

K1

- a) Suomennetaan lause $A \wedge B$.

” $\overbrace{\text{Ajattelen joululomaa}}^A$ $\wedge$ $\overbrace{\text{hymyilen}}^B$.”

- b) Suomennetaan lause $A \Rightarrow B$.

” $\text{Jos } \overbrace{\text{ajattelen joululomaa}}^A$, $\text{niin } \overbrace{\text{hymyilen}}^B$.”

- c) Suomennetaan lause $\neg B \vee (A \wedge B)$.

” $\overbrace{\text{En hymyile}}^{\neg B}$ $\vee$ $\overbrace{\text{ajattelen joulomaa}}^A$ $\wedge$ $\overbrace{\text{hymyilen}}^B$.”

Vastaus

- a) Ajattelen joululomaa ja hymyilen.
b) Jos ajattelen joululomaa, niin hymyilen.
c) En hymyile, tai sitten ajattelen joululomaa ja hymyilen.

K2

- a) Suomennetaan lause $A \wedge B \wedge \neg C$.

” $\overbrace{\text{Osallistut testiin}}^A$ $\wedge$ $\overbrace{\text{ja saat palkinnon}}^B$ $\wedge$ $\overbrace{\text{et ole iloinen.}}^{\neg C}$.”

Lauseen voidaan ilmaista myös toisin:

Osallistut testiin ja saat palkinnon, mutta et ole iloinen.

- b) Suomennetaan lause $(A \vee C) \Rightarrow B$.

”**Jos** $\overbrace{\text{osallistut testiin}}^A$ $\vee$ $\overbrace{\text{tai olet iloinen,}}^C$ **niin** $\overbrace{\text{saat palkinnon.}}^B$.”

- c) Suomennetaan lause $B \Leftrightarrow (A \wedge C)$.

” $\overbrace{\text{Saat palkinnon,}}^B$ $\Leftrightarrow$ $\overbrace{\text{jos ja vain jos}}^{\text{ }}$
 $\overbrace{\text{osallistut testiin}}^A$ $\wedge$ $\overbrace{\text{tai olet iloinen.}}^C$.”

Vastaus

- a) Osallistut testiin ja saat palkinnon, mutta et ole iloinen.
b) Jos osallistut testiin tai olet iloinen, niin saat palkinnon.
c) Saat palkinnon, jos ja vain jos osallistut testiin ja olet iloinen

K3

A : ”Markus on välitunnilla.”, $\neg A$: ”Markus ei ole välitunnilla.”

B : ”Markus on ulkona.”, $\neg B$: ”Markus on sisällä.”

a) Formalisoidaan lause ”Markus on sisällä tai välitunnilla.”

$$\neg B \vee A$$

b) Formalisoidaan lause ”jos Markus on ulkona, niin hän on välitunnilla.”

$$B \Rightarrow A$$

c) Formalisoidaan lause ”ei pidä paikkaansa, että Markus on välitunnilla ja ulkona.”

$$\neg(A \wedge B)$$

Vastaus

a) $\neg B \vee A$

b) $B \Rightarrow A$

c) $\neg(A \wedge B)$

K4

Määritellään atomilauseet seuraavasti:

A : ”Laura on lukiolainen”,

B : ”Laura opiskelee pitkää matematiikkaa” ja

C : ”Laura opiskelee fysiikkaa”.

- a) Formalisoidaan lause ”Laura on lukiolainen ja opiskelee pitkää matematiikkaa, mutta ei fysiikkaa.”

$$A \wedge B \wedge \neg C$$

- b) Formalisoidaan lause ”jos Laura opiskelee pitkää matematiikkaa ja fysiikkaa, niin hän on lukiolainen.”

$$(B \wedge C) \Rightarrow A$$

- c) Formalisoidaan lause ”Laura on lukiolainen täsmälleen silloin, kun hän opiskelee fysiikkaa tai pitkää matematiikkaa.”

$$A \Leftrightarrow (C \vee B)$$

Vastaus

a) $A \wedge B \wedge \neg C$

b) $(B \wedge C) \Rightarrow A$

c) $A \Leftrightarrow (C \vee B)$

K5

Täydennetään totuustaulu.

A	B	$\neg B$	$A \vee \neg B$	$(A \vee \neg B) \Rightarrow A$
1	1	0	1	1
1	0	1	1	1
0	1	0	0	1
0	0	1	1	0

Lause on tosi, kun ainakin toinen lauseista A ja B on tosi.

Vastaus

Lause on tosi, kun ainakin toinen lauseista A ja B on tosi.

K6

- a) Laaditaan lauseelle $(A \wedge B) \vee (\neg A \wedge B)$ totuustaulu.

A	B	$\neg A$	$A \wedge B$	$\neg A \wedge B$	$(A \wedge B) \vee (\neg A \wedge B)$
1	1	0	1	0	1
1	0	0	0	0	0
0	1	1	0	1	1
0	0	1	0	0	0

Lause on tosi, kun lause B on tosi.

- b) Laaditaan lauseelle $(\neg A \wedge B) \Leftrightarrow A$ totuustaulu.

A	B	$\neg A \wedge B$	$(\neg A \wedge B) \Leftrightarrow A$
1	1	0	0
1	0	0	0
0	1	1	0
0	0	0	1

Lause on tosi, kun molemmat lauseet A ja B ovat epätosia.

vastaus

- a) Lause on tosi, kun lause B on tosi.
b) Lause on tosi, kun molemmat lauseet A ja B ovat epätosia.

K7

Laaditaan lauseen $(A \vee \neg B) \vee C$ totuustaulu.

A	B	C	$\neg B$	$A \vee \neg B$	$(A \vee \neg B) \vee C$
1	1	1	0	1	1
1	1	0	0	1	1
1	0	1	1	1	1
1	0	0	1	1	1
0	1	1	0	0	1
0	1	0	0	0	0
0	0	1	1	1	1
0	0	0	1	1	1

Lause on epätosi, kun A ja C ovat epätosia ja B on tosi.

Vastaus

Lause on epätosi, kun A ja C ovat epätosia ja B on tosi.

K8

Laaditaan lauseen $(\neg A \vee B) \vee (B \wedge C)$ totuustaulu.

A	B	C	$\neg A$	$\neg A \vee B$	$B \wedge C$	
1	1	1	0	1	1	1
1	1	0	0	1	0	1
1	0	1	0	0	0	0
1	0	0	0	0	0	0
0	1	1	1	1	1	1
0	1	0	1	1	0	1
0	0	1	1	1	0	1
0	0	0	1	1	0	1

Lause on epätosi, kun A on tosi ja B on epätosi.

Vastaus

Lause on epätosi, kun A on tosi ja B on epätosi

K9

Olkoon A : ”Petteri pitää koirista” ja B : ”Petteri pitää kissoista”.

- a) Formalisoidaan lause ”jos Petteri pitää koirista tai kissoista, niin hän pitää koirista mutta ei kissoista.”

$$(A \vee B) \Rightarrow (A \wedge \neg B)$$

Laaditaan lauseelle $(A \vee B) \Rightarrow (A \wedge \neg B)$ totuustaulu.

A	B	$\neg B$	$A \vee B$	$A \wedge \neg B$	$(A \vee B) \Rightarrow (A \wedge \neg B)$
1	1	0	1	0	0
1	0	1	1	1	1
0	1	0	1	0	0
0	0	1	0	0	1

Lause on tosi, kun Petteri ei pidä kissoista (eli hän pitää vain koirista tai ei kissoista eikä koirista).

- b) Formalisoidaan lause ”Petteri pitää kissoista, jos ja vain jos ei pidä paikkaansa, että hän pitää kissoista ja koirista.”

$$B \Leftrightarrow \neg(B \wedge A)$$

Laaditaan lauseelle $B \Leftrightarrow \neg(B \wedge A)$ totuustaulu.

A	B	$B \wedge A$	$\neg(B \wedge A)$	$B \Leftrightarrow \neg(B \wedge A)$
1	1	1	0	0
1	0	0	1	0
0	1	0	1	1
0	0	0	1	0

Lause on tosi, kun Petteri pitää vain kissoista.

Vastaus

- a) $(A \vee B) \Rightarrow (A \wedge \neg B)$

Lause on tosi, kun Petteri ei pidä kissoista.

- b) $B \Leftrightarrow \neg(B \wedge A)$

Lause on tosi, kun Petteri pitää vain kissoista.

K10

Olkoon

H : ”Helmi pääsee veneretkelle”,

K : ”Kimmo pääsee uimaan” ja

T : ”Timo pääsee tivoliin”.

Formalisoidaan lasten vaatimukset.

Timo: $\neg(H \wedge K)$

Kimmo: $\neg H \wedge T$

Helmi: $\neg K \wedge (H \Rightarrow T)$

Tutkitaan lauseita totuustaulun avulla.

H	K	T	$\neg H$	$\neg K$	$H \wedge K$	$H \Rightarrow T$	$\neg(H \wedge K)$	$\neg H \wedge T$	$\neg K \wedge (H \Rightarrow T)$
1	1	1	0	0	1	1	0	0	0
1	1	0	0	0	1	0	0	0	0
1	0	1	0	1	0	1	1	0	1
1	0	0	0	1	0	0	1	0	0
0	1	1	1	0	0	1	1	1	0
0	1	0	1	0	0	1	1	0	0
0	0	1	1	1	0	1	1	1	1
0	0	0	1	1	0	1	1	0	1

Taulukosta löytyy 1 rivi, jossa kaikki lauseet ovat yhtä aikaa tosia. Tämä rivi toteuttaa kaikkien lasten toiveet:

Helmi ei pääse veneretkelle, Kimmo ei pääse uimaan, mutta Timo pääsee Tivoliin.

Vastaus

Helmi ei pääse veneretkelle, Kimmo ei pääse uimaan, mutta Timo pääsee Tivoliin.

K11

Lause on tautologia, jos se on aina tosi.

a) Tutkitaan lausetta $A \vee B$ totuustaulun avulla.

A	B	$A \vee B$
1	1	1
1	0	1
0	1	1
0	0	0

Lause ei ole tautologia, koska se ei ole aina tosi.

b) Tutkitaan lausetta $(A \vee \neg B) \vee (C \vee B)$ totuustaulun avulla.

A	B	C	$\neg B$	$A \vee \neg B$	$C \vee B$	$(A \vee \neg B) \vee (C \vee B)$
1	1	1	0	1	1	1
1	1	0	0	1	1	1
1	0	1	1	1	1	1
1	0	0	1	1	0	1
0	1	1	0	0	1	1
0	1	0	0	0	1	1
0	0	1	1	1	1	1
0	0	0	1	1	0	1

Lause on tautologia, koska se on aina tosi.

Vastaus

a) ei ole tautologia

b) on tautologia

K12

P : ”Paavon päivänä käy lämmin tuuli.”

M : ”Marjasato on hyvä.”

Formalisoidaan lause ”jos Paavon päivänä käy lämmin etelätuuli, niin kesällä tulee hyvä marjasato. Koska nyt on huono marjasato, niin Paavon päivänä ei ollut lämmin etelätuuli.”

$$((P \Rightarrow M) \wedge \neg M) \Rightarrow \neg P$$

Tutkitaan lausetta totuustaulun avulla.

P	M	$\neg P$	$\neg M$	$P \Rightarrow M$	$(P \Rightarrow M) \wedge \neg M$	$((P \Rightarrow M) \wedge \neg M) \Rightarrow \neg P$
1	1	0	0	1	0	1
1	0	0	1	0	0	1
0	1	1	0	1	0	1
0	0	1	1	1	1	1

Lause on tosi kaikilla lauseiden P ja M totuusarvoilla, joten kyseessä on tautologia.

Vastaus

$$((P \Rightarrow M) \wedge \neg M) \Rightarrow \neg P, \text{ on tautologia}$$

K13

a) Muodostetaan lauseen $A \square (A \square B)$ totuustaulu.

A	B	$A \square B$	$A \square (A \square B)$
1	1	0	1
1	0	1	0
0	1	0	0
0	0	0	0

b) Lause $A \square (A \square B)$ on totta, kun A ja B ovat molemmat tosia. Lause on siis loogisesti yhtä pitävä esimerkiksi lauseen $A \wedge B$ kanssa.

A	B	$A \square (A \square B)$	$A \wedge B$
1	1	1	1
1	0	0	0
0	1	0	0
0	0	0	0

Vastaus

b) esimerkiksi $A \wedge B$

K14

Lause P on tosi silloin, kun C on tosi, tai kun A on epätosi ja B on tosi.

Lauseen P kanssa loogisesti yhtäpitävä lause on esimerkiksi $(\neg A \wedge B) \vee C$.

A	B	C	P	$\neg A$	$\neg A \wedge B$	$(\neg A \wedge B) \vee C$
1	1	1	1	0	0	1
1	1	0	0	0	0	0
1	0	1	1	0	0	1
1	0	0	0	0	0	0
0	1	1	1	1	1	1
0	1	0	1	1	1	1
0	0	1	1	1	0	1
0	0	0	0	1	0	0

Vastaus

esimerkiksi $(\neg A \wedge B) \vee C$

K15.

Tapa 1. Ilman laskinta

Määritetään kongruenssin avulla jakojäännös, kun päivien lukumäärä jaetaan luvulla 7.

Päivien lukumäärä voidaan esittää summana $1001 = 700 + 280 + 21$.

$$700 \equiv 0 \pmod{7}$$

$$280 \equiv 0 \pmod{7}$$

$$21 \equiv 0 \pmod{7}$$

$$1001 \equiv 700 + 280 + 21$$

$$\equiv 0 + 0 + 0$$

$$\equiv 0 \pmod{7}$$

Koska jakojäännös on nolla, 1001 yön kuluttua on myös sunnuntai.

Tapa 2. Laskimen avulla

Laskimella saadaan $1001 : 7 = 143$.

