.argv) == 3:

    kategorija = sys.argv[2]

else:

    kategorija = None

for i in pomLista:

    if kategorija is None or i[2] == kategorija:

        print(" ".join(i)) #se pecati cel red

odbrani\_jadenja = []

vkupno=0

print("Vnesi 3 jadenja: ")

for \_ in range(3): # \_ SE KORISTI ZA KOGA SAKAME NESTO DA POFTORIME 3 PATI

    broj = int(input()) #VNESI REDEN BROJ PREKU TASTATURA

    if broj>0:

        najden = False

        for jadenje in pomLista:

         if broj == int(jadenje[0]):

            jadenje = pomLista[broj-1] #INDEKSI GLEDAME ZATOA E -1

            odbrani\_jadenja.append(jadenje[1])

            vkupno += int(jadenje[3]) #na indeks 3 se naoga cenata

            najden=True

            break

         if not najden:

             print("Nevazecki")

    else:

        print("nevazecki broj")

for jadenje in odbrani\_jadenja:

    print(jadenje)

print(f"vkupno: {vkupno}")

4.# Да се напише Python 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 „kraj“). 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.

#!/usr/bin/python3

import sys

if len(sys.argv) < 2: #skripta, lista

    print("Nemate dovolno argumenti!")

    quit()

validni\_korisnicki = []

for element in sys.argv[1:]: #imam lista koja ne znae kolku argumenti ima

    if element.startswith('@'):

        validni\_korisnicki.append(element)

if len(validni\_korisnicki) == 0:

    print("Nema validni korisnicki iminja!")

    quit()

statistika = {}

while True:

    ime\_datoteka = input("Vnesi ime na datoteka - napisi kraj za da zavrsis: ").strip()

    if ime\_datoteka == 'kraj':

        break

    dat = open(ime\_datoteka, "r")

    Brojlinija = 0

    for linija in dat:

        Brojlinija += 1

        pronajdeni = []

        for ime in validni\_korisnicki:

            if ime in linija:

                pronajdeni.append(ime)

        if len(pronajdeni) > 0: # ako imame barem edno korisnicko ime vo red od datotekata

            if Brojlinija not in statistika: #dali tekovniot red postoi vo recnikot

                statistika[Brojlinija] = [] #koga ne postoi go dodavame i mu davame prazen spisok koj ke gi cuva site korisnicki iminja koi se pojavile vo redot

                statistika[Brojlinija] += pronajdeni #go dodavam spisokot pronajdeni vo toj red

    dat.close()

for linija in sorted(statistika):

    print("Linija", linija, ": ", " ".join(statistika[linija]))

5.# Да се напише Python скрипта со име „prva.py3“ која како аргументи добива две влезни датотеки. Првата датотека е код напишан во Python, додека пак втората датотека е документација која треба да се вметне во кодот. Документацијата е организирана така што секоја линија од датотеката најпрво има број, а потоа текст. Скриптата треба да ги спои двете датотеки, така што втората датотека (документацијата) треба да ја додаде во програмскиот код (првата датотека). Додавањето се прави така што бројката на почеток на секоја линија во документацијата кажува на која линија во кодот треба да биде истата вметната, а потоа, на крајот на линијата во кодот, се додава линијата од документацијата, започнувајќи со #. Новоформираниот код се печати на екран.

#!/usr/bin/python3

import sys

if len(sys.argv) != 3:  #skripta, datoteka 1, datoteka 2

    print("Gresno vnesuvanje!")

    quit()

dat2 = open(sys.argv[2], "r")

pomosnaLista = []

for linija in dat2:

    element = linija.split(" ", 1) #go delam tekstot na 2 dela, 1 znaci deka ke odi do prvoto prazno mesto samo so ova imam element[0]=brojot i element [1]=celiot tekst

    if len(element) == 2: #znam deka tocno na 2 dela mi e

        broj\_linija = int(element[0])

        tekst\_dokument = element[1].strip() # go spojuvam celiot tekst bez praznini

    pomosnaLista.append((broj\_linija, tekst\_dokument)) # isto kako da ja povikam 2 pati

    # pomosnaLista.append(broj\_linija)

    # pomosnaLista.append(tekst\_dokument)

dat2.close()

dat1 = open(sys.argv[1], "r")

site\_linii = dat1.readlines() #site linii se citaat

dat1.close()

nov\_kod = []

for i,line in enumerate(site\_linii, 1): #od kade gi zimam liniite i od koj indeks ke zapocne vo slucajov 1

    linija = line.strip() #go trgam null terminatorot

    nov\_kod.append(linija) #gi stavam vo noviot kod site linii

    for pom in pomosnaLista:

        if pom[0] == i: #i e iterator vo site\_listi a pom iterira vo pom\_lista samo na elementite [0] pozicija

            nov\_kod[-1] += " #" + pom[1] #so -1 sme na tekovniot

for line in nov\_kod:

    print(line)

6.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: „$“ „€“ „£“ „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.

