

EX01 - Cashier

Kaust Gitis: [EX/ex01_cashier](#)

Faili nimi: [cashier.py](#)

Kujuta ette, et oled kassapidaja. Iga kord, kui on vaja kliendile sente tagasi anda, mingi summa x , tahad sa teada, mitu senti (täpsemalt münti) on sul minimaalselt vaja kliendile tagasi anda.

Sinu ülesandeks on luua programm, mis saab sisendiks numbri (alati vahemikus 1-100) ning prindib konsooli, mitu senti on vaja minimaalselt tagastada, et sisestatud summat katta. Kassas on ühikuga sendid järgmiste väärtustega: **(1,5,10,20,50)**

Näited

Näide sisendiga 63:

```
Enter a sum: 63
```

```
Amount of coins needed: 5
```

Näide sisendiga 6:

```
Enter a sum: 6
```

```
Amount of coins needed: 2
```

NB! Programmis peavad olema sõned täpselt samal kujul. Sisendi küsimise puhul: "Enter a sum: " ja väljundi puhul: "Amount of coins needed: (_sentide arv_)"

Abiks

Sisendid ja vastavad väljund

Antud ülesandes testitakse kõiki sisendeid vahemikus 1-100. Siin on mõned sisendid, et oma programmi kontrollida.

Sum Coins needed Coins used

1	1	1
11	2	10, 1
45	3	20, 20, 5
27	4	20, 5, 1, 1
39	7	20, 10, 5, 1, 1, 1, 1

Algoritm

Seda ülesannet saab lahendada nõ ahnelt. See tähendab, et igal sammul valime me suurima väärtusega sendi, mida juba valitud ei ole, ning proovime seda ette antud summa sisse mahutada. Algoritmi, mis niimodi ahnelt käitub, nimetataksegi ahneks algoritmiks. Antud probleemi puhul annab selline algoritm alati optimaalseima lahenduse.

Meie algoritm võiks töötada nii:

1. Eelmise näite puhul, kui meil algseks sisendiks on 63, kontrolliks meie algoritm, mitu korda 50 selle arvu sisse mahub. Mahub täpselt 1 kord ja jäägiks jääb 13 ($63 - 50 * 1$).
2. Järgmisena, mitu 20 sendist mahub nüüd selle jäägi sisse: 13 sisse ei mahu 20 **mitte ühtegi korda**.
3. Mitu 10 sendist: jääk on endiselt 13. 13 sisse mahub 10 täpselt **1 kord**, uueks jäägiks on 3 ($13 - 10$).

4. Mitu 5 sendist: 5 sendiseid ei mahu 3 sisse **mitte ühtegi korda**.
5. Mitu 1 sendist: 3 sisse mahub ühe sendiseid 3 korda.
6. Meie algoritm on jõudnud lõpuni ja tulemuseks on 5 (1+0+1+0+3).

NB! Kuigi selles ülesandes annab selline ahne algoritm alati optimaalseima tulemuse, siis iga probleemi puhul see nii ei ole.

Jääk

Pythonis saame jäägi leida järgmiselt:

```
c = a % b
```

c väärtus on arvu a jagamisel b-ga tekkiv jääk.

Näiteks jagades arv 7 arvuga 5 on jäägiks 2. Võime seda lihtsa tehtega Pythonis kontrollida:

```
print(7 % 5)
```

[Copy](#)

Meie programm annab vastuseks samuti 2.

Mõtleme, kuidas kirjutada programm, mis savestaks nii jäägi kui ka selle, mitu korda üks arv teise sisse mahub.

7 / 5 annab meile tulemuseks 1.4. Näeme, et arvu 1.4 täisosa on 1 ning saime kätte oma vastuse - Arv 5 mahub arvu 7 sisse ühe korra ning jäägiks on 2.

Kuidas Pythonis selline programm välja näeks? Jääki me oskame juba leida:

```
remainder = 7 % 5
```

[Copy](#)

Täisosa saab Pythonis leida lihtsalt: jagame arvu 7 arvuga 5 ning salvestame tulemuse täisarvuna. Pythonis saame ujukomaarve muuta täisarvuks `int()` funktsiooni abil.

```
float_number = 7 / 5
print(float_number) # 1.4
whole_number = int(float_number)
print(whole_number) # 1
```

[Copy](#)

tähisega tähistame Pythonis [kommentaare](#). Kommentaarid ei mõjuta seda, kuidas kood jookseb. Üleval näites on kasutatud kommentaare, et näidata `print()` funktsiooni väljundit

samaväärne tehtega `int(7 / 5)`, on tehe `7 // 5`. Kasutades topelt jagamismärki saame tulemuseks jagamisel tekkiva täisosa. Kirjutame oma eelmise programmi ümber kasutades `//` märki.

```
whole_number = 7 // 5
print(whole_number) # 1
```

[Copy](#)

Kasulikud lingid

[Pydoc](#)

[Tsükkel](#)

[Matemaatilised avaldised Pythonis](#)

Hinded

Hinde nimi	Sinu punktid
ex01_cashier - Tests	0p / 15p
ex01_cashier - Style	0p / 1p
ex01_cashier - Defense	0p / 1p
Kokku:	0p / 15p

Tähtajad

Pärast	Protsent	Grupp
2020-09-15 09:00:00	20%	Kõik grupid
2020-09-16 09:00:00	10%	Kõik grupid
2020-09-17 09:00:00	0%	Kõik grupid

Esitused ↺

My registrations

◀ PR01 - Python

Mine...

1. nädala quiz ▶

- Abi
- eesti (et)
- eesti (et)
- English (en)

Lae alla mobiilirakendus

Kasutustingimused