Koska 1001 yöhon sisältyy tasan 143 viikkoa, viikonpäivä on sunnuntai.

Vastaus

sunnuntai

K16.

a) Laskimella saadaan $127 : 6 = 21,166\dots$

Jakolaskun $127 : 6$ osamäärä on 21
ja jakojäännös $127 - 21 \cdot 6 = 1$.

Jakoyhtälö on $127 = 6 \cdot 21 + 1$.

b) Laskimella saadaan $423 : 8 = 22,875$.

Jakolaskun $423 : 8$ osamäärä on 52
ja jakojäännös $423 - 52 \cdot 8 = 7$.

Jakoyhtälö on $423 = 8 \cdot 52 + 7$.

c) Laskimella saadaan $1000 : 11 = 90,9090\dots$

Jakolaskun $1000 : 11$ osamäärä on 90
ja jakojäännös $1000 - 90 \cdot 11 = 10$.

Jakoyhtälö on $1000 = 11 \cdot 90 + 10$.

Vastaus

a) osamäärä 21, jakojäännös 1, jakoyhtälö $127 = 6 \cdot 21 + 1$

b) osamäärä 52, jakojäännös 7, jakoyhtälö $423 = 8 \cdot 52 + 7$

c) osamäärä 90, jakojäännös 10, jakoyhtälö $1000 = 11 \cdot 90 + 10$

K17.

a) Laskimella saadaan $-111 : 37 = -3$.

Jakoyhtälö on $-111 = 37 \cdot (-3) + 0$.

b) Laskimella saadaan $-109 : 37 = -2,945\dots$.

Etsitään suurin luvun 37 monikerta, joka on pienempi kuin -109 .

$$37 \cdot (-2) = -74 > -109$$

$$37 \cdot (-3) = -111 < -109$$

Lasketaan lukujen -109 ja -111 erotus: $-109 - (-111) = 2$.

Jakoyhtälö on $-109 = 37 \cdot (-3) + 2$.

c) Laskimella saadaan $-998 : 37 = -26,972\dots$.

Etsitään suurin luvun 37 monikerta, joka on pienempi kuin -998 .

$$37 \cdot (-26) = -962 > -998$$

$$37 \cdot (-27) = -999 < -998$$

Lasketaan lukujen -998 ja -999 erotus: $-998 - (-999) = 1$.

Jakoyhtälö on $-998 = 37 \cdot (-27) + 1$.

Vastaus

a) $-111 = 37 \cdot (-3) + 0$

b) $-109 = 37 \cdot (-3) + 2$

c) $-998 = 37 \cdot (-27) + 1$

K18.

a) **Tapa 1.** Luetellaan lukujen 36 ja 60 tekijät ja valitaan niistä suurin.

Luku 36 on jaollinen luvuilla 1, 2, 3, 4, 6, 9, 12, 18 ja 36.

Luku 60 on jaollinen luvuilla 1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30 ja 60.

Lukujen 36 ja 60 suurin yhteinen tekijä on 12.

Tapa 2. Määritetään lukujen 36 ja 60 suurin yhteinen tekijä Eukleideen algoritmilla.

$$60 = 36 \cdot 1 + 24$$

$$36 = 24 \cdot 1 + 12$$

$$24 = 12 \cdot 2 + 0$$

$$\text{syt}(36, 60) = 12$$

b) Määritetään lukujen 36 ja 60 pienin yhteinen monikerta.

$$12 \cdot \text{pym}(36, 60) = 36 \cdot 60 \quad |:12 \qquad \text{syt}(a, b) \cdot \text{pym}(a, b) = ab$$

$$\begin{aligned} \text{pym}(36, 60) &= \frac{36 \cdot 60}{12} \\ &= 180 \end{aligned}$$

Vastaus

a) 12

b) 180

K19.

Muodostetaan jakoyhtälöt laskimen avulla.

$$5538 = 455 \cdot 12 + 78$$

$$455 = 78 \cdot 5 + 65$$

$$78 = 65 \cdot 1 + 13$$

$$65 = 13 \cdot 5 + 0$$

Suurin yhteinen tekijä on viimeinen nollasta eroava jakojäännös.

$$\text{syt}(5538, 455) = 13$$

Vastaus

13

K20.

Määritetään lukujen 1386 ja 546 suurin yhteinen tekijä Eukleideen algoritmilla.

$$1386 = 546 \cdot 2 + 294$$

$$546 = 294 \cdot 1 + 252$$

$$294 = 252 \cdot 1 + 42$$

$$252 = 42 \cdot 6 + 0$$

$$\text{syt}(1386, 546) = 42$$

Siten 42 on suurin luku, jolla murtoluku $\frac{546}{1386}$ voidaan supistaa.

$$\frac{546}{1386} \stackrel{(42)}{=} \frac{13}{33}$$

Vastaus

voidaan supistaa luvulla 42, saadaan $\frac{13}{33}$

K21.

- a) Osoitetaan, että lukujen 398 ja 326 erotus on jaollinen luvulla 9.

$$398 - 326 = 72 = 9 \cdot 8$$

Siis $398 \equiv 326 \pmod{9}$. $\square$

- b) Osoitetaan, että lukujen 128 ja 0 erotus on jaollinen luvulla 32.

$$128 - 0 = 128 = 32 \cdot 4$$

Siis $128 \equiv 0 \pmod{32}$. $\square$

- c) Osoitetaan, että lukujen 443 ja -1 erotus on jaollinen luvulla 111.

$$443 - (-1) = 444 = 111 \cdot 4$$

Siis $443 \equiv -1 \pmod{111}$. $\square$

K22.

- a) Määritetään jakojäännös kongruenssin avulla.

Jakolaskun $16 : 5$ jakojäännös on 1, joten $16 \equiv 1 \pmod{5}$.

Korvataan luvut kongruenteilla luvuilla modulo 5.

$$16^{2018} \equiv 1^{2018} \equiv 1 \pmod{5}$$

Jakojäännös on 1.

- b) Jakolaskun $125 : 6$ jakojäännös on 5, joten $125 \equiv 5 \pmod{6}$.

Erotus $5 - (-1) = 6$ on jaollinen luvulla 6, joten $5 \equiv -1 \pmod{6}$.

Korvataan luvut kongruenteilla luvuilla modulo 6.

$$125^{101} \equiv (-1)^{101} \equiv -1 \equiv 5 \pmod{6}$$

Jakojäännös on 5.

Vastaus

a) 1

b) 5

K23.

Todistetaan jaollisuus kongruenssin avulla. Luku on jaollinen luvulla 16, jos se on kongruentti nollan kanssa modulo 16.

Erotus $20 - 4 = 16$ on jaollinen luvulla 16, joten $20 \equiv 4 \pmod{16}$.

Erotus $17 - 1 = 16$ on jaollinen luvulla 16, joten $17 \equiv 1 \pmod{16}$.

Erotus $19 - 3 = 16$ on jaollinen luvulla 16, joten $19 \equiv 3 \pmod{16}$.

Erotus $15 - (-1) = 16$ on jaollinen luvulla 16, joten $15 \equiv -1 \pmod{16}$.

Korvataan laskutoimituksen luvut kongruenteilla luvuilla modulo 16.

$$\begin{aligned} 20 \cdot 17^{2016} + 19 \cdot 15^{2017} - 1 &\equiv 4 \cdot 1^{2017} + 3 \cdot (-1)^{2017} - 1 \\ &\equiv 4 \cdot 1 + 3 \cdot (-1) - 1 \\ &\equiv 4 - 3 - 1 \\ &\equiv 0 \pmod{16} \end{aligned}$$

Siis luku $20 \cdot 17^{2016} + 19 \cdot 15^{2017} - 1$ on jaollinen luvulla 16. $\square$

K24

a) Jaetaan luku 96 tekijöihin vaiheittain, kunnes päädytään alkulukujen tuloon.

$$\begin{aligned} 96 &= 8 \cdot 12 \\ &= 2 \cdot 4 \cdot 2 \cdot 6 \\ &= 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 3 \\ &= 2^5 \cdot 3 \end{aligned}$$

b) Jaetaan luku 244 tekijöihin vaiheittain, kunnes päädytään alkulukujen tuloon.

$$\begin{aligned} 244 &= 4 \cdot 61 \\ &= 2 \cdot 2 \cdot 61 \\ &= 2^2 \cdot 61 \end{aligned}$$

c) Jaetaan luku 1200 tekijöihin vaiheittain, kunnes päädytään alkulukujen tuloon.

$$\begin{aligned} 1200 &= 30 \cdot 40 \\ &= 3 \cdot 10 \cdot 4 \cdot 10 \\ &= 3 \cdot 2 \cdot 5 \cdot 2 \cdot 2 \cdot 5 \\ &= 2^3 \cdot 3 \cdot 5^2 \end{aligned}$$

Vastaus

- a) $2^5 \cdot 3$
- b) $2^2 \cdot 61$
- c) $2^3 \cdot 3 \cdot 5^2$

K25

a) Muodostetaan lukujen 18, 63 ja 270 alkutekijähajotelmat.

$$18 = 6 \cdot 3 = 2 \cdot 3 \cdot 3 = 2 \cdot 3^2$$

$$63 = 9 \cdot 7 = 3 \cdot 3 \cdot 7 = 3^2 \cdot 7$$

$$270 = 10 \cdot 27 = 2 \cdot 3 \cdot 3 \cdot 3 \cdot 5 = 2 \cdot 3^3 \cdot 5$$

b) Suurin yhteinen tekijä on se yhteinen osa, joka sisältyy jokaiseen alkutekijähajotelmaan. Jokaisesta hajotelmasta löytyy 3^2 .

$$\text{syt}(18, 63, 270) = 3^2 = 9$$

c) Pienin yhteinen monikerta sisältää jokaisen alkutekijähajotelmissa esiintyvän alkuluvun eli luvut 2, 3, 5 ja 7. Jokaista alkulukua otetaan mukaan suurimman eksponentin ilmaisema määrä.

$$\text{pym}(18, 63, 270) = 2 \cdot 3^3 \cdot 5 \cdot 7 = 1890$$

Vastaus

a) $18 = 2 \cdot 3^2$, $63 = 3^2 \cdot 7$ ja $270 = 2 \cdot 3^3 \cdot 5$

b) 9

c) 1890

K26

a) Koska $\sqrt{379} = 19,467\dots$ täytyy tutkia jaollisuutta alkuluvuilla 2, 3, 5, 7, 11, 13, 17 ja 19.

$$379 : 2 = 189,5$$

$$379 : 3 = 126,333\dots$$

$$379 : 5 = 75,8$$

$$379 : 7 = 54,142\dots$$

$$379 : 11 = 34,454\dots$$

$$379 : 13 = 29,153\dots$$

$$379 : 17 = 22,294\dots$$

$$379 : 19 = 19,947\dots$$

Koska mikään jakolaskuista ei mene tasan, luku 379 on alkuluku.

b) Koska $\sqrt{8671} = 93,118\dots$ täytyy tutkia jaollisuutta alkuluvuilla, jotka ovat pienempiä tai yhtä suuria kuin luku 93.

$$8671 : 2 = 4335,5$$

$$8671 : 3 = 2890,333\dots$$

$$8671 : 5 = 1734,2$$

$$8671 : 7 = 1238,714\dots$$

$$8671 : 11 = 788,272\dots$$

$$8671 : 13 = 667$$

Koska 13 on luvun 8671 tekijä, niin 8671 ei ole alkuluku.

c) Koska $\sqrt{79625} = 282,179\dots$ täytyy tutkia jaollisuutta alkuluvuilla, jotka ovat pienempiä tai yhtä suuria kuin luku 282.

$$79625 : 2 = 39812,5$$

$$79625 : 3 = 26541,666\dots$$

$$79625 : 5 = 15925$$

Koska 5 on luvun 79625 tekijä, niin 79625 ei ole alkuluku.

(Tämä nähdään toisaalta jo suoraan siitä, että luku 79625 päättyy numeroon 5.)

Vastaus

- a) On. Luvuilla 2, 3, 5, 7, 11, 13, 17 ja 19 jakaminen riittää.
- b) Ei ole. Luku on jaollinen luvulla 13.
- c) Ei ole. Luku on jaollinen luvulla 5.

K27

Yhtälön $a \cdot b = \text{syt}(a, b) \cdot \text{pym}(a, b)$ perusteella

$$a \cdot b = 6 \cdot 60 = 2 \cdot 3 \cdot 6 \cdot 10 = (2 \cdot 3) \cdot (2 \cdot 3) \cdot 2 \cdot 5.$$

Kummankin luvun tekijät löytyvät tästä hajotelmasta. Koska lisäksi luku $6 = 2 \cdot 3$ on kummankin luvun tekijä, niin erilaiset vaihtoehdot saadaan selville jakamalla loput tekijät 2 ja 5 luvuille a ja b kaikilla mahdollisilla tavoilla.

$$a = 6 \cdot 2 \cdot 5 = 60$$

$$b = 6$$

$$a = 6 \cdot 2 = 12$$

$$b = 6 \cdot 5 = 30$$

$$a = 6 \cdot 5 = 30$$

$$b = 6 \cdot 2 = 12$$

$$a = 6$$

$$b = 6 \cdot 2 \cdot 5 = 60$$

Luvut a ja b ovat siis joko 6 ja 60 tai 12 ja 30.

Vastaus

6 ja 60 tai 12 ja 30

K28.

Oletetaan, että n on kokonaisluku.

Kirjoitetaan oletus
(eli lähtötiedot)
näkyviin.

Tehtävänä on osoittaa, että luku $2n(n+1)$ on
jaollinen luvulla 4.

Kirjoitetaan väite
näkyviin.

Jakoyhtälön nojalla jokainen kokonaisluku n voidaan
esittää muodossa

Aloitetaan päättely
oletuksesta.

$$n = 4q, n = 4q + 1, n = 4q + 2 \text{ tai } n = 4q + 3,$$

missä q on kokonaisluku.