#!/usr/bin/python3

import sys

if len(sys.argv) < 2:  #skripta, datoteka 1,

    print("Gresno vnesuvanje!")

    quit()

dat1 = open(sys.argv[1], "r")

valuti = {"$": 0, "€": 0, "£": 0, "kn": 0, "kr": 0, "Rp": 0, "CHF": 0}

for linija in dat1:

    zbor = linija.split(" ")

    for i in range(len(zbor) -1):

        if zbor[i] in valuti and zbor[i+1].isdigit():

            valuti[zbor[i]] += zbor[i+1]

dat1.close()

vkupno = 0

for val, iznos in valuti.items():

    if iznos > 0:

        kurs = input(f"Vnesete vrednost na {val} vo denari")

        vkupno += iznos \* kurs

print("Pronajdeni valuti: ")

for val, iznos in valuti.items():

    if iznos > 0:

        print(f"{val}: {iznos}")

print(f"Vkupno vo denari: {vkupno}")

7.# . Да се напише PYTHON скрипта за работа со датотеки. Скриптата, како аргумент од командна линија добива име на една влезна датотека. Скриптата треба да ги најде сите телефонски броеви во датотеката и сопствениците на тие телефонски броеви. Секој телефонски збор е составен од 9 цифри, притоа започнува со 07. Скриптата треба да ги најде сите различни телефонски броеви и на крај да ги отпечати. Зборот пред секој телефонски број означува кој е корисникот на кој припаѓа тој телефонски број. Скриптата треба да го отпечати на екран името на сопственикот на телефонскиот број и самиот телефонски број. Доколку во датотеката истиот телефонски број се појавува со различни корисници, тогаш се печати само последниот корисник што ќе биде пронајден во датотеката за тој телефонски број. Секој збор од друг збор во датотеката е одделен со празно место.

#!/usr/bin/python3

import sys

if len(sys.argv) < 2:  #skripta, datoteka 1

    print("Gresno vnesuvanje!")

    quit()

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

telefon = {}

for linija in dat:

    zbor = linija.split(" ")

    if len(zbor) == 2:

        ime = zbor[0]

        broj = zbor[1]

        if len(broj) == 9 and broj.isdigit() and broj.startswith("07"):

            telefon[broj] = ime

dat.close()

for broj, sopstvenik in telefon.items():

    print(f"Sopstvenik: {sopstvenik}, Telefon: {broj}")

8.# Да се напише PYTHON скрипта која како аргументи од командна линија добива имиња на две датотеки датотеки. Во првата датотека, во секоја линија одделно, се запишани наредби кои треба да ги препознава скриптата (секоја наредба е еден збор – низа од букви).Скриптата треба да ги најде сите наредби (дефинирани во првата датотека) во втората датотека. Секоја

наредба започнува со името на наредбата, после која, одделени со празно место, се наведени аргументите на таа наредба. За аргументи на дадена команда се сметаат сите зборови после командата (одделени со празно место) кои започнуваат со цртичка (-). Низата аргументи завршува кога ќе се најде првиот збор различен од цртичка, притоа може да се случи некои аргументи да се простираат и во повеќе линии од датотеката. За сите најдени наредби (со нивните аргументи) ги печати на екран. Доколку некоја наредба се појавува повеќе пати во датотеката, на екран се печати еднаш само, притоа аргументите се додаваат на првото појавување на наредбата, притоа започнуваат со „-е“ аргумент.

