

HARJOITUSTEHTÄVÄT 4

TEHTÄVÄ 1



Kirjoita proseduri `huuda(lause)`, joka saa parametrikseen lauseen ja tulostaa sen ruudulle kokonaan isoilla kirjaimilla ja huutomerkillä varustettuna.

Kysy pääohjelmassa käyttäjältä lause ja käydä aliohjelmalla `huuda()` annetulle lauseelle.

```
Anna lause: moikka  
MOIKKA!
```

VINKKI:

Metodi `upper()` muuttaa annetun merkkijonon kirjaimet isoiksi kirjaimiksi.

```
>>> miono = "Moi kaikki!"  
>>> miono2 = miono.upper()  
>>> print(miono2)  
MOI KAIKKI!
```

TEHTÄVÄ 2



Kirjoita funktio `monista(miono, maara)`, joka saa parametrikseen merkkijonon ja kokonaisluvun ja palauttaa uuden merkkijonon, johon on monistettu parametrina annettua merkkijonoa niin monta kertaa peräkkäin kuin mitä parametrina saatu määrä on.

```
m = monista("moi", 3)  
print(m) # tulostaa moimoimoi
```

TEHTÄVÄ 3



Kirjoita proseduuri kolmionala(kanta, korkeus), joka laskee ja tulostaa ruudulle annetun kolmion alan. Kolmion ala lasketaan kaavalla

$$\frac{kanta \cdot korkeus}{2}$$

Voit kysyä kannan ja korkeuden käyttäjältä ja käyttää proseduuria käyttäjän syötteille.

Anna kanta: 3
Anna korkeus: 4
Kolmion pinta-ala on 6.0.

TEHTÄVÄ 4



Kirjoita proseduuri, joka tulostaa ruudulle halutun määrän välilyöntejä ja tähtiä. Kutsu sitä kahdessa eri proseduurissa siten, että toinen proseduuri tulostaa salmiakkikuvion ja toinen S-kirjaimen.

tulostaSalmiakki() tulostaa:

```
*
***
*****
*****
*****
***
*
```

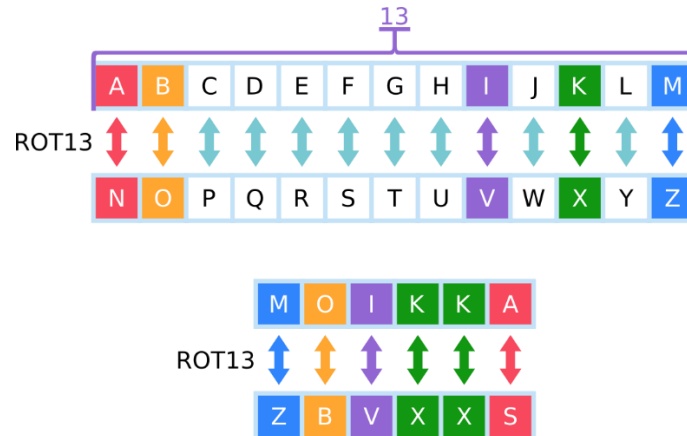
tulostaS() tulostaa:

```
*****
**
*****
**
*****
```

TEHTÄVÄ 5



ROT13 on hyvin yksinkertainen salausmenetelmä. Ideana on, että salattavan merkkijonon jokainen kirjain korvataan kirjaimella, joka löytyy aakkosista 13 merkkiä edempää. Seuraava kuva esittää salauksen toimintaa:



Kirjoita funktio `rot13(merkkijono)`, joka salaa annetun merkkijonon ROT13-menetelmällä ja palauttaa salatun merkkijonon.

```

sana = "moikka"
salattu = rot13(sana)
print(salattu) # tulostaa zbvxxn

```

TEHTÄVÄ 6



Kirjoita ohjelma, joka kysyy käyttäjältä kaksi lukua ja kutsuu proseduuria, joka tulostaa ruudulle annettujen lukujen suurimman yhteisen tekijän ja pienimmän yhteisen jaettavan. Proseduurin täytyy käyttää kahta apufunktiota, jotka laskevat tekijän ja jaettavan.

```

Anna eka luku: 3
Anna toka luku: 4
Luvut ovat 3 ja 4.
Lukujen suurin yhteinen tekijä on 1.
Lukujen pienin yhteinen jaettava on 12.

```

Kahden luvun suurin yhteinen tekijä on suurin sellainen luku, jolla voi jakaa molemmat luvut niin, että jakolasku menee tasan (eli jakojäännös on nolla). Esimerkiksi lukujen 20 ja 30 suurin yhteinen tekijä on 10.

Kahden luvun pienin yhteinen jaettava taas on pienin sellainen luku, jonka voi jakaa molemmilla luvuilla niin, että jakolasku menee tasan (eli jakojäännös on nolla). Esimerkiksi lukujen 2 ja 3 pienin yhteinen jaettava on 6.