1) Jos $n = 4q$, niin

$$\begin{aligned} 2n(n+1) &= 2 \cdot 4q(4q+1) \\ &= 4 \cdot 2q(4q+1) \end{aligned}$$

2) Jos $n = 4q + 1$, niin

$$\begin{aligned} 2n(n+1) &= 2 \cdot (4q+1)((4q+1)+1) \\ &= 32q^2 + 24q + 4 \\ &= 4 \cdot (8q^2 + 6q + 1) \end{aligned}$$

Sievennetään CAS-
laskimella.

3) Jos $n = 4q + 2$, niin

$$\begin{aligned} 2 \cdot (4q + 2)((4q + 2) + 1) &= (8q + 4)(4q + 3) \\ &= 4 \cdot (2q + 1)(4q + 3) \end{aligned}$$

4) Jos $n = 4q + 3$, niin

$$\begin{aligned} 2 \cdot (4q + 3)((4q + 3) + 1) &= 2 \cdot (4q + 3)(4q + 4) \\ &= 2 \cdot (4q + 3) \cdot 4 \cdot (q + 1) \\ &= 4 \cdot 2 \cdot (4q + 3)(q + 1) \end{aligned}$$

Jokaisessa tapauksessa luku $2n(n + 1)$ voidaan esittää kokonaislukujen tulona niin, että yksi tulon tekijöistä on luku 4.

Perustellaan, miksi väite on tosi.

Siis luku $2n(n + 1)$ on jaollinen luvulla 4. $\square$

K29

Jokainen kokonaisluku on muotoa $3q$, $3q + 1$ tai $3q + 2$ riippuen jakojäännöksestä luvulla 3 jaettaessa.

Sijoitetaan kukin näistä vaihtoehdoista vuorollaan luvun n paikalle ja osoitetaan, että jokin kolmesta luvusta n , $n + 2$ ja $n + 4$ jakautuu aina tekijöihin.

$$n = 3q:$$

$$\begin{array}{ll} n = 3q & \text{Yhdistetty luku} \\ n + 2 = 3q + 2 \\ n + 4 = 3q + 4 \end{array}$$

$$n = 3q + 1:$$

$$\begin{array}{ll} n = 3q + 1 \\ n + 2 = (3q + 1) + 2 = 3q + 3 = 3(q + 1) & \text{Yhdistetty luku} \\ n + 4 = (3q + 1) + 4 = 3q + 5 \end{array}$$

$$n = 3q + 2:$$

$$\begin{array}{ll} n = 3q + 2 \\ n + 2 = (3q + 2) + 2 = 3q + 4 \\ n + 4 = (3q + 2) + 4 = 3q + 6 = 3(q + 2) & \text{Yhdistetty luku} \end{array}$$

Kaikissa tapauksissa jokin luvuista n , $n + 2$ ja $n + 4$ on yhdistetty luku.

K30.

Oletetaan, että n on luonnollinen luku.

Kirjoitetaan oletus
(eli lähtötiedot)
näkyviin.

Tehtävänä on osoittaa, että luku $3^{2n+1} + 5^n$ on jaollinen luvulla 4.

Kirjoitetaan väite
näkyviin.

Todistetaan jaollisuus kongruenssin avulla.

Erotukset $3 - (-1) = 4$ ja $5 - 1 = 4$ ovat jaollisia luvulla 4, joten

$$3 \equiv -1 \pmod{4}$$

$$5 \equiv 1 \pmod{4}.$$

Korvataan laskutoimituksen luvut kongruenteilla luvuilla modulo 4.

$$3^{2n+1} + 5^n$$

$$\equiv (-1)^{2n+1} + 1^n$$

$$\equiv -1 + 1$$

$$\equiv 0 \pmod{4}$$

$(-1)^{2n+1} = -1$,
koska $2n+1$ on
pariton luku.

Koska $3^{2n+1} + 5^n \equiv 0 \pmod{4}$,

niin luku $3^{2n+1} + 5^n$ on jaollinen luvulla 4. $\square$

Perustellaan, miksi
väite on tosi.

K31.

a) 8530002

$$\begin{aligned} &= 8 \cdot 10^6 + 5 \cdot 10^5 + 3 \cdot 10^4 + 0 \cdot 10^3 + 0 \cdot 10^2 + 0 \cdot 10 + 2 \\ &= 8 \cdot 10^6 + 5 \cdot 10^5 + 3 \cdot 10^4 + 2 \end{aligned}$$

Koska $10 \equiv 1 \pmod{3}$, niin

$$\begin{aligned} &8 \cdot 10^6 + 5 \cdot 10^5 + 3 \cdot 10^4 + 2 \\ &\equiv 8 \cdot 1^6 + 5 \cdot 1^5 + 3 \cdot 1^4 + 2 \\ &\equiv 8 \cdot 1 + 5 \cdot 1 + 3 \cdot 1 + 2 \\ &\equiv 8 + 5 + 3 + 2 \\ &\equiv 18 \equiv 0 \pmod{3}. \end{aligned}$$

Siis luku 8 530 002 on jaollinen luvulla 3. $\square$

b) 13080627

$$\begin{aligned} &= 1 \cdot 10^7 + 3 \cdot 10^6 + 0 \cdot 10^5 + 8 \cdot 10^4 + 0 \cdot 10^3 + 6 \cdot 10^2 + 2 \cdot 10 + 7 \\ &= 1 \cdot 10^7 + 3 \cdot 10^6 + 8 \cdot 10^4 + 6 \cdot 10^2 + 2 \cdot 10 + 7 \end{aligned}$$

Koska $10 \equiv 1 \pmod{9}$, niin

$$\begin{aligned} &1 \cdot 10^7 + 3 \cdot 10^6 + 8 \cdot 10^4 + 6 \cdot 10^2 + 2 \cdot 10 + 7 \\ &\equiv 1 \cdot 1^7 + 3 \cdot 1^6 + 8 \cdot 1^4 + 6 \cdot 1^2 + 2 \cdot 1 + 7 \\ &\equiv 1 \cdot 1 + 3 \cdot 1 + 8 \cdot 1 + 6 \cdot 1 + 2 \cdot 1 + 7 \\ &\equiv 1 + 3 + 8 + 6 + 2 + 7 \\ &\equiv 27 \equiv 0 \pmod{9}. \end{aligned}$$

Siis luku 13 080 627 on jaollinen luvulla 9. $\square$

K32

a) Hyödynnetään Eratostheneen seulaa (laadittu esimerkiksi tehtävässä 7.3), jossa on korostettuna kaikki alkuluvut lukuun 120 saakka.

	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100
101	102	103	104	105	106	107	108	109	110
111	112	113	114	115	116	117	118	119	120

Taulukon perusteella kahdeksan ensimmäistä alkulukuparia ovat:

(3, 5), (5, 7), (11, 13), (17, 19), (29, 31), (41, 43), (59, 61), (71, 73)

b) Jokainen lukua 3 suurempi kokonaisluku on muotoa $6n - 2$, $6n - 1$, $6n$, $6n + 1$, $6n + 2$ tai $6n + 3$, missä n on jokin positiivinen kokonaisluku (esimerkiksi $10 = 6 \cdot 1 + 4 = 6 \cdot 2 - 2$ ja $21 = 6 \cdot 3 + 3$).

Huomataan, että muotoa $6n - 2$, $6n$, $6n + 2$ ja $6n + 3$ olevat luvut voidaan jakaa tekijöihin seuraavasti:

$$6n - 2 = 2 \cdot (3n - 1)$$

$$6n = 2 \cdot (3n)$$

$$6n + 2 = 2 \cdot (3n + 1)$$

$$6n + 3 = 3 \cdot (2n + 1)$$

Siispä vain luvut $6n - 1$ ja $6n + 1$ voivat olla alkulukuja ja siten muodostaa alkulukuparin.

Vastaus

a) (3, 5), (5, 7), (11, 13), (17, 19), (29, 31), (41, 43), (59, 61), (71, 73)

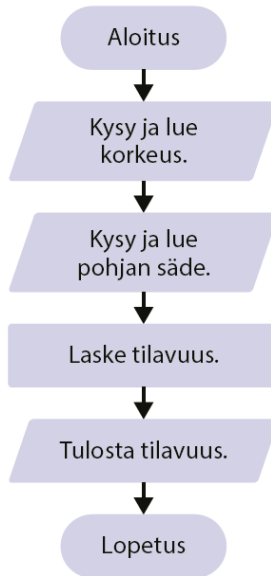
K33

Esitetään ohjelman toiminta pseudokoodina ja vuokaaviona.

Pseudokoodi

- Kysy ja lue korkeus.
- Kysy ja lue pohjan säde.
- Laske tilavuus
- Tulosta tilavuus näytölle.

Vuokaavio



K34

Kirjoitetaan ohjelma.

```
#Ohjelma kysyy ja lukee käyttäjältä suoran ympyrälieriön  
#korkeuden ja pohjan säteen, laskee ympyrälieriön tilavuuden  
#ja tulostaa sen näytölle.
```

```
import math  
#Kysytään ympyrän säde ja muutetaan se liukuluvuksi.  
h = float(input("Anna ympyrälieriön korkeus: "))  
r = float(input("Anna ympyrälieriön pohjan säde: "))  
  
#Lasketaan ympyrälieriön tilavuus.  
V = math.pi * r ** 2 * h  
  
#Tulostetaan ympyrälieriön tilavuus.  
print("Ympyrälieriön tilavuus:", round(V, 2))
```

Suoritetaan ohjelma.

```
Anna ympyrälieriön korkeus: 14  
Anna ympyrälieriön pohjan säde: 6.2  
Ympyrälieriön tilavuus: 1690.68
```

K35

Kirjoitetaan ohjelma.

```
#Ohjelma kysyy ja lukee käyttäjältä lämpötilan celsiusasteinta,  
#muuntaa lämpötilan fahrenheitasteiksi ja tulostaa lämpötilan  
#sekä celsius- että fahrenheitasteina.
```

```
#Kysytään lämpötila celsiusasteina.  
#Muutetaan merkkijono liukuluvuksi.  
c = float(input("Anna lämpötila celsiusasteina: "))
```

```
#Muunnetaan lämpötila fahrenheitasteiksi.  
f = 9 / 5 * c + 32
```

```
#Tulostetaan lämpötila sekä celsius- että fahrenheitasteina.  
print("Lämpötila", c, "°C on", round(f, 1), "°F.")
```

Suoritetaan ohjelma.

Anna lämpötila celsiusasteina: 12
Lämpötila 12.0 °C on 53.6 °F.

K36

Kirjan 1. painoksessa esimerkkitulostuksessa on virhe. Tulostuksen pitäisi olla:

Ohjelma tutkii, onko kokonaisluku jaollinen luvulla 3 tai 7.
Anna luku: 21
Luku 21 on jaollinen luvulla 3.
Luku 21 on jaollinen luvulla 7.
Ohjelman suoritus päättyi.

Ohjelma tutkii, onko kokonaisluku jaollinen luvulla 3 tai 7.
Anna luku: 10
Ohjelman suoritus päättyi.

Kirjoitetaan ohjelma.

```
#Ohjelma kysyy käyttäjältä kokonaisluvun ja tutkii, onko luku
#luvulla 3 tai luvulla 7. Mikäli luku on jaollinen, ohjelma tulostaa
#tiedon näytölle. Lopuksi ohjelma tulostaa: "Ohjelman suoritus päättyi".
```

```
print("Ohjelma tutkii, onko kokonaisluku jaollinen luvulla 3 tai 7.")
```

```
#Kysytään luku ja muutetaan merkkijono kokonaisluvuksi.
luku = int(input("Anna luku: "))
```

```
#Tutkitaan, onko luku jaollinen luvulla 3.
if luku % 3 == 0:
    print("Luku", luku, "on jaollinen luvulla 3.")
```

```
#Tutkitaan, onko luku jaollinen luvulla 7.
if luku % 7 == 0:
    print("Luku", luku, "on jaollinen luvulla 7.")
```

```
print("Ohjelman suoritus päättyi.")
```

Suoritetaan ohjelma.

Ohjelma tutkii, onko kokonaisluku jaollinen luvulla 3 tai 7.
Anna luku: 21
Luku 21 on jaollinen luvulla 3.
Luku 21 on jaollinen luvulla 7.
Ohjelman suoritus päättyi.

K37

Kirjoitetaan ohjelma.

```
#Ohjelma kysyy käyttäjältä kokonaisluvun ja tutkii, onko luku
#luvulla 3 ja luvulla 7. Ohjelma tulostaa tiedon, onko luku
#jaollinen vai ei. Lopuksi ohjelma tulostaa: "Ohjelman suoritus
#päättyi".

print("Ohjelma tutkii, onko kokonaisluku jaollinen luvulla 3 ja 7.")

#Kysytään luku ja muutetaan merkkijono kokonaisluvuksi.
luku = int(input("Anna luku: "))

#Tutkitaan, onko luku jaollinen luvulla 3.
if luku % 3 == 0:
    print("Luku", luku, "on jaollinen luvulla 3.")
else:
    print("Luku", luku, "ei ole jaollinen luvulla 3.")

#Tutkitaan, onko luku jaollinen luvulla 7.
if luku % 7 == 0:
    print("Luku", luku, "on jaollinen luvulla 7.")
else:
    print("Luku", luku, "ei ole jaollinen luvulla 7.")

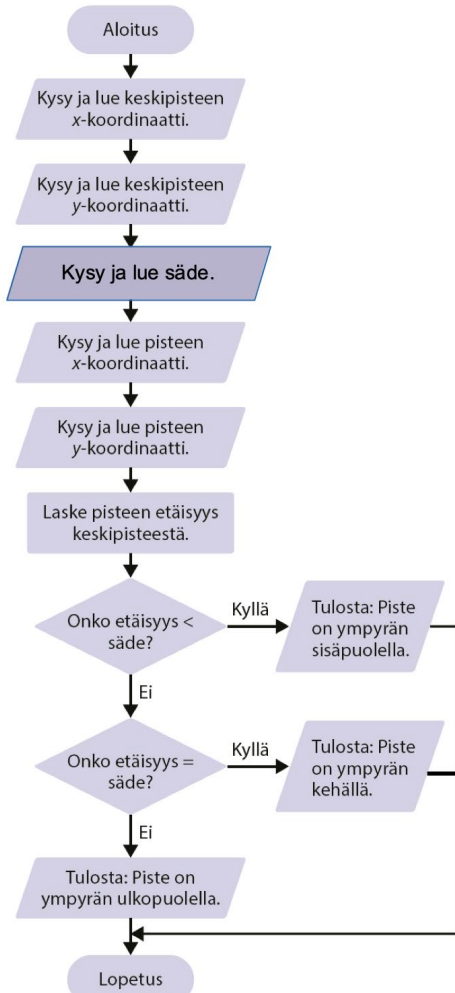
print("Ohjelman suoritus päättyi.")
```

Suoritetaan ohjelma.