#!/usr/bin/python3

import sys

if len(sys.argv) != 3:

    print("Gresno vnesuvanje!")

    quit()

dat1 = open(sys.argv[1], "r")

dat2 = open(sys.argv[2], "r")

Datoteka1\_komandi = []

for linija in dat1:

    element = linija.strip() # ke gi spojam site novi linii vo edna niza

    Datoteka1\_komandi.append(element)

dat1.close()

najdeni\_komandi = []

for linija in dat2:

    element=linija.split(" ")

    if len(element) == 0: #ako listata mi e prazna nemam elementi

        continue

    #[stop, a, i, o u]

    Datoteka2\_komanda = element[0] #prviot element vo nizata da bide komanta

    if Datoteka2\_komanda in Datoteka1\_komandi:

        najdena = False

        for i in najdeni\_komandi:

            if i[0] == Datoteka2\_komanda:

                najdena=True

                for el in element[1:]: #iterira niz celata linija od datoteka2

                 if el.startswith("-"):

                    i[1].append("-e" + el[1:])

                 else:

                    break

                break

        if not najdena:

            argumenti = []

            for el in element[1:]: #trgnuvame od 1 bidejki 0 ni e komandata

                if el.startswith("-"):

                    argumenti.append("-e" + el[1:])

                else:

                    break

            najdeni\_komandi.append([Datoteka2\_komanda, argumenti])

dat2.close()

#Ispis na rezultatite

for Datoteka2\_komanda, argumenti in najdeni\_komandi:

    print(Datoteka2\_komanda, " ".join(argumenti))

9.# Да се напише PYTHON скрипта која како аргументи од командна линија добива 2 датотеки (имиња на датотеки). Доколку има помалку од 2 датотеки, тогаш треба да се испише порака за грешка и да заврши скриптата. Првата датотека содржи парови зборови, по два зборови во секој ред посебно, втората датотека е датотека со случаен текст. Паровите зборови на првата датотека се граничници т.е. првиот збор е почетната граница а вториот збор е крајот. Скриптата треба да ги најде и отпечати сите зборови од втората датотека кои се наоѓаат од првиот граничен збор до вториот и да ги отпечати на екран. Ова треба да се направи за сите парови гранични зборови од првата датотека. Доколку некој од граничните зборови го нема во втората датотека или првиот не е пред вториот,во тој случај не се печати за тој пар гранични зборови ништо.

#!/usr/bin/python3

import sys

if len(sys.argv) < 3:  #skripta, datoteka 1, datoteka 2

    print("Gresno vnesuvanje!")

    quit()

dat1 = open(sys.argv[1], "r")

parovi = []

for line in dat1:

    element = line.split(" ")

    if len(element) == 2:

        start = element[0]

        end = element[1]

        parovi.append((start, end))

dat1.close()

dat2 = open(sys.argv[2], "r")

for linija in dat2:

    zborovi = linija.strip().split(" ")

    for start,end in parovi:

        if start in zborovi and end in zborovi:

            start\_indeks = zborovi.index(start) #go naogam indeksot

            end\_indeks = zborovi.index(end) #i ovde indeks vraka

            if start\_indeks < end\_indeks:

                print(" ".join(zborovi[start\_indeks:end\_indeks +1])) #+1 e za dodavanje na posledniot zbor bidejki ne se vklucuva vo granicite

dat2.close()

10.# Да се напише PERL скрипта со име „prva.plx“ која како аргументи добива најпрво збор кој ќе гопребарува, 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.

#!/usr/bin/python3

import sys

if len(sys.argv) != 4:  # ime na py skripta (prva.plx), zbor, zname, datoteka

    quit()

zname = sys.argv[2]

if zname != 'l' and zname != 'w' and zname != 'f':

    quit()

Tocenzbor = sys.argv[1]

brojacL =0 #broevi od liniite vo koj go ima zborot

brojacW = 0 # brojac za zborovi pred zborot sto se bara

brojacF =0 # kolku bati e pronajden zborot vo datotekata

dat = open(sys.argv[3], "r")

for i in dat:  # dali treba da pisuva dat

    brojacL = brojacL + 1

    zborovi = i.split(" ")

    for zborce in zborovi:

        if zborce == Tocenzbor:

            brojacF = brojacF + 1

            if zname=='w':

                print(brojacW)

            elif zname == 'l':

                print(brojacL)

        else:

            brojacW = brojacW + 1

if zname == 'f':

    print(brojacF)

dat.close()

