TARTU ÜLIKOOL. ARVUTITEADUSE INSTITUUT

# Programmeerime endale Lahendaja

## Programmeerimisülesannete kogu

**Tekst ja ülesanded: Rein Prank . Näidislahendused: Tiiu Kaljuste, Rein Prank**

**10.01.2020**

# Sisukord

# 1 Sissejuhatus

## 1.1 Miks, kellele

Käesolev ülesannete kogu on mõeldud õpilastele, kes on omandanud programmeerimise algkursuse materjali mingis programmeerimiskeeles ja vajavad ülesandeid saadud oskuste rakendamiseks. Selle ülesannete kogu ülesanded on koostatud põhikooli matemaatika ülesannete teemadel.

Kogu maailmas ja ka Eestis räägitakse juba pikemat aega infotehnoloogia spetsialistide puudusest. Meie infotehnoloogia- ja programmeerimisfirmade juhid väidavad, et nad oleksid suutelised rakendama rohkem töötajaid, eksportima rohkem produkti /teenuseid, maksma töötajatele kõrget palka ja riigile makse. Ülikoolid püüavad IT erialadele vastu võtta rohkem tudengeid ja koolitada neist haritud IT-spetsialiste. Et see võimalik oleks, on õpilastele vaja infotehnoloogia ja sealhulgas programmeerimisega tutvumise võimalusi juba eelmistel kooliastmetel. Räägitakse programmeerimise õpetamise laiendamisest gümnaasiumis, aga ka programmeerimisega tutvumisest juba põhikoolis. See vajab väljendamist õppekavades, õpetajate koolitamist ja vajalike õppematerjalide loomist.

Programmeerimise õppimist alustatakse tavaliselt tehnilist laadi sissejuhatava programmeerimis-kursusega. Sissejuhatavas kursuses tutvutakse tavaliselt programmeerimiskeelte lausetega (omistamine, lugemine ja kirjutamine, tingimuslaused, tsüklid, rekursioon, funktsioonid), andmetüüpide ja -struktuuridega (arvud, sõned, järjendid, listid, failid jne), sisendi ja väljundi kasutamisega, graafika vahenditega. Kasutatakse ka mingit vaadeldava programmeerimiskeele jaoks mõeldud programmeerimiskeskkonda. Käesoleva ülesannete kogu sihtrühmana on mõeldud õpilasi, kes on mingis vormis sellise algkursuse materjali omandanud (valikkursusena või ka kõigile õpilastele õpetatava materjalina põhikoolis või gümnaasiumis, MOOC-i vormis, mõne programmeerija abi kasutades või päris iseõppijana). Kui on tegemist õpilasega, kelles on tekkinud huvi programmeerimise vastu ja ta mõtleb võimalusest vastavale erialale õppima minna või mingis muus valdkonnas arvutite rakendamisega tegelda, siis ta tahaks programmeerimisega tegelemist peale algse tehnika omandamist jätkata. Nii koolides toimuvaks kui ka iseseisvaks tööks on nüüd eelkõige vaja mitmesuguse suunitlusega ülesannete kogusid, et asjast huvitatud õpilased saaksid edasi minna, saades tuge õpetajalt ja veebist.

## 1.2 Kes/mis on Lahendaja?

Matemaatika ülesannete lahendamise harjutamiseks on programmeeritud suur hulk õpilastele või üliõpilastele mõeldud ülesannete lahendamise keskkondi. Enamik nendest tegeleb avaldiste teisendamise ülesannetega. Mõned nendest (näiteks MathXpert[9], Aplusix [8], T-algebra [10]) on üsna universaalsed ja võimaldavad väga paljude erinevate ülesandetüüpide lahendamist. Paljud tegelevad aga ainult mingi kitsama valdkonna ülesannetega nagu tehted murdudega, ühikute teisendamine, lineaarvõrrandi lahendamine, lineaarsete võrrandisüsteemide lahendamine või funktsioonide diferentseerimine. Lahendamiskeskkond peaks sisaldama avaldiste toimetit sammu/vastuse sisestamiseks ja oskama anda tagasisidet selle õigsuse kohta (samaväärsus eelmise reaga, sammu vastavus õpitavale lahendusalgoritmile, vastusele esitatavate tingimuste täidetus).

Kui keskkond nõuab õppurilt ainult ülesande lõppvastuse sisestamist, siis ei tarvitse ta ise olla võimeline ülesandeid lahendama. Ülesannete koostaja võib koos ülesandega anda ette ka õige(d) vastuse(d) ja programmi poolt piisab vähem või rohkem intelligentsest õpilase ja õpetaja vastuse võrdlemisest. Palju täiuslikumat tagasisidet ja lahendussammude tegemiseks vajalikke nõuandeid andva keskkonna saame aga kokku panna siis, kui meie käsutuses on Lahendaja - programmimoodul, mis suudab õppuritele antavaid ülesandeid ise (eelistatavalt sammukaupa) lahendada. Niisiis kujutab selle ülesannete kogu ülesannete lahendamine endast Lahendaja osade programmeerimist põhikooli matemaatika õpikute eri valdkondade ülesannete jaoks. Muidugi sobivad paljude ülesandetüüpide jaoks Lahendajaks arvutialgebra ja/või dünaamilise geomeetria süsteemid (Wiris, Geogebra jne) ning meie programmeerimisülesanded tähendavad ka nende süsteemide väikeste komponentide programmeerimist.

## 1.3 Ülesannete kogu suunitlus

Käesolevas ülesannete kogus olevate ülesannete teemad on saadud põhikooli matemaatika õpikute ülesannetest. Niisugusel teemavalikul programmeerimisest huvitatud õpilaste jaoks on mitu põhjust:

1) Põhikooli matemaatikal on olemas tõsine algoritmiline sisu, mida tulevased programmeerijad peaksid tundma. Suur osa algoritmilisest sisust on õpikutes olemas ilmutatud kujul mitmesuguste algebra ülesannete lahendamise algoritmidena: tehted mitmekohaliste arvudega, naturaalarvude algtegurite ja ühistegurite leidmine, tehted murdudega, lineaarvõrrandi lahendamine, võrrandisüsteemide lahendamise meetodid, tehted üksliikmete ja hulkliikmetega. Põhikoolis õpetatakse ka geomeetria algoritme: lõigu ja nurga poolitamine, ristsirge ja paralleelsirge konstrueerimine jms. Algoritmilise sisu peidetud osa moodustavad põhikooli õpikutes ülesanded, mida esitatakse ainult konkreetsete näidete kujul (kuidas mõõta 3-liitrise ja 5-liitrise anuma abil 7 liitrit vett?), milles nõutakse lahendust suuruste antud väärtuste jaoks, aga ei räägita vaadeldavat tüüpi ülesande üldisest lahendusalgoritmist. Algoritmidest, mis niisuguseid ülesandeid üldkujul lahendavad, saame rääkida alles koos programmeerimisega, sest vähegi mahukamate näidete korral on lahendamisel tekkiv arvutuste maht kirjaliku lahendamise jaoks liiga suur.
2) Arvuteooria ja ka geomeetria olulised algoritmid tuuakse matemaatika kursuses sisse väga vara (V-VI klassis) ja nad jäävad siis tihti meelde mingi lubatud operatsioonide hulgana, mida võib rakendada suvalises järjekorras. Selle materjali ühendamiseks programmeerimises vaadeldavate algoritmidega on kasulik need teemad vahepeal lisandunud hariduse baasil üle vaadata ja olulisemad algoritmid ka läbi programmeerida. Teiselt poolt võib aga öelda, et teemad 3.5.2 ja 3.5.3 moodustavad selle ülesannete kogu kõige sisukama osa.
3) Programmi kirjutamiseks on vaja detailsemat algoritmide kirjeldust, kui õpikud inimkasutaja jaoks esitavad. Et programm toimiks üheselt, peab programmeerija osa detaile täiendavalt fikseerima (kumma teguri kümnendkohad korrutada teise teguriga, milline tundmatu on asendusvõtet kasutades mõistlik esimesena avaldada, millal teha algoritmi kirjelduses mitteesitatud avaldiste lihtsustamise samme jne). Detailide programmeerimine sunnib õpilast läbi mõtlema definitsioonide detailid ja võimalikud lahendustes tekkivaid olukorrad, avaldiste kirjutamise konventsioonid jms.

Ülesanded on ülesannete kogus toodud niisuguses järjestuses, nagu vastavad teemad õpikutes esimest korda sisse tuuakse. Paljudel juhtudel on mingi klassi õpikust pärinevale ülesandele lisatud keerukamaid variante, mis on pärit järgmiste klasside materjalist. Konkreetsete arvuliste väärtustega ülesandeid on üldistatud suvaliste arvuliste algandmete puhul töötava programmi koostamise ülesanneteks.

## 1.4 Kuidas ülesannete kogu kasutada

Ülesannete kogus on nii väga lihtsaid kui ka olümpiaadiprogrammeerimise tasemega ülesandeid. See võimaldab ülesannete kogu ka mitmel tasemel nii õpetajate kui õpilaste poolt kasutada:

1) Ülesannete leidmine valikainetes ja programmeerimise ringides lahendamiseks põhikoolist kuni täppisteadustele orienteeritud gümnaasiumiteni.
2) Ülesannete leidmine iseseisvaks programmeerimiseks pärast esialgsete tehniliste oskuste omandamist.
3) Informaatika olümpiaadideks valmistumise esimene tase, eriti põhikooli õpilaste jaoks.
4) Ideede leidmine koolides toimuvate programmeerimise olümpiaadide ülesannete jaoks.

Kindlasti ei ole see ülesannete kogu mõeldud kaanest kaaneni järjestikku lahendamiseks. Ülesannete kogu põhimaterjalina kasutav ringijuht või üksiklahendaja peaks lahendatavad ülesanded valima vastavalt klassi/enda algtasemele ja jõukohasele liikumiskiirusele. Sobiv võib olla liikumine mingi teemaga läbi erinevate klasside materjali. Ülesannete valiku juures võib abistada tabel, kus on antud hinnangud ülesande üldise raskuse ja kolme lahendamisel sagedamini vajaliku komponendi raskuse kohta: avaldiste süntaksi töötlemine, arvutustes vajalik matemaatika (valemid, õpikualgoritmid jms) ja lahendusalgoritmi väljatöötamine. Süntaksi juures tähendab tase 1 arvude, tehtemärkide jms eraldamist fikseeritud süntaksiga avaldises, tase 2 lineaarse ehitusega avaldise (näiteks suvalise liikmete arvuga summa) mõistmist, tase 3 suvalise struktuuriga avaldise analüüsi (erineva prioriteediga tehted, sulud jms). Teiste komponentide raskuse hinnangud on rohkem subjektiivsed.

Olümpiaadideks valmistumisel pakub käesolev ülesannete kogu algajate taset. Informaatika olümpiaadidel on meil põhikooli võistlusklass küll olemas, aga seal on seni üsna vähe osavõtjaid. Peamine põhjus on kindlasti programmeerimisõpetuse puudumine koolis ja vähene õpetajatelt saadav toetus. Iseseisva töö teeb raskeks ka sobivate ülesannete puudumine. Kui õpilane suudab juba toimuvate olümpiaadide ülesandeid lahendada, võib ta leida palju allikaid edasiseks sobivate ülesannetega. Neid ei ole püütud selles ülesannete kogus veel kord pakkuda. Aga matemaatikat ja programmeerimist siduvate ülesannete jaoks vajalik elementaarne tehnika on püütud samm-sammult läbi võtta.

Mõnedes keeltes on standardfunktsioone (näiteks eval ja sort Pythonis), mis võimaldavad mingeid selle ülesannete kogu ülesandeid enam-vähem ühe pöördumise teel lahendada. On olemas ka teeke, kus on realiseeritud algoritmid paljude tihti lahendatavate matemaatika ülesannete jaoks (vt näiteks [4]). Mõnede ülesannete õige vastuse võib kätte saada ka oma programmist mõne arvutialgebra süsteemi poole pöördudes. Õppimise jaoks tuleks selle ülesannete kogu ülesanded lahendada programmeerimiskeelte madala taseme funktsioone kasutades, ilma spetsiaalseid matemaatika vahendeid rakendamata. Küll aga võiks oma programmi testimisel ülesannete vastuseid võrrelda mingil muul usaldataval viisil saadud vastustega.

Kui ülesanne on lahendatud, siis soovitame uurida ka selle või samateemaliste ülesannete näidislahendusi. Sealt võib leida alternatiivseid programmeerimiskeele vahendeid ja algoritmilisi võtteid antud valdkonna ülesannete lahendamiseks.

## 1.5 Millest koosneb ülesanne

Selles ülesannete kogus võib ülesanne sisaldada mitmeid komponente:

1. Teoreetiline sissejuhatus. Mõistete definitsioonid, kasulikud valemid, teoreemid jms.

2. Ülesande lühisõnastus: antud-leida.
3. Detailsem sõnastus: lahenduses lubatud sammude kirjeldused, sisendi ja väljundi formaat, sisendandmete võimalikud muutumispiirid ja väljundandmete vormistuselt nõutavad tingimused.
4. Sisendi ja talle vastava väljundi näited.
5. Soovitused lahendusalgoritmi ja/või algoritmi parameetrite valikuks.
6. Soovitused lahenduse testimiseks/mõned esialgsed testid peamiste juhtude jaoks.
7. Sisendandmete komplektid põhjalikuks testimiseks.
8. Näidislahendus (programm).

Teoreetiline sissejuhatus on tihti ühine mitmest ülesandest koosnevale teemale. Lihtsamad või mõnede eelnevatega analoogilised ülesanded võivad ülesannete kogus koosneda ainult lühisõnastusest. Sisendandmete täpsema formaadi ja väljundi esituse peab siis programmi autor ise enne programmeerimist fikseerima. Mõnede ülesannete sõnastus sisaldab mitut erinevate sisend- ja/või väljundtingimustega määratud varianti. Need variandid on tähistatud väikeste ladina tähtedega: a), b), … . Lahendusalgoritmi soovitused on tekstis paigutatud ülesannetest eraldi.

Soovitame programmid koostada nii, et sisendandmed loetakse tekstifailist ja vastus salvestatakse tekstifaili. See annab võimaluse pärast ootamatu vastuse saamist kontrollida, kas sisendandmed vastasid kavatsusele. Saame ka lahendada samu teste pärast programmi korrigeerimist uuesti, ilma et teste peaks uuesti sisestama. Samuti saab keerulisemaid sisendandmed (keerukamad avaldised, arvude massiivid jms) tekstifaili sisestada käsureast mugavamat toimetamist võimaldavas keskkonnas ja neid enne kasutamist üle vaadata. Failidega suhtlev programm võimaldab ka testimist testimisserveris, nagu seda tehakse mitmetes programmeerimiskursustes ja informaatika olümpiaadidel.

Soovitusi testimiseks ei ole antud kõigi ülesannete jaoks. Kui neid ei ole, siis on nad tihti olemas juba mõne eelneva sarnase ülesande juures.

Suur osa ülesandeid on sõnastatud nii, et sisendandmetelt võib eeldada teatud matemaatilistele tingimustele vastamist (on võimalik antud arvuga taandada, leidub parajasti üks lahend vms). Nendele tingimustele vastamist ei pea programm kontrollima, kui seda pole nõutud ülesande sõnastuses. Selles ülesannete kogus on palju lihtsaid ülesandeid, kus sisendtingimuste kontroll võib olla isegi keerulisem, kui on ülesande enda lahendamine. Paljudel juhtudel aga on kõige otsem tee andmete sobilikkuse kontrolliks põhijuhu läbilahendamine. Lahendusalgoritmi programmeerimisel võib aga nendest eeldustest ka loobuda ja koostada programmi nii, et ta kontrollib tarvilikke tingimusi ja annab vajadusel veateate sisendandmete kohta.

## 1.6 Ülesannete raskused

Ülesanded on üsna erineva raskusega: elementaarsetest kuni olümpiaadideks sobivate ülesanneteni. Sealjuures on iga teema sees nii kergeid kui ka raskemaid ülesandeid. Esitame raskuse hinnangud tabelina, milles on neli veergu. Esimeses veerus on ülesande summaarne raskus. Järgmised kolm veergu sisaldavad hinnanguid kolmele komponendile, mis on mängus paljudes (aga mitte kõigis) ülesannetes:  avaldise süntaksi töötlemine, matemaatiline keerukus, algoritmi programmeerimise keerukus. Tühi lahter tähendab, et antud ülesande lahendamiseks ei ole vastav komponent vajalik (näiteks ei töötata avaldistega) või on triviaalne.