Ohjelma tutkii, onko kokonaisluku jaollinen luvulla 3 ja 7.
Anna luku: 12
Luku 12 on jaollinen luvulla 3.
Luku 12 ei ole jaollinen luvulla 7.
Ohjelman suoritus päättyi.

K38

Suunnitellaan ohjelma vuokaavion avulla.



K39

Kirjoitetaan ohjelma.

```
#Ohjelma kysyy käyttäjältä xy-koordinaatistoon sijoitetun
#ympyrän keskipisteen koordinaatit, ympyrän säteen sekä
#tutkittavan pisteen koordinaatit. Ohjelma laskee pisteen
#etäisyyden ympyrän keskipisteestä ja tutkii etäisyyden
#avulla, onko piste ympyrän sisäpuolella, kehällä vai
#ulkopuolella. Ohjelma tulostaa tutkimuksen tuloksen.

import math

print("Ohjelma tutkii, onko piste ympyrän sisäpuolella,")
print("kehällä vai ulkopuolella.")
print()

#Kysytään keskipisteen x- ja y-koordinaatit, säde ja pisteen
#x- ja y-koordinaatit.
kp_x = float(input("Ympyrän keskipisteen x-koordinaatti: "))
kp_y = float(input("Ympyrän keskipisteen y-koordinaatti: "))

r = float(input("Ympyrän säde: "))

piste_x = float(input("Tutkittavan pisteen x-koordinaatti: "))
piste_y = float(input("Tutkittavan pisteen y-koordinaatti: "))

#Lasketaan pisteen etäisyys keskipisteestä.
etaisyys = math.sqrt((kp_x - piste_x) ** 2 + (kp_y - piste_y) ** 2)

#Tutkitaan, onko etäisyys pienempi, suurempi vai yhtä suuri
#kuin säde.
if etaisyys < r:
    print("Piste on ympyrän sisäpuolella.")
elif etaisyys == r:
    print("Piste on ympyrän kehällä.")
else:
    print("Piste on ympyrän ulkopuolella.")
```

Suoritetaan ohjelma.

```
Ympyrän keskipisteen x-koordinaatti: 0
Ympyrän keskipisteen y-koordinaatti: 0
Ympyrän säde: 5
Tutkittavan pisteen x-koordinaatti: -3
Tutkittavan pisteen y-koordinaatti: 0
Piste on ympyrän sisäpuolella.
```

K40

Kirjoitetaan ohjelma.

#Ohjelma pelaa yhden erän kivi-paperi-sakset-peliä
#käyttäjää vastaan.

```
import random
print("Pelataan kivi-paperi-sakset-peliä.")

#Kysytään käyttäjältä syöte.
ihminen = int(input("Kivi (1), paperi (2), sakset (3)? "))
#Arvotaan koneen valinta.
kone = random.randint(1,3)

#Tutkitaan, mitä käyttäjä valitsi.
if ihminen == 1:
    print("Sinä valitsit kiven.")
elif ihminen == 2:
    print("Sinä valitsit paperin.")
elif ihminen == 3:
    print("Sinä valitsit sakset.")
else:
    print("Sinä valitsit virheellisesti.")

#Tutkitaan, mitä kone valitsi.
if kone == 1:
    print("Minä valitsin kiven.")
elif kone == 2:
    print("Minä valitsin paperin.")
else:
    print("Minä valitsin sakset.")

#Tutkitaan, kumpi voittaa.
if kone == 1 and ihminen == 3:
    print("Voitin! Kivi voittaa sakset")
elif kone == 1 and ihminen == 2:
    print("Sinä voitit. Paperi voittaa kiven.")
elif kone == 2 and ihminen == 1:
    print("Voitin! Paperi voittaa kiven.")
elif kone == 2 and ihminen == 3:
    print("Sinä voitit. Sakset voittaa paperin.")
elif kone == 3 and ihminen == 1:
    print("Sinä voitit. Kivi voittaa sakset")
elif kone == 3 and ihminen == 2:
    print("Voitin! Sakset voittaa paperin.")
elif kone == ihminen:
    print("Tasapeli.")
else:
    print()
```

Suoritetaan ohjelma.

Pelataan kivi-paperi-sakset-peliä.
Kivi (1), paperi (2), sakset (3)? 3
Sinä valitsit sakset.
Minä valitsin sakset.
Tasapeli.

K41

Kirjoitetaan ohjelma.

```
#Ohjelma pelaa yhden erän kivi-paperi-sakset-peliä
#käyttäjää vastaan. Käyttäjä tekee valintansa antamalla
#merkin k (kivi), p (paperi) tai s (sakset).

import random
print("Pelataan kivi-paperi-sakset-peliä.")

#Kysytään käyttäjältä syöte.
ihminen = input("Kivi (k), paperi (p), sakset (s)? ")
#Arvotaan koneen valinta.
kone = random.randint(1,3)

#Tutkitaan, mitä käyttäjä valitsi ja muutetaan valinta luvuksi.
if ihminen == "k":
    print("Sinä valitsit kiven.")
    ihminen = 1
elif ihminen == "p":
    print("Sinä valitsit paperin.")
    ihminen = 2
elif ihminen == "s":
    print("Sinä valitsit sakset.")
    ihminen = 3
else:
    print("Sinä valitsit virheellisesti.")

#Tutkitaan, mitä kone valitsi.
if kone == 1:
    print("Minä valitsin kiven.")
elif kone == 2:
    print("Minä valitsin paperin.")
else:
    print("Minä valitsin sakset.")

#Tutkitaan, kumpi voittaa.
if kone == 1 and ihminen == 3:
    print("Voitin! Kivi voittaa sakset")
elif kone == 1 and ihminen == 2:
    print("Sinä voitit. Paperi voittaa kiven.")
elif kone == 2 and ihminen == 1:
    print("Voitin! Paperi voittaa kiven.")
elif kone == 2 and ihminen == 3:
    print("Sinä voitit. Sakset voittaa paperin.")
elif kone == 3 and ihminen == 1:
    print("Sinä voitit. Kivi voittaa sakset")
elif kone == 3 and ihminen == 2:
    print("Voitin! Sakset voittaa paperin.")
elif kone == ihminen:
    print("Tasapeli.")
else:
    print()
```

Suoritetaan ohjelma.

Pelataan kivi-paperi-sakset-peliä.

Kivi (k), paperi (p), sakset (s)? k

Sinä valitsit kiven.

Minä valitsin sakset.

Sinä voitit. Kivi voittaa sakset.

K42

Kirjoitetaan ohjelma.

```
#Ohjelma kysyy kokonaislukuja ja laskee niiden summan.
#Ohjelma jatkaa lukujen kysymistä ja summan laskemista,
#kunnes käyttäjä syöttää luvun 0.

#Asetetaan muuttujan luku arvoksi nollasta poikkeava arvo.
luku = -1
#Asetetaan muuttujan summa arvoksi 0.
summa = 0

#Tutkitaan, onko luku eri suuri kuin 0.
while luku != 0:
    luku = float(input("Anna luku: "))
    summa += luku

print("Lukujen summa:", summa)
```

Suoritetaan ohjelma.

```
Anna luku: 9
Anna luku: -9
Anna luku: 10
Anna luku: 0.4
Anna luku: 0.7
Anna luku: -1000
Anna luku: 0
Lukujen summa: -988.9
```

K43

Kirjoitetaan ohjelma.

```
#Ohjelma kysyy käyttäjältä positiivisen kokonaisluvun ja
#laskee luvun numeroiden summan. Ohjelma tarkastaa, onko
#käyttäjän antama kokonaisluku positiivinen.

print("Ohjelma kysyy käyttäjältä positiivisen kokonaisluvun")
print("ja laskee luvun numeroiden summan.")

luku = int(input("Anna positiivinen kokonaisluku: "))
#Tutkitaan, onko käyttäjän syöte virheellinen.
while luku <= 0:
    print("Et antanut positiivista kokonaislukua.")
    luku = int(input("Anna positiivinen kokonaisluku: "))

#Asetetaan muuttujan summa arvoksi 0.
summa = 0

#Muutetaan luku merkkijonoksi.
merkkijono = str(luku)

#Luetaan merkkijonoa merkki kerrallaan.
for merkki in merkkijono:
    #Lisätään merkki kokonaisluvuksi muutettuna summaan.
    summa += int(merkki)

print("Luvun", luku, "numeroiden summa:", summa)
```

Suoritetaan ohjelma.

Ohjelma kysyy käyttäjältä positiivisen kokonaisluvun
ja laskee luvun numeroiden summan.
Anna positiivinen kokonaisluku: 90909023123
Luvun 90909023123 numeroiden summa: 38

K44

Kirjoitetaan ohjelma.

```
#Ohjelma simuloi kolikon heittoa. Ohjelma kysyy
#käyttäjältä heittojen lukumäärän, heittää kolikkoa ja
#laskee, kuinka monta kertaa saatiin kruuna ja kuinka
#monta kertaa klaava. Ohjelma laskee, kuinka monta
#prosenttia heitoista oli kruunia ja kuinka monta
#prosenttia klaavoja.

import random

print("Ohjelma simuloi kolikon heittoa.")
lkm = int(input("Kuinka monta kertaa kolikkoa heitetään? "))

#Asetetaan summa-muuttujiin alkuarvoksi 0.
summa_kr = 0
summa_kl = 0

#Kolikkoa heitetään lkm kertaa.
for i in range(lkm):
    #Heitetään kolikkoa.
    kolikko = random.randint(1,2)
    #Tutkitaan, onko kruuna vai klaava.
    if kolikko == 1: #kruuna
        summa_kr += 1
    else:            #klaava
        summa_kl += 1

print("Kruunia yhteensä:", summa_kr)
print("Klaavoja yhteensä:", summa_kl)

#Lasketaan ja tulostetaan yhden desimaalin tarkkuudella,
#kuinka monta prosenttia oli kruunia ja kuinka monta klaavoja.
print("Kruunia:", round(summa_kr / lkm * 100, 1), "%")
print("Klaavoja:", round(summa_kl / lkm * 100, 1), "%")
```

Suoritetaan ohjelma.

```
Ohjelma simuloi kolikon heittoa.
Kuinka monta kertaa kolikkoa heitetään? 100000
Kruunia yhteensä: 50332
Klaavoja yhteensä: 49668
Kruunia: 50.3 %
Klaavoja: 49.7 %
```

K45

Kirjoitetaan ohjelma.

#Ohjelma kysyy käyttäjältä lukua 1 suuremman kokonaisluvun ja
#tulostaa kaikki alkuluvut siihen saakka.

```
import math
```

```
luku = int(input("Anna lukua 1 suurempi kokonaisluku: "))  
#Tutkitaan, onko käyttäjän syöte virheellinen.  
while luku <= 1:  
    print("Et antanut luku 1 suurempaa kokonaislukua.")  
    luku = int(input("Anna lukua 1 suurempi kokonaisluku: "))
```

```
print("Alkuluvut lukuun", luku, "saakka ovat:")
```

```
#Ulompi silmukka käy läpi luvut 2, ..., luku  
for i in range(2, luku + 1):  
    #math.isqrt pyöristää neliöjuuren alaspäin lähimpään  
    #kokonaislukuun  
    loppu = math.isqrt(i)  
    alkuluku = True #Asetetaan totuusarvoksi True.  
  
    #Sisempi silmukka käy läpi luvut 2, ..., loppu.  
    for j in range(2, loppu + 1):  
        #Tutkitaan, onko luku i jaollinen luvulla j.  
        if i % j == 0:  
            #Jos jaollinen, asetetaan totuusarvoksi False.  
            alkuluku = False  
  
    #Tutkitaan, onko muuttujan alkuluku arvo True.  
    if alkuluku == True:  
        print(i) #Tulostetaan luku.
```

Suoritetaan ohjelma.