11.# Потребно е да се напише Python скрипта за работа со крипто-валути. Скриптата добива на влез текстуална датотека во која, покрај другиот текст, се содржат вредности на крипто-валути. Скриптата треба да ја измине датотеката и за секоја одделна крипто-вредност да отпечати кое ќе биде најголемото зголемување од претходната вредност а кое ќе биде најголемото намалување од предходната вредност. На крај да се отпечатат, за секоја криптовалута одделно, најголемото зголемување на валутата и помеѓу кои вредности е истото, најголемото намалување на валутата и помеѓу кои вредности е истото. Крипто-валути кои се земаат во предвид се: BTC, Doge, EOS, ETH и XPM. Во текстот најпрво се наоѓа кратенката за крипто-валутата а потоа вредноста во долари. Сите зборови се одделени помеѓу себе со празно место. Доколку некоја крипто-валута ја нема во датотеката или се појавува еднаш, не се печати ништо за таа крипто-валута

#!/usr/bin/python3

import sys

if len(sys.argv) < 2:  #skripta, datoteka 1, dat2

    print("Gresno vnesuvanje!")

    quit()

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

kripto\_valuti = {"BTC" : [], "Doge" :[], "EOS" : [], "ETH" : [], "XPM" : []} #NA OVOJ NACIN KORISTAM RECNIK SO VREDNOSTI I SEKOJA VREDNOST IMA SVOJA LISTA

for linija in dat:

    zbor = linija.strip().split(" ")

    for i in range(len(zbor)): #odi do kraj na nizaata

        if zbor[i] in kripto\_valuti:

            vrednost = zbor[i+1].replace("$", "") #zimam vrednost od sledniot zbor i go cistam od $ za da dobijam samo broj

            if vrednost.isdigit(): #se odiguram za vrednosta

                kripto\_valuti[zbor[i]].append(float(vrednost))

dat.close()

print(kripto\_valuti) #vaka ke go ispecatam cel so site vrednosti

for kv, vrednosti in kripto\_valuti.items(): #so kv mi se kripto valutite zemeni, a so vrednosti iteriram niz listite od vrednosti

   #items vraka parovi na klucevi

   #ako nemam items ke gi dobijam samo nivnite iminja bez cenite

    if len(vrednosti) < 2:

        print(f"{kv} Nema dovolno vrednosti")

        continue

    max\_pokacuvanje = 0

    max\_namaluvanje = 0

    vrednost\_pokacuvanje = ()

    vrednost\_namaluvanje = ()

    for i in range(len(vrednosti) -1): # mora da ima -1 bidejki zimam vrednosti so i+1 sto znaci deka ke otide vo null ako ne odzemam

        razlika = vrednosti[i+1] - vrednosti[i]

        if razlika > max\_pokacuvanje:

            max\_pokacuvanje=razlika

            vrednost\_pokacuvanje =(vrednosti[i], vrednosti[i+1])

        if razlika < max\_namaluvanje:

            max\_namaluvanje = razlika

            vrednost\_namaluvanje = (vrednosti[i], vrednosti[i+1])

    print(f"{kv} najgolemo pokacuvanje {max\_pokacuvanje} so vrednosti {vrednost\_pokacuvanje}, najgolemo namaluvanje {max\_namaluvanje} so vrednosti {vrednost\_namaluvanje}")

12.#  Да се напише PYTHON 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.

#!/usr/bin/python3

import sys

if len(sys.argv) != 2:

print("Nemate dovolno argumenti!")

quit()

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

tel\_broevi = {} # ako treba kluc da sozdadam

#      tel\_broevi = {"071234567": "Petar Petrov"}

#          print(tel\_broevi["071234567"])  # Дава: Petar Petrov

for linija in dat:

    element = linija.split(" ")

    if len(linija) >= 3: #moze da ima 2 preziminja

        telefon = element[1]

    if len(telefon) == 9 and telefon.isdigit() and telefon.startswith("07"):

     ime\_prezime = " ".join(element[2:]) # so join gi spojuvam praznite mesta

        tel\_broevi[telefon] = ime\_prezime # vaka se postavuva kluc

# lista = ["Ana", "Marija", "Anova"]

# rezultat = " ".join(lista)  # Спојување со празно место

# print(rezultat)  # "Ana Marija Anova"

# bez join ke dobijam samo ANA

dat.close()

while True:

print("1 - prebaruvaj po tel broj")

print("2 - prebaruvaj po ime i prezime")

print("3 - preku cel imenik ")

komanda = input("Vnesi komanda 1, 2 ili 3: ")

if komanda == "1":

telefon = input("Vnesi broj: ")

if telefon in tel\_broevi:

print("Ime i prezime: {tel\_broevi[telefon]}")

else:

print("Tel broj ne postoi!")

elif komanda == "2":

ime\_prezime = input("Vnesi ime i prezime: ")

najdeno = False

for telefon, ime in tel\_broevi.items():

if(ime == ime\_prezime):

print("Telefonskiot broj e: " {telefon})

najdeno = True

break

if not najdeno:

print("Telefonski broj ne postoi")

elif komanda == "3":

if tel\_broevi:

for telefon, ime in tel\_broevi.items():

print("Tel broj: {telefon}, Ime i prezime: {ime}")

else:

print("Nevalidna opcija")

break

13.# 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 вo 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 сe 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.