Kui uurida informaatika olümpiaadide ülesandeid, siis neist suurema osa raskuspunkt on just (piisava kiirusega töötava) algoritmi koostamine. Aga piisava kiirusega töötava algoritmi koostamine on just see materjal, mille jaoks maailmas on olemas palju olümpiaadidest osavõtjatele mõeldud allikaid, sealhulgas ka eestikeelne kaheosaline võistlusprogrammeerimise õpik [4,5]. Käesoleva ülesannete kogu ülesannetes arvutamise kiirust eriti ei rõhutata. Piisab õige vastuse saamisest. Spetsiaalsete graafidel põhinevate lahendusalgoritmide koostamine piirdub ainult nn laiuti otsinguga (vt ülesanded jaotise 3.1 lõpus).

| **Ptk** | **Yl. arv** | **Yles nr** | **Ülesande teema** | **Ras kus** | **Avald sünt** | **Matem** | **Progr** |
|---|---|---|---|---|---|---|---|
| **Kokku** | **189** | | | | | | |
| **I klass** | **18** | **1** | Arvude võrdlemine | 1 | | 1 | |
| | | **2** | Liitmistehe | 1 | 1 | 1 | |
| | | **3** | a+b=c liikme leidmine | 1 | 1 | 1 | |
| | | **4** | Lahutamistehe | 1 | 1 | 1 | |
| | | **5** | a-b=c liikme leidmine | 1 | 1 | 1 | |
| | | **6** | Võrdlusmärgi valimine | 1 | 1 | 1 | |
| | | **7** | Tehtemärgi panemine | 1 | 1 | 1 | 1 |
| | | **8** | Tehte- ja võrdlusmärgi valimine | 2 | 1 | 2 | |
| | | **9** | Lõikude summa pikkus (ühikutega) | 2 | 2 | 1 | |
| | | **10** | Summa arvutusrahadega | 4 | | | 4 |
| | | **11** | Liitmistabel | 1 | | | 1 |
| | | **12** | Väljendada antud ühikutes | 1 | | 1 | 1 |
| | | **13** | Nädalapäevade tehted | 1 | 1 | 1 | |
| | | **14** | Vee mõõtmine anumatega | 4 | | 2 | 4 |
| | | **15** | Kaalumine kangkaaludega | 4 | | 3 | 4 |
| | | **16** | Järeldamine kahest võrratusest | 3 | 1 | 3 | 3 |
| | | **17** | Võrratuste tabel | 4 | 1 | 3 | 4 |
| | | **18** | Kivimite jagamine pooleks | 3 | | | 3 |
| **II klass** | **4** | **1** | Summa maksmine müntidega | 4 | | | 4 |
| | | **2** | Järjestamine pikkuse järgi | 2 | | 2 | 2 |
| | | **3** | Kellaaeg+minutid | 2 | | 2 | |
| | | **4** | Võrdsete summa väljendamine korrutisena | 2 | 2 | 2 | |
| **III klass** | **10** | **1** | Järjestamine | 2 | | | 2 |
| | | **2** | Lõikude summa ja vahe | 1 | | 1 | |
| | | **3** | Ostu eest tasumise viisid | 3 | | | 3 |
| | | **4** | Arvud ruutudesse | 1 | | | 1 |
| | | **5** | Võluruudu kontroll | 1 | | | 1 |
| | | **6** | Võluruudu ehitamine | 4 | | | 4 |
| | | **7** | Ühikute teisendamine | 1 | | | 1 |
| | | **8** | Korrutustabel | 1 | | | 1 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | **9** | Jäägiga jagamine | 1 | | 1 | |
| | | **10** | Homne kuupäev | 2 | | | 2 |
| **IV klass** | **6** | **1** | Numbrite leidumine arvudes | 1 | | | 1 |
| | | **2** | Suuruselt teine arv | 2 | | | 2 |
| | | **3** | Tingimustele vastavad arvud | 2 | | 2 | |
| | | **4** | Kuupidest kujundi vaade | 2 | | 2 | |
| | | **5** | Tehted ajavahemikega | 1 | | 1 | |
| | | **6** | Nuppudega mxn mäng | 4 | | | 4 |
| **V-1 Liitmine** | **21** | **1** | Arv järkarvude summana | 1 | | 1 | |
| | | **2** | Naturaalarv sõnadega | 2 | | 2 | 2 |
| | | **3** | Kirjalik liitmine | 2 | | | 2 |
| | | **4** | Aditiivse avaldise väärtus | 2 | 2 | | |
| | | **5** | Aditiivne sulgudega avaldis | 3 | 3 | | 3 |
| | | **6** | Ristküliku pindala ja ümbermõõt | 1 | | 1 | |
| | | **7** | Ümardamine | 1 | | 1 | |
| | | **8** | Ühikuga suurus antud ühikuks | 1 | | 1 | |
| | | **9** | Araabia ja rooma numbrid | 2 | | 2 | 2 |
| | | **10** | Järjestamine 2-dim | 3 | | | 3 |
| | | **11** | Muutuja asendam väärtusega | 2 | 2 | | |
| | | **12** | Muutujatega avaldise väärtus | 2 | 2 | | |
| | | **13** | Lin funkts väärtuste tabel | 1 | | 1 | |
| | | **14** | Üldise funkts väärtuste tabel | 3 | 3 | | |
| | | **15** | Tehete järjekorra lubatavus | 3 | 3 | | 3 |
| | | **16** | Aditiivsete arvavaldiste võrdlemine | 2 | 2 | | |
| | | **17** | Aritm progr summa tsükliga | 1 | | | 1 |
| | | **18** | Võrrand x±a=b | 2 | | 2 | |
| | | **19** | Aditiivne võrrand | 2 | 2 | 2 | |
| | | **20** | Tunniplaan | 1 | | | 1 |
| | | **21** | Avaldis nulliks ± valiku abil | 2 | 2 | | 2 |
| **V-2 korrutamine** | **17** | **1** | Korrutise arvutamine | 2 | 2 | | |
| | | **2** | Jagamistehte tulemus | 1 | 1 | | |
| | | **3** | Kirjalik korrutamine | 2 | | 2 | 2 |
| | | **4** | Kirjalik jagamine | 3 | | 2 | 3 |
| | | **5** | Korrutamise ja jagamisega arvavaldise väärtus | 3 | 3 | 2 | 2 |
| | | **6** | Nelja tehtega arvavaldise väärtus | 4 | 4 | 2 | 3 |
| | | **7** | Võrrand ax=b | 1 | | | 1 |
| | | **8** | Koondamine lin hulkliikmes | 2 | 2 | | 2 |
| | | **9** | Lineaarse hulkliikme korrutamine arvuga | 2 | 2 | | |
| | | **10** | Tärnide asendamine | 3 | | 3 | 3 |
| | | **11** | Tehtemärgid arvude vahele | 3 | | | 3 |
| | | **12** | Tegurite leidmine | 2 | | 2 | 2 |
| | | **13** | Algarvulisuse kontroll | 2 | | 2 | 2 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | **14** | Algteguriteks lahutus | 2 | | 2 | 2 |
| | | **15** | Eratosthenese sõel | 2 | | 2 | 2 |
| | | **16** | Eukleidese algoritm | 2 | | 2 | 2 |
| | | **17** | SÜT, VÜK | 2 | | 2 | 2 |
| **V-3 Konstrukts** | **23** | **1** | Lõik otspunktide järgi | 1 | | 1 | |
| | | **2** | Kiir punktide järgi | 1 | | 1 | |
| | | **3** | Sirge punktide järgi | 1 | | 1 | |
| | | **4** | Nurk suuruse järgi | 1 | | 1 | |
| | | **5** | Ringjoone joonestamine | 1 | | 1 | |
| | | **6** | Kahe sirge lõikepunkt | 2 | | 2 | |
| | | **7** | Sirge ja ringjoone lõikepunkt | 3 | | 3 | 3 |
| | | **8** | Antud pikkusega lõik kiirele | 2 | | 2 | |
| | | **9** | Kahe ringjoone lõikepunkt | 4 | | 4 | 3 |
| | | **10** | Ristsirge joonestamine | 2 | | 2 | |
| | | **11** | Paralleelsirge joonestamine | 3 | | 3 | |
| | | **12** | Nurga teisaldamine | 3 | | 3 | 2 |
| | | **13** | Kolmnurk külgede järgi | 2 | | 2 | 2 |
| | | **14** | Kolmnurk 2 külje ja nurga j. | 2 | | 2 | 2 |
| | | **15** | Kolmnurk külje ja 2 nurga j. | 2 | | 2 | 2 |
| | | **16** | Ruudu joonestamine | 1 | | 1 | |
| | | **17** | Ristküliku joonestamine | 1 | | 1 | |
| | | **18** | Rombi joonestamine | 2 | | 2 | |
| | | **19** | Lõigu poolitamine | 3 | | 3 | 3 |
| | | **20** | Nurga poolitamine | 3 | | 3 | 3 |
| | | **21** | Ringi puutuja joonestamine | 3 | | 3 | |
| | | **22** | Kolmnurga ümberringjoon | 3 | | 3 | 3 |
| | | **23** | Kolmnurga siseringjoon | 3 | | 3 | 3 |
| **V-4 Kellaosutid** | **4** | **1** | Minutiosuti liikumine ja aeg | 2 | | 2 | |
| | | **2** | Tunniosuti liikumine ja aeg | 2 | | 2 | |
| | | **3** | Nurk osutite vahel | 2 | | 2 | |
| | | **4** | Aeg osutite asendi järgi | 2 | | 2 | |
| **V-5 Statistika** | **5** | **1** | Aritmeetiline keskmine | 1 | | 1 | |
| | | **2** | Osahulkade aritm keskmised | 2 | | 1 | 2 |
| | | **3** | Aritmeetiline keskmine sagedustabeli järgi | 1 | | 1 | |
| | | **4** | Moodi arvutamine | 1 | | 1 | |
| | | **5** | Mediaani arvutamine | 1 | | 1 | |
| **VI-1 Murrud** | **29** | **1** | Lihtmurrud/liigmurrud | 1 | | 1 | |
| | | **2** | Laiendamine antud laiendajaga | 1 | | 1 | |
| | | **3** | Laiendamine antud nimetajani | 1 | | 1 | |
| | | **4** | Taandamine antud arvuga | 1 | | 1 | |
| | | **5** | Taandamine SÜT-ga | 2 | | 2 | 2 |
| | | **6** | Osamäär meetrist, … | 1 | | 1 | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | **7** | Tundmatu leidmine võrdest | 1 | | 1 | |
| | | **8** | Ühenimeliseks teisendamine | 2 | | 2 | |
| | | **9** | Murdude võrduse kontroll | 1 | | 1 | |
| | | **10** | Murdude võrdlemine | 1 | | 1 | |
| | | **11** | Murd segaarvuks | 1 | | 1 | |
| | | **12** | Segaarv liigmurruks | 1 | | 1 | |
| | | **13** | Ühenimeliste murdude ± | 2 | | 2 | |
| | | **14** | Ühenimeliste segaarvude liitmine | 2 | | 2 | |
| | | **15** | Ühenimeliste segaarvude lahutamine | 2 | | 2 | |
| | | **16** | Erinimel murdude ± | 3 | 2 | 3 | 2 |
| | | **17** | Murdudega aditiivne võrrand | 3 | 2 | 3 | 2 |
| | | **18** | 10-ndmurd harilikuks | 2 | | 2 | |
| | | **19** | Harilik murd 10-ndmurruks | 3 | | 3 | 3 |
| | | **20** | Hariliku murru 10-ndlähend | 2 | | 2 | |
| | | **21** | Harilik ± kümnendmurd | 2 | | 2 | |
| | | **22** | *-ülesanded | 2 | | 2 | 2 |
| | | **23** | Harilike murdude korrutamine | 2 | | 2 | |
| | | **24** | Korrutamine, ka segaarvud | 2 | | 2 | |
| | | **25** | Osa leidmine arvust | 1 | | 1 | |
| | | **26** | Pöördarvu leidmine | 2 | | 2 | |
| | | **27** | Harilike murdude jagamine | 2 | | 2 | |
| | | **28** | Arv osa ja osamäära järgi | 2 | | 2 | |
| | | **29** | Murdudega avaldise väärtus | 2 või 3 | 2 või 3 | 2 | |
| **VI-1 Kujundid** | **17** | **1** | Lükke parameetrite leidmine | 2 | | 2 | |
| | | **2** | Pöörde nurga leidmine | 2 | | 2 | |
| | | **3** | Peegelduse leidmine | 2 | | 2 | |
| | | **4** | Telgsümmeetrilisuse kontroll | 2 | | 2 | |
| | | **5** | $\pi$ alumine ja ülemine hinnang | 2 | | 2 | 2 |
| | | **6** | Tingimused kolmnurga külgedele ja nurkadele | 1 | | 1 | |
| | | **7** | Kolmnurga määramine 3 elemendiga | 4 | | 4 | 2 |
| | | **8** | Kuubi pinnalaotus? | 4 | | 4 | 4 |
| | | **9** | Küljed -kolmnurga liik (nurgad) | 1 | | 1 | |
| | | **10** | Nurgad -kolmnurga liik (küljed) | 1 | | 1 | |
| | | **11** | Nurgad - kolmnurga liik (nurgad) | 1 | | 1 | |
| | | **12** | 3 punkti - kolmnurk? | 1 | | 1 | |
| | | **13** | 4 punkti - nelinurga tüüp | 2 | | 2 | |
| | | **14** | Punktid - ruutude arv | 3 | | 3 | 3 |
| | | **15** | Punktid - ristkülikute arv | 3 | | 3 | 3 |
| | | **16** | Punktid - rombide arv | 3 | | 3 | 3 |
| | | **17** | Punktid - rööpkülikute arv | 3 | | 3 | 3 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **VII Lin. võrrandid** | **19** | **1** | Võrrand ax=b | 1 | 1 | 1 | |
| | | **2** | Tundm vas-le, arvud par-le | 2 | 2 | 1 | |
| | | **3** | Sarnaste liikmete koondam | 2 | 2 | 1 | |
| | | **4** | Murdudest vabanemine lin-v | 3 | 3 | 2 | 3 |
| | | **5** | Sulgude avamine | 2 | 2 | 1 | |
| | | **6** | Lin-v standardalgoritm | 4 | 3 | 3 | 4 |
| | | **7** | Samasus ja vastuolul võrrand | 3 | 3 | 3 | 3 |
| | | **8** | Abs-väärtusega lin-võrrand | 4 | 3 | 3 | 4 |
| | | **9** | Lin võrratuse tõeväärtus | 2 | 2 | 2 | |
| | | **10** | Teheted lin võrratusega | 2 | 2 | 2 | |
| | | **11** | Lin võrratuse lahendamine | 4 | 3 | 3 | 4 |
| | | **12** | Abs-väärtusega lin-võrratuse lah | 4 | 3 | 3 | 4 |
| | | **13** | Lin-2 süsteem liitmisvõttega | 3 | 2 | 3 | 3 |
| | | **14** | Lin-2 süsteem asendusvõttega | 3 | 2 | 3 | 3 |
| | | **15** | Lin-2 süsteemi uurimine | 3 | 2 | 3 | 3 |
| | | **16** | Lin-3 süsteem liitmisvõttega | 3 | 2 | 3 | 3 |
| | | **17** | Lin-3 süsteem asendusvõttega | 3 | 2 | 3 | 3 |
| | | **18** | Lin-4 süsteem liitmisvõttega | 3 | 2 | 3 | 3 |
| | | **19** | Lin-4 süsteem asendusvõttega | 3 | 2 | 3 | 3 |
| **VIII-1 Yksliikmed** | **5** | **1** | Üksliikmete korrutamine | 1 | 2 | 2 | |
| | | **2** | Üksliikmete astendamine | 1 | | 1 | |
| | | **3** | Üksliikmete korrutamine, normaalkuju | 2 | 2 | 2 | |
| | | **4** | Üksliikmete jagamine | 2 | 2 | 2 | |
| | | **5** | Vabanemine negatiivsetest astmetest | 2 | 2 | 2 | |
| **VIII-2 Hulkliikmed** | **10** | **1** | Sarnaste liikmete koondamine | 3 | 2 | 2 | 3 |
| | | **2** | Hulkliikmete ± | 3 | 2 | 2 | 3 |
| | | **3** | Hulkl korrutamine üksl-ga | 2 | 2 | 2 | 2 |
| | | **4** | Hulkl jagamine üksl-ga | 2 | 2 | 2 | 2 |
| | | **5** | Teguri toomine sulgude ette | 2 | 2 | 2 | 2 |
| | | **6** | Kaksliikmete korrutamine | 2 | 2 | 2 | 2 |
| | | **7** | Hulkliikmete korrutamine | 3 | 2 | 2 | 3 |
| | | **8** | Rühmitamisvõte tegurdamiseks | 3 | 2 | 2 | 3 |
| | | **9** | Abivalemite kasutamine | 3 | 3 | 3 | 3 |
| | | **10** | 1 muutuja hulkliikmete jagamine | 4 | 3 | 3 | 4 |

## 1.7 Näidislahendused

Ülesannete koguga on kaasas ka mitmete erinevaid tehnikaid kasutavate ülesannete näidislahendused keeles Python. Vastava ülesande juures on siis antud ka näidislahenduse faili nimi.

Näidislahendustes on kasutatud (mõnikord ühe ja sama töö tegemiseks) erinevaid Pythoni vahendeid, et näidata erinevaid võimalusi. Enamus näidislahendusi on pärit T.Kaljuste bakalaureusetööst.

Näidislahendused loevad sisendandmeid tekstifailist. Iga näidislahendusega on kaasas üks sisendfaili formaati demonstreeriv sisendfaili näide. Oma programmi testimiseks tuleb muidugi koostada hulk erinevate nimedega sisendfaile ja nende kasutamiseks modifitseerida programmi tekstis sisendfaili nime. Näidisprogrammid väljastavad töö tulemused ekraanile, lisades töö paremaks jälgimiseks tihti ka vahetulemusi, mida ülesande tekst ei tarvitse nõuda.

Õppimise eesmärgil soovitame ülesandeid lahendada esialgu ilma näidislahendusi vaatamata ja pöörduda nende poole alles siis, kui tekib vajadus mõne konkreetse tehnilise probleemi lahenduse leidmiseks. Ülesannete koguga töötades on kavas lisada uusi näidislahendusi.

# 2 Testimisest

Paljudel selle ülesannete kogu kasutajatel tuleb ise otsustada oma programmide korrektsuse üle. Sellepärast räägime enne ülesannete juurde asumist natuke ka programmide testimisest. Testide koostamise ja testimise kirjaoskus on vajalikud selleks, et leida oma programmist vigu ja need parandada. Olümpiaadidest osavõtjatel on vaja ka teada, kuidas koostatakse olümpiaadidel lahenduste hindamise testid. Seal muidugi testitakse paljude ülesannete juures peale vastuse õigsuse ka programmi töökiirust, andes ette erineva mahuga algandmeid. Meie siin programmi töökiirust eesmärgiks ei sea ega testi.

Peatükis 3 on enamiku ülesannete juures ka soovitused põhijuhtude testimiseks. Paljude ülesannete jaoks on ülesannete kogule lisatud ka fail, mis sisaldab musta kasti meetoditega (vt 2.1.3) koostatud põhjalikumat testide komplekti. Oma programmi tõsisemaks testimiseks tuleks sellele lisada veel valge kasti teste (vt 2.1.4), mis arvestavad seda, milliseid meetodeid on programmis kasutatud.

## 2.1 Testimise teooriast

Erialakirjanduses defineeritakse testimist järgmiselt:
*Testimine on programmi täitmise protsess eesmärgiga avastada vigu.*

Niimoodi defineeritud testimine on destruktiivne tegevus ja võib programmeerijale olla üsna vastumeelne, kui testida tuleb enda kirjutatud programmi. Niisugune eesmärk seatakse aga selleks, et testija asuks tehtava töö suhtes õigele positsioonile. Kui tahame leida vigu, siis ka leiame neid rohkem. Kui soovime ennast veenda, et meie programm töötab õigesti, siis võib see veendumus tekkida ka sellel põhjusel, et testimine pole olnud piisavalt põhjalik. On ka selliseid programmeerimisfirmasid, kus „huvide konflikti“ vältimiseks autor ise oma programme ei testi või koguni on olemas spetsiaalne testijate üksus.

### 2.1.1 Testide koostamise meetodid

Testide koostamise meetodid on suhteliselt formaalsed ja nende järgimine võimaldab vähendada riske, mis tekivad enda kirjutatud programmi testimisest. Alltoodud kaks meetodite klassi erinevad selle poolest, millist informatsiooni kasutatakse testide koostamisel.

1. **Musta kasti meetodid.** Testid koostatakse lahendatava ülesande põhjal, programmi vaadatakse kui musta kasti. Väga lihtsate programmide/moodulite korral on võimalik ammendav testimine

(näiteks menüüs olevate kõigi valikute toimimise läbiproovimine). Üldiselt jagatakse võimalikud sisendandmed sellisel viisil nn. ekvivalentsiklassidesse, et vähegi mõistlikult kirjutatud programm peaks samasse klassi kuuluvate sisendandmete korral töötama ühesugusel viisil.

2. **Valge kasti meetodid**. Teste koostatakse programmi sisemist ehitust (kasutusel olevad arvude, sõnede jms andmetüübid, massiivide suurused, hargnemised, tüüpiliste alamülesannete lahendamise meetodid jms) arvestades.