Anna lukua 1 suurempi kokonaisluku: 23
Alkuluvut lukuun 23 saakka ovat:

```
2  
3  
5  
7  
11  
13  
17  
19  
23
```

K46

Kirjoitetaan ohjelma.

```
#Ohjelma pelaa yhden erän kivi-paperi-sakset-peliä
#käyttäjää vastaan. Käyttäjä tekee valintansa antamalla
#merkin k (kivi), p (paperi) tai s (sakset). Ohjelma
#tarkastaa syötteen oikeellisuuden, pelaa peliä niin kauan
#kuin käyttäjä haluaa ja laskee, kuinka monta kertaa
#tietokone voittaa, kuinka monta kertaa käyttäjä voittaa
#ja kuinka monta kertaa tulee tasapeli.

import random

jatketaanko = "k"

#Asetetaan muuttujien, joihin lasketaan voittojen ja
#tasapelien määrä, alkuarvoiksi 0.
voittaja_i = 0
voittaja_k = 0
tasapeli = 0

while jatketaanko != "e":
    print("Pelataan kivi-paperi-sakset-peliä.")
    #Kysytään käyttäjältä syöte.
    ihminen = input("Kivi (k), paperi (p), sakset (s)? ")
    #Tutkitaan, onko syöte virheellinen.
    while ihminen != "k" and ihminen != "p" and ihminen != "s":
        print("Annoit virheellisen syötteen. Syötä uudestaan.")
        ihminen = input("Kivi (k), paperi (p), sakset (s)? ")
    #Arvotaan koneen valinta.
    kone = random.randint(1,3)

    #Tutkitaan, mitä käyttäjä valitsi ja muutetaan valinta luvuksi.
    if ihminen == "k":
        print("Sinä valitsit kiven.")
        ihminen = 1
    elif ihminen == "p":
        print("Sinä valitsit paperin.")
        ihminen = 2
    else:
        print("Sinä valitsit sakset.")
        ihminen = 3

    #Tutkitaan, mitä kone valitsi.
    if kone == 1:
        print("Minä valitsin kiven.")
    elif kone == 2:
        print("Minä valitsin paperin.")
    else:
        print("Minä valitsin sakset.")
```

```

#Tutkitaan, kumpi voittaa. Lasketaan voittojen ja
#tasapelien lukumäärä.
if kone == 1 and ihminen == 3:
    print("Voitin! Kivi voittaa sakset")
    voittaja_k += 1
elif kone == 1 and ihminen == 2:
    print("Sinä voitit. Paperi voittaa kiven.")
    voittaja_i += 1
elif kone == 2 and ihminen == 1:
    print("Voitin! Paperi voittaa kiven.")
    voittaja_k += 1
elif kone == 2 and ihminen == 3:
    print("Sinä voitit. Sakset voittaa paperin.")
    voittaja_i += 1
elif kone == 3 and ihminen == 1:
    print("Sinä voitit. Kivi voittaa sakset")
    voittaja_i += 1
elif kone == 3 and ihminen == 2:
    print("Voitin! Sakset voittaa paperin.")
    voittaja_k += 1
else:
    print("Tasapeli.")
    tasapeli += 1

jatketaanko = input("Jatketaanko? (k/e) ")

print("Sinun voittojesi lukumäärä:", voittaja_i)
print("Minun voittojeni lukumäärä:", voittaja_k)
print("Tasapelien lukumäärä:", tasapeli)

```

Suoritetaan ohjelma.

```

Pelataan kivi-paperi-sakset-peliä.
Kivi (k), paperi (p), sakset (s)? k
Sinä valitsit kiven.
Minä valitsin sakset.
Sinä voitit. Kivi voittaa sakset
Jatketaanko? (k/e) k
Pelataan kivi-paperi-sakset-peliä.
Kivi (k), paperi (p), sakset (s)? p
Sinä valitsit paperin.
Minä valitsin kiven.
Sinä voitit. Paperi voittaa kiven.
Jatketaanko? (k/e) k
Pelataan kivi-paperi-sakset-peliä.
Kivi (k), paperi (p), sakset (s)? s
Sinä valitsit sakset.
Minä valitsin sakset.
Tasapeli.
Jatketaanko? (k/e) e
Sinun voittojesi lukumäärä: 2
Minun voittojeni lukumäärä: 0
Tasapelien lukumäärä: 1

```


K47

a) Muutetaan luku kymmenjärjestelmään.

$$10011_2$$

Luvussa on 5 numeroa, joten
2:n suurin eksponentti on 4.

$$\begin{aligned} &= 1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 \\ &= 19 \end{aligned}$$

Siis $10011_2 = 19_{10}$.

b) Kirjoitetaan luvun 49 jakoyhtälö 2:lla jaettaessa ja sen jälkeen saadun osamäärän jakoyhtälö ja niin edelleen, kunnes osamäärä on 0.

$$49$$

$$\begin{aligned} &= 2 \cdot 24 + 1 \\ &= 2 \cdot (2 \cdot 12 + 0) + 1 \\ &= 2^2 \cdot 12 + 2 \cdot 0 + 1 \\ &= 2^2 \cdot (2 \cdot 6 + 0) + 2 \cdot 0 + 1 \\ &= 2^3 \cdot 6 + 2^2 \cdot 0 + 2 \cdot 0 + 1 \\ &= 2^3 \cdot (2 \cdot 3 + 0) + 2^2 \cdot 0 + 2 \cdot 0 + 1 \\ &= 2^4 \cdot 3 + 2^3 \cdot 0 + 2^2 \cdot 0 + 2 \cdot 0 + 1 \\ &= 2^4 \cdot (2 \cdot 1 + 1) + 2^3 \cdot 0 + 2^2 \cdot 0 + 2 \cdot 0 + 1 \\ &= 2^5 \cdot 1 + 2^4 \cdot 1 + 2^3 \cdot 0 + 2^2 \cdot 0 + 2 \cdot 0 + 1 \\ &= 2^5 \cdot (2 \cdot 0 + 1) + 2^4 \cdot 1 + 2^3 \cdot 0 + 2^2 \cdot 0 + 2 \cdot 0 + 1 \\ &= 2^6 \cdot 0 + 2^5 \cdot 1 + 2^4 \cdot 1 + 2^3 \cdot 0 + 2^2 \cdot 0 + 2 \cdot 0 + 1 \\ &= 110001_2 \end{aligned}$$

Siis $49_{10} = 110001_2$.

Vastaus

a) 19

b) 110001_2

K48

Kirjoitetaan ohjelma.

```
#Ohjelma muuntaa kaksijärjestelmän luvun 101011
#kymmenjärjestelmään.

#Asetetaan kaksijärjestelmän luku merkkijonona.
luku2 = "101011"

#Muunnetaan kaksijärjestelmän luku kymmenjärjestelmän luvuksi.
#Asetetaan luvun 2 eksponentin arvo.
eksponentti = len(luku2) - 1

#Asetetaan kymmenjärjestelmän luvun arvoksi aluksi 0.
luku10 = 0

#Luetaan muuttujan luku2 arvoa merkki kerrallaan.
for merkki in luku2:
    #Jos merkki on 1, lisätään kymmenjärjestelmän lukuun
    #merkkiä vastaava luvun 2 potenssi.
    if merkki == "1":
        luku10 += 2 ** eksponentti
    #Vähennetään eksponentin arvoa yhdellä.
    eksponentti -= 1

print("2-järjestelmän luku", luku2, "10-järjestelmän lukuna:", luku10)
```

Suoritetaan ohjelma.

2-järjestelmän luku 101011 10-järjestelmän lukuna: 43

K49

Kirjoitetaan ohjelma.

```
#Ohjelma muuntaa kymmenjärjestelmän luvun 157
#kaksijärjestelmään.

#Asetetaan kymmenjärjestelmän luku.
luku10 = 157

#Asetetaan jaettavaksi luvuksi kymmenjärjestelmän luku.
jaettava = luku10

#Muunnetaan 10-järjestelmän luku kaksijärjestelmään jakojäännösten avulla.
#Asetetaan kaksijärjestelmän luvuksi tyhjä merkkijono.
luku2 = ""
while jaettava != 0:
    #Lasketaan luvun osamäärä ja jakojäännös.
    osam = jaettava // 2
    jakoj = jaettava % 2
    #Lisätään merkkijonon luku2 alkuun jakojäännöksen arvo merkkijonona.
    luku2 = str(jakoj) + luku2
    #Asetetaan tutkittavaksi luvuksi osamäärän arvo.
    jaettava = osam

print("10-järjestelmän luku", luku10, "kaksijärjestelmän lukuna:", luku2)
```

Suoritetaan ohjelma.

10-järjestelmän luku 157 2-järjestelmän lukuna: 10011101

K50

a) Muutetaan luvut kymmenjärjestelmään.

$$1000111_2 = 1 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 71$$

$$11001_2 = 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 25$$

Lasketaan lukujen summa kymmenjärjestelmässä.

$$1000111_2 + 11001_2 = 71 + 25 = 96$$

b) Muutetaan summan arvo 96 kaksijärjestelmään

$$96$$

$$= 2 \cdot 48 + 0$$

$$= 2 \cdot (2 \cdot 24 + 0) + 0$$

$$= 2^2 \cdot 24 + 2 \cdot 0 + 0$$

$$= 2^2 \cdot (2 \cdot 12 + 0) + 2 \cdot 0 + 0$$

$$= 2^3 \cdot 12 + 2^2 \cdot 0 + 2 \cdot 0 + 0$$

$$= 2^3 \cdot (2 \cdot 6 + 0) + 2^2 \cdot 0 + 2 \cdot 0 + 0$$

$$= 2^4 \cdot 6 + 2^3 \cdot 0 + 2^2 \cdot 0 + 2 \cdot 0 + 0$$

$$= 2^4 \cdot (2 \cdot 3 + 0) + 2^3 \cdot 0 + 2^2 \cdot 0 + 2 \cdot 0 + 0$$

$$= 2^5 \cdot 3 + 2^4 \cdot 0 + 2^3 \cdot 0 + 2^2 \cdot 0 + 2 \cdot 0 + 0$$

$$= 2^5 \cdot (2 \cdot 1 + 1) + 2^4 \cdot 0 + 2^3 \cdot 0 + 2^2 \cdot 0 + 2 \cdot 0 + 0$$

$$= 2^6 \cdot 1 + 2^5 \cdot 1 + 2^4 \cdot 0 + 2^3 \cdot 0 + 2^2 \cdot 0 + 2 \cdot 0 + 0$$

$$= 1100000_2$$

Vastaus

a) 96

b) 1100000_2

K51

a) Kirjoitetaan molemmille puolille kantalukujen potenssien summat ja sievennetään yhtälöä.

$$2X0_{10} = 11111010_2$$

$$2 \cdot 10^2 + X \cdot 10^1 + 0 = 1 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0$$

$$10X + 200 = 250$$

$$10X = 50$$

$$X = 5$$

Saatu ratkaisu $X = 5$ kelpaa kymmenjärjestelmän numeromerkiksi.

b) Kirjoitetaan molemmille puolille kantalukujen potenssien summat ja sievennetään yhtälöä.

$$4X_{10} = 10100X_2$$

$$4 \cdot 10^1 + X = 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + X$$

$$X + 40 = X + 40$$

$$0 = 0$$

Yhtälö on aina tosi, joten numeromerkiksi X :n paikalle kelpaa mikä hyvänsä kaksijärjestelmästä ja kymmenjärjestelmästä löytyvä merkki. Tällaisia ovat $X = 0$ ja $X = 1$.

Vastaus

a) $X = 5$

b) $X = 0$ tai $X = 1$

K52

Kirjoitetaan ohjelma.

```
#Ohjelma etsii yhtälön  $x^{1/2} - 1/2 * x^2 = 0$   
#välillä ]1, 2[ olevan ratkaisun kahden desimaalin  
#tarkkuudella.
```

```
#Asetetaan aloitusvälin päätepisteet..  
a = 1  
b = 2
```

```
#Lasketaan välin keskipiste.  
c = 0.5 * (a + b)  
#Lasketaan välin pituus.  
vali = abs(a - b)  
#Lasketaan funktion arvo kohdassa a.  
f_a = a ** (1 / 2) - 1 / 2 * a ** 2  
#Lasketaan funktion arvo kohdassa b.  
f_b = b ** (1 / 2) - 1 / 2 * b ** 2  
#Lasketaan funktion arvo kohdassa c.  
f_c = c ** (1 / 2) - 1 / 2 * c ** 2
```

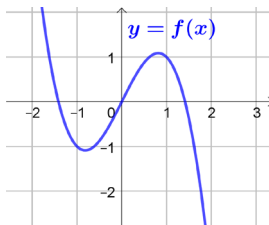
```
#Tutkitaan, onko välin pituus liian suuri ja ettei välin  
#keskipiste ole funktion nollakohta.  
while vali >= 0.001 and f_c != 0:  
    #Tutkitaan saako funktio erimerkkiset arvot välin  
    #alkupisteessä ja keskipisteessä.  
    if f_a * f_c < 0:  
        #Asetetaan keskipiste uudeksi loppupisteeksi.  
        b = c  
    else:  
        #Asetetaan keskipiste uudeksi alkupisteeksi.  
        a = c  
    #Lasketaan välin uusi keskipiste.  
    c = 0.5 * (a + b)  
    #Lasketaan välin uusi pituus.  
    vali = abs(a - b)  
    #Lasketaan funktion arvo kohdassa a.  
    f_a = a ** (1 / 2) - 1 / 2 * a ** 2  
    #Lasektaan funktion arvo kohdassa c.  
    f_c = c ** (1 / 2) - 1 / 2 * c ** 2  
  
print("Yhtälön ratkaisu:  $x \approx$ ", round(c, 2))
```

Suoritetaan ohjelma.

Yhtälön ratkaisu: $x \approx 1.59$

K53

a) Piirretään funktion f kuvaaja.



