

## Viikkotehtävät 3

### Ehtolauseet (if / elif / else)

1. Kirjoita ohjelma, joka kysyy käyttäjältä kaksi kokonaislukua ja tulostaa niistä **ainoastaan suuremman luvun** näytölle. Jos numerot ovat yhtä suuria, tulosta ainoastaan teksti *"Numerot ovat yhtä suuria."*. Noudata **täysin** samaa kirjoitusasua kuin alla olevissa esimerkkikuvissa!

Esimerkkejä ohjelman toiminnasta:

```
Anna ensimmäinen luku:  
5  
Anna toinen luku:  
7  
Suurempi luku = 7
```

```
Anna ensimmäinen luku:  
11  
Anna toinen luku:  
4  
Suurempi luku = 11
```

```
Anna ensimmäinen luku:  
5  
Anna toinen luku:  
5  
Numerot ovat yhtä suuria.
```

Tehtävän tiedostonimi = *exercise3\_1.py*

Tyypillinen koodimäärä: **8-12 riviä** (tyhjiä rivejä ja kommentteja ei lasketa)

2. Tee ohjelma, jossa kysytään kesäpäivän lämpötila kokonaislukuna. Tulosta käyttäjän syöttämän lämpötilan mukaan vain yksi seuraavista vaihtoehtoista:
- "KYLMAÄ", jos lämpötila on 0-10 astetta.
  - "KOLEAA", jos lämpötila on 11-15 astetta.
  - "NORMAALI LÄMPÖTILA", jos lämpötila on 16-20 astetta.
  - "LÄMMINTÄ ", jos lämpötila on 21-25 astetta.
  - "HELLETTÄ", jos lämpötila on 26-30 astetta.

Noudata ohjelmassa **täysin** samaa kirjoitusasua kuin alla olevissa esimerkkikuvissa!

Esimerkkejä ohjelman toiminnasta:

```
Syötä päivän lämpötila:
```

```
14
```

```
KOLEAA
```

```
Syötä päivän lämpötila:
```

```
20
```

```
NORMAALI LÄMPÖTILA
```

```
Syötä päivän lämpötila:
```

```
28
```

```
HELLETTÄ
```

Tehtävän tiedostonimi = **exercise3\_2.py**

Tyypillinen koodimäärä: **11-16 riviä** (tyhjiä rivejä ja kommentteja ei lasketa)

**Vinkki:** Tässä tehtävässä **if-elif-elif**-ym. -rakenne on käytännöllisempi kuin täysin erilliset **if**-lauseet, koska kaikki ehdot käsittelevät samaa muuttujaa (= lämpötilaa).



3. Kirjoita ohjelma, joka kysyy käyttäjältä viikon työtuntimäärän sekä tuntipalkan, ja laskee niiden perusteella työntekijän yhden viikon palkka-ansiot.

Jos työntekijä on tehnyt viikon aikana ylityötä, ylityötunneista maksetaan 50 % korotus tuntipalkkaan. Ylityötä ovat kaikki 40 tunnin yli menevät tunnit.

Pyöristä tulos kahteen desimaaliin.

**Huom:** Ylityökorotus koskee **vain ylimeneviä tunteja**, ei kaikkia tunteja! (eli hyvä idea on laskea esim. ensimmäisten 40:n tunnin palkka erikseen ja ylimenevien tuntien palkka erikseen, ja summata ne lopuksi yhteen)

**Esimerkkejä ohjelman toiminnasta:**

```
Syötä viikon työtunnit:
37
Syötä tuntipalkkasi:
15.2
Viikon ansiosi ovat: 562.4€
```

```
Syötä viikon työtunnit:
47
Syötä tuntipalkkasi:
17.6
Viikon ansiosi ovat: 888.8€
```

Lisää esimerkkiarvoja oman ohjelmakoodin testaamiseen:

- Työtunnit: **30**, tuntipalkka: **15** = Viikon ansiot: **450.0€**
- Työtunnit: **40**, tuntipalkka: **15** = Viikon ansiot: **600.0€**
- Työtunnit: **45**, tuntipalkka: **15** = Viikon ansiot: **712.5€**

**Vinkki:** Tee oma laskulogiikka eri tapauksille (if/elsen avulla), joista toinen käsittelee tapaukset, jossa ylityötunteja ei ole, ja toinen tapaukset, jossa työtunnit menevät yli 40h

Tehtävän tiedostonimi = **exercise3\_3.py**

Tyypillinen koodimäärä: **10-16 riviä** (tyhjiä rivejä ja kommentteja ei lasketa)

4. Tee ohjelma, jossa kysytään asiakkaan antama rahamäärä sekä ostoksen hinta kokonaislukuina. Ohjelman tarkoitus on tutkia, riittääkö annettu rahamäärä.

- Jos rahat riittävät heti alussa, tulosta "Kiitos." ja lisäksi paljonko rahaa annetaan takaisin.
- ...mutta jos rahat eivät riittäneetkään, pyydetään käyttäjää syöttämään vielä yhden kerran lisää rahaa.
  - Tarkista tämän jälkeen uudelleen riittävätkö rahat ostoksen hankkimiseen. Jos eivät edelleenkään riittäneet, tulosta näytölle "Sinulla ei ole tarpeeksi rahaa.", mutta jos rahat riittivät, tulosta "Kiitos." ja paljonko annetaan rahaa takaisin.

Ohjelmakoodi ei saa tulostaa ylimääräisiä tekstejä käyttäjälle, vaan ainoastaan ne mitä tietyssä tilanteessa tarvitsee tulostaa. (ks. alla olevat kuvat)

Esimerkkejä ohjelman toiminnasta:

```
Anna rahaa:
100
Ostosten hinta:
85
Kiitos. Annetaan takaisin 15 €
```

```
Anna rahaa:
100
Ostosten hinta:
150
Rahat eivät riitä, anna lisää rahaa:
60
Kiitos. Annetaan takaisin 10 €
```

```
Anna rahaa:
50
Ostosten hinta:
100
Rahat eivät riitä, anna lisää rahaa:
20
Sinulla ei ole tarpeeksi rahaa.
```



**Vinkkejä:** Noudata **täysin** samaa kirjoitusasua kuin yllä olevan kuvan esimerkeissä, silloin **codePost** tunnistaa parhaiten ohjelmasi toiminnan!

**Tämä tehtävä on helpoin tehdä sisäkkäisillä ehtolauseilla:**

**if / else** (oliko rahaa alussa tarpeeksi), ja **else**-lauseen sisällä: kysytään lisää rahaa ja tarkistetaan uudella sisäkkäisellä if-lauseella tilanne (katsotaan onko rahaa tarpeeksi toisella kerralla). Esim.:

**Jos** alussa annettu raha riittää, **kiitä käyttäjää ja kerro vaihtorahan määrä.**

**...mutta jos** alussa annettu raha ei riittänyt, silloin logiikka on tämä:

**Pyydetään käyttäjältä lisää rahaa**

**Jos kokonaisrahamäärä riittää nyt, kiitä käyttäjää ja kerro vaihtorahan määrä**

**...mutta jos kokonaisrahamäärä ei riittänyt vielä, ilmoitetaan asiasta käyttäjälle**

Tehtävän tiedostonimi = **exercise3\_4.py**

Tyypillinen koodimäärä: **12-18 riviä** (tyhjiä rivejä ja kommentteja ei lasketa)

5. Opettaja laskee kokeen arvosanat käyttäen oheista taulukkoa.

Tee ohjelma, joka pyytää käyttäjältä pistemäärän ja tulostaa sitä vastaavan arvosanan.