Tõsisema testimise jaoks koostatakse kõigepealt algne hulk teste musta kasti meetoditega ja lisatakse siis täiendavalt teste valge kasti meetodite abil. Mõnedel juhtudel on ainult musta kasti meetoditega piirdumine enam-vähem paratamatu. Äärmine juht on siin see, kui kasutatakse automaatset testimist, kus on eelnevalt koostatud teatud hulk sisendandmete komplekte ja testimist juhtiv programm käivitab testitava programmi iga komplektiga ning kontrollib väljundandmete vastavust nõuetele. Automaatset testimist saab rakendada selliste ülesannete korral, kus programmi töö tulemuseks on mingite muutujate väärtused vms, mida testiv programm „oskab“ kontrollida. Kui kontrollitakse suure hulga õpilaste või tudengite kodutöid, siis tuleb kontrollijal samuti põhiliselt jääda eelnevalt koostatud testide juurde, sest tal pole piisavalt aega igasse programmi süvenemiseks. Muidugi võib selline testimine osutuda ebaadekvaatseks, kui õppuri programm ei ole koostatud meetoditega, mida testide koostamisel eeldati. Programmeerimise olümpiaadidel testitakse tavaliselt kõiki lahendusi varem fikseeritud testide hulga peal. Igale testile on omistatud punktide arv, mida selle testi läbimine annab.

### 2.1.2 Testimise printsiibid

Testide koostamisel ja testimisel on kasulik arvesse võtta järgmisi põhimõtteid, mis kuuluvad tarkvara tootmise klassikasse [2]:

1. Testi kohustuslik osa on oodatav õige vastus. Kui meil enne testi käivitamist oodatavat vastust valmis ei ole ja püüame programmi poolt arvutatud vastuse õigsust tagantjärele „silma järgi“ hinnata, siis loeme tihti ka vigase vastuse õigeks.
2. Iga testi tulemusi tuleb põhjalikult uurida. See käib eriti selliste ülesannete kohta, kus tulemus on keerulisema struktuuriga (meie temaatika puhul avaldised, jadad, tabelid ja eriti geomeetria konstruktsioonid).
3. Testid tuleb kirjutada nii lubatud kui mittelubatud sisendandmete (meie teemade korral juhud, kus antakse negatiivne vastus) kohta.
4. Teste ei tohi ära visata enne, kui programm on lõplikult valmis. Kui oleme programmi muutnud, siis tuleb ka testid uuesti käivitada. See muidugi tähendab, et testid peavad olema salvestatud faili (programm võiks sisendandmeid sealt lugeda). Vastasel korral me peame samu andmeid korduvalt sisestama ja me unustame osa teste ka ära.
5. Mingis programmi osas veel leiduvate vigade arv kipub olema proportsionaalne seal juba leitud vigade arvuga, st programmi nõrgaks osutunud osi tuleb põhjalikumalt testida.
6. Kui testite oma programmi, siis on kasulik arvestada, et vaevaliselt tekkinud või uute meetodite õppimist nõudnud programmi osas on tõenäoliselt vigu rohkem.
7. Hea test võimaldab mitte ainult vigu avastada, vaid ka lokaliseerida ja parandada.

### 2.1.3 Musta kasti meetodid

#### Meetod 1: Ekvivalentsiklassid

Ülesande järgi saab tavaliselt leida teatud „oskused“, mis programmil peaksid olema. Näiteks ülesannet 3.1.9 lahendav programm peab oskama eraldada avaldises arve ja küsimärki, leida sõnena esitatud arvu väärtuse, vaadelda võrdusmärgi ja tehtemärkide võimalikke asendeid, teha otsust lahendi puudumise kohta.

Iga sellist oskust vaadeldes saab võimalikud sisendandmed jagada klassidesse nii, et samasse klassi kuuluvate andmete korral teeb programm vaadeldavat tööd tõenäoliselt (s.t. kui ta kasutab vähegi mõistlikku meetodit) samal viisil. Järelikult selle oskuse kontrolliks piisab anda igast sisendandmete klassist üks test.
**Näide:** Tehtemärgi panemine $\oplus$ asemele
**Antud:** sõne kujul $a \oplus b \lesseqgtr c$ kus $a, b$ ja $c$ on naturaalarvud ning $\lesseqgtr$ on võrdlusmärk ($<, \leq, =, \geq$ $või$ $>$).
**Leida:** sobiv tehtemärk $\oplus$ kohale (+ või -).
**Märkus.** Võib leiduda ka mitte ühtegi või mitu õiget vastust.
**Põhijuhtude testimine** (ideed ekvivalentsiklasside jaoks)**.** Vaadata igal kohal läbi nii ühe- kui mitmekohalised arvud, samuti 0. Kõik võimalikud võrdlusmärgid. Õige vastus: pole/-/+/mõlemad.
**Oskused**:
1. Arvude ja märkide asukoha eraldamine
2. Arvude väärtuse leidmine.
3. Võrdlusmärgi kehtivuse kontroll.
4. Sobiva tehtemärgi valik või mittelahenduvuse otsuse tegemine.

Esimese oskuse põhjal peame vaatlema klasse, kus on ühekohalised/mitmekohalised arvud igal positsioonil, samuti 0. Teise oskuse kontrolliks tuleb samuti vaadelda nii ühekohalisi/mitmekohalisi arve ja nulli. Kolmanda oskuse jaoks on vajalikud testid, kus on kõik võimalikud võrdlusmärgid. Neljanda oskuse kontrolliks on vajalikud juhud, kus sobiv märk: ei leidu/+/-/mõlemad.

Pärast klasside määramist tuleb:
1) Katta lubatud klassid minimaalse arvu testidega (üks test võib katta iga oskuse jaoks mingi klassi).
2) Koostada iga mittelubatud klassi jaoks oma test. Kui negatiivse otsuse jaoks on mitu võimalikku põhjust, siis negatiivne otsus esimesena kontrollitaval põhjusel tähendab tavaliselt, et programm järgmisi põhjusi enam ei kontrolli ja tagapool asuvad programmi osad jäävad testimata.

### Meetod 2: Piiride kontroll

Programm võib klasside piiril asuvad andmed paigutada valesse klassi ja arvutada tulemuse vale reegli järgi. Seda võib põhjustada näiteks:
1) eksimine range ja mitterange võrratusmärgi vahel,
2) ±1 tüüpi vead tsüklite piiridel,
3) arvutusvead reaalarvudega arvutamisel,
4) probleemid väga suurte ja väga väikeste arvudega.

Testimisvajadused:
1) Testid iga piiri jaoks täpselt piiril oleva väärtusega ja kummalgi pool piiri lähedal.
2) Kui on tegemist mingi jadaga, siis testida esimest ja viimast elementi,
3) Suured, väikesed ja ligikaudu piiril asuvad väärtused.

Piiride kontroll võib mõnede ülesannete puhul nõuda üsna suurt arvu teste. Vaadelda tuleb nii sisend- kui väljundandmete piire.

Meetod 3: Oletused vigade kohta
Kogenud testijad leiavad peale üldiste meetodite veel antud ülesande jaoks sobivaid meetodeid vigade leidmiseks. Näiteks kellaaegade puhul teisendused 60-ndsüsteemi ja 10-ndsüsteemi vahel, trigonomeetrias teisendused kraadide ja radiaanide vahel.

### 2.1.4 Valge kasti meetodid

Programmi teksti uurides selgitatakse välja programmi struktuur: hargnemised, suunamised, tsüklid, alamprogrammid. Selle põhjal koostatakse testid, mis arvestavad programmi ehitust.

Meetod 1: Harude katmine. Testimisel peab iga käsk programmis olema töötanud vähemalt ühe korra.

Meetod 2: Otsustuste katmine. Iga otsustus peab testimise ajal olema tehtud igas võimalikus suunas vähemalt ühe korra.

Selliste algandmete konstrueerimine, et suure programmi sees satutaks teatud harusse või vahetulemused viiksid teatud otsustusele, võib olla keeruline. Test ise võib tulla vigane. Testi kontrollimiseks saab silumisvahenditega programmi täitmise enne otsustuse tegemist (alamprogrammi täitmist jne) peatada ja vaadata, kas muutujatel on oodatud väärtused. Saab kasutada ka lokaalset kontrolli, kirjutades silumiseks/testimiseks kestprogramme, mis söödavad programmi osale ette vajalikke vahetulemuste väärtusi.

### 2.1.5 Tüüpvead

Testimise ja silumise alases kirjanduses soovitatakse vigade lokaliseerimiseks pöörata tähelepanu järgmistele küsimustele:

1) Kas kõik muutujad on algväärtustatud? Kas tsükli korduval läbimisel tuleb muutujaid uuesti algväärtustada?

2) Kas massiivide kasutamisel indeksite väärtused on alati deklareeritud/kavatsetud piirides? Mõnedes programmeerimiskeskondades saab kompileerimisel indeksite väärtuste lubatavuse kontrolli sisse/välja lülitada.

3) Kas kasutusel on antud alamprogrammis või väljaspool deklareeritud muutuja? Tihti on paljukasutatud muutujad I, J, … olemas ka väljaspool alamprogrammi ja võidakse kasutada/muuta nende väärtusi, kui see muutuja alamprogrammis deklareeritud pole..

4) Kas sisendandmed loetakse õigest failist?

5) Kas jagaja väärtus saab olla 0?

6) Kas täisarvulise muutuja väärtus saab ületada tüübi piiri?

7) Kas vahetulemused on tüüpide piirides? Millist tüüpi on vahetulemused/osaavaldiste väärtused (reeglid sõltuvad programmeerimiskeelest)?

Matemaatika ülesannete programmeerimisel tekib tihti olukord, kus meil on mitu muutujat (muutujate paari jne), mis võivad ülesandes mängida sama rolli. Näiteks kui kontrollime, kas tippude koordinaatidega ette antud kolmnurk on võrdhaarne, siis võivad haaradeks olla suvalised kaks külge. Keerulisemate ülesannete korral võib aga variante olla mitte kolm, vaid palju rohkem. On olemas oht, et põhimõtteliselt oskame ülesannet küll lahendada, aga jätame kontrollitavasse tingimusse mõne muutujate kombinatsiooni kirjutamata või teeme vigu, kui paljundame tingimuse osi Copy/Paste abil. Testimise jaoks tähendab see, et testid peavad ühtviisi katma kõiki variante (mõnikord kümneid). Oma programmi teksti inspekteerimisel tuleks aga veenduda, et tingimustes on kirjas õige arv variante.

# 3 Ülesanded

Ülesanded on ülesannete kogus toodud niisuguses järjestuses, nagu vastavad teemad matemaatika õpikutes alates I klassist esimest korda sisse tuuakse. Paljudel juhtudel on ülesande lihtsaimale variandile lisatud keerukamaid variante, mis võivad olla pärit järgmiste klasside materjalist või saadud esialgse variandi üldistamise teel. Mis puutub programmi koostamiseks vajalikke teadmisi, siis paljud ülesanded on nooremate klasside õpikutes esitatud lahendamiseks mingite konkreetsete arvuliste andmetega. Aga selles ülesannete kogus nõutakse programmi, mis lahendab ülesande suvaliste (antud piirides asuvate) algandmete korral. Üldise algoritmi koostamiseks võib olla tarvis tunda keskkooli vanemate klasside või koguni ülikooli matemaatikat. Koos ülesannetega on tihti ka vajalik teoreetiline materjal.

Paljude loomulike ülesannete puhul on võimalikud erijuhud, kus ülesandes nõutud suurusi ei saagi leida (kahte punkti läbiva sirge tõusu leidmine, kui punktid ühtivad) või neil pole ülesandes vaadeldavat mõtet (kahte punkti läbiva sirge konstrueerimine samas olukorras). Selliste olukordade ettenägemine ja rakendusprogrammidele vastavate reaktsioonide programmeerimine on programmeerijate jaoks tähtis oskus, mis võimaldab vältida tõrgete tekkimist programmide töös. Selles ülesannete kogus on mõnede ülesannete tekstis öeldud, et sisendandmed ei sisalda niisuguseid erijuhte. Mõned ülesanded nõuavad otseselt kontrolli, kas on tegemist erijuhuga või mitte. Paljudel juhtudel pole aga ülesande tekstis erijuhtude võimalikkust soovitud meelde tuletada. Nende olemasolust arusaamine ning käsitlemise viis (mida sellisel juhul väljastada jms) on jäetud ülesande lahendajale. Mõnede selliste ülesannete puhul on erijuhtudest räägitud testimissoovitustes. Lahendust programmeerides oleks soovitav neid näpunäiteid vaadata alles pärast programmi valmimist ja esialgset oma testidega testimist.

## 3.1 I klass

Esimese klassi matemaatikas on algul vaatluse all ühekohalised positiivsed arvud, edasi null ja kahekohalised arvud. Programmeerimise jaoks pole need kitsendused tegelikult olulised. Loomulikul viisil kirjutatud programm lahendab sama ülesande ka suuremate arvude jaoks. Suurem osa jaotise 3.1 ülesannetest on üsna lihtsad.

Osutub, et juba esimeses klassis lahendatakse ka mitmete üsna oluliste algoritmiliste ülesannete väikesi erijuhte. Ülesanded 10, 14, 15 ja 18 seavad eesmärgiks koostada programm, mis lahendab antud tüüpi ülesande suvaliste arvuliste algandmete korral. Need ülesanded on eelmistest palju raskemad ja enne nende juurde asumist tuleks lahendada hulk järgmiste klasside matemaatikal baseeruvaid vahepealse keerukusega ülesandeid. Nendele ülesannetele on lisatud ka soovitusi, kuidas valida andmestruktuur ja kuidas panna tööle etappide kaupa töötav algoritm.

1. Arvude võrdlemine
   **Antud:** kaks täisarvu a ja b.
   **Leida:** Milline võrdlusmärk (' $<$ ',' $=$ ' või ' $>$ ') sobib nende vahele.

2. Liitmistehte tulemuse leidmine
   **Antud:** avaldis $a+b$, kus $a$ ja $b$ on naturaalarvud.
   **Leida:** summa $a+b$.

3. Võrduse $a + b = c$ puuduva liikme leidmine
   **Antud:** võrdus kujul $a + b = c$, kus kolmest liikmest $a, b$ ja c kaks on naturaalarvud ja kolmas on tundmatu $x$.
   **Asendada** $x$ sellise arvuga, et võrdus kehtiks (negatiivne väärtus teise liidetava jaoks panna sulgudesse).
   **Näited:** sisendi $1 + x = 10$ puhul väljastada $1 + 9 = 10$, sisendi $10 + x = 1$ puhul $10 + (-9) = 1$, sisendi $x + 10 = 1$ puhul $-9 + 10 = 1$.

4. Lahutamistehte tulemuse leidmine
   **Antud:** avaldis $a - b$, kus a ja b on naturaalarvud.
   **Leida:** vahe $a - b$.

5. Võrduse $a - b = c$ puuduva liikme leidmine (koos mittenegatiivsuse kontrolliga)
   **Antud:** võrdus kujul $a - b = c$ , kus kolmest liikmest a, b ja c kaks on täisarvud (võivad vajadusel olla sulgudes) ja kolmas on tundmatu $x$.
   **Asendada** $x$ sellise arvuga, et võrdus kehtiks (vajadusel panna x väärtus sulgudesse).

6. Võrdlusmärgi panemine tehte tulemuse ja arvu vahele
   **Antud:** sõne kujul $a \oplus b \lesseqgtr c$ kus $a$, $b$ ja c on naturaalarvud ning ⊕on pluss or miinus.
   **Leida:** sobiv võrdlusmärk $\lesseqgtr$ kohale (<, = või >).
   **Lahendus.** (I_6) Võrdlusmärgi valimine.py

7. Tehtemärgi panemine ⊕ asemele
   **Antud:** sõne kujul $a \oplus b \lesseqgtr c$ kus $a, b$ ja $c$ on naturaalarvud ning $\lesseqgtr$ on võrdlusmärk ($<, \leq, =, \geq$ $või >$).
   **Leida:** sobiv tehtemärk ⊕ kohale (+ või -).
   **Märkus.** Võib leiduda ka mitte ühtegi või mitu õiget vastust.

8. Tehtemärgi ja võrdlusmärgi valik
   **Antud:** sõne kujul $a?\,b?\,c$, kus $a, b$ ja $c$ on naturaalarvud.
   **Panna** ühe küsimärgi asemele tehtemärk + või – ja teise küsimärgi asemele võrdusmärk, nii et võrdus kehtiks.

9. Lõikude summa pikkus (ühikutega)
   **Antud:** kahe lõigu pikkused. Formaat: a meetrit b detsimeetrit c sentimeetrit d millimeetrit, kus $a, b, c$ ja $d$ on naturaalarvud, nulliga võrduvad komponendid võivad puududa ja $b, c, d < 10$.
   **Leida:** nende lõikude summa pikkus (samas formaadis).

10. Summa saamine arvutusrahadega
    Esimese klassi õpikus on ülesanne:
    Laduda arvutusrahadega lauale 31 senti. Kasutada on arvutusrahad väärtusega 50, 20, 10, 5 ja 2 senti.
    **Kuidas otsida lahendust?**
    Vaatame ülesannet pisut laiemalt ja selgitame välja kõik summad, mida üldse antud arvutusrahadest saab moodustada (ja mis ei ületa 31). Leiame järjestikku summad, mida on võimalik saada 0, 1, 2, … arvutusrahaga. Kui mingi rahade arvu juures saame summa 31, on ülesanne lahendatud ja vastus positiivne.
    Rahade arv 0 annab ainsa summa 0. Ühe rahaga saab lauale panna summad 20, 10, 5 ja 2 senti

(50 on suurem kui 31). Esimesest ühe rahaga summast saame 2 rahaga summad 30, 25 ja 22 senti. Summast 10 saame uute summadena 15 ja 12 senti, summast 5 saame 7 ja summast 2 saame lisaks 4. Kahe rahaga summadest saame kolmandat münti lisades uued summad 27, 24, 17, 14, 9 ja 6. Neljanda rahaga lisanduvad 29, 26, 19, 16, 11 ja 8. Viiendat raha lisades jõuame otsitava summani 31. Arvutuslikult võime protsessi organiseerida nii, et omistame järjendi $Vajalik[0], \dots, Vajalik[i], \dots\ Vajalik[31]$ elementidele väärtuseks minimaalse müntide arvu, millega on võimalik saada summa $i$. Algväärtusteks võime panna näiteks 32, mis vastusena on võimatu. Iga müntide arvu $k$ juures vaatame üle kõik summad, mida saab moodustada $k-1$ mündiga, ja asendame väärtusega $k$ kõik järjendi elemendid $Vajalik[i] = 32$, kus $i$ võib saada $(k-1)$-liikmelisest summast, lisades mingi mündi väärtuse. Et pärast oleks lihtsam arvutusrahade komplekti esitada, võib eraldi järjendisse meelde jätta ka summa $i$ saamiseks viimasena lisatud mündi väärtuse.
On ka võimalik, et ülesandes lubatud arvutusrahadest ei saagi nõutavat summat moodustada. Otsingu võime katkestada, kui mingil etapil $k$ ei teki ühtegi uut summat, mille jaoks minimaalne rahade arv oleks $k$.
Kirjeldatud otsingumeetod (seda nimetatakse **laiuti otsinguks**) võimaldab lahendada nõutava summa saamise ülesannet suvaliste algandmete korral ja annab õpiku ülesandele „optimaalse“ lahendi, milles on vähim võimalik arv arvutusrahasid. Oleme valmis järgmise programmeerimis-ülesande lahendamiseks:
**Antud:** nõutav lõppsumma ja arvutusrahade väärtused.
**Leida:** (võimalikult lihtne) arvutusrahade valik, mis annab nõutud summa.
**Lahendus.** (I_10) Muutujaga avaldise väärtus.py

11. Leida summade tabel 10-ni.
    a) **Trükkida** summade a+b tabel liidetavate $a, b = 1 \dots 10$ jaoks.

12. Väljenda detsimeetrites ja sentimeetrites
    **Antud:** objekti pikkus meetrites, detsimeetrites ja sentimeetrites.
    **Väljendada** see pikkus detsimeetrites ja sentimeetrites.
    **Näide.** Kui sisend on 13 meetrit 7 detsimeetrit 6 sentimeetrit,
    siis peab väljund olema 137 detsimeetrit 6 sentimeetrit.