Kuvaajan perusteella positiivinen nollakohta on välillä $[1, 2]$.
Sievennetään yhtälö $f(x) = 0$ muotoon $x = g(x)$.

$$2x - x^3 = 0$$

$$x^3 = 2x$$

$$x = \sqrt[3]{2x}$$

Iteroidaan funktiota $g(x) = \sqrt[3]{2x}$ alkuarvolla $x_1 = 1$.

$$x_1 = 1$$

$$x_2 = g(1) = 1,2599210\dots$$

$$x_3 = g(1,259\dots) = 1,3607900\dots$$

$$x_4 = g(1,360\dots) = 1,3961766\dots$$

$$x_5 = g(1,396\dots) = 1,4081755\dots$$

$$x_6 = g(1,408\dots) = 1,4121980\dots$$

$$x_7 = g(1,412\dots) = 1,4135413\dots$$

$$x_8 = g(1,413\dots) = 1,4139894\dots$$

$$x_9 = g(1,413\dots) = 1,4141388\dots$$

$$x_{10} = g(1,414\dots) = 1,4141886\dots$$

$$x_{11} = g(1,414\dots) = 1,4142052\dots$$

$$x_{12} = g(1,414\dots) = 1,4142107\dots$$

Iteraatioista havaitaan, että funktion g kiintopisteen ja funktion f positiivisen nollakohdan likiarvo on neljän desimaalin tarkkuudella 1,4142.

b) Kirjoitetaan ohjelma.

```
#Ohjelma etsii funktion  $g(x) = (2 * x) ** (1 / 3)$ 
#ainoan positiivisen kiintopisteen.

#Asetetaan iteroinnin alkuarvo.
x = 1
#Lasketaan funktion arvo kohdassa x.
g_x = (2 * x) ** (1 / 3)

#Tutkitaan, onko arvo g_x liian kaukana arvosta x.
while abs(g_x - x) >= 0.00001:
    #Sijoitetaan muuttujan x uudeksi arvoksi g_x.
    x = g_x
    #Lasketaan funktion arvo kohdassa x.
    g_x = (2 * x) ** (1 / 3)

print("Kiintopiste:", round(g_x, 4))
```

Suoritetaan ohjelma.

Kiintopiste: 1.4142

Vastaus

$$x \approx 1,4142$$

K54

a) Kirjoitetaan ohjelma.

```
#Ohjelma etsii yhtälön  $x^2 - 3x - 4 = 0$ 
#välillä ]1, 2[ olevan ratkaisun kahden desimaalin
#tarkkuudella.

#Asetetaan aloitusvälin päätepisteet.
a = 1
b = 2

#Lasketaan välin keskipiste.
c = 0.5 * (a + b)
#Lasketaan välin pituus.
vali = abs(a - b)
#Lasketaan funktion arvo kohdassa a.
f_a = a ** (5) - 3 * a - 4
#Lasketaan funktion arvo kohdassa b.
f_b = b ** (5) - 3 * b - 4
#Lasketaan funktion arvo kohdassa c.
f_c = c ** (5) - 3 * c - 4

#Tutkitaan, onko välin pituus liian suuri ja ettei välin
#keskipiste ole funktion nollakohta.
while vali >= 0.001 and f_c != 0:
    #Tutkitaan saako funktio erimerkkiset arvot välin
    #alkupisteessä ja keskipisteessä.
    if f_a * f_c < 0:
        #Asetetaan keskipiste uudeksi loppupisteeksi.
        b = c
    else:
        #Asetetaan keskipiste uudeksi alkupisteeksi.
        a = c
    #Lasketaan välin uusi keskipiste.
    c = 0.5 * (a + b)
    #Lasketaan välin uusi pituus.
    vali = abs(a - b)
    #Lasketaan funktion arvo kohdassa a.
    f_a = a ** (5) - 3 * a - 4
    #Lasketaan funktion arvo kohdassa c.
    f_c = c ** (5) - 3 * c - 4

print("Yhtälön ratkaisu:  $x \approx$ ", round(c, 2))
```

Suoritetaan ohjelma.

Yhtälön ratkaisu: $x \approx 1.54$

b) Muokataan yhtälö $x^5 - 3x - 4 = 0$ muotoon $x = g(x)$.

$$x^5 + x - 3 = 0 \quad | +3x + 4$$

$$x^5 = 3x + 4$$

$$x = \sqrt[5]{3x + 4}$$

Kirjoitetaan ohjelma.

```
#Ohjelma etsii funktion g(x) = (3 * x + 4) ** (1 / 5)
#kiintopisteen alkuarvolla x = 1.

#Asetetaan iteroinnin alkuarvo.
x = 1
#Lasketaan funktion arvo kohdassa x.
g_x = (3 * x + 4) ** (1 / 5)

#Tutkitaan, onko arvo g_x liian kaukana arvosta x.
while abs(g_x - x) >= 0.001:
    #Sijoitetaan muuttujan x uudeksi arvoksi g_x.
    x = g_x
    #Lasketaan funktion arvo kohdassa x.
    g_x = (3 * x + 4) ** (1 / 5)

print("Kiintopiste:", round(g_x, 2))
```

Suoritetaan ohjelma.

Kiintopiste: 1.54

K55

CAS-laskimella saadaan yhtälön $x^3 - 2x = 1$ ratkaisujen likiarvot.

$$x^3 - 2x = 1$$

$$x = -1 \text{ tai } x \approx -0,61803 \text{ tai } x \approx 1,61803$$

a) Iteroidaan funktiota $f(x) = \frac{x^3 - 1}{2}$ alkuarvoilla $-1,5$; $-0,5$ ja $1,5$.

Käytetään taulukkolaskentaohjelmaa.

	A	B	C
1	-1,5	-0,5	1,5
2	-2,1875	-0,5625	1,1875
3	-5,7337646484375	-0,5889892578125	0,3372802734375
4	-94,7517868285095	-0,602162644566306	-0,480815838219314
5	-425336,585938204	-0,609172042451552	-0,555578433186343
6	-3,84740784941329E+16	-0,613029002455583	-0,585744473962074
7	-2,8475718066822E+049	-0,615189546609041	-0,600483465455714
8	-1,1545003198431E+148	-0,616411757546215	-0,60826128176755
9	#NUM!	-0,617106970501002	-0,612522797964462
10	#NUM!	-0,617503650830404	-0,614904431238306
11	#NUM!	-0,617730392826522	-0,616249976182777
12	#NUM!	-0,617860129196818	-0,617014788190425
13	#NUM!	-0,617934404109361	-0,617451001254537
14	#NUM!	-0,617976941021169	-0,617700281619423
15	#NUM!	-0,618001306326749	-0,617842894795799
16	#NUM!	-0,618015264377888	-0,617924535506303
17	#NUM!	-0,618023260964385	-0,617971288727012
18	#NUM!	-0,618027842382423	-0,617998068477812
19	#NUM!	-0,618030467229716	-0,618013409461438
20	#NUM!	-0,618031971109867	-0,618022198259414
21	#NUM!	-0,618032832748798	-0,618027233528839
22	#NUM!	-0,618033326421442	-0,618030118394936
23	#NUM!	-0,618033609269866	-0,618031771247713
24	#NUM!	-0,618033771327319	-0,618032718238758
25	#NUM!	-0,618033864177887	-0,618033260813285

Iteroimalla löydetään ratkaisun $x = -0,618...$ likiarvo.

b) Iteroidaan funktiota $g(x) = \sqrt[3]{2x+1}$ alkuarvoilla $-1,5$; $-0,5$ ja $1,5$. Käytetään taulukkolaskentaohjelmaa.

	A	B	C
1	-1,5	-0,5	1,5
2	-1,25992104989487	0	1,5874010519682
3	-1,14973960072352	1	1,61019558391094
4	-1,091247121073	1,44224957030741	1,61603551859011
5	-1,05746583496037	1,57197273764448	1,6175249301586
6	-1,03692994703961	1,60621869897872	1,61790434964357
7	-1,0240375320917	1,61501968402268	1,61800097625479
8	-1,01577486647583	1,61726604969857	1,61802558227217
9	-1,01040787792036	1,61783841403383	1,61803184808789
10	-1,00689099045616	1,61798418528748	1,61803344364296
11	-1,00457304898206	1,61802130649724	1,61803384994167
12	-1,00303945169501	1,61803075928381	1,61803395340321
13	-1,0020222090441	1,61803316638529	1,61803397974907
14	-1,00134632595571	1,61803377933965	1,61803398645789
15	-1,00089674624294	1,61803393542483	1,61803398816625
16	-1,00059747378261	1,61803397517099	1,61803398860127
17	-1,00039815730479	1,61803398529211	1,61803398871205
18	-1,00026536777691	1,61803398786939	1,61803398874026
19	-1,0001768805627	1,61803398852568	1,61803398874744
20	-1,00011790647265	1,6180339886928	1,61803398874927
21	-1,00007859813727	1,61803398873536	1,61803398874974
22	-1,00005239601279	1,61803398874619	1,61803398874985
23	-1,00003492945511	1,61803398874895	1,61803398874988
24	-1,00002328576118	1,61803398874965	1,61803398874989
25	-1,0000155235998	1,61803398874983	1,61803398874989

Iteroimalla löydetään ratkaisujen $x = -1$ ja $x = 1,618...$ likiarvot.

Vastaus

a) $x \approx -0,62$

b) $x \approx -1,00$ tai $x \approx 1,62$

K56

a) Sievennetään yhtälö muotoon $x = g(x)$.

$$3^x - x^4 + 1 = 0$$

$$3^x + 1 = x^4$$

$$x = \sqrt[4]{3^x + 1} \quad \text{tai} \quad x = -\sqrt[4]{3^x + 1}$$

Tarkoitus on löytää negatiivinen ratkaisu, joten määritellään

$$g(x) = -\sqrt[4]{3^x + 1} \quad \text{ja iteroidaan funktiota } g \text{ alkuarvolla } x_1 = -2.$$

$$x_1 = -2$$

$$x_2 = g(-2) = -1,0266900\dots$$

$$x_3 = g(-1,026\dots) = -1,0726239\dots$$

$$x_4 = g(-1,072\dots) = -1,0693822\dots$$

$$x_5 = g(-1,069\dots) = -1,0696066\dots$$

$$x_6 = g(-1,069\dots) = -1,0695910\dots$$

$$x_7 = g(-1,069\dots) = -1,0695921\dots$$

Iteraatioista havaitaan, että funktion g kiintopisteen ja yhtälön negatiivisen ratkaisun likiarvo on neljän desimaalin tarkkuudella $-1,070$. Ensimmäinen iteraatio, joka pyöristyy oikein tähän tarkkuuteen, on x_5 , eli tarvitaan neljä iteraatiokierrosta.

b) Kirjoitetaan ohjelma.

```
#Ohjelma etsii funktion  $g(x) = -(3 ** x + 1) ** (1 / 4)$ 
#kiintopisteen alkuarvolla  $x = -2$ . Ohjelma laskee
#iterointikierrosten määrän.

#Asetetaan iteroinnin alkuarvo.
x = -2
#Lasketaan funktion arvo kohdassa x.
g_x = -(3 ** x + 1) ** (1 / 4)

summa = 0

#Tutkitaan, onko arvo g_x liian kaukana arvosta x.
while abs(g_x - x) >= 0.0001:
    #Sijoitetaan muuttujan x uudeksi arvoksi g_x.
    x = g_x
    #Lasketaan funktion arvo kohdassa x.
    g_x = -(3 ** x + 1) ** (1 / 4)
    summa += 1

print("Kiintopiste:", round(g_x, 3))
print("Iterointikierroksia:", summa)
```

Suoritetaan ohjelma.

Kiintopiste: -1.07
Iterointikierroksia: 4

Vastaus

$x \approx -1,070$; tarvitaan vähintään 4 iterointikierrosta

A1.

- a) Huomataan, että $8 \cdot 20 = 160$ ja $163 - 160 = 3$.

Luku 8 menee 163:een 20 kertaa, ja yli jää 3.

Jakolaskun $163 : 8$ osamäärä on 20 ja jakojäännös 3.

Jakoyhtälö on $163 = 8 \cdot 20 + 3$.

- b) Huomataan, että $11 \cdot 9 = 99$ ja $100 - 99 = 1$.

Luku 11 menee 100:aan 9 kertaa, ja yli jää 1.

Jakolaskun $100 : 11$ osamäärä on 9 ja jakojäännös 1.

Jakoyhtälö on $100 = 11 \cdot 9 + 1$.

Vastaus

- a) osamäärä 20, jakojäännös 3, jakoyhtälö $163 = 8 \cdot 20 + 3$

- b) osamäärä 9, jakojäännös 1, jakoyhtälö $100 = 11 \cdot 9 + 1$

A2.

Osoitetaan lauseet loogisesti ekvivalenteiksi totuustaulun avulla.

A	B	$\neg A$	$\neg B$	$A \Rightarrow \neg B$	$\neg A \vee \neg B$
1	1	0	0	0	0
1	0	0	1	1	1
0	1	1	0	1	1
0	0	1	1	1	1

Koska lauseiden $A \Rightarrow \neg B$ ja $\neg A \vee \neg B$ totuustaulut ovat samat, lauseet ovat loogisesti ekvivalentit. $\square$

A3.

a) Etsitään luvuille 18 ja 42 pienin yhteinen monikerta.

Määritetään ensin lukujen suurin yhteinen tekijä Eukleideen algoritmillä.

$$42 = 18 \cdot 2 + 6 \qquad \text{sy}(42, 18) = \text{sy}(18, 6)$$

$$18 = 6 \cdot 3 + 0 \qquad \text{sy}(18, 6) = 6$$

Saatiin $\text{sy}(42, 18) = 6$.

Lasketaan seuraavaksi lukujen pienin yhteinen monikerta.

$$\text{pym}(42, 18) \cdot 6 = 42 \cdot 18 \qquad \text{pym}(a, b) \cdot \text{sy}(a, b) = ab$$

$$\text{pym}(42, 18) = \frac{42 \cdot 18}{6} = 126$$

b) Murtolukujen yhteiseksi nimittäjäksi tulee laventaa $\text{pym}(42, 18) = 126$. Laventajat saadaan selville jakolaskulla.

$$\frac{126}{42} = 3 \qquad \frac{126}{18} = 7$$

Lavennetaan murtoluvut.

$$^7) \frac{1}{18} = \frac{7}{126} \qquad ^3) \frac{1}{42} = \frac{3}{126}$$

Vastaus

a) 126

$$\text{b) } \frac{1}{18} = \frac{7}{126} \text{ ja } \frac{1}{42} = \frac{3}{126}$$

A4.

