

**Ülesanne:**

Tekstifailis tabel.txt on andmed paljude inimeste lotovõitude kohta: nimi, isikukood, elukoht, võitmise kuu ja võidetud summa. Iga isiku andmed (kirje) on eraldi reas (rea lõpus \n – reavahetussümbol). Ühe kirje piires on andmeväljad eraldatud tabulaatoriga (\t). Teha andmetest väljavõtteid ja kokkuvõtteid.

Failist loetud andmeid võib hoida ja kasutada erinevas vormingus:

1. loendite loend (kahemõõtmeline andmehulk — tabel) võimaldab kõige kiiremini ja mugavamalt teha erinevaid päringuid, sest iga väärtus on eraldi viidatav (täiendavaid operatsioone pole vaja, kui kohe teisendada arvud arvuvormingusse), kokkuvõtete tegemisel on see variant kindlasti mugavam.
2. mitu loendit (t\_nimi, t\_kood, t\_linn, t\_kuu, t\_summa), sama isiku andmed on sama järjenumbriga.
3. loend kirjetest — iga isiku andmed (tekst) on üks loendi element
4. loendi kasutamine kõigi andmete hoidmiseks pole hädavajalik; andmeid saab lugeda näiteks kirje kaupa (erinevad võimalused) ja teha nendega vajalikud operatsioonid, seejärel lugeda failist järgmine ports andmeid.

Tulemuse leidmiseks kontrollitakse järjest iga isiku andmeid, vajadusel kirjutatakse (kopeeritakse) need mujale või muudetakse vastavate muutujate väärtusi (summad, loenduri vm).

Tulemuse leidmisel on samuti variante:

1. Päringu tulemused võib korjata algul loendisse ja lõpuks kirjutada faili. See variant on oluline siis, kui andmetega on vaja teha veel mingeid operatsioone, näiteks sortida.
2. Tulemused kirjutatakse kohe teise faili (saab kasutada väljavõtete tegemisel).

Päringute tegemist võiks alustada lihtsamatest. Siin on väljavõtte tegemine lihtsam kui kokkuvõtte.

**Näiteks olgu ülesanne: eraldada uude faili nende inimeste andmed, kes on sündinud etteantud kuul.**

**#variant 1 (lühike ja lihtne)**

```

mis = int(input("Sisesta kuu: "))
f = open('tabel.txt', 'r')
t = open('tulemus.txt', 'w')

for rida in f:
    isik = rida.split('\t')
    kuu_nr = int(isik[1][3:5])
    if kuu_nr == mis:
        t.write(rida)

t.close()
f.close()

```

# kasutaja antud kuul sündinud inimeste andmed teise faili  
# andmetega faili avamine  
# uue faili avamine tulemuste kirjutamiseks  
# failist loetakse rea (kirje) kaupa järgmise isiku andmed  
# rida jaotatakse väärtusteks (tekstid), eraldajaks tabulaator  
# isik[1] on isikukood, kuu on isikukoodi neljas ja viies number  
# kui järjekordse isiku sünnikuu on sama, mis kasutaja antud nr,  
# kirjutatakse tema andmed (kogu tekst) teise faili  
# failide sulgemine

**#variant 2 (pikem, keerukam). Failist on andmed loetud loendisse kogu, mille iga element on omakorda tekstidest koosnev loend (nimi, isikukood, elukoht, kuu, summa).**

```
kuud = ["Jaanuuar", "Veebruar", "Märts", "Aprill", "Mai", "Juuni", "Juuli", "August", "September", "Oktoober", "November", "Detsember"]
```

**#sellist kuude nimekirja võib edaspidigi vaja minna**

```

f = open("tabel.txt", 'r')
kogu=[]
for kirje in f.readlines():
    kogu.append(kirje.strip().split('\t'))
f.close()

kuu_nr=int(input("Millisel kuul sündinud inimesed kirjutada faili see_kuu? "))
see_kuu=[]
for isik in kogu:
    if int(isik[1][3:5])==kuu_nr:
        see_kuu.append(isik)

faili_nimi=kuud[kuu_nr-1]+' .txt'
kuu_fail=open(faili_nimi,'w')
for rrr in see_kuu:
    print('\t'.join(rrr), file=kuu_fail)
kuu_fail.close()