13. Nädalapäevade liitmine ja lahutamine
    **Antud:** nädalapäev ja päevade arv $a$.
    **Leida:** milline nädalapäev
    a) on $a$ päeva pärast,
    b) oli $a$ päeva eest.

14. Veehulga mõõtmine antud mahuga anumaid kasutades
    Vett võetakse kraanist ja on vaja valada potti teatud hulk vett (poti maht pole teada, aga ta on piisavalt suur). Vee mõõtmiseks saame kasutada kahte teadaoleva mahuga anumat. Mõõtmise igal sammul võime kallata mingi anuma teisest anumast/potist või kraanist täis või kallata mingis anumas oleva kogu vee ära/teise anumasse/potti.
    **Näitena** vaatleme 1. klassi õpikus olevat konkreetset ülesannet: Kuidas saada potti 7 liitrit vett, kui saame kasutada 3-liitrist ja 5-liitrist anumat?
    Kui kujutame anumates ja potis olevaid veehulki arvude kolmikutena, siis võime mõõtmisel teha järgmised sammud:
    (0,0,0)

(0,5,0)
(0,0,5)
(0,5,5)
(3,2,5)
(3,0,7)
**Kuidas otsida lahendust?** On selge, et anumates olevad veemahud on alati täisarvulised, seega võime mõõtmisprotsessi võimalikke seisundeid kujutada naturaalarvude kolmikutega (0,0,0),…, (3,5,7). Otsingul püüame iga sammude arvu jaoks leida seisundid, milleni võime sellise sammude arvuga jõuda. Seisundi (0,0,0) sammude arv on 0. Ühe sammuga võime jõuda seisunditeni (3,0,0) ja (0,5,0). Nendest seisunditest saab teise sammuga jõuda seisunditeni (0,3,0), (0,0,3), (3,5,0), (3,2,0) ja (0,0,5). Nendest seisunditest kõikvõimalikke samme tehes saame leida kolme sammuga saavutatavad seisundid jne. Kui jõuame seisundini, kus kolmas komponent on 7, oleme saanud lahenduse (ja selleks vajaliku minimaalse sammude arvu). On ka võimalik, et etteantud anumate mahud ei võimaldagi eesmärki saavutada. Negatiivne vastus tuleb anda, kui mingil etapil $n$ ei teki uusi seisundeid.
Üldjuhu programmeerimise ülesande võime sõnastada järgmiselt:
**Antud:** vajalik vee maht potis $a$ ja
a) kahe mõõtmisanuma mahud $b$ ja $c$,
b) kolme mõõtmisanuma mahud $b$, $c$ ja $d$,
c) mõõtmisanumate arv $m$ ja nende $m$ anumate mahud.
**Leida:** sellise võimalikult lühikese mõõtmisprotsessi käik (seisundite jada), mille tulemusena mõõdetakse $a$ liitrit vett või saadakse negatiivne vastus.

15. Kangkaaludega kaalumine
Esimese klassi õpikus on ülesanne: Kuidas saab 5 $kg$ ja 2 $kg$ vihiga kaaluda 3 $kg$ tangu?
Selles ülesandes on mõeldud kaalumist kahe kaalukausiga kangkaaludega, kus saab vihte panna mõlemale kaalukausile. Antud näite korral võime ühele kaalukausile panna 5 $kg$ vihi, teisele 2 $kg$ vihi ja kaaluda siis sellel kausil 3 $kg$ tangu. Üldjuhul tuleb otsida vihtide sellist paigutust, et kaalukaussidel olevate vihtide kaalude vahe oleks võrdne kaalumisele kuuluva raskusega .
**Antud:** vajalik kaalumisel saadav kogus $a$ ja
a) kahe kaaluvihi kaalud,
b) kaaluvihtide kaalud (võib olla rohkem kui 2 erinevat vihti),
c) kaaluvihtide kaalud ja antud kaaluga vihtide arvud.
**Leida:** vihtide sellised mittelõikuvad osahulgad, et nende kaalude vahe oleks $a$ ühikut (mõned vihid võivad ka kasutamata jääda).
**Kommentaar.** Ülesannet saab mitmes suunas üldistada.
1) Saab lubada kaalumist osade kaupa, näiteks saame kaaluda $6 = 3 + 3$ kilogrammi.
2) Vähemalt puisteainete puhul saaks lubada „lisavihtide“ valmistamist kaalutavast ainest (müüjad turul teevad seda). Ülatoodud näite korral saaks tekitada isegi 1 $kg$ vihi ja edasi saaks juba kaaluda suvalise täisarvulise koguse.

16. Järeldamine kahest võrratusest
**Antud:** kaks võrratust või võrdust arvude $a, b$ ja $c$ vahel, mis kirjeldavad kahte erinevat erinevate komponentidega arvude paari.
**Leida:** Sisendist järelduv maksimaalselt tugev võrratus/võrdus kolmanda paari kohta (kui võimalik).
**Näited.** Kui näiteks $a < c$ ja $a > b$, siis saame $b < c$. Kui aga $a < b$ ja $a < c$, siis võivad $b$ ja $c$ olla

suvalises vahekorras.
**Kommentaar.** Üldiselt võib nii ühest kui kahest erinevast sisendseosest koos tuleneda vastuolu. Näiteks $a < a$ või $a < b$ ja $b = a$ on vastuolulised sisendid. Selle ülesande tingimused muutujate kohta sisendseostes ei võimalda vastuolu. Ülesande lahendamiseks tuleb selgeks teha, millisel juhul saame kahest seosest midagi uut järeldada. Üks muutuja esineb mõlemas seoses ja midagi uut võib järelduda kahe ülejäänud muutuja seose kohta. Kui vähemalt üks seostest on võrdus, siis on järelduse tegemine lihtne. Näiteks kui $a < b$ ja $b = c$, siis ka $a < c$. Kaks võrratust paigutavad kaks ühekordselt esinevat muutujat kas ühele poole kolmandat või teise teisele poole. Viimasel juhul saame rakendada võrratuse transitiivsust.

17. Võrduste ja võrratuste tabel
**Sisendi** esimesel real on naturaalarvud $m$ ja $n$, kus $m \leq 6$ ja $n \leq 10$. Teisel real on $n$ üksteisest komaga eraldatud võrdust/võrratust/mittevõrdust muutujate vahel, kus muutujateks on $m$ esimest ladina tähestiku väiketähte, võrdlusmärgid on kujul $<<, <=, ==, >=, >>, !=$.
a) sisend koosneb ainult võrdusseostest,
b) sisend koosneb ainult võrduse ja mittevõrdumise seostest,
c) sisend koosneb ainult rangetest ja mitterangetest võrratustest,
d) sisend koosneb ainult võrdustest ja võrratustest,
e) sisendis võivad esineda kõik ülaltoodud seosed.
**Leida** iga muutujate paari jaoks sisendist järelduv tugevaim seos ja väljastada tulemus $m \times m$ tabelina vastavalt näitele, seoste vahel reas on koma. Kui mingite muutujate $x$ ja $y$ seose kohta ei järeldu sisendist midagi, siis kirjutada tabelisse $x??y$. Kui sisend on vastuoluline, siis kirjutada väljundi ainsale reale VASTUOLU.
**Näide.** Kui sisend on
4 3
```
a<=b,a<=c,c<<d,
```
siis väljund peab olema
```
a==a,a<=b,a<=c,a<<d
b>=a,b==b,b??c,b??d
c>=a,c??b,c==c,c<<d
d>>a,d??b,d>>c,d==d
```
**Kuidas lahendada?** Tabeli jaoks võiks moodustada kahetähelistest võrdlusmärkidest koosneva $m \times m$ sõnemassiivi ja kirjutada sinna küsimärgid ja diagonaalil asuvad võrdused.
Edasi võiks programm koosneda etappidest, kus vaadeldakse järjekordset sisendis olevat seost ja lisatakse tabelisse kõik võimalikud järeldused või teatatakse vastuolust. Järelduste saamiseks tuleb uut seost võrrelda vastavas lahtris juba olemasoleva seosega, rakendada sümmeetriat, rakendada transitiivsust (vaadeldes kõiki juba tabelis olevaid seoseid, kus esineb vähemalt üks uue seose muutujatest). Kui mingisse lahtrisse saadakse uus tugevam seos, siis tuleb ka temast lähtudes teha kõikvõimalikud järeldused. Uuest seosest lähtuva etapi saame lõpetada, kui uusi järeldusi enam juurde ei tule. Iga järeldus tähendab tabeli lahtris oleva seose asendamist tugevamaga ja järelduste ahelas saab olla ainult lõplik arv seoseid
($?? \rightarrow mitterange\ järjestus \rightarrow range\ järjestus\ või\ võrdus \rightarrow vastuolu$).

18. Kivimite jagamine kaalult võrdseteks hulkadeks
**Antud:** kivimite arv, igaühe kaal (täisarv).
**Leida:** Kas ja kuidas saab kivimid jagada kahte võrdse kaaluga ossa?
**Algoritmist.** Otsime sellist osahulka, mille kaal on pool summaarsest kaalust. Osahulga otsimisel võime kasutada rekursiooni või üksteisesse sisestatud tsükleid, kus järjekordset kivimit

paigutatakse osahulka 0 korda või 1 kord. Kivimite lisamise võib lõpetada, kui osahulk kaalub juba liiga palju. On selge, et nii vaadatakse läbi kuni $2^n$ varianti, kus $n$ on kivimite arv.
**Lahendus.** Rekursiooni kasutav lahendus on failis (I_18) Kivimid pooleks.py

## 3.2 II klass

1. Summa maksmine antud müntidega
   **Antud:** maksmisele kuuluv summa $a$ ja maksjal olevate müntide hinnad (ühesuguseid münte võib olla rohkem kui üks).
   **Leida:** kõik võimalused, kuidas saab maksta summa $a$.
   **Näide.** Vajalik summa on 21 senti, maksjal on mündid 20, 20, 10, 5, 5, 2, 2, 2, 2 senti.
   **Märkus.** Erinevalt varasemast arvutusrahade ülesandest tuleb siin arvestada ka ühesuguse väärtusega müntide arvudega.

2. Järjestamine pikkuse järgi
   **Antud:** laste arv, laste nimed ja pikkused.
   **Järjestada** lapsed pikkuse kasvamise järgi.
   Selle ülesande lahendamine kuulub tavaliselt programmeerimiskursusesse.

3. Kellaajale minutite liitmine
   **Antud:** kellaaeg (tunnid ja minutid) ja ooteaeg $a$ (minutites, väiksem kui 24 tundi).
   **Leida:** kellaaeg $a$ minuti pärast.
   **Lahendus.** (II-3) Kellaaeg+minutid.py

4. Võrdsete liikmetega summa väljendamine korrutisena
   **Antud:** avaldis kujul $a_1 + \cdots + a_k$, kus $a_1, \ldots, a_k$ on naturaalarvud, k>0.
   **Väljendada** summa kujul
   1) $k * a$, kui summa kõik liikmed on võrdsed,
   2) $k_1 * a_1 + \cdots + k_n * a_n$, kui summas esineb $n$ erinevat liidetavate väärtust.

## 3.3 III klass

1. Suvalise järjendi sorteerimine
   **Antud:** järjestatavate arvude arv $n$ ja arvud $a_1, \dots, a_n$.
   **Järjestada** arvud mittekahanevalt.

2. Tehted detsimeetrite ja sentimeetritega
   **Antud:** kahe lõigu pikkused (detsimeetrid ja sentimeetrid).
   **Leida:** lõikude summa ja vahe pikkused.
   **Testimisest.** Peale peajuhu testide veel: testid, kus lõigu sentimeetrite/detsimeetrite arv on null; testid, kus sama tekib vastuses; kaks võrdset lõiku.

3. Mitmel erineval viisil saab ostu eest tasuda?
   **Antud:** ostu hind (sentides), ostjal olevate müntide arv $n$ ($n \leq 10$) ja müntide väärtused $m_1, \dots, m_n$ (sentides, ühesuguseid võib olla mitu).
   **Leida:** mitmel erineval viisil saab ostu eest nende müntidega tasuda (sama väärtusega mündid loetakse siinjuures võrdseteks).
   **Testimisest.** On vaja teste erineva ostjal olevate müntide arvuga, võimalusega moodustada sama summa erinevatest kahest/kolmest mündist, teste erinevate vastustega.

4. **Kirjutada** arvud 1,1,1,2,2,2,3,3,3 3x3 ruudustiku ruutudesse, nii et
   a) ridade ja veergude summad on võrdsed,
   b) ridade, veergude ja diagonaalide summad on võrdsed,
   c) igas reas, igas tulbas ja igas diagonaalis oleks 1,2 ja 3
   või anda vastus VÕIMATU.

5. Võluruudu kontroll
   **Antud:** a) 3x3 naturaalarvu,
   b) 4x4 naturaalarvu,
   c) 5x5 naturaalarvu.
   **Leida:** kas arvud moodustavad võluruudu, kus
   a) ridade ja veergude summad on võrdsed,
   b) ridade, veergude ja diagonaalide summad on võrdsed.
   **Testimisest.** Vaja on nii positiivse kui negatiivse vastusega teste. Negatiivne vastus võiks tulla erinevate ridade/veergude/diagonaalide tõttu. Variandi b) jaoks on vaja ka teste, kus variandi a) tingimus on täidetud.
   Lahendus. (III-5) Võluruudu kontroll.py

6. Võluruudu ehitamine antud arvudest
   **Antud:** $n^2$ naturaalarvu (mõned võivad olla ka võrdsed).
   **Paigutada** need arvud $n \times n$ ruudustikku nii, et tekiks võluruut, kus
   a) ridade ja veergude summad on võrdsed,
   b) ridade, veergude ja diagonaalide summad on võrdsed.
   **Sisend.** Sisendfaili ainsal real on $n^2$ tühikutega eraldatud naturaalarvu.
   a) $n = 3$,
   b) $n = 4$.
   **Väljund.** Väljundfaili esimesele reale väljastada tühikutega eraldatuna ühe võimaliku võluruudu esimese rea elemendid, teisele reale võluruudu teise rea elemendid jne. Kui antud arvudest ei saa võluruutu moodustada, siis väljastada EI SAA.

**Nõuanne.** Ülesannet on $n = 3$ jaoks võimalik üsna lihtsalt lahendada variantide läbiproovimise teel, aga huvitavam oleks meetod, mida saaks rakendada ka suurema $n$ korral.

7. Teisendamine: meetrid, detsimeetrid, sentimeetrid, millimeetrid
   **Antud:** Mingi objekti pikkus (meetrid, detsimeetrid, sentimeetrid, millimeetrid) ja mõõtühik, millesse tuleb pikkus teisendada. Pikkuse komponendid on esitatud naturaalarvudena koos ühikuga, nulliga võrduvad komponendid võivad puududa.
   **Leida:** Objekti pikkus sisendis nõutud ühikutes (ei tarvitse olla täisarv).
   **Näide.** Kui sisendiks on $3\ m\ 25\ cm\ 6\ mm$ ja nõutav ühik $dm$, siis väljund peab olema $32{,}56\ dm$.
   **Testimisest.** On vaja teste, kus esinevad/võrduvad nulliga/puuduvad sisendi erinevad komponendid

8. Korrutustabel
   **Trükkida** korrutustabel teguritega $1 \dots 10$ nii, et vastavad kümnendkohad oleks üksteise all.

9. Jäägiga jagamine
   **Sisend.** Avaldis $a\!:b$, kus $a$ ja $b$ on naturaalarvud, $b > 0$.
   **Väljund.** Kirjutada väljundfaili esimesele reale jagatise täisosa ja kui arvud ei jagu täpselt, siis lisada JÄÄK … .

10. Homne kuupäev
    **Antud:** tänane kuupäev (formaadis päev. kuu) ja informatsioon aasta kohta (+ või – vastavalt sellele, kas on lisapäeva-aasta).
    **Leida:** homne kuupäev.

## 3.4 IV klass

1. Kas kõiki numbreid on kasutatud?
   Leida, kas antud arvudes on kasutatud kõiki numbreid.
   **Sisend:** Sisendfaili real pikkusega kuni 80 sümbolit on tühikuga eraldatud naturaalarvud.
   **Väljund:** Kui arvudes on kasutatud kõiki numbreid, siis JAH.
   Kui pole, siis: PUUDUVAD <puuduvad numbrid kasvavas järjekorras>.

2. Suuruselt teine arv
   Leida arvujada suuruselt teine arv.
   **Sisend:** Sisendfaili real pikkusega kuni 80 sümbolit on tühikuga eraldatud naturaalarvud.
   **Väljund.** Suuruselt teine arv arvujadas.
   **Nõuanne.** See ülesanne on nn tsükli invariandi kohta. Tsükli invariant on see info, mida programm säilitab seni läbitud jada osa kohta. Kindlasti peab ta sisaldama jada senist teise suurusega liiget, aga ka võimaldama selle muutmist kõikidel juhtudel, mis järgmiste liikmete väärtused võivad tekitada.
   **Testimisest.** Testid peaksid sisaldama juhte, kus
   - jada suuremad liikmed on kõik erinevad,
   - mõned suuremad liikmed (erinevates kombinatsioonides) on omavahel võrdsed,
   - vastus tekib juba tsükli invariandi algväärtustamisel,
   - tsükli invariandi muutujate väärtusi tuleb muuta mõned korrad või palju kordi.
   **Lahendus.** (IV-2) Suuruselt teine arv.py

3. Tingimustele vastavate arvude moodustamine
   Moodustada antud numbrikaartidest võimalikult suur 5-kohaline arv ning vähim 2-ga jaguv arv.
   **Sisend:** Sisendfaili real pikkusega kuni 80 sümbolit on tühikuga eraldatuna numbrikaartidel olevad numbrid (üks number võib esineda mitmel kaardil).
   **Väljund:** Kirjutada väljundfaili esimesele reale neist numbritest moodustatud võimalikult suur 5-kohaline arv ja teisele reale vähim 5-kohaline 2-ga jaguv arv.
   **Soovitus.** Moodustada selle ülesande jaoks ise erinevaid tingimusi.