Ilmoita käyttäjälle, jos annettu pistemäärä ei ole mahdollinen (pienempi kuin 0, tai suurempi kuin 100)

Pisteet	Arvosana
0 - 50	0
51 - 60	1
61 - 70	2
71 - 80	3
81 - 90	4
91 - 100	5

Esimerkkejä ohjelman toiminnasta:

```
Anna pistemäärä:
36
Arvosana: 0
```

```
Anna pistemäärä:
72
Arvosana: 3
```

```
Anna pistemäärä:
87
Arvosana: 4
```

```
Anna pistemäärä:
60
Arvosana: 1
```

```
Anna pistemäärä:
-15
Pistemäärä ei ole mahdollinen.
```

```
Anna pistemäärä:
150
Pistemäärä ei ole mahdollinen.
```

Tehtävän tiedostonimi = *exercise3\_5.py*

Tyypillinen koodimäärä: **12-18 riviä** (tyhjiä rivejä ja kommentteja ei lasketa)

## Lisätehtäviä!

6. Karkausvuosi on jaollinen 4:llä, mutta ei 100:lla. Kuitenkin vuosi, joka on jaollinen 400:lla, on aina karkausvuosi. Tee ohjelma, joka päättelee käyttäjän syöttämästä vuodesta, onko kyseinen vuosi karkausvuosi vai ei. Jos vuosi on karkausvuosi, tulosta **Karkausvuosi: KYLLÄ**. Jos vuosi ei ole karkausvuosi, tulosta **Karkausvuosi: EI**. Älä tulosta ylimääräistä tekstiä (ks. esimerkkikuvat alla.). Älä hyödynnä tässä tehtävässä mitään Pythonin lisämoduulia, joka laskee karkausvuositiedon sinun puolestasi, vaan toteuta tämä tehtävä ehtolauseilla!

### Vinkkejä:

Tämän voi tehdä joko yhdellä if/else-lausekkeella, tai usealla sisäkkäisellä if-lausekkeella. Valitse se tapa kumpi tuntuu miellyttävämmältä!

#### Esimerkkejä vuosista, jotka ovat karkausvuosia:

1996 (jaollinen 4:llä)  
2000 (jaollinen 400:lla)  
2016 (jaollinen 4:llä)

#### Esimerkkejä vuosista, jotka eivät ole karkausvuosia:

2015 (ei jaollinen 4:llä tai 400:lla)  
1994 (ei jaollinen 4:llä tai 400:lla)  
1900 (jaollinen 100:lla, mutta ei 400:lla)

**Pieni lisätehtävä:** Hyödynnä Boolean –muuttujaa tässä harjoituksessa (True / False). Booleania odotetaan käytettävän samalla tavalla kuin viikkotehtävässä 4-5.

### Esimerkkejä ohjelman toiminnasta:

```
Anna vuosiluku:  
1996  
Karkausvuosi: KYLLÄ
```

```
Anna vuosiluku:  
2017  
Karkausvuosi: EI
```

```
Anna vuosiluku:  
2000  
Karkausvuosi: KYLLÄ
```

```
Anna vuosiluku:  
1800  
Karkausvuosi: EI
```

Tehtävän tiedostonimi = *exercise3\_6.py*

Tyypillinen koodimäärä: **10-20 riviä** (tyhjiä rivejä ja kommentteja ei lasketa)

Vinkki: yksi tapa ratkaista tämä tehtävä on noudattaa tällaista rakennetta:

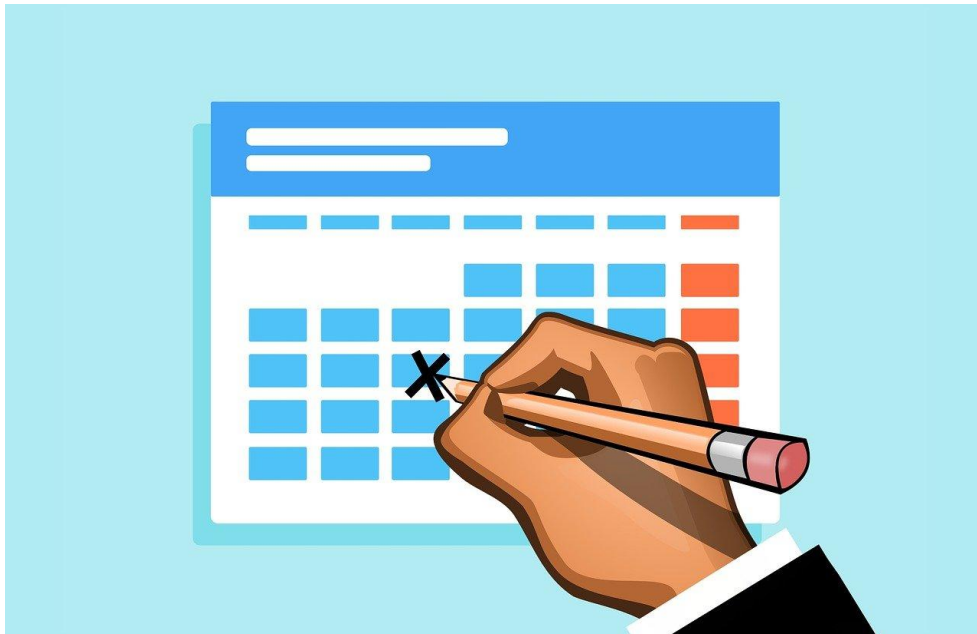
**Jos** vuosiluku on jaollinen 400:lla, *kyseessä on aina karkausvuosi.*

**...mutta jos** vuosiluku ei ole jaollinen 400:lla, silloin logiikka on tämä:

**Jos** vuosiluku on jaollinen 100:lla, *kyseessä ei ole karkausvuosi*

**...mutta jos** vuosiluku on jaollinen 4:llä, *kyseessä on karkausvuosi*

Muussa tapauksessa, *kyseessä ei ole karkausvuosi*



7. Tehtävänä on toteuttaa postihintalaskuri, jolla voi laskea kirjeiden ja pakettien lähetyshintoja alla olevan kuvitteellisen taulukon mukaan. Pyydä käyttäjältä ensin lähetyksen tyyppi, ja sitten paino.

Tyyppi	Perushinta	Painoluokka		
		< 200g	200g - 500g	> 500g
Kirje	50snt	ei lisämaksua	4snt / 100g	7snt / 100g
Paketti	2eur	ei lisämaksua	8snt / 100g	14snt / 100g

- Painon mukaan määräytyvät lisämaksut lasketaan täyttyneitä satoja grammoja kohden, esimerkiksi 499g kirjeen painon mukaan määräytyvä lisämaksu on  $400g * 4snt/100g = 16snt$
- Yli 500g **kirjeistä** peritään kahden euron lisämaksu, **mikäli kirje ei mahdu postilaatikkoon** (kysy käyttäjältä mahtuuko kirje postilaatikkoon)
- Esimerkkihintoja:**  
Kirje 400g =  $0.5 \text{ €} + (4 * 0.04\text{€}) = 0.66\text{€}$  (tai 66snt)  
Kirje 650g =  $0.5 \text{ €} + (6 * 0.07\text{€})$ , **mahtuu postilaatikkoon** =  $0.92\text{€}$  (tai 92snt)  
Kirje 650g =  $0.5 \text{ €} + (6 * 0.07\text{€})$ , **ei mahdu postilaatikkoon (+ 2€)** =  $2.92\text{€}$   
Paketti 400g =  $2 \text{ €} + (4 * 0.08\text{€}) = 2.32\text{€}$   
Paketti 650g =  $2 \text{ €} + (6 * 0.14\text{€}) = 2.84\text{€}$



Tehtävän tiedostonimi = *exercise3\_7.py*

Tyypillinen koodimäärä: **20-36 riviä** (tyhjiä rivejä ja kommentteja ei lasketa)

**Huom:** Tämän voi ratkaista myös alle 20:llä rivillä, mutta se vaatii apumuuttujien hyödyntämistä! (vinkki: tee oma apumuuttuja painokertoimelle, joka pitää kirjaa oikeaa kerrointa riippuen onko kyseessä kirje/paketti ja minkä painoinen lähetys. esim. jos käyttäjä lähettää paketin 400g, silloin kerroin olisi 0.08, eli 8snt jne)



8. Toteuta ohjelma, joka pyytää käyttäjältä ostosten kokonaissumman euroina, onko asiakas opiskelija (K / E) ja onko asiakas kanta-asiakas (K / E). Asiakas voi olla yhtä aikaa sekä opiskelija- että kanta-asiakas.