```

# andmetega faili avamine  
# kogu on loend loenditest, mille iga element sisaldab ühe isiku andmeid  
# uus tühi loend tulemuse hoidmiseks  
# isik on stringidest (tekstidest) koosnev loend ühe isiku andmetest  
# isik[1] on isikukood, kuu on isikukoodi neljas ja viies number (jrk 3 ja 4)  
# sobivate andmete kirjutamine loendisse, NB! tegelikult võiks kohe kirjutada faili  
# faili nime jaoks valitakse loendist kuud sobiva järjenumbriga kuu nimetus  
# uue faili avamine tulemuste kirjutamiseks  
# tulemuse kirjutamine faili, rrr on loend, mille kõik väärtused on tekstid  
# loendi väärtuste (tekstide) ühendamine ja kirjutamine faili, fn print lisab reavahetuse

Eelneva näite jätkuks kokkuvõtte summadest linnade kaupa. Tulemus väljastatakse tabeli kujul *shell*i aknasse.

```
f = open("tabel.txt", 'r')
kogu = []
t_nimi = []; t_kood = []
t_linn = []; t_kuu = []; t_summa = []
for kirje in f.readlines():
    loend = kirje.strip().split('\t')
    kogu.append(loend)
    t_nimi.append(loend[0])
    t_kood.append(loend[1])
    t_linn.append(loend[2])
    t_kuu.append(loend[3])
    t_summa.append(float(loend[4]))
f.close()

print('variant1: ')
d = {}
for nr in range(len(t_linn)):
    if t_linn[nr] in d:
        d[t_linn[nr]] += t_summa[nr]
    else:
        d[t_linn[nr]] = t_summa[nr]
for v in d: print(v, '\t', d[v], sep=")

print('variant2')
d = {}
for nr in range(len(kogu)):
    if kogu[nr][2] in d:
        d[kogu[nr][2]] += float(kogu[nr][4])
    else:
        d[kogu[nr][2]] = float(kogu[nr][4])
loend = list(d.keys())
loend.sort()
for v in loend: print(v, '\t', d[v], sep=")

print('variant3: eraldi loendites linnad ja summad')
linnad = []; summad = []
for isik in kogu:
    if linnad.count(isik[2])==0:
        linnad.append(isik[2])
        summad.append(float(isik[4]))
    else:
        nr = linnad.index(isik[2])
        summad[nr] += float(isik[4])
for i in range(len(linnad)):
    print(linnad[i]+' \t'+str(summad[i]))

print('kokkuvõtte sortimiseks moodustatakse loend, mille elemendid on linn ja vastav summa')
linn_sum = []
for i in range(len(linnad)): linn_sum.append([linnad[i], summad[i]])
linn_sum.sort(reverse=True)
for i in range(len(linnad)): print(linn_sum[i][0]+' \t'+str(linn_sum[i][1]))
```

```
# andmetega faili avamine
# tühi loend, sellesse paigutatakse hiljem kõikide inimeste andmed
# tühjad loendid, tabeli iga veerg eraldi loendis
# kirje on rida tekstifailist
# reast moodustatakse väärtuste loend (tekstid)
# kogu on loend loenditest, mille iga element sisaldab ühe isiku andmeid
# teistesse loenditesse kirjutatakse iga andmeelement (tabeli veerg) eraldi loendis
# summa teisendatakse enne loendisse lisamist arvuku
# uus tühi sõnastik - väärtuste paarid: linn (võtmeväärtus) ja summade summa
# nr on isiku andmete järjenumbr loendites
# isik[2] on linn; kui selline linn on juba sõnastikus,
# lisatakse selle isiku summa vastava linna summale
# kui linn sõnastikus puudub,
# lisatakse sõnastikku linn ja vastav summa
# iga võti sõnastikus: printitakse võtmeväärtus, tabulaator ja väärtus
# uus tühi sõnastik
# nr on isiku andmete järjenumbr loendis kogu
# isik[nr][2] on linn; kui selline linn on juba sõnastikus,
# lisatakse selle isiku summa vastava linna summale
# kui linn sõnastikus puudub,
# lisatakse sõnastikku linn ja vastav summa
# moodustatakse loend sõnastiku võtmeväärtustest
# loendi elementide järjestamine
# printitakse võtmeväärtus (võetud järjestatud loendist), tabulaator ja väärtus
```