4. Värvitud kuupidest koosneva kujundi vaade
   Kujund on moodustatud värvilistest ühikkuupidest, kus iga kuubi kõik tahud on sama värvi. Värvid on tähistatud numbritega 1-9. Kujund asub koordinaatteljestiku osas, kus kuupide koordinaadid rahuldavad võrratusi $1 \leq x, y, z \leq 5$.
   **Sisend.** Sisendfaili esimesel real on kuupide andmeid esitavate ridade arv n ja järgmisel n real komaga eraldatuna üksikute kuupide andmed (kuubi koordinaadid ja värv), kus igal real kuni eelviimaseni on kümne kuubi andmed, viimasel võib olla vähem.
   **Väljund.** Kirjutada väljundfaili esimesele viiele reale värvid, mis paistavad ülalt, st z-telje positiivse osa poolt vaadates. Number 0 tähistab värvilise tahu puudumist antud positsioonis.
   **Näide.** Kui sisendiks on
   2
   (1,1,1)3,(1,2,1)2,(1,3,1)2,( 1,4,1)4,(1,5,1)5,(2,1,1)1, (2,3,1)3,(2,5,1)7,(3,2,1)6,(4,3,1)4,
   (1,2,2)1 ,(1,5,3)3
   siis väljund peab olema
   (1,1)3,(1,2)1,(1,3)2,(1,4)4,(1,5)3
   (2,1)1,(2,2)0,(2,3)3,(2,4)0,(2,5)7

(3,1)0,(3,2)6,(3,3)0,(4,4)0,(4,5)0
(4,1)0,(4,2)0,(4,3)4,(4,4)0,(4,5)0
(5,1)0,(5,2)0,(5,3)0,(5,4)0,(5,5)0

5. Ajavahemike summa
Antud: kaks ajavahemikku (tunnid, minutid, sekundid).
**Leida:** nende ajavahemike summa samas formaadis.

6. Üldistatud 15-mäng
$m \times n$ ruudustiku ruutudes on mingis järjekorras nupud arvudega $1, 2, 3, \ldots, m \times n - 1$ ja üks ruut on tühi. Seise tähistame jadana, kus on järjestikku kirjutatud esimeses, teises jne ridades olevad arvud, kusjuures tühja ruutu tähistame numbriga 0. Mängu üks käik tähendab nupu nihutamist naaberruudult (alla, üles, vasakule või paremale) tühja ruutu. Mängija ülesandeks on jõuda etteantud algseisust lõppseisu $(1,2,3, \ldots, m \times n - 1,0)$. Näites vaatleme siin 3x2 mängu, kus nuppudel olevad arvud on ühekohalised ja seise saab tähistada 6-numbriliste sõnede abil.
**Sisend.** Sisendfaili esimesel real on mängu algseis, näiteks 3x2 mängus võib selleks olla 231054.
**Väljund.** Kui lõppasendisse jõudmine on võimalik, siis väljastada järjestikuste seisude jada algseisust lõppseisuni, paigutades iga seisu eraldi reale. Kui lõppseisu jõudmine ei ole võimalik, siis väljastada EI SAA.
**Variandid.** Erinevad ruutude arvud ja/või ruudustiku kujud: 4x2, 3x3, 5x2, 4x3, 3x4, 6x2, … . Selle mängu tuntuim variant on muidugi 15 nupuga mäng 4x4 laual, mida on kirjeldatud ka Wikipedias: https://en.wikipedia.org/wiki/15_puzzle. Soovitame proovida, kui suure ruudustiku korral lahendab Teie programm ülesande vastuvõetava aja jooksul.
**Lahenduse olemasolu.** Lahendust ei ole pooltel juhtudest. Selle tõestamiseks tähistame mänguseisu arvude jadana, kus tühja kohta tähistab 0. Vaatleme selle jada inversioonide („vales järjekorras“ olevate paaride) arvu ja tühja lahtri viimasele positsioonile viimiseks vajalikku käikude arvu. Näiteks jadas 231054 on 5 inversiooni (paarid 21, 20, 31, 30, 10, 54) ja tühja lahtri viimasele positsioonile viimiseks on vaja 2 käiku. On kerge kontrollida, et ülaltoodud kahe suuruse summa säilitab igal käigul oma paarsuse. Lõppseisu puhul on vaadeldav summa 5, seega paaritu summaga seisudest (nagu oli näites) saab liikuda lõppseisu.
**Kuidas lahendada?** Kui võimalike seisude arv pole liiga suur, saame kasutada laiuti otsingut. Näiteks $3x2$ mängus on $6! = 720$ võimalikku seisu. Omistame algseisule käikude arvu 0. Edasi vaatleme etapil i kõiki seise, kuhu võib jõuda $i - 1$ käiguga. Seisudele, kuhu niisugustest seisudest saab liikuda 1 käiguga (ja millele pole veel käikude arvu omistatud), omistame käikude arvu $i$. Jätkame protsessi seni, kuni mingil etapil $i$ ei teki ühtegi $i$ käiku nõudvat seisu. Kui ruudustik on suur, siis töötab laiuti otsing liiga aeglaselt. Kui pole nõutud minimaalse käikude arvuga lahendust, siis võime modelleerida „loomulikku“ algoritmi, mida inimene selle ülesandega kohtudes tavaliselt leiutab. Püüame paigutada õigele kohale numbrid 1, 2, jne. Saame kõigepealt täita esimesed $m - 2$ rida numbrite järjekorras (vastava võtte võime välja töötada „füüsilisi“ nuppe kasutades). Sealjuures saame kasutada kahte järgmist rida. Viimased kaks rida võime täita laiuti otsingu abil, kui see on võimalik.

i

## 3.5 V klass

### 3.5.1 Naturaalarvude liitmine ja lahutamine

1. Naturaalarvu esitamine järkarvude summana
   **Antud**: positiivne naturaalarv $a$.
   **Esitada** arv $a$ järkarvude summana (alates sobivast järgust).
   **Sisend:** Sisendfaili ainsal real on naturaalarv $a$, kus $0 < a \leq 1000000$.
   **Väljund:** Kirjutada väljundfaili esimesele reale võrdus $a =< järkarvude\ summa >$.
   **Väljundi näited:** 3201=3000+200+1; 7=7.

2. Naturaalarvu kirjutamine sõnadega
   **Antud**: positiivne naturaalarv $a$.
   **Esitada** arv $a$ sõnadega kirjutatuna
   **Sisend:** Sisendfaili ainsal real on naturaalarv $a$, kus $0 < a \leq 1000000$.
   **Väljund:** Kirjutada väljundfaili esimesele reale võrdus $a =< a\ esitus\ sõnadega >$.
   **Näide:** 3201=kolm tuhat kakssada üks.

3. Kirjaliku liitmise vormistamine
   **Antud:** avaldis (positiivsetest naturaalarvudest plussmärkide abil moodustatud summa).
   **Esitada** nende liitmine tavalise kirjaliku liitmisalgoritmi järgi, paigutades kõik arvud õigetele positsioonidele.
   **Sisend:** Sisendfaili ainsal real on naturaalarvudest plussmärgi abil moodustatud avaldis. Avaldise pikkus võib olla kuni 80 sümbolit, ta sisaldab vähemalt kaks liidetavat. Arvud on kuni 6-kohalised.
   **Väljund.** Kirjaliku liitmisalgoritmi rakendamise tulemus, kus
   1) arvud asuvad õigetel positsioonidel üksteise all, saadud summa algab esimesest positsioonist,
   2) liidetavad ja tulemus on eraldatud miinustest koosneva lõpptulemuse pikkuse joonega,
   3) liitmismärki pole väljastatud.
   **Väljundi näide:**
   ```
    234567
       246
      5982
         4
     56789
    876543
   -------
   1174131
   ```
   **Nõuanne.** Väljundfaili vaatamiseks võiks kasutada fonti, kus kõikide sümbolite laiused on võrdsed (näiteks Courier).

4. Pluss- ja miinusmärke sisaldava avaldise väärtuse leidmine
   **Antud:** naturaalarvudest pluss- ja miinusmärkide abil moodustatud avaldis.
   **Leida:** avaldise väärtus.
   **Sisend:** Sisendfaili ainsal real on naturaalarvudest pluss- ja miinusmärkide abil moodustatud

avaldis. Avaldise pikkus võib olla kuni 80 sümbolit. Arvud on kuni 4-kohalised. Avaldises võib olla ka ainult 1 liige.
**Väljund:** Kirjutada väljundfaili esimesele reale avaldis, võrdusmärk ja avaldise väärtus.
**Väljundi näide:** 2345-738-5356+8=-3741

5. Naturaalarvudest pluss- ja miinusmärkide ning sulgude abil moodustatud avaldise väärtuse leidmine
**Antud:** naturaalarvudest pluss- ja miinusmärkide ning sulgude abil moodustatud avaldis.
**Leida:** avaldise väärtus.
**Sisend:** Sisendfaili ainsal real on naturaalarvudest pluss- ja miinusmärkide ning sulgude abil moodustatud avaldis. Avaldise pikkus võib olla kuni 80 sümbolit. Arvud on kuni 4-kohalised. Avaldises võib olla ka ainult 1 liige.
**Väljund:** Kirjutada väljundfaili esimesele reale avaldis, võrdusmärk ja avaldise väärtus.
**Väljundi näide:** 2345-((738-5356)+8)=6955.
**Soovitus.** Võib teisendada avaldist, asendades sisemistest sulgudest alates sulgavaldised nende väärtustega, tehes sulgudes tehted vasakult paremale. Võib ka taandada eelmisele ülesandele, jättes sulge ära ja muutes sealjuures sulgudes olevate liikmete märke, kui sulgude ees on miinus.

6. Ristküliku pindala ja ümbermõõdu leidmine
**Antud:** ristküliku küljed (ühesugustes ühikutes - mm, cm, dm, või m).
**Leida:** ristküliku pindala ja ümbermõõt.
**Sisend:** Sisendfaili esimesel real on ristküliku pikkus koos tühikuga eraldatud mõõtühikuga, teisel real laius samas formaadis (sama ühikuga). Pikkus ja laius on kuni 3-kohalised positiivsed naturaalarvud.
**Väljund:** Kirjutada väljundfaili esimesele reale ristküliku pindala ning tühikuga eraldatud ühik (vastavalt sisendandmete ühikule, ruut kirjutada kujul **2),
**Näide.** Olgu sisendfailis:
15 cm
8 cm
Siis väljundfaili tuleb kirjutada:
120 cm **2
46 cm

7. Ümardamine
**Antud:** Ümardatav täis- või reaalarv $a$ ja järguühik, milleni tuleb arv $a$ ümardada.
**Leida:** Ümardamise tulemus.
**Sisend.** Sisendfaili esimesel real on ümardatav täisarv või reaalarv $a$ (absoluutväärtusega kuni 1 miljon), teisel real järguühik, milleni tuleb ümardada (…, 0.001, 0.01, 0.1, 1, 10, 100, …).
**Väljund.** Kirjutada väljundfaili esimesele reale
$a\sim<\text{ümardamise tulemus}>$
**Näited.** Olgu sisendfailis read
23456.789
0.1
Siis väljundfaili tuleb kirjutada
23456.789~23456.8

Kui teisel real on

```
100
```

siis peab väljund olema

```
23456.789~23500
```

8. Ühikuga suuruse teisendamine vastavalt etteantud ühikule
   **Antud**: ühikuga suurus (pikkus, pindala või ruumala) ja ühik, milles tuleb anda vastus.
   **Leida:** sisendsuurus nõutud ühikutes.
   **Sisend:** Sisendfaili esimesel real on naturaal- või reaalarv, millele järgneb tühik ja selle taga suuruse ühik, teisel real on sama suuruse jaoks sobiv ühik, milles tuleb anda vastus. Ruut või kuup ühikutes esitatakse sümbolite ** abil, näiteks 12 cm**2 ja 5 cm**3. Ühikuteks võivad olla pikkuse korral mm, cm, dm, m, km ja pindala või ruumala korral vastavad ruut- ja kuupühikud.
   **Väljund.** Kirjutada väljundfaili esimesele reale sisendi esimese rea sisu, võrdusmärk, nõutud ühikuteks teisendatud sisendsuuruse arvuline väärtus (täis- või reaalarv), tühik ja ühik sisendfaili teiselt realt.
   **Näide.** Olgu sisendiks
   125 mm**2
   cm**2
   Siis väljundiks tuleb kirjutada
   125 mm**2=1.25 cm**2

9. Araabia ja rooma numbrid
   Teisendada araabia numbritega antud arv rooma numbriteks ja vastupidi.
   a) **Antud:** arv araabia numbritega kirjutatuna.
   **Leida:** sama arv rooma numbrina.
   **Sisend:** Sisendfaili ainsal rea l on araabia numbritega kirjutatud naturaalarv $A$, kus $1 \leq A \leq 3000$.
   **Väljund:** sama arv rooma numbritega kirjutatuna.

   b) **Antud:** arv rooma numbritega kirjutatuna.
   **Leida:** sama arv araabia numbritega kirjutatud kujul.
   **Sisend:** Sisendfaili ainsal real on rooma numbritega kirjutatud naturaalarv $A$, kus $1 \leq A \leq 3000$.
   **Väljund:** sama arv araabia numbritega kirjutatuna.

10. Järjestamine (2-dimensionaalne)
    **Antud:** Koordinaatidega esitatud tasandipunktide loend.
    **Järjestada** punktid $x$-koordinaadi järgi mittekahanevalt, võrdsete $x$-koordinaatide korral $y$-koordinaatide järgi mittekahanevalt.
    **Sisend:** Sisendfaili esimesel real on punktide arv $n$,
    järgmisel $n$ real $n$ punkti koordinaadid formaadis $(x; y)$, kus $x$ ja $y$ on täis- või reaalarvud.
    **Väljund:** Väljundfaili järjestikustele ridadele kirjutada punktide koordinaadid ülesandes nõutud järjekorras. Iga punkt väljastada ainult ühes eksemplaris.

11. Muutujate asendamine väärtustega
**Antud:** muutujaid sisaldav avaldis $A(x_1, \dots, x_k)$ ja muutuja väärtus(ed) $a_1, \dots, a_k$.
**Asendada** avaldises kõik muutuja esinemised nende väärtus(t)ega.
**Sisend:** Sisendfaili esimesel real pikkusega kuni 80 sümbolit on sulgudeta avaldis, mis on koostatud reaalarvudest ja muutujatest plusse ja miinuseid kasutades. Järgmistel ridadel on võrdused, mis esitavad avaldises esinevate kõigi muutujate väärtused (täis- või reaalarvud).
**Väljund:** Kirjutada väljundfaili esimesele reale avaldis $A$, kus muutujad on asendatud nende väärtustega. Vajadusel panna väärtus sulgudesse.

12. Muutujatega avaldise väärtuse leidmine
**Antud:** muutujaid sisaldav avaldis $A(x_1, \dots, x_k)$ ja muutuja väärtus(ed) $a_1, \dots, a_k$.
**Leida:** avaldise väärtus.
**Sisend:** Sisendfaili esimesel real pikkusega kuni 80 sümbolit on sulgudeta avaldis, mis on koostatud reaalarvudest ja muutujatest plusse ja miinuseid kasutades. Järgmistel ridadel on võrdused, mis esitavad avaldises esinevate kõigi muutujate väärtused (täis- või reaalarvud).
**Väljund:** Kirjutada väljundfaili esimesele reale avaldise väärtus muutujate antud väärtuste korral.
**Kuidas arvutada.** Pärast muutujate asendamist nende väärtustega on avaldise väärtuse arvutamiseks kaks loomulikku teed rekursiooni rakendamiseks. Soovitav oleks mõlemad läbi proovida:
1) rekursiivne programm, mis leiab tehte, mida võib teha esimesena, teostab selle ja pöördub siis iseenda poole ühe tehtemärgi võrra lihtsama avaldisega,
2) rekursiivne programm, mis leiab viimase tehte asukoha, pöördub iseenda poole vasaku ja parema osaavaldise väärtuse arvutamiseks ning rakendab neile siis viimast tehet.

13. Lineaarse funktsiooni väärtuste tabel
**Antud:** kordajad A ja $B$, vaadeldava lõigu otspunktid $a$ ja $b$ ($a < b$) ning tabeli samm s ($s < b - a$).
**Leida:** $Ax + B$ väärtuste tabel $x \epsilon [a, b]$ jaoks sammuga $s$ ($s < b - a$).
Tabeli vormistus tuleb täiendavalt fikseerida (valida ise).

14. Üldkujulise funktsiooni väärtuste tabel
**Antud:** funktsiooni $f(x)$ esitav avaldis, mis võib sisaldada tehtemärke, sulge ja elementaarfunktsioonide tähiseid; vaadeldava lõigu otspunktid $a$ ja $b$ ($a < b$) ning tabeli samm $s$ ($s < b - a$)
**Leida:** funktsiooni $f(x)$ väärtuste tabel $x \epsilon [a, b]$ jaoks sammuga $s$.
Tabeli vormistus tuleb täiendavalt fikseerida.
**Nõuanded.** Funktsiooni väärtuste arvutamiseks võib kasutada üleeelmise ülesande juures kirjeldatud skeeme. Kui avaldis sisaldab mitte kõikjal määratud funktsioone (näiteks jagamine, tangens), peab algoritm sisaldama kontrolle, mis väldivad arvutuse katkemist, ja programm peab tabelisse funktsiooni väärtuse asemel kirjutama vastava märke.

15. Tehete järjekorra lubatavuse kontroll
**Antud:** Arve, sulge ja tehtemärke sisaldav avaldis ja uuritav tehete järjekord.
**Kontrollida**, kas antud tehete järjekord on lubatav.
**Sisend:** Sisendfaili esimesel real on arve, sulge ja tehtemärke sisaldav avaldis (kuni 9 tehtemärki).

Teisel real on avaldise tehtemärkide all kontrollitavale tehete järjekorrale vastavad tehete järjekorranumbrid 1…9 (võib eeldada korrektset paigutust).
**Väljund.** Kui tehete järjekord on lubatav, siis kirjutada väljundfaili esimesele reale LUBATAV. Vastasel juhul kopeerida väljundfaili kahele esimesele reale sisendfaili read ja kirjutada kolmandale reale vähim sellise tehte number, mida ei saa pakutud järjekorras teha (sest tehte mõni argumentidest on veel välja arvutamata).
**Märkus.** Antud sõnastuses pole ülesanne veel üheselt määratud. Tuleks veel fikseerida, kas võrdse prioriteediga tehteid (liitmine-lahutamine, korrutamine-jagamine) peab osaavaldises tegema vasakult paremale või lubame neid teha suvalises järjekorras (nii, et see arvutuse tulemust ei muuda).