#!/usr/bin/python3

import sys

if len(sys.argv) != 2:

    print("Pogresen broj na argumenti")

    quit()

dat = open(sys.argv[1], "r") # si ja otvaram datotekata za citanje

izlez = open(sys.argv[1] + "\_out", "a") #imeto + \_out ,  so "a" otvarame datoteka za dopisuvanje

# ako taa postoi novite podatoci ke bidat dodadeni na kraj

for linija in dat:

    zbor = linija.split(" ") # vraka lista so zborovi

datum = []

ostanato = []

for zborce in zbor:

    delovi = zborce.strip().split("-")  #strip za brisenje prazni delovi, split"-" gi deli zborovite so crticka pr: 01-01-2021   => ['01', '01', '2021']

    if len(delovi) == 3:

        if delovi[0].isdigit() and delovi[1].isdigit() and delovi[2].isdigit():

            den = int(delovi[0])

            mesec = int(delovi[1])

            godina = int(delovi[2])

            if(1<= den <=31) and (1<= mesec <= 12) and (1000<= godina <= 9999):

                datum.append(zborce.strip())

            else:

                ostanato.append(zborce.strip())

        else:

            ostanato.append(zborce.strip())

    else:

        ostanato.append(zborce.strip())

    if len(datum) == 0: # ako nemame nieden datum vo listata

        izlez.write(linija)

    else:

        tekst = " ".join(ostanato) #site zborovi vo edna linija

        for d in datum: #gi zima datum po datum od linija

            nova\_linija = f"{d} {tekst}\n" #format na zapis datum + ostanato

            izlez.write(nova\_linija)

izlez.close()

dat.close()

14.# Да се напише Python скрипта која од командна линија добива три датотеки. Скриптата треба да ги испреплети двете датотеки и излезот да го запише во трета датотека. Најпрво ќе се смести првата линија од првата датотека, па првата линија од втората и т.н. Дополнително, паралелно да се изврши печатење на екран излезот што се запишува во третата датотека.

#!/usr/bin/python

import sys

dat1 = open(sys.argv[1],"r") #ime na prva datoteka

dat2 = open(sys.argv[2],"r") #ime na vtora datoteka

dat3 = open(sys.argv[3], "w") # tretata e samo za zapisuvanje

dat1=0 # brojac za linii

dat2=0 #brojac za linii

pole1 = [] #listi za liniite

pole2 = [] #lista za linijata

for i in dat1: #niz eden red

    d1= d1+ 1  #brojac se zgolemuva

    pole1.append(i)  #kako prv element na nizata ke bide prviot red od dat1

for i in dat2:

    d2= d2+ 1

    pole2.append(i)

if d1 > d2:  # gi sporeduva brojacite

    min = pole2

    max = pole1

else:

    min = pole1

    max = pole2

j=0 # za pristap do elementite od max

for i in min: #niz site redovi od pomalata, ovde se zgolemuva za minimalnata

    print(i) #prviot red od minimalnata

    print.(max[j]) # prviot red od maksimalnata

    dat3.write(i) #zapisi gi dvata reda vo dat3

    dat3.write(max[j])

    j= j+1 # ovde ke se zgolemi za maksimalnata niza

////////////////////////////////////////////////////////////////////////

#!/usr/bin/python3

import sys

if len(sys.argv) != 3:

    print("Gresno vnesuvanje!")

    quit()

dat1 = open(sys.argv[1], "r") #kod napisan

dat2 = open(sys.argv[2], "r") #dokument

niza=[]

for linija in dat2:

    element = linija.split(" ", 1)

    if len(element) == 2:

     broj = element[0]

     tekst = element[1].strip()

    niza.append((broj,tekst))

dat2.close()

sekoja\_linija\_vo\_niza = dat1.readline()

dat1.close()