Muodostetaan jakoyhtälöt.

$$59\,052 = 3822 \cdot 15 + 1722$$

$$\text{syt}(59\,052, 3822) = \text{syt}(3822, 1722)$$

$$3822 = 1722 \cdot 2 + 378$$

$$\text{syt}(3822, 1722) = \text{syt}(1722, 378)$$

$$1722 = 378 \cdot 4 + 210$$

$$\text{syt}(1722, 378) = \text{syt}(378, 210)$$

$$378 = 210 \cdot 1 + 168$$

$$\text{syt}(378, 210) = \text{syt}(210, 168)$$

$$210 = 168 \cdot 1 + 42$$

$$\text{syt}(210, 168) = \text{syt}(168, 42)$$

$$168 = 42 \cdot 4 + 0$$

$$\text{syt}(168, 42) = 42$$

Siiis $\text{syt}(59\,052, 3822) = 42$.

Vastaus

42

A5

Kirjan 1. painoksessa tehtävänannon esimerkkitulostuksessa on virhe.

Tulostuksen pitäisi olla:

Anna kokonaisluku a: 5
Anna kokonaisluku b: 21
Anna moduloluku n: 4
Luvut 5 ja 21 ovat kongruentit modulo 4.

Kirjoitetaan ohjelma.

```
#Ohjelma kysyy käyttäjältä kaksi kokonaislukua a ja b sekä
#positiivisen kokonaisluvun n. Ohjelma tutkii, ovat luvut
#a ja b kongruentteja modulo n ja tulostaa tiedon näytölle.

print("Ohjelma tutkii ovatko luvut a ja b kongruentit modulo n.")

#Kysytään luvut käyttäjältä.
a = int(input("Anna kokonaisluku a: "))
b = int(input("Anna kokonaisluku b: "))

n = int(input("Anna moduloluku n: "))

#Tutkitaan, onko käyttäjän syöte virheellinen.
while n <= 0:
    print("Annoit virheellisen arvon.")
    print("Moduloluvun n tulee olla positiivinen.")
    n = int(input("Anna moduloluku n: "))

#Tutkitaan, ovatko luvut kongruentit modulo n.
if (a - b) % n == 0:
    print(f"Luvut {a} ja {b} ovat kongruentit modulo {n}.")
else:
    print(f"Luvut {a} ja {b} eivät ole kongruentit modulo {n}.")
```

Suoritetaan ohjelma.

Ohjelma tutkii ovatko luvut a ja b kongruentit modulo n.
Anna kokonaisluku a: 5
Anna kokonaisluku b: 21
Anna moduloluku n: 4
Luvut 5 ja 21 ovat kongruentit modulo 4.

A6.

Määritetään jakojäännös kongruenssin avulla.

$$17 \equiv -1 \pmod{18}$$

$$15 \equiv -3 \pmod{18}$$

$$183 \equiv 3 \pmod{18}$$

Korvataan laskutoimituksen luvut kongruenteilla luvuilla modulo 18.

$$\begin{aligned} 17^{19} + 15 \cdot 183 &\equiv (-1)^{19} + (-3) \cdot 3 \\ &\equiv -1 - 9 \\ &\equiv -10 \\ &\equiv 8 \pmod{18} \end{aligned}$$

Kongruenteilla luvuilla on sama jakojäännös.

Kun luku $17^{19} + 15 \cdot 183$ jaetaan luvulla 18, niin jakojäännös on 8.

Vastaus

Jakojäännös on 8.

A7

Kirjoitetaan ohjelma.

```
#Ohjelma kysyy käyttäjältä vallitsevan tuulen nopeuden (m/s),
#kertoo tuulen vaikuttavuuden ja antaa tarvittaessa
#kovan tuulen varoituksen tai myrskyvaroituksen.

nopeus = int(input("Anna tuulen nopeus kokonaislukuna (m/s): "))

print("Tuulen vaikuttavuus: ")

#Tutkitaan ja tulostetaan tuulen vaikuttavuus.

if nopeus < 0:
    print("-")
    print("Annoit virheellisen arvon.")
elif nopeus < 1:
    print("tyyntä")
elif nopeus < 4:
    print("heikkoa tuulta")
elif nopeus < 8:
    print("kohtalaista tuulta")
elif nopeus < 14:
    print("navakkaa tuulta")
elif nopeus < 21:
    print("kovaa tuulta")
    print("Annetaan kovan tuulen varoitus.")
elif nopeus < 33:
    print("myrskyä")
    print("Annetaan myrskyvaroitus.")
else:
    print("hirmumyrskyä")
    print("Annetaan myrskyvaroitus.")
```

Suoritetaan ohjelma.

```
Anna tuulen nopeus kokonaislukuna (m/s): 35
Tuulen vaikuttavuus:
hirmumyrskyä
Annetaan myrskyvaroitus.
```

A8.

a) Muutetaan luku 1101011_2 kymmenjärjestelmään.

$$\begin{aligned} 1101011_2 &= 1 \cdot 2^6 + 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 \\ &= 107_{10} \end{aligned}$$

Siis $1101011_2 = 107_{10}$.

b) Muutetaan luku 352_{10} kaksijärjestelmään.

$$352 = 2 \cdot 176 + 0$$

$$176 = 2 \cdot 88 + 0$$

$$88 = 2 \cdot 44 + 0$$

$$44 = 2 \cdot 22 + 0$$

$$22 = 2 \cdot 11 + 0$$

$$11 = 2 \cdot 5 + 1$$

$$5 = 2 \cdot 2 + 1$$

$$2 = 2 \cdot 1 + 0$$

$$1 = 2 \cdot 0 + 1$$

Siis $352_{10} = 101100000_2$.

Vastaus

a) 107

b) 101100000_2

A9

Puolitusmenetelmä:

Kirjoitetaan ohjelma.

```
#Ohjelma funktion  $f(x) = 4 * x ** 3 - 7 * x ** 2 - 1$ 
#välillä ]1, 2[ olevan nollakohdan kolmen desimaalin
#tarkkuudella.

#Asetetaan aloitusvälin päätepisteet.
a = 1
b = 2

#Lasketaan välin keskipiste.
c = 0.5 * (a + b)
#Lasketaan välin pituus.
vali = abs(a - b)
#Lasketaan funktion arvo kohdassa a.
f_a = 4 * a ** 3 - 7 * a ** 2 - 1
#Lasketaan funktion arvo kohdassa b.
f_b = 4 * b ** 3 - 7 * b ** 2 - 1
#Lasketaan funktion arvo kohdassa c.
f_c = 4 * c ** 3 - 7 * c ** 2 - 1

#Tutkitaan, onko välin pituus liian suuri ja ettei välin
#keskipiste ole funktion nollakohta.
while vali >= 0.0001 and f_c != 0:
    #Tutkitaan saako funktio erimerkkiset arvot välin
    #alkupisteessä ja keskipisteessä.
    if f_a * f_c < 0:
        #Asetetaan keskipiste uudeksi loppupisteeksi.
        b = c
    else:
        #Asetetaan keskipiste uudeksi alkupisteeksi.
        a = c
    #Lasketaan välin uusi keskipiste.
    c = 0.5 * (a + b)
    #Lasketaan välin uusi pituus.
    vali = abs(a - b)
    #Lasketaan funktion arvo kohdassa a.
    f_a = 4 * a ** 3 - 7 * a ** 2 - 1
    #Lasektaan funktion arvo kohdassa c.
    f_c = 4 * c ** 3 - 7 * c ** 2 - 1

print("Funktion nollakohta:  $x \approx$ ", round(c, 3))
```

Suoritetaan ohjelma.

Funktion nollakohta: $x \approx 1.825$.

Kiintopistemenetelmä:

Funktion $f(x) = 4x^3 - 7x^2 - 1$ nollakohdat ovat yhtälön $4x^3 - 7x^2 - 1 = 0$ ratkaisut.

Muokataan yhtälö $4x^3 - 7x^2 - 1 = 0$ muotoon $x = g(x)$

$$4x^3 - 7x^2 - 1 = 0 \quad \Big| +7x^2 + 1$$

$$4x^3 = 7x^2 + 1 \quad \Big| : 4$$

$$x^3 = \frac{7x^2 + 1}{4}$$

$$x = \sqrt[3]{\frac{7x^2 + 1}{4}}$$

Valitaan iteroinnin alkuarvoksi $x_1 = 1$.

Kirjoitetaan ohjelma.

```
#Ohjelma etsii funktion g(x) = ((7 * x ** 2 + 1) / 4) ** (1 / 3)
#kiintopisteen alkuarvolla x = 1.
```

```
#Asetetaan iteroinnin alkuarvo.
x = 1
#Lasketaan funktion arvo kohdassa x.
g_x = ((7 * x ** 2 + 1) / 4) ** (1 / 3)
```

```
#Tutkitaan, onko arvo g_x liian kaukana arvosta x.
while abs(g_x - x) >= 0.0001:
    #Sijoitetaan muuttujan x uudeksi arvoksi g_x.
    x = g_x
    #Lasketaan funktion arvo kohdassa x.
    g_x = ((7 * x ** 2 + 1) / 4) ** (1 / 3)
```

```
print("Kiintopiste:", round(g_x, 3))
```

Suoritetaan ohjelma.

Kiintopiste: 1.825

A10.

Oletetaan, että n on luonnollinen luku.

On osoitettava, että luku $6^{2n+1} + 1$ on jaollinen luvulla 7.

Osoitetaan jaollisuus kongruenssin avulla.

Koska $6 \equiv -1 \pmod{7}$, niin

$$\begin{aligned} 6^{2n+1} + 1 &\equiv (-1)^{2n+1} + 1 & (-1)^{2n+1} &= -1, \text{ koska } 2n+1 \text{ on pariton.} \\ &\equiv -1 + 1 \\ &\equiv 0 \pmod{7}. \end{aligned}$$

Kun luku $6^{2n+1} + 1$ jaetaan luvulla 7, jakojäännös on 0.

Luku $6^{2n+1} + 1$ on siten jaollinen luvulla 7, kun n on luonnollinen luku. $\square$

B1.

a) ”Saadaan kolme klaavaa” voidaan kirjoittaa toisin:

”ensimmäisellä heitolla saadaan klaava
ja toisella heitolla saadaan klaava
ja kolmannella heitolla saadaan klaava.”

Lause on formalisoituna $A \wedge B \wedge C$.

b) ”Saadaan ainakin yksi klaava” voidaan kirjoittaa toisin:

”ensimmäisellä heitolla saadaan klaava
tai toisella heitolla saadaan klaava
tai kolmannella heitolla saadaan klaava.”

Lause on formalisoituna $A \vee B \vee C$.

c) ”Saadaan tasan yksi klaava” voidaan kirjoittaa toisin:

”ensimmäisellä heitolla saadaan **klaava**

ja toisella heitolla saadaan **kruuna**

ja kolmannella heitolla saadaan **kruuna**

TAI

ensimmäisellä heitolla saadaan **kruuna**

ja toisella heitolla saadaan **klaava**

ja kolmannella heitolla saadaan **kruuna**

TAI

ensimmäisellä heitolla saadaan **kruuna**

ja toisella heitolla saadaan **kruuna**

ja kolmannella heitolla saadaan **klaava.**”

Lause on formalisoituna

$$(A \wedge \neg B \wedge \neg C) \vee (\neg A \wedge B \wedge \neg C) \vee (\neg A \wedge \neg B \wedge C).$$

Vastaus

a) $A \wedge B \wedge C$

b) $A \vee B \vee C$

c) $(A \wedge \neg B \wedge \neg C) \vee (\neg A \wedge B \wedge \neg C) \vee (\neg A \wedge \neg B \wedge C)$

B2.

a) Saadaan $-35 : 7 = -5$.

$$\text{Siis } -35 = 7 \cdot (-5) + 0.$$

b) Laskimella saadaan $-36 : 7 = -5,142\dots$

Etsitään suurin luvun 7 monikerta, joka on pienempi kuin -36 .

$$7 \cdot (-5) = -35 > -36$$

$$7 \cdot (-6) = -42 < -36$$

Lasketaan lukujen -36 ja -42 erotus: $-36 - (-42) = 6$

$$\text{Siis } -36 = 7 \cdot (-6) + 6.$$

Vastaus

a) $-35 = 7 \cdot (-5) + 0$

b) $-36 = 7 \cdot (-6) + 6$

B3

Kirjoitetaan ohjelma.

```
#Ohjelma kysyy käyttäjältä neliön pinta-alan. Ohjelma
#tarkastaa, että käyttäjän antama pinta-ala on positiivinen
#luku. Ohjelma laskee neliön sivun pituuden ja piirin kolmen
#desimaalin tarkkuudella.
```

```
A = float(input("Anna neliön pinta-ala: "))
```

```
while A <= 0:
    print("Annoit virheellisen syötteen.")
    A = float(input("Anna neliön pinta-ala: "))
```

```
#Lasketaan sivun pituus.
```

```
x = A ** (1 / 2)
```

```
#Lasketaan piiri.
```

```
p = 4 * x
```

```
print("Neliön sivun pituus:", round(x, 3))
```

```
print("Neliön piiri:", round(p, 3))
```

Suoritetaan ohjelma.

Anna neliön pinta-ala: 9

Neliön sivun pituus: 3.0

Neliön piiri: 12.0

B4.

- a) Muodostetaan lukujen 24 ja 150 alkutekijähajotelmat.

$$24 = 3 \cdot 8 = 3 \cdot 2 \cdot 2 \cdot 2 = 2^3 \cdot 3$$

$$150 = 10 \cdot 15 = 2 \cdot 5 \cdot 3 \cdot 5 = 2 \cdot 3 \cdot 5^2$$

- b) Suurin yhteinen tekijä on se yhteinen osa, joka sisältyy molempiin alkutekijähajotelmaan.