16. Aditiivsete arvavaldiste võrdlemine
**Antud:** avaldised $A_1 \oplus \ldots \oplus A_k$ ja $B_1 \oplus \ldots \oplus B_m$, kus tehtemärkideks on plussid ja miinused, avaldiste liikmed on naturaalarvud.
**Leida:** kehtiv võrdus või võrratus $A_1 \oplus \ldots \oplus A_k \lesseqgtr B_1 \oplus \ldots \oplus B_m$, kus sisendiks olnud avaldiste vahel on märk <, = või >.

17. Aritmeetilise progressiooni summa tsükliga
**Antud:** aritmeetilise progressiooni esimene liige, vahe ja liikmete arv.
**Leida:** progressionni summa, liites tsükliga kokku kõik jada liikmed.

18. Võrrand $x \oplus a = b$
**Antud:** a) võrrand $x + a = b$ kus a ja b on reaalarvud,
b) võrrand $x - a = b$, kus a ja b on reaalarvud.
**Leida:** võrrandi lahend.

19. Aditiivne võrrand
**Antud:** võrrand $X_1 \oplus \ldots \oplus X_k = 0$, milles märkide kohal on plussid ja miinused, mõnede $i$ väärtuste korral on $X_i$ tundmatu $x$, ülejäänud liikmed $X_i$ on naturaalarvud.
a) võrrandil on üks lahend,
b) erijuhud on võimalikud.
**Leida:** võrrandi lahend.

20. Tunniplaan
**Antud:** tundide algus (tund, minut), tunni pikkus, tundide arv ja vahetunni pikkus (minutites, täisarvulised).
**Leida:** tunniplaan.

21. Nulliga võrduva avaldise moodustamine
Panna arvude vahele plussid ja miinused, nii et saadud avaldise väärtus oleks 0.
**Antud:** sõne pikkusega kuni 80 sümbolit, milles on tühikuga eraldatud naturaalarvude jada (kuni 20 arvu).
**Panna** arvude vahele märgid + ja – nii, et saadud avaldise väärtus oleks 0. Kui see pole võimalik, siis anda vastus Ei SAA.

### 3.5.2 Naturaalarvude korrutamine ja jagamine. Tegurid, algarvud

Naturaalarvude korrutamise ja jagamise teema all vaadeldakse väga tähtsaid arvuteooria mõisteid ja algoritme, mida iga haritud inimene peaks tundma. Muuhulgas kasutatakse neid ka krüptograafias. Viiendas klassis sisse toodud mõisted tuleks siin hoolikalt definitsioone jälgides läbi mõelda ja ka läbi programmeerida.

1. Korrutamistehete tulemuse arvutamine

   **Antud:** positiivsetest naturaalarvudest korrutamismärkide abil moodustatud avaldis (korrutis).

   **Leida:** avaldise väärtus.

   **Sisend.** Sisendfaili ainsal real on kuni kolmekohalistest positiivsetest naturaalarvudest tärni abil moodustatud korrutis, kus on 1-4 liiget.

   **Väljund.** Kirjutada väljundfaili esimesele reale sisendrida, võrdusmärk ja korrutise väärtus.

2. Jagamistehte tulemuse arvutamine

   **Antud:** kaks positiivset naturaalarvu.

   **Leida:** Nende jagatis ja jääk.

   **Sisend.** Sisendfaili ainsal real on kaks kooloniga eraldatud kuni 6-kohalist positiivset naturaalarvu.

   **Väljund.** Kopeerida väljundfaili esimesele reale sisend, mille järele kirjutada võrdusmärk, jagatise täisosa ja kui esimene arv ei jagu teisega täpselt, siis lisada ' JÄÄK …'.

3. Kirjalik korrutamine

   **Antud:** kaks kuni 5-kohalist naturaalarvu.

   **Esitada** nende korrutamine tavalise kirjaliku algoritmi järgi, paigutades kõik arvud õigetele positsioonidele.

   **Sisend.** Sisendfaili ainsal real on kaks tühikuga eraldatud kuni 5-kohalist positiivset naturaalarvu.

   **Väljund.** Kirjaliku korrutamisalgoritmi rakendamise tulemus, kus esimene tegur on korrutatud teise teguri järjestikuste kümnendkohtadega ja

   1) tegurid, osakorrutised ja tulemus on eraldatud miinustest koosnevate lõpptulemuse pikkuste joontega,

   2) arvud asuvad õigetel positsioonidel üksteise all, lõpptulemus algab esimesest positsioonist,

   3) nulliga võrduvad korrutised on ära jäetud,

   4) korrutamis- ja liitmismärki pole väljastatud.

   **Näide.**

   ```
      3456
      2030
   -------
    10368x
   6912x
   -------
   7015680
   ```

   **Märkus:** Enamus õpikutest jätab näites $x$-ga tähistatud kohad tühjaks, aga mõni võib soovitada sinna nullide kirjutamist. Valida üks stiil ja realiseerida see programmis.

   **Soovitus.** Väljundfaili vaatamiseks võiks kasutada fonti, kus kõikide sümbolite laiused on võrdsed (näiteks Courier).

   Lahendus. (V-NK-3) Kirjalik korrutamine.py

4. Kirjalik jagamine
**Antud:** kaks kuni 8-kohalist naturaalarvu.
**Esitada** nende jagamine tavalise kirjaliku algoritmi järgi, paigutades kõik arvud õigetele positsioonidele.
**Sisend.** Sisendfaili ainsal real on kaks kooloniga eraldatud kuni 8-kohalist positiivset naturaalarvu, kus esimene arv jagub teisega täpselt.
**Väljund.** Kirjaliku jagamisalgoritmi rakendamise tulemus, kus esimene arv on jagatud teise arvuga ja
1) iga lahutamistehte all on miinustest koosnev vähendatavast ja lahutatavast pikima pikkune joon (esimese tehte juures lahutatava pikkune),
2) arvud asuvad õigetel positsioonidel üksteise all, jagatav algab esimesest positsioonist,
3) näidatakse ka jagatisse tekkivatele nullidele vastavaid lahutamistehteid,
4) tehtemärkidest kirjutada väljundfaili ainult esimesel real olev jagamismärk (koolon).
**Näide.** Kui sisendfailis on rida
1603882:782
siis väljundfaili tuleb kirjutada

```
1603882:782=2051
1564
----
  398
    0
  ---
  3988
  3910
  ----
    782
    782
    ---
```

**Variant.** Võib vaadelda ka ülesannet, kus esimene arv ei jagu teisega. Sellisel juhul on viimase lahutamise tulemus positiivne ja väljundfaili esimese rea lõppu tuleb lisada JÄÄK …'.

5. Korrutamise ja jagamisega arvavaldised
**Antud:** avaldis, mis on moodustatud naturaalarvudest
a) korrutamise ja jagamise märke kasutades,
b) jagamise märke ja sulge kasutades,
c) korrutamise ja jagamise märke ning sulge kasutades.
**Leida:** avaldise väärtus.

6. Nelja tehtega arvavaldised
**Antud:** avaldis, mis on moodustatud naturaalarvudest
a) nelja tehtemärki kasutades,
b) nelja tehtemärki ja sulge kasutades,
**Leida:** avaldise väärtus.

7. Võrrandi $ax = b$ lahendamine
**Antud:** Võrrand kujul $ax = b$, kus $a$ ja $b$ on täisarvud.

**Leida:** võrrandi lahend.
**Sisend.** Sisendfaili ainsal real on võrrand $ax = b$, kus $a$ ja $b$ on täisarvud ja $x$ tundmatu, kusjuures arv $b$ jagub $a$-ga täpselt.
**Väljund.** Kirjutada väljundfaili esimesele reale võrrandi lahend.

8. Koondamine lineaarses hulkliikmes
**Antud:** Liikmetest kujul $ax$ ja $b$ plusside ja miinuste abil moodustatud hulkliige, kus $a$ ja $b$ on positiivsed täisarvud, $x$ muutuja.
**Koondada** hulkliige.
**Sisend.** Sisendfaili ainsal real pikkusega kuni 80 sümbolit on hulkliige $X_1 \oplus \ldots \oplus X_k$, kus üksliikmed $X_i$ on kujul $ax$ või $b$, kus $a$ ja $b$ on positiivsed kuni 3-kohalised täisarvud ja $x$ on muutuja.
**Väljund.** Kirjutada väljundfaili esimesele reale koondamise tulemus kaksliikmena $cx + d$, kus $c$ ja $d$ on täisarvud (tulemus võib olla ka üksliige).

9. Lineaarse hulkliikme korrutamine arvuga
**Antud:** avaldis $a(X_1 \oplus \ldots \oplus X_k)$, kus $a$ on täisarv, $\oplus$ tähistab plussi või miinust, üks liikmetest $X_i$ on täisarv ja ülejäänud on lineaarsed üksliikmed kujul $b_i x_i$, kus $b_i$ on naturaalarvud ja $x_i$ on üksteisest erinevad muutujad.
**Avada** selles avaldises sulud.

10. Tärnide/tähtede asendamine kirjaliku korrutamise tehtes vm tuntud struktuuris
**Antud:** matemaatika kursusest tuntud struktuur, näiteks kirjaliku korrutamise esitus, milles osa kümnendkohti on asendatud tärnide või tundmatuid tähistavate tähtedega.
**Leida:** kõik struktuuri kuuluvad elemendid.
**Näide**. Pane tärnide asemele numbrid nii, et kirjaliku korrutamise tehe oleks õige

```
 **
 52
---
 **
**
---
*7*
```

**Variandid.** a) erinevate tähtede asemel peavad/ei pea olema erinevad numbrid,
b) leida üks lahend/kõik lahendid

11. Tehtemärkide panemine arvude vahele
**Antud:** sõne $A = B$, kus $A$ ja $B$ koosnevad tühikutega eraldatud naturaalarvudest.
**Asendada** tühikud tehtemärkidega … nii, et võrdus oleks tõene.
**Näide:** Kui sisend on 9 8 7= 6 54 3 2 ning lubatud tehted on +, - ja *, siis väljundiks sobib
9*8-7= 6+ 54+3+2
**Variandid:** a) erinevad lubatud tehtemärkide hulgad,
b) tühiku võib ka ära jätta, st moodustada kahest naabrist ühe arvu,
c) sisendis ei ole võrdusmärki, ka see tuleb panna ühe tühiku kohale.
**Töökiirusest.** Loomuliku algoritmi tööaeg sõltub valitavate tehtemärkide arvust

eksponentsiaalselt.

12. Tegurite leidmine
**Def.** Antud positiivse naturaalarvu $a$ **teguriks** nimetatakse iga naturaalarvu, millega arv $a$ jagub.
Näiteks arvu 7 tegurid on 1 ja 7,
arvu $144$ tegurid on $1, 2, 3, 4, 6, 8, 9, 12, 16, 18, 24, 36, 48, 72$ ja $144$.
Arvu 0 tegurite hulk ei paku huvi, sest 0 jagub iga naturaalarvuga.
On selge, et igal positiivsel naturaalarvul on vähemalt kaks tegurit: 1 ja arv ise.
**Antud:** positiivne naturaalarv $a$.
**Leida:** arvu $a$ kõik tegurid.
**Sisend.** Sisendfaili ainsas reas on kuni 10-kohaline positiivne naturaalarv $a$.
**Väljund.** Kirjutada väljundfaili esimesele reale tühikuga eraldatuna arvu $a$ kõik tegurid kasvavas järjekorras.
**Arvutusalgoritmist.** On selge, et arvu $a$ tegurite leidmiseks võiksime läbi kontrollida kõik naturaalarvud $1\ \dots\ a$. Suure $a$ korral või vajaduse korral seda ülesannet paljude $a$ väärtuste jaoks lahendada võiks selline moodus töötada liiga aeglaselt. Tegelikult saame tegurid kätte ka jaguvuse kontrolliga ainult kuni ruutjuureni $a$ -st, sest kõik arvu $a$ ruutjuurest suuremad tegurid saame leida kujul $a/b$, kus $b$ on ruutjuurest väiksem tegur.
**Lahendus.** (V-NK-12) tegurite leidmine.py

13. Algarvulisuse kontroll
**Def.** Naturaalarvu $a$, millel on kaks tegurit (arv 1 ja arv $a$ ise), nimetatakse **algarvuks**.
Kui arvul on rohkem kui kaks tegurit, siis nimetatakse seda arvu **kordarvuks**.
Algarvud on 2, 3, 5, 7, 11 jne, kordarvud on 4, 6, 8, 9, 10 jne. Arvul 1 on ainult üks tegur, ta pole algarv ega kordarv. Vanakreeka matemaatik Eukleides näitas, et algarve on lõpmata palju.
Algarvudel on väga suur tähtsus klassikalises arvuteoorias ja praegusel ajal järjest suuremat tähtsust omandavas krüptograafias.
Ka algarvulisuse kontrolliks piisab tegurite otsimisest kuni ruutjuureni arvust $a$.
**Antud:** positiivne naturaalarv $a$.
**Kontrollida**, kas $a$ on algarv.
**Sisend.** Sisendfaili ainsal real on positiivne naturaalarv $a$.
**Väljund.** Kui $a$ on algarv, siis kirjutada väljundfaili esimesele reale arv $a$, millele on lisatud ON ALGARV. Kui $a$ pole algarv, siis lisada $a$ järele võrdusmärk ja mingi esitus korrutisena.

14. Algtegurite ja nende astmete leidmine (kanooniline kuju)
**Aritmeetika põhiteoreem** ütleb, et iga 1-st suurem naturaalarv on algarv või esitatav algarvude korrutisena ning see esitus on ühene kuni tegurite järjekorra täpsuseni. See võimaldab esitada arvu $a$ nn **kanoonilisel kujul** algarvude astmete korrutisena
$a = p_1^{n_1} \cdot p_2^{n_2} \cdot \dots \cdot p_k^{n_k}$, kus $k \geq 1$ ja $p_1 < p_2 < \cdots < p_k$
Näiteks kui tähistame kanoonilises kujus korrutamist tärniga ja astendamist kahe tärniga, siis
$7 = 7 ** 1$ ja $144 = 2 ** 4 * 3 ** 3$.
**Antud:** naturaalarv $a$.
**Leida:** arvu $a$ kanooniline esitus algarvude astmete korrutisena.
**Sisend.** Sisendfaili ainsal real on naturaalarv $a \geq 2$.
**Väljund.** Kirjutada väljundfaili esimesele reale arv $a$, võrdusmärk ja arvu $a$ esitus kanoonilisel

kujul, kus korrutamist tähistab tärn ja astendamist kaks tärni. Ühega võrduvad astendajad jätta kirjutamata.

15. Eratosthenese sõel
    Eratosthenese sõel võimaldab leida naturaalarvude $2, \dots, M$ hulgast kõik algarvud.
    **Algoritmi idee.** Kasutame arvude $2, \dots, M$ jaoks massiivi, mille $i$-s element näitab, kas arv $i$ on alles märkimata, algarv või kordarv. Töö alguses pole ükski arvudest $2, \dots, M$ märgitud algarvuks ega kordarvuks.
    Pärast etappi $t$ on märgitud $t$ vähimat algarvu ja kõigi nende algarvude kordsed on märgitud kordarvudeks. Etapil 1 märgitakse algarvuks 2 ja kordarvudeks tema kordsed $4, 6, 8, \dots$ . Igal järgmisel etapil märgitakse algarvuks vähim arv, mis pole veel algarvuks ega kordarvuks märgitud (sest ta ei jagu ühegi temast väiksema algarvuga), ja kordarvudeks kõik tema kordsed. Algoritm lõpetab töö, kui järjekordse etapi alguses ei leita enam märkimata arve.
    **Antud:** naturaalarv $M > 1$
    **Leida:** kõik algarvud, mis pole suuremad kui $M$.
    **Lahendus.** (V-NK-15) Eratosthenes.py

16. Ühistegurid ja ühiskordsed. Eukleidese algoritm
    Arvu $c$ nimetatakse arvude $a$ ja $b$ **ühisteguriks**, kui ta on nii $a$ kui ka $b$ tegur. Analoogiliselt nimetatakse arvu $c$ arvude $a$ ja $b$ **ühiskordseks**, kui ta on nii $a$ kui ka $b$ kordne. Vaadeldakse ka rohkem kui kahe arvu ühistegureid ja ühiskordseid (näiteks mitme murru liitmise juures). Olulise tähtsusega on **suurim ühistegur**, mida lühendatult tähistatakse $SÜT(a, b)$ ja tihti ka ingliskeelse lühendiga $gcd(a, b)$ ning vähim ühiskordne, mida tähistatakse $VÜK(a, b)$ja $lcm(a, b)$.
    Kuuenda klassi matemaatikast teame, et
    1) Murdu saab taandada parajasti nende arvudega, mis on lugeja ja nimetaja ühistegurid.
    2) Taandumatu murru saame, kui taandame murdu lugeja ja nimetaja suurima ühisteguriga.
    3) Murdude liitmisel ja lahutamisel on sobiv võtta ühiseks nimetajaks kõigi operandide nimetajate vähim ühiskordne.
    On selge, et arvude suurima ühisteguri leidmiseks piisab kõige väiksema arvu tegurite uurimisest. Nende leidmiseks piisab otsimisest kuni väiksema arvu ruutjuureni ja vastavast jagamisest. Samuti saab suurima ühisteguri kokku panna arvude $a$ ja $b$ lahutustest algarvulisteks teguriteks, võttes iga algarvu astmeks minimaalse kahest astmest.
    Aga juba Eukleidese „Elementides" on esitatud väiksema töömahuga nn. *Eukleidese algoritm*.
    **Eukleidese algoritm** naturaalarvude $a$ ja $b$ suurima ühisteguri leidmiseks:
    *Etapp t (t=1, 2, …):*
    *1) Kui b=0, siis lõpetada etapp.*
    *2) Leida jagamisel a:b tekkiv jääk r.*
    *3) a:=b; b:=r.*
    *Väljund: muutuja a väärtus.*
    On selge, et jäägid vähenevad iga etapiga ja mingil etapil saame $b = 0$ ning algoritm lõpetab töö. Eukleidese algoritmi korrektsuse ja etappide arvu hinnangu $1 + log_2 a + log_2 b$ tõestuse eesti keeles võib leida näiteks loengukonspektist [3] . Sealt võib leida ka teisi kasulikke teoreeme SÜT ja VÜK kohta.
    Vähima ühiskordse saab kokku panna arvude $a$ ja $b$ lahutustest algarvulisteks teguriteks, võttes iga algarvu astmeks maksimaalse kahest astendajast. Aga SÜT ja VÜK saab leida ka teineteise

kaudu, kasutades üsna ilmset võrdust $SÜT(a,b) \cdot VÜK(a,b) = ab$, milles võib kergesti veenduda algarvulisteks teguriteks lahutusi vaadates.

**Antud:** kaks positiivset naturaalarvu $a$ ja $b$.
**Leida** Eukleidese algoritmi abil arvude $a$ ja $b$ suurim ühistegur.