Jos kyseessä on kanta-asiakas, kysytään lisäksi kuinka paljon kanta-asiakaspisteitä asiakkaalla on entuudestaan. Kysy lopuksi mahdollinen alennuskoodi. Hyväksytyt alennuskoodit ovat:

- **FALL24**
- **BK2SCHOOL**

### Kysymysten jälkeen, muuta tilauksen loppusummaa seuraavanlaisella alennuslogiikalla:

- Ensimmäisenä, käsittele mahdollinen alennuskoodi:
  - Alennuskoodi **FALL24** vähentää ostosten kokonaissummasta 10%. Ei koske postimaksua.
  - Alennuskoodi **BK2SCHOOL** vähentää kokonaissummasta 20% ainoastaan, jos kyseessä on opiskelija-asiakas. Ei koske postimaksua.
  - Asiakas voi käyttää vain yhtä alennuskoodia tilausta tehdessä.
- Seuraavaksi, jos kyseessä on kanta-asiakas: annetaan tämän hetkisestä kokonaissummasta 100 asiakaspistettä per tilauksen jokaista alkavaa 10€:a kohti (pisteet lasketaan vasta mahdollisen alennuskoodialennuksen jälkeen, ei koske postimaksua)
- Kanta-asiakkaan tilauksen loppusummasta vähennetään 5€ jokaista täyttä 1000:tta kanta-asiakaspistettä kohden
  - Mukaan otetaan myös uudesta ostoksesta tulevat uudet kanta-asiakaspisteet
- Lopuksi kokonaissummaan lisätään postimaksut: 7,95€
  - Jos kokonaissumma tässä vaiheessa ylittää 99€, postimaksu on 0€

**Tulosta lopullinen tilauksen loppusumma postikuluineen.**



### Huomioita / vinkkejä:

- Asiakkaan ei ole pakko syöttää alennuskoodia ollenkaan
- Älä anna asiakkaalle kanta-asiakaspisteitä alkuperäisestä hinnasta, vaan vasta alennuskoodin jälkeisestä hinnasta!
- Tarkista kokonaissumma postikulualennuksen osalta vasta aivan viimeisenä!

### Esimerkkituloksia:

<b>Summa:</b> 100€, <b>Opiskelija:</b> Kyllä <b>Kanta-asiakas:</b> Kyllä (alussa 1000 pistettä) <b>Alennuskoodi:</b> BK2SCHOOL <b>Tilauksen loppusumma</b> = 82,95 €	<b>Summa:</b> 200€, <b>Opiskelija:</b> Kyllä <b>Kanta-asiakas:</b> Ei <b>Alennuskoodi:</b> BK2SCHOOL <b>Tilauksen loppusumma</b> = 160 €
<b>Summa:</b> 100€ <b>Opiskelija:</b> Ei <b>Kanta-asiakas:</b> Kyllä (alussa 1500 pistettä) <b>Alennuskoodi:</b> FALL24 <b>Tilauksen loppusumma</b> = 87,95 €	<b>Summa:</b> 300€ <b>Opiskelija:</b> Ei <b>Kanta-asiakas:</b> Kyllä (alussa 3500 pistettä) <b>Ei alennuskoodia</b> <b>Tilauksen loppusumma</b> = 270 €

Tehtävän tiedostonimi = *exercise3\_8.py*

Tyypillinen koodimäärä: **20-40 riviä** (tyhjiä rivejä ja kommentteja ei lasketa)