Molemmista hajotelmistä löytyy $2 \cdot 3$.

$$\text{Siis } \text{syt}(24, 150) = 2 \cdot 3 = 6.$$

- c) Pienin yhteinen monikerta sisältää jokaisen alkulukuhajotelmissa esiintyvän alkuluvun eli luvut 2, 3, 5. Jokaista alkulukua otetaan mukaan suurimman eksponentin ilmaisema määrä.

$$\text{Siis } \text{pym}(24, 150) = 2^3 \cdot 3 \cdot 5^2 = 600.$$

Vastaus

a) $24 = 2^3 \cdot 3$ ja $150 = 2 \cdot 3 \cdot 5^2$

b) 6

c) 600

B5.

Olkoon

A : ”Heimo on kelmi”,

B : ”Miia on kelmi” ja

C : ”Silja on kelmi”.

Formalisoidaan lauseet.

Heimo sanoo: ”Miia ja Silja ovat molemmat kelmejä”
eli ”Miia on kelmi ja Silja on kelmi”.

Heimon lause on formalisoituna $B \wedge C$.

Miia sanoo: ”Heimo ja Silja ovat molemmat ritareita”
eli ”Heimo on ritari ja Silja on ritari”.

Miian lause on formalisoituna $\neg A \wedge \neg C$.

Silja sanoo: ”Heimo on ritari tai Miia on kelmi”.

Siljan lause on formalisoituna $\neg A \vee B$.

Laaditaan formalisoitujen lauseiden totuustaulu.

	H	M	S	Heimo:			Miia:	Silja:
	A	B	C	$\neg A$	$\neg C$	$B \wedge C$	$\neg A \wedge \neg C$	$\neg A \vee B$
1	1	1	1	0	0	1	0	1
2	1	1	0	0	1	0	0	1
3	1	0	1	0	0	0	0	0
4	1	0	0	0	1	0	0	0
5	0	1	1	1	0	1	0	1
6	0	1	0	1	1	0	1	1
7	0	0	1	1	0	0	0	1
8	0	0	0	1	1	0	1	1

Kelmit valehtelevat aina, ja ritarit puhuvat aina totta.

Jos Heimo on kelmi, hänen väitteensä ei ole totta, joten rivit 2, 3, ja 4 ovat mahdollisia.

Jos Heimo on ritari, hänen väitteensä on totta. Siten myös rivi 5 on mahdollinen. Eli rivit 2, 3, 4 ja 5 ovat mahdollisia.

Jos Miia on kelmi, hänen väitteensä ei ole totta, joten rivit 2 ja 5 ovat edelleen mahdollisia.

Jos Miia on ritari, hänen väitteensä on totta ja rivit 3 ja 4 eivät voi pitää paikkaansa.

Jos Silja on kelmi, hänen väitteensä ei ole totta, joten rivi 5 ei ole mahdollinen.

Jos Silja on ritari, hänen väitteensä on totta, joten rivi 2 on mahdollinen.

Rivi 2 on ainoa mahdollinen rivi. Sen perusteella Heimo on kelmi, Miia on kelmi ja Silja on ritari.

Vastaus

Heimo ja Miia ovat kelmejä, Silja on ritari.

B6

Kirjoitetaan ohjelma.

```
#Ohjelma tutkii, kuinka monta kokonaisluvulla n jaollista
#lukua on välillä [a, b].

n = int(input("Anna kokonaisluku n: "))

a = int(input("Anna välin alaraja: "))
b = int(input("Anna välin yläraja: "))

#Asetetaan muuttujan summa arvoksi 0.
summa = 0

#Silmukka käy läpi luvut a, ..., b.
for luku in range(a, b + 1):
    #Tutkitaan, onko luku jaollinen luvulla n.
    if luku % n == 0:
        summa += 1

print(f"Välillä [{a}, {b}] on {summa} kokonaislukua,")
print(f"jotka ovat jaollisia luvulla {n}.")
```

Suoritetaan ohjelma.

```
Anna kokonaisluku n: 9
Anna välin alaraja: 123
Anna välin yläraja: 987
Välillä [123, 987] on 96 kokonaislukua,
jotka ovat jaollisia luvulla 9.
```

B7.

Oletetaan, että luku n on positiivinen kokonaisluku.

On osoitettava, että luku $n^3 + 11n$ on jaollinen luvulla 6.

Osoitetaan jaollisuus kongruenssin avulla.

Jokainen kokonaisluku n toteuttaa yhden kongruensseista $n \equiv 0$, $n \equiv 1$, $n \equiv 2$, $n \equiv 3$, $n \equiv 4$ tai $n \equiv 5 \pmod{6}$.

Osoitetaan, että kaikissa tapauksissa $n^3 + 11n \equiv 0 \pmod{6}$.

1) Jos $n \equiv 0 \pmod{6}$, niin

$$n^3 + 11n \equiv 0^3 + 11 \cdot 0 \equiv 0 \pmod{6}.$$

2) Jos $n \equiv 1 \pmod{6}$, niin

$$n^3 + 11n \equiv 1^3 + 11 \cdot 1 \equiv 12 \equiv 0 \pmod{6}.$$

3) Jos $n \equiv 2 \pmod{6}$, niin

$$n^3 + 11n \equiv 2^3 + 11 \cdot 2 \equiv 30 \equiv 0 \pmod{6}.$$

4) Jos $n \equiv 3 \pmod{6}$, niin

$$n^3 + 11n \equiv 3^3 + 11 \cdot 3 \equiv 60 \equiv 0 \pmod{6}.$$

5) Jos $n \equiv 4 \pmod{6}$, niin

$$n^3 + 11n \equiv 4^3 + 11 \cdot 4 \equiv 108 \equiv 0 \pmod{6}.$$

6) Jos $n \equiv 5 \pmod{6}$, niin

$$n^3 + 11n \equiv 5^3 + 11 \cdot 5 \equiv 180 \equiv 0 \pmod{6}.$$

Huomataan, että aina $n^3 + 11n \equiv 0 \pmod{6}$.

Siis kaikilla kokonaisluvuilla n luku $n^3 + 11n$ on jaollinen luvulla 6. $\square$

B8

a) Kirjoitetaan ohjelma.

```
#Ohjelma etsii funktion  $f(x) = x^5 - (2x + 1)^2$ 
#välillä ]0, 3[ olevan nollakohdan kuuden desimaalin
#tarkkuudella.

#Asetetaan aloitusvälin päätepisteet.
a = 0
b = 3

#Lasketaan välin keskipiste.
c = 0.5 * (a + b)
#Lasketaan välin pituus.
vali = abs(a - b)
#Lasketaan funktion arvo kohdassa a.
f_a = a ** 5 - (2 * a + 1) ** 2
#Lasketaan funktion arvo kohdassa b.
f_b = b ** 5 - (2 * b + 1) ** 2
#Lasketaan funktion arvo kohdassa c.
f_c = c ** 5 - (2 * c + 1) ** 2

#Tutkitaan, onko välin pituus liian suuri ja ettei välin
#keskipiste ole funktion nollakohta.
while vali >= 0.0000001 and f_c != 0:
    #Tutkitaan saako funktio erimerkkiset arvot välin
    #alkupisteessä ja keskipisteessä.
    if f_a * f_c < 0:
        #Asetetaan keskipiste uudeksi loppupisteeksi.
        b = c
    else:
        #Asetetaan keskipiste uudeksi alkupisteeksi.
        a = c
    #Lasketaan välin uusi keskipiste.
    c = 0.5 * (a + b)
    #Lasketaan välin uusi pituus.
    vali = abs(a - b)
    #Lasketaan funktion arvo kohdassa a.
    f_a = a ** 5 - (2 * a + 1) ** 2
    #Lasketaan funktion arvo kohdassa c.
    f_c = c ** 5 - (2 * c + 1) ** 2

print("Funktion nollakohta:  $x \approx$ ", round(c, 6))
```

Suoritetaan ohjelma.

Funktion nollakohta: $x \approx 1.860399$

b) Funktion $f(x) = x^5 - (2x + 1)^2$ nollakohdat ovat yhtälön $x^5 - (2x + 1)^2 = 0$ ratkaisut.

Muokataan yhtälö $x^5 - (2x + 1)^2 = 0$ muotoon $x = g(x)$

$$x^5 - (2x + 1)^2 = 0 \quad | + (2x + 1)^2$$

$$x^5 = (2x + 1)^2$$

$$x = \sqrt[5]{(2x + 1)^2} = (2x + 1)^{2/5}$$

Valitaan iteroinnin alkuarvoksi $x_1 = 0$.

Kirjoitetaan ohjelma.

```
#Ohjelma etsii funktion g(x) = (2 * x + 1) ** (2 / 5)
#kiintopisteen alkuarvolla x = 0.
```

```
#Asetetaan iteroinnin alkuarvo.
```

```
x = 0
```

```
#Lasketaan funktion arvo kohdassa x.
```

```
g_x = (2 * x + 1) ** (2 / 5)
```

```
#Tutkitaan, onko arvo g_x liian kaukana arvosta x.
```

```
while abs(g_x - x) >= 0.0000001:
```

```
    #Sijoitetaan muuttujan x uudeksi arvoksi g_x.
```

```
    x = g_x
```

```
    #Lasketaan funktion arvo kohdassa x.
```

```
    g_x = (2 * x + 1) ** (2 / 5)
```

```
print("Kiintopiste:", round(g_x, 6))
```

Suoritetaan ohjelma.

Kiintopiste: 1.860399

Nollakohta on $x \approx 1,860\,399$

B9

1. Kirjoitetaan ohjelma.

```
#Tasojoukon A pisteet määräytyvät epäyhtälöistä  $0 \leq x \leq 2$ ,  
# $0 \leq y \leq 4$  ja  $y \geq x ** 2$ .  
#Ohjelma arpoo 1000 pistettä suorakulmiosta B, jonka määrittävät  
#epäyhtälöt  $0 \leq x \leq 2$  ja  $0 \leq y \leq 4$ . Ohjelma tulostaa niiden  
#pisteiden lukumäärän, jotka kuuluvat joukkoon A.
```

```
import random
```

```
osuma = 0
```

```
#Arvotaan 1000 pistettä suorakulmiosta B.  
for i in range(1000):  
    x = random.uniform(0,2)  
    y = random.uniform(0,4)  
    #Tutkitaan, onko piste myös tasojoukossa A.  
    if y >= x ** 2:  
        osuma += 1
```

```
print("Tasojoukkoon A kuuluvien pisteiden lukumäärä:", osuma)
```

Suoritetaan ohjelma.

Tasojoukkoon A kuuluvien pisteiden lukumäärä: 677

2.2 Lasketaan Hillen tulosten keskiarvo.

$$\frac{673 + 664 + 672 + 679 + 667 + 650 + 640 + 678 + 660 + 667}{10} = 665$$

Suorakulmion B pinta-ala on $2 \cdot 4 = 8$. Merkitään kirjaimella x joukon A pinta-alaa.

Pinta-ala	Todennäköisyys
x	665/1000
8	1

Todennäköisyys on suoraan verrannollinen pinta-alaan. Muodostetaan verrantoyhtälö ja ratkaistaan x .

$$\frac{x}{7} = \frac{\frac{665}{1000}}{1}$$

Ratkaistaan CAS-laskimella.

$$x = 5,32 \approx 5,3$$

Pinta-ala on noin 5,3.

Vastaus

2. keskiarvo 665, pinta-ala noin 5,3

B10.

Olkoon n ja m kokonaislukuja.

On osoitettava seuraava väite:

n on jonkin luvun m kuutio $\Leftrightarrow$ luvun n alkutekijähajotelmassa jokaisen tekijän eksponentti on jaollinen luvulla 3.

Tulee siis osoittaa, että

1) " $\Leftarrow$ " Jos luvun n alkutekijähajotelmassa jokaisen tekijän eksponentti on jaollinen luvulla 3, niin kokonaisluku n on jonkin luvun m kuutio.

ja

2) " $\Rightarrow$ " Jos luku n on jonkin luvun m kuutio, niin luvun n alkutekijähajotelmassa jokaisen tekijän eksponentti on jaollinen luvulla 3.

Osoitetaan ensin kohta 1.

Jos luvun n alkutekijähajotelmassa jokaisen tekijän eksponentti on jaollinen luvulla 3, luvun n alkutekijähajotelma on muotoa:

$$n = p_1^{3 \cdot q_1} \cdot p_2^{3 \cdot q_2} \cdot \dots \cdot p_k^{3 \cdot q_k}.$$

Sievennetään alkutekijähajotelma.

$$n = p_1^{3 \cdot q_1} \cdot p_2^{3 \cdot q_2} \cdot \dots \cdot p_k^{3 \cdot q_k} = \underbrace{(p_1^{q_1} \cdot p_2^{q_2} \cdot \dots \cdot p_k^{q_k})}_=m^3 = m^3$$

Tällöin kokonaisluku n on jonkin toisen kokonaisluvun m kuutio.

Osoitetaan kohta 2.

Jos kokonaisluku n on jonkin toisen kokonaisluvun m kuutio, niin $n = m^3$. Luvun m alkutekijähajotelma on muotoa

$$m = s_1^{t_1} \cdot s_2^{t_2} \cdot \dots \cdot s_k^{t_k}.$$

Sijoitetaan tämä lausekkeeseen $n = m^3$.

$$n = m^3 = (s_1^{t_1} \cdot s_2^{t_2} \cdot \dots \cdot s_k^{t_k})^3 = s_1^{3 \cdot t_1} \cdot s_2^{3 \cdot t_2} \cdot \dots \cdot s_k^{3 \cdot t_k}$$

Tällöin luvun n alkutekijähajotelmassa jokaisen tekijän eksponentti on jaollinen luvulla 3.

Koska molemmat kohdat 1 ja 2 pätevät, niin väite on todistettu. $\square$