17. SÜT, VÜK
**Antud:** naturaalarv $n > 1$ ja naturaalarvud $a_1, \dots, a_n > 0$ .
**Leida:** a) arvude $a_1, \dots, a_n$ suurim ühistegur,
b) arvude $a_1, \dots, a_n$ vähim ühiskordne.
**Lahendus.** (V-NK-17) SÜT-VÜK-2.py

### 3.5.3 Geomeetria konstruktsioonide algoritmid

**Joonestamisalgoritmid tasandil**

Meie järgnevates ülesannetes koostame programme, mis realiseerivad põhikoolis õpetatavaid geomeetriliste objektide konstruktsiooni baasalgoritme või mis kasutavad neid konstruktsioone uute ülesannete lahendamiseks. Ülesannete eesmärk on realiseerida programmina koolimatemaatika sirkli ja joonlaua algoritme. Kui konstruktsioonialgoritm kasutab abijooni, siis tuleb ka need joonisele kanda, kuigi mõne ülesande puhul on võimalik otsitavate punktide koordinaate algebraliselt välja arvutada. Erinevates programmeerimiskeeltes on programmeeritud mitmesuguseid joonestuskäskude teeke, mis võimaldavad mõnesid meie ülesandeid ka ühe käsuga täita (mitte ainult paari esimest, vaid ka keerulisemaid). Õppimise eesmärgil peaksite programmides kasutama ainult keele standardprotseduure. Selles jaotises tuleks kõigepealt lahendada ülesanded 1-9, mis annavad vajalikud tööriistad elementaarsete alamülesannete lahendamiseks. Lahendused võiks vormistada alamprogrammidena (funktsioonidena), et neid saaks kasutada keerulisemate ülesannete lahendamisel.
Lõikude pikkusi mõõdame selle jaotise ülesannetes pikselites, nurki kraadides. Joonise jaoks võiks ekraanil valida teatud suurusega akna. Programm võiks joonistel tähistada punktid väikese täidetud ringiga, mille juurde on kirjutatud punkti tähis ja koordinaatidega punkti korral ka koordinaadid, näiteks $\bullet A(120{,}150)$. Sisendandmete kohta võib eeldada, et punktid paigutuvad üksteisest piisavalt kaugele ja tähised ei kata üksteist. Aga võib ka panna programmi hoolitsema selle eest, et näiteks tähised paigutatakse sobivalt. Programm võiks kontrollida, et algandmetena antud objektid ja nõutav konstruktsioon paigutuvad joonise jaoks valitud aknasse. Kui see pole nii, siis võiks programm anda vastava teate.
Kolmnurkade kohta lepime kokku, et küljed a, b, ja c on vastavalt tippude $A$, $B$ ja $C$ vastasküljed.
Kui konstruktsioone tehakse sirkli ja joonlauaga, siis võime sirkli teraviku või joonlaua serva paigutada mingisse joonisel olevasse punkti ja kasutada seda punkti ringjoone keskpunktina või punktina joonestataval sirgel ilma tema koordinaate teadmata. Programmi kirjutades tuleb meil aga ringi või sirge joonestamiseks kasutada funktsioone, mille sisendparameetrite seas on ka joont määravate punktide koordinaadid. Need tuleb siis eelnevalt välja arvutada.
Konstruktsioonides kasutatakse peamiselt punkte, mis on mingite sirkli ja joonlaua abil konstrueeritavate joonte (sirgete või ringjoonte) lõikepunktid. Kooligeomeetria annab meile sirge ja ringjoone võrrandid. Lõikepunktide koordinaadid on siis joonte võrranditest tekkivate võrrandisüsteemide lahendid. Kahe sirge lõikepunkti saame kahe tundmatuga lineaarse

võrrandisüsteemi lahendamisel (ülesanne 6), mida õpetatakse juba põhikoolis. Gümnaasiumi õpikutest võib leida rida joonte võrrandeid. Suuruste arvutamise valemeid ja algoritme on toodud õpiku „Võistlusprogrammeerimine“ 2. osa [7] 12-ndas peatükis „Arvutusgeomeetria“: punkti koordinaatide muutumine tasandi pöördel, kahe punkti vaheline kaugus, paralleelsuse kontroll, sirgete lõikepunkti leidmine, hulknurga ümbermõõt ja pindala, punkti kuulumine hulknurka.

Selles peatükis lisame veel valemid/algoritmid sirge ja ringjoone ning kahe ringjoone lõikepunktide leidmiseks (ülesanded 7 ja 9). Jaotise lõpus on ülesandeid, millel ei ole kooligeomeetrias õpitavat lahendusalgoritmi. Nad nõuavad lahendajalt oma lahendusalgoritmi konstrueerimist.

Joonestamise teema juures soovitame ülesannete lahendamisel koostatud programmid vormistada funktsioonidena, mida saab järgmiste ülesannete lahendamisel kasutada.

1. Lõigu joonestamine punktide järgi
   **Antud:** kahe erineva punkti $A$ ja $B$ koordinaadid.
   **Joonestada:** lõik $AB$.
   **Lahendus.** (V-G-1) Lõik kahe punktiga.py

2. Kiire joonestamine punktide järgi
   **Antud:** kahe erineva punkti $A$ ja $B$ koordinaadid, kus $A$ on kiire otspunkt ja $B$ kiirel asuv punkt.
   **Joonestada:** kiir, mille otspunkt on $A$ ja mis läbib punkti $B$ (joonestatud osa kiirest peaks ulatuma akna servani).

3. Sirge joonestamine punktide järgi
   **Antud:** kahe erineva punkti $A$ ja $B$ koordinaadid.
   **Leida** neid punkte läbiva sirge võrrand.
   **Joonestada:** selline osa punkte $A$ ja $B$ läbivast sirgest, mis ulatub mõlemalt poolt joonistamisakna servani.
   **Lahendus.** (V-G-3) Sirge kahe punktiga.py

4. Nurga joonestamine
   **Antud:** nurga tipu $A$ ja ühe haaral asuva punkti $B$ koordinaadid ning nurga suurus kraadides $x$.
   **Joonestada** nurk, mille üheks haaraks on kiir $AB$ ja suurus on $x$ kraadi.

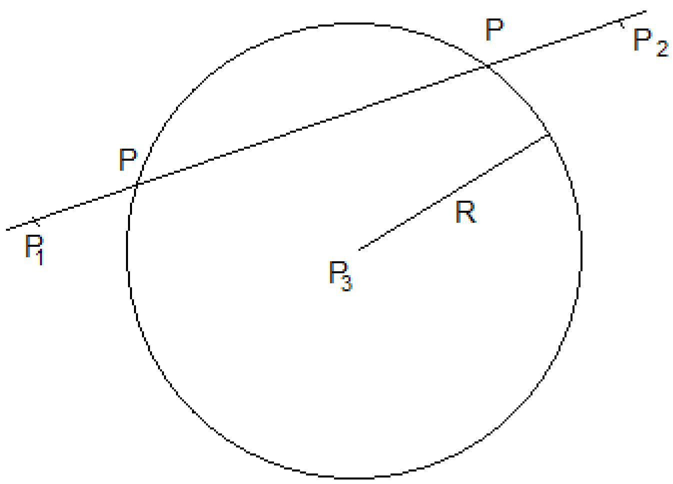
5. Ringjoone joonestamine keskpunkti ja ringjoonel oleva punkti/raadiuse järgi
   **Antud:** ringi keskpunkti $O$ koordinaadid ja
   a) ringjoonel asuva punkti $A$ koordinaadid,
   b) ringi raadius $R$.
   **Joonestada** ringjoon, mille keskpunkt on $O$ ja mis läbib punkti $A$ /on raadiusega $R$.
   **Kommentaar.** Keerulisemate konstruktsioonide osana kasutamiseks võib olla vajalik joonestada ka ringjooni, millest ainult teatud kaar(ed) asub (asuvad) joonestamisaknas. Selliste ringjoonte joonestamine võib erinevates programmeerimiskeeltes olla erineva raskusega ülesanne.

**Lahendus.** (V-G-5) Ringjoon.py

6. Kahe sirge lõikepunkti leidmine
   **Antud:** kahe erineva punkti koordinaadid sirgel $s$ ja kahe erineva punkti koordinaadid sirgel $t$
   a) $s$ ja $t$ on lõikuvad sirged,
   b) üldjuht.
   **Leida** sirgete s ja t lõikepunkti koordinaadid, joonistada sirged ja kirjutada koordinaadid lõikepunkti juurde. Erijuhtudel anda teade.

7. Sirge ja ringjoone lõikepunktide/puutepunkti leidmine
   Näitame kõigepealt, kuidas saab leida lõikepunktide/puutepunkti koordinaadid. Lahendamiseks vajalikku informatsiooni võib leida ka Wolframi MathWorldist http://mathworld.wolfram.com/Circle-LineIntersection.html või Stackexchange veebilehelt https://math.stackexchange.com/questions/228841/how-do-i-calculate-the-intersections-of-a-straight-line-and-a-circle. Wikipedia annab lahenduse juhuks, kui ringi keskpunkt on koordinaatide alguspunktis:
   https://en.wikipedia.org/wiki/Intersection_%28Euclidean_geometry%29.

   Olgu sirge määratud kahe erineva punktiga $P_1(x_1, y_1)$ ja $P_2(x_2, y_2)$, ringjoone keskpunkt olgu $P_3(x_3, y_3)$ ja raadius $R$.



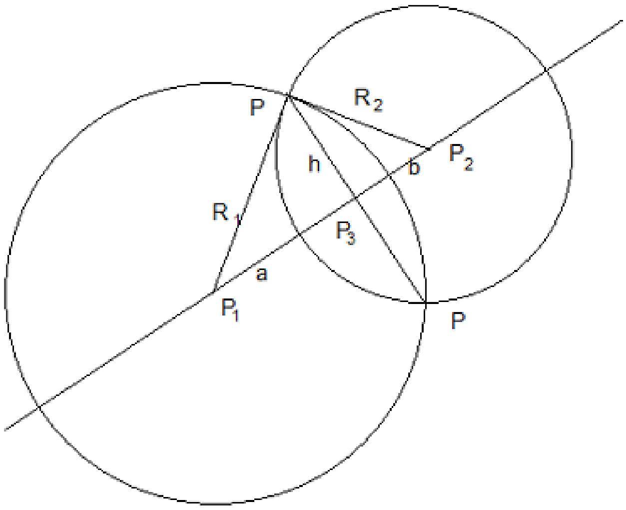

Siis sirgel asuvat punkti $P(x, y)$ kirjeldavad parameetrilised võrrandid
$x = x_1 + u(x_2 - x_1)$,
$y = y_1 + u(y_2 - y_1)$.
Ringjoone punktide kohta kehtib ringjoone võrrand
$(x - x_3)^2 + (y - y_3)^2 = R^2$.
Asendades ringjoone võrrandisse esimesest kahest võrdusest x ja y, saame u leidmiseks ruutvõrrandi kujul $au^2 + bu + c = 0$, kus
$a = (x_2 - x_1)^2 + (y_2 - y_1)^2$,
$b = 2((x_2 - x_1)(x_1 - x_3) + y_2 - y_1)(y_1 - y_3))$,
$c = {x_3}^2 + {y_3}^2 + {x_1}^2 + {y_1}^2 - 2(x_3x_1 + y_3y_1) - R^2$
Ringjoone ja sirge ühiste punktide arv sõltub ruutvõrrandi diskriminandist $D = b^2 - 4ac$.
Kui D<0, siis ühiseid punkte ei ole.

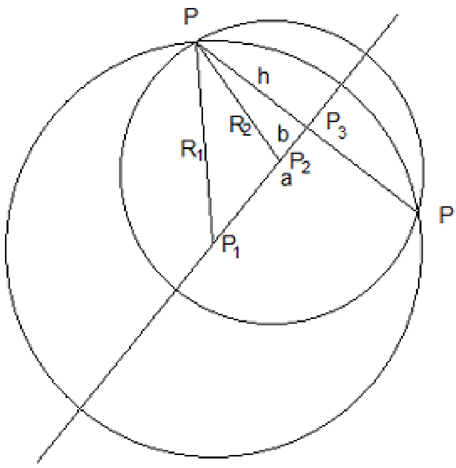
Kui D=0, siis sirge puutub ringjoont.
Kui D>0, siis sirge lõikab ringjoont kahes punktis.

**Antud:** sirget määrava kahe erineva punkti $P_1(x_1, y_1)$ ja $P_2(x_2, y_2)$koordinaadid, ringjoone keskpunkti $P_3(x_3, y_3)$koordinaadid ja ringjoone raadius $R$.
**Leida** sirge ja ringjoone lõikepunktide/puutepunkti koordinaadid (kui lõikuvad/puutuvad) ning kirjutada lõikepunktide/puutepunkti juurde koordinaadid.
**Joonistada** sirge ja ringjoon sobivas mõõtkavas.

8. Antud pikkusega lõigu joonestamine kiirele
**Antud:** Lõigu üks otspunkt $A$, kiirel asuv teine punkt $C$ ja lõigu pikkus $l$.
**Joonestada** kiirele punkt $B$, nii et lõigu $AB$ pikkus oleks $l$.

9. Kahe ringjoone lõikepunktide/puutepunkti leidmine
Kahe ringjoone lõikepunkti leidmine ei kuulu gümnaasiumi matemaatika ainekavasse ja ei ole vaatluse all ka paljudes ülikooli analüütilise geomeetria kursustes (kuigi ei ole tehniliselt keeruline). Näitame siin, kuidas saab lõikepunkti(de) koordinaate avaldada. Erijuhul, kus ringjoonte keskpunktideks on (0;0) ja x-telje punkt (d;0) on vastav tuletus esitatud Wolframi MathWorldis http://mathworld.wolfram.com/Circle-CircleIntersection.html. Üldjuhu jaoks võib tuletuse leida veebilehtedelt http://paulbourke.net/geometry/circlesphere/ ja https://stackoverflow.com/questions/3349125/circle-circle-intersection-points, kus küll on vaadeldud ainult alltoodud joonisel vasakul olevat juhtu.
Olgu ringjoonte keskpunktid $P_1(x_1, y_1)$ ja $P_2(x_2, y_2)$ ning ringjoonte raadiused $R_1$ ja $R_2$.
Keskpunktide vahekauguse ja raadiuste kaudu saab üsna lihtsalt välja eraldada juhud, kus
1) ringjooned ei lõiku, sest vahekaugus on liiga suur,
2) ringjooned ei lõiku, sest üks ringjoon asub teise sees,
3) ringjooned ühtivad.
Vaatleme peajuhtu, kus leidub kaks lõikepunkti või üks puutepunkt. Tähistagu $P(u, v)$ otsitavat lõikepunkti ning $P_3(x_3, y_3)$ lõikepunktide vahelise lõigu keskpunkti (puutumise korral nad kõik ühtivad). Vastavalt väiksema ringi keskpunkti $P_2$ asukohale punkti $P_3$ suhtes tuleb meil vaadelda kahte joonistel toodud juhtu.



Olgu a kaugus esimese ringi keskpunktist $P_1$punktini $P_3$ , b kaugus teisest keskpunktist $P_2$ punktini $P_3$ , h kaugus punkti(de)st P punktini $P_3$ ja d ringjoonte keskpunktide vaheline kaugus .

Vaatleme kõigepealt vasakpoolsel joonisel kujutatud juhtu. Kolmnurkadele $P_1PP_3$ ja $P_2PP_3$ Pythagorase teoreemi rakendades saame

$a^2 + h^2 = {R_1}^2$ ja $b^2 + h^2 = {R_2}^2$.

Lahutades esimesest võrrandist teise, saame

${R_1}^2 - {R_2}^2 = a^2 - b^2$.

Kui liidame selle võrrandi mõlemale poolele $d^2$ Ja arvestame, et praegu vaadeldaval juhul kehtib $d = a + b$, on tulemuseks võrrand

${R_1}^2 - {R_2}^2 + d^2 = a^2 - b^2 + a^2 + 2ab + b^2$,

mille paremat poolt võime edasi teisendada nii:

$= 2a^2 + 2ab = 2ad$.

Seega saame võrrandist avaldada kauguse $a$ algselt teadaolevate suuruste kaudu:

$a = \frac{{R_1}^2 - {R_2}^2 + d^2}{2d}$ .

Puutumise korral, kus $d = R_1 + R_2$, saame sellest valemist $a = R_1$.

Uurides kahe lõikepunkti juhtu edasi, saame oma arutluse esimesest võrrandist

$h^2 = {R_1}^2 - a^2$,

mis võimaldab arvutada h, sest paremal olevad suurused on juba teada.

Lõpuks saame leida ka punkti $P_3$koordinaadid. Kui tähistame punktide $P_1$, $P_2$ ja $P_3$ kohavektorid nendesamade tähtedega, siis ilmselt kehtib

$P_3 = P_1 + \frac{a}{d}(P_2 - P_1)$.

Siit saame:

$u = x_1 \pm h(y_2 - y_1)$/d,

$v = y_1 \mp h(x_2 - x_1)$/d.

Parempoolsele joonisele vastaval juhul kehtib $d = a - b$. Kui aga teeme uuesti läbi äsjatoodud arutluse, selgub, et kauguste a ja h ning koordinaatide u ja v jaoks saame samad avaldised, samuti kehtib puutumise korral $a = R_1$

Nüüd oleme valmis ülesande lahendamiseks.

**Antud:** kahe ringjoone keskpunktid $(x1, y1)$, $(x2, y2)$ ja ringjoonte raadiused $R_1$ ja $R_2$.

**Leida** ringjoonte lõikepunktide koordinaadid (kui lõikuvad)/puutepunkti koordinaadid (kui puutuvad)/negat vastus/otsus ühtimise kohta. **Joonistada** ringjooned ning kirjutada lõikepunktide/puutepunkti juurde koordinaadid.

**Soovitus.**

Lõikepunti koordinaatide valem koos tuletusega on erijuhuks, kus ringjoonte keskpunktideks on (0;0) ja x-telje punkt (d;0) esitatud Wolframi MathWorldis http://mathworld.wolfram.com/Circle-CircleIntersection.html. Sellest saab leida ka valemi üldjuhu jaoks.

10. Ristsirge joonestamine

**Antud:** sirgel $s$ asuvate erinevate punktide $A$ ja $B$ koordinaadid ning punkti $C$ koordinaadid.

**Joonestada:** punkti $C$ läbiv sirge, mis on risti sirgega s (koos konstruktsiooni abijoontega).

**Kommentaar.** Standardse konstruktsiooniga tekib lisaks antud punktile $C$ kaks ristsirge punkti. Nende punktide saamiseks joonistatavate ringjoonte raadius tuleks valida nii, et vähemalt üks punkt asuks joonistamisaknas.

11. Paralleelsirge joonestamine
**Antud:** sirgel s asuvate erinevate punktide $A$ ja $B$ koordinaadid ning väljaspool sirget s asuva punkti $C$ koordinaadid
**Joonestada:** punkti $C$ läbiv sirge, mis on paralleelne sirgega s (koos konstruktsiooni abijoontega)

12. Nurga teisaldamine
**Antud:** etteantud nurka $ABC$ moodustavate punktide $A$, $B$ ja $C$ koordinaadid ning teisealdatud nurga ühe haara punktide X ja Y koordinaadid, kus Y on nurga tipp.
**Konstrueerida** nurk $ABC$ ja kiir YX, punkti Z leidmiseks vajalikud ringjooned nurgal $ABC$ ja kiire $YX$ juures, arvutada punkti Z koordinaadid, joonistada kiir punktist Y läbi Z, nii et nurgad $ABC$ ja $XYZ$ on võrdsed.
**NB!** Enne programmeerimist on vaja otsustada, kas soovitakse säilitada ka orientatsiooni.

13. Kolmnurga joonestamine kolme külje järgi
**Antud:** kolm positiivset reaalarvu $a, b$ ja $c$.
**Joonestada:** kolmnurk külgedega $a, b$ ja $c$ (kui leidub), koos konstruktsiooni abijoontega.
**Soovitus.** Ühe külje võiks paigutada standardsel viisil , näiteks otspunktidega $(0, 0)$ ja $(a, 0)$.

14. Kolmnurga joonestamine kahe külje ja nurga järgi
**Antud:** positiivsed reaalarvud $a, b$ ja $C$
**Joonestada:** kolmnurk, millel on küljed pikkustega a ja b ning nende vahel nurk $C$ kraadi (koos konstruktsiooni abijoontega)
**Soovitus.** Ühe külje võiks paigutada standardsel viisil , näiteks otspunktidega $(0, 0)$ ja $(a, 0)$.

15. Kolmnurga joonestamine külje ja kahe nurga järgi
**Antud:** positiivsed reaalarvud $a$, $B$ ja $C$
**Joonestada:** kolmnurk, millel on külg pikkusega $a$ ning tema lähisnurgad on $B$ ja $C$ kraadi (kui leidub), koos konstruktsiooni abijoontega.
**Soovitus.** Ühe külje võiks paigutada standardsel viisil , näiteks otspunktidega $(0, 0)$ ja $(a, 0)$.

16. Ruudu joonestamine
**Antud:** ruudu ühe külje otspunktide $A$ ja $B$ koordinaadid (täisarvud).
**Joonestada** ruut, mille üks külg on $AB$ (koos konstruktsiooni abijoontega).

17. Ristküliku joonestamine
**Antud:** ristküliku külgede pikkused $a$ ja $b$.
**Joonestada** ristkülik külje pikkustega $a$ ja $b$.
**Soovitus.** Ühe külje võiks paigutada standardsel viisil , näiteks otspunktidega $(0, 0)$ ja $(a, 0)$.

18. Rombi joonestamine
**Antud:** rombi külje pikkus $a$ ja üks rombi nurk $A$ (kraadides).

**Joonestada** vastav romb.

19. Lõigu poolitamine
    **Antud:** kahe erineva punkti $A$ ja $B$ koordinaadid
    **Joonestada:** lõik $AB$ ja tema keskristsirge (koos konstruktsiooni abijoontega)

20. Nurga poolitamine
    **Antud:** kolme mitte ühel sirgel asuva punkti $A$, $B$ ja $C$ koordinaadid
    **Joonestada:** lõigud $A\ B$ ja $BC$ ning nurga $ABC$ poolitaja (koos konstruktsiooni abijoontega)

21. Ringi puutuja joonestamine
    **Antud:** ringi keskpunkti $O(x, y)$ koordinaadid ja raadius $R$ ning punkti $A(a, b)$ koordinaadid.
    **Joonestada** ring ja punkt $A$.
    **Leida** punkti $A$ läbivate ringi puutujate puutepunktid ringjoonega ja joonestada need puutujad (koos konstruktsiooni abijoontega).

22. Kolmnurga ümberringjoone joonestamine
    **Antud:** kolmnurga tippude $A$, $B$ ja $C$ koordinaadid.
    **Joonestada** kolmnurga tipud ja küljed ning ümberringjoone keskpunkti saamiseks vajalikud abijooned, arvutades eelnevalt vajalike punktide koordinaadid. Leida ümberringjoone keskpunkti koordinaadid ja joonestada ümberringjoon.

23. Kolmnurga siseringjoone joonestamine
    **Antud:** kolmnurga tippude $A$, $B$ ja $C$ koordinaadid.
    **Joonestada** kolmnurga tipud ja küljed ning siseringjoone keskpunkti saamiseks vajalikud abijooned, arvutades eelnevalt vajalike punktide koordinaadid. Leida siseringjoone keskpunkti koordinaadid ja raadius või puutepunkt kolmnurga mingi küljega ning joonestada siseringjoon.

### 3.5.4 Nurgad kella osutite vahel

Uurime seoseid kellaaja ja osutite asendi (nurk vertikaalasendi suhtes kraadides) vahel. Lubatud kellaaegadeks loeme 00.00 kuni 24.00.

1. Minutiosuti liikumine ja kellaaeg
   **Antud:** kellaaeg (tunnid, minutid) ja nurk $A$ (kraadides), mille võrra pöördub minutiosuti.
   **Leida:** Kui palju näitab kell, kui minutiosuti on pöördunud $A$ kraadi võrra.

2. Tunniosuti liikumine ja kellaaeg
   **Antud:** kellaaeg (tunnid, minutid) ja nurk $A$ (kraadides), mille võrra pöördub tunniosuti.
   **Leida:** Kui palju näitab kell, kui tunniosuti on pöördunud $A$ kraadi võrra.

3. Nurk osutite vahel
   **Antud:** kellaaeg (tunnid, minutid).
   **Leida:** nurk tunni- ja minutiosuti vahel (kraadides).

4. Kellaaja leidmine osutite asendi järgi
   **Antud:** nurgad tunniosuti ja minutiosuti ning vertikaalasendi vahel (kraadides).
   **Leida:** Kui sellisele asendile vastab kellaaeg, siis leida see. Kui ei, siis väljastada VÕIMATU ASEND.

### 3.5.5 Statistika mõisted

1. Aritmeetilise keskmise leidmine
   **Antud:** klassi õpilaste arv $n$ $(n \leq 40)$ja kõigi õpilaste pikkused.
   **Leida:** õpilaste pikkuste aritmeetiline keskmine.

2. Osahulkade aritmeetilised keskmised
   Antud: klassi õpilaste arv $n$ $(0 < n \leq 40)$ ja iga õpilase kohta tema sugu (M/N) ning pikkus (cm).
   Leida: klassi kõigi õpilaste, poiste ja tüdrukute pikkuste aritmeetilised keskmised.

3. Aritmeetiline keskmine sagedustabeli järgi
   Antud: klassi õpilaste pikkuste sagedustabel (pikkus, sellise pikkusega õpilaste arv).
   Leida: õpilaste pikkuste aritmeetiline keskmine.

4. Moodi arvutamine
   **Antud:** klassi õpilaste arv $n$ $(n \leq 40)$ja kõigi õpilaste pikkused.
   **Leida:** õpilaste pikkuste mood.

5. Mediaani arvutamine
   **Antud:** klassi õpilaste arv $n$ $(n \leq 40)$ja kõigi õpilaste pikkused.
   **Leida:** õpilaste pikkuste mediaan.

## 3.6 VI klass

### 3.6.1 Harilikud murrud

Järgnevas paneme murde kirja kujul x/y, kus x on murru lugeja, y murru nimetaja ja sümbol '/' tähistab murrujoont. Programmeerimisel võib eeldada, et sisendandmeteks olevate murdude lugejad ja nimetajad on mõistliku suurusega, näiteks ülimalt kolmekohalised.
Eesti koolides nõutakse tavaliselt hariliku murru kujul esitatud lõppvastuse teisendamist segaarvuks, mille murdosa on taandumatu murd (ka siis, kui algandmed neid tingimusi ei rahulda). Nulliga võrduvad segaarvu osad jäetakse kirjutamata:
1) kui murdosa on null, siis antakse vastuseks ainult täisosa (see võib olla ka 0),
2) kui täisosa on null, siis antakse vastuseks ainult murdosa.

Kui ülesande tekst ei nõua vastuse esitamist mingil muul kujul (näiteks osalise taandamise, laiendamise jms ülesanded), siis eeldame vastuse esitamist ülalkirjeldatud kujul.

1. Liht- ja liigmurrud
   **Def.** Kui hariliku murru lugeja on nimetajast väiksem, siis nimetatakse murdu **lihtmurruks**. Kui lugeja on nimetajaga võrdne või sellest suurem, siis nimetatakse murdu **liigmurruks**.
   **Antud:** murd kujul $x/y$, kus $x$ ja $y$ on positiivsed naturaalarvud.
   **Otsustada**, kas $x/y$ on lihtmurd või liigmurd.
2. Murru laiendamine
   **Jagatise põhiomadus.** Jagatis ei muutu, kui korrutada või jagada jagatavat ja jagajat ühe ja sama nullist erineva arvuga.
   **Def.** Murru lugeja ja nimetaja korrutamist ühe ja sama positiivse naturaalarvuga $a$ nimetatakse murru **laiendamiseks** ja arvu $a$ nimetatakse sealjuures **laiendajaks**.
   Üksiku murru laiendamine laiendajaga 1 ei oma mõtet, aga kui näiteks murdude liitmise juures laiendatakse liidetavaid murde ühise nimetajani, siis võib mõne liidetava laiendaja olla ka 1.

   **Antud:** murd kujul $x/y$ (kus x ja y on positiivsed naturaalarvud) ja laiendaja (positiivne naturaalarv a).
   **Laiendada** murdu $x/y$ laiendajaga $a$.

3. Laiendamine antud nimetajani
   **Antud:** murd kujul $x/y$ ja nõutav uus nimetaja (positiivne naturaalarv $d$, mis on $y$ kordne).
   **Leida:** nimetajani $d$ laiendatud murd.
   **Lahendus.** (VI-M-3) Laiendamine nimetajani.py

4. Murru taandamine antud arvuga
   **Def.** Murru lugeja ja nimetaja jagamist ühe ja sama ühest suurema teguriga nimetatakse murru **taandamiseks**.
   Kui murru lugejal ja nimetajal puudub 1-st erinev ühistegur, siis murdu taandada ei saa ja sellist murdu nimetatakse **taandumatuks murruks**.
   Taandumatu murd on antud murru lihtsaim esitus.
   Murdu saab taandada
   a) järk-järgult, valides lugeja ja nimetaja ühistegureid seni, kuni jõutakse taandumatu murruni,
   b) korraga, jagades lugeja ja nimetaja nende suurima ühisteguriga.

**Antud:** Murd $x/y$ ja naturaalarv $a \geq 2$, mis on $x$ ja $y$ ühisteguriks.
**Taandada** murd $x/y$ arvuga $a$.

5. Taandamine suurima ühisteguriga
   **Antud:** Murd $x/y$.
   **Taandada** murd $x/y$ arvude $x$ ja $y$ suurima ühisteguriga.

6. Osa meetrist, ruutmeetrist, kuupmeetrist
   Missuguse osa meetrist/ruutmeetrist/kuupmeetrist moodustab x …?
   **Antud**: suurus (pikkus/pindala/ruumala) koos ühikuga ja tervikuks olev ühik.
   **Leida:** osamäär hariliku murruna.
   Näide. Kui sisend on: 4 cm, m,
   siis väljund võib olla 4/100 või 1/25.

7. Tundmatu $x$ leidmine võrdest
   **Antud:** võrdus $a/b = c/d$, kus üks lugeja või nimetaja on muutuja $x$, ülejäänud on positiivsed naturaalarvud.
   **Leida:** selline $x$ väärtus, mis muudab võrduse tõeseks.

8. Ühenimeliseks teisendamine
   Murdude võrdlemiseks, liitmiseks ja lahutamiseks on tavaliselt otstarbekohane teisendada nad laiendamise teel ühenimeliseks. Ühiseks nimetajaks saab olla arv, mis on laiendatavate murdude nimetajate ühiskordne. Et uued murrud oleks võimalikult lihtsad, tuleks ühiseks nimetajaks valida laiendatavate murdude nimetajate vähim ühiskordne.
   **Antud**: kaks või rohkem murdu kujul $x/y$.
   **Teisendada** murrud ühenimelisteks, valides ühiseks nimetajaks algsete murdude nimetajate vähima ühiskordse.
   **Lahendus.** (VI-1-8)Yhisele nimetajale.py

9. Millised murdude võrdused on tõesed?
   **Antud:** murdude võrdus kujul $a/b = c/d$, kus $a, b, c$, ja $d$ on naturaalarvud, $b, d > 0$.
   **Leida:** kas võrdus kehtib?

10. Harilike murdude võrdlemine
    **Antud:** kaks murdu kujul $x/y$
    **Leida:** Panna murdude vahele sobiv võrdlusmärk <, = või >.
    a. Ühenimelised,
    b. Laiendamine,
    c. Võrdse lugejaga,
    d. Taanduvad murrud.

11. Murru teisendamine segaarvuks
    **Antud:** murd kujul $x/y$, kus $x$ ja $y$ on naturaalarvud, $y > 0$.

**Teisendada** murd segaarvuks (lihtmurru korral jääb murd samaks).

12. Segaarvu teisendamine liigmurruks
    **Antud:** segaarv $x\frac{y}{z}$, kus $x, y$ ja $z$ on naturaalarvud ja $x, z > 0$. Kui $y = 0$, siis võib murdosa ka puududa.
    **Teisendada** segaarv liigmurruks.

13. Ühenimeliste murdude liitmine ja lahutamine
    **Antud:** kahe ühenimelise murru summa või vahe kujul $a/c \oplus b/c$, kus $\oplus$ on pluss või miinus.
    **Leida** nende murdude summa/vahe.

14. Ühenimeliste segaarvude liitmine
    **Antud:** kaks või rohkem ühenimelist positiivset segaarvu .
    **Leida** nende segaarvude summa.

15. Ühenimeliste segaarvude lahutamine
    **Antud:** kaks ühenimelist positiivset segaarvu.
    **Leida** nende segaarvude vahe .

16. Erinimeliste murdude liitmine ja lahutamine
    **Antud:** kahe murru summa või vahe kujul $a/b \oplus c/d$, kus $a, b, c$ ja $d$ on naturaalarvud, $b, d > 0$ ja $\oplus$ on pluss või miinus
    **Leida** nende murdude summa/vahe
    **Lahendus.** (VI-1-16)Murdude liitmine-lahutamine.py

17. Lahendada murdudega aditiivne võrrand
    **Antud:** võrrand $x \oplus y = z$, kus üks liikmetest on tundmatu ja teised on murrud või segaarvud.
    **Leida** võrrandi lahend (murd või segaarv).

18. Kümnendmurd harilikuks murruks
    **Antud:** kümnendmurd.
    **Teisendada** kümnendmurd harilikuks murruks/segaarvuks.

19. Harilik murd kümnendmurruks (ka perioodilised kümnendmurrud)
    Hariliku murru saab esitada lõpliku kümnendmurruna parajasti siis, kui murru nimetajal (taandatud kujul) ei ole muid algtegureid peale 2 ja 5, ülejäänud juhtudel saame perioodilise kümnendmurru. Perioodi leidmiseks võime kõigepealt eraldada jagatise täisosa. Murdosa jagamisel jagajaga vaatleme samme, kus jagatava nullist erinevad kümnendkohad on juba läbitud, st jagatise järgmise kümnendkoha leidmiseks lisatakse eelmise sammu jäägi lõppu 0. Tekkivad jäägid on väiksemad kui jagaja, st mingil sammul peab tekkima jääk, mis on juba esinenud. Sealt alates hakkavad jagatises kümnendkohad korduma – oleme leidnud perioodi. Kui tekib jääk 0, siis oleme saanud lõpliku kümnendmurru, sest korduva nulli võime kümnendmurru lõpust ära jätta.
    **Antud:** harilik murd $a/b$, kus $a$ ja $b$ on naturaalarvud, $b > 0$.

**Teisendada** harilik murd kümnendmurruks (lõplikuks või perioodiliseks).

20. Hariliku murru kümnendlähend
**Antud:** harilik murd ja nõutav lähendi täpsus järguühikuna.
**Leida:** murru kümnendlähend.

21. Avaldise väärtuse arvutamine (kümnendmurd ja harilik murd)
**Antud:** avaldis $x \oplus y$, kus $\oplus$ on pluss või miinus ning üks alamavaldistest $x$ ja $y$ on harilik murd, teine lõplik kümnendmurd.
**Leida:** Kui harilik murd avaldub lõpliku kümnendmurruna, siis teisendada ta kümnendmurruks ja leida vastus kümnendmurruna. Vastasel korral leida vastus hariliku murruna.

22. a) Moodusta numbritest 4, 5 ja 6 kõikvõimalikud kolmekohalised arvud (iga number on igas arvus üks kord). Leia saadud arvude suurim ühistegur.
b) Kirjuta kõik sellised kolmekohalised arvud, mille kümneliste number on 5 ja mis jaguvad 18-ga.

23. Harilike murdude korrutamine
**Antud:** avaldis $a/b * c/d$, kus $a, b, c$, ja $d$ on naturaalarvud, $b, d > 0$ ja $*$ mõistame kui korrutamismärki.
**Leida:** murdude $a/b$ ja $c/d$ korrutis.

24. Korrutamine, kus on ka segaarve
**Antud:** avaldis $A * B$, kus $A$ ja $B$ on harilikud murrud, naturaalarvud või segaarvud
**Leida:** korrutis $A * B$.

25. Osa leidmine arvust
**Antud:** arv $x$ ja osamäär hariliku murruna $a/b$.
**Leida:** $a/b$ osa arvust $x$.

26. Pöördarvu leidmine
**Antud:** arv $x$ (kümnendmurd, harilik murd või segaarv).
**Leida:** arvu $x$ pöördarv hariliku murru või segaarvu kujul.

27. Harilike murdude jagamine
**Antud:** avaldis $a/b : c/d$, kus $a, b, c$, ja $d$ on naturaalarvud, $b, d > 0$ ja : mõistame kui jagamismärki.
**Leida:** murdude $a/b$ ja $c/d$ jagatis.

28. Arvu leidmine osa ja osamäära järgi
**Antud:** osamäär kujul $a/b$, kus a ja b on naturaalarvud, b>0, ja sellele osamäärale vastav osa (arv x).
**Leida:** arv, millest $c$ moodustab $a/b$ osa.

29. Murdudega avaldise väärtus
**Antud:** avaldis, mis on moodustatud täisarvudest, harilikest murdudest ja segaarvudest

a) plusside ja miinuste abil,
b) plusside, miinuste ja sulgude abil,
c) plusside, miinuste, korrutamis- ja jagamismärkide abil,
d) plusside, miinuste, korrutamis- ja jagamismärkide ja sulgude abil.
**Leida:** avaldise väärtus.

### 3.6.2 Tasandi teisendused. Geomeetrilised kujundid

Vaatleme tasandi teisendusi $F((x, y)) = (x_1, y_1)$ , mis teisendavad tasandi kõigi punktide hulga üksüheselt tasandi kõigi punktide hulgaks. Meid huvitavad siin nn isomeetrilised teisendused, mis säilitavad kõigi punktipaaride korral nende punktide vahelise kauguse. Sellised teisendused teisendavad iga kujundi temaga kongruentseks kujundiks.

Teisendust $F$ nimetatakse **lükkeks** vektori (a, b) võrra, kui iga punkti (x,y) korral
$F((x, y)) = (x + a, y + b)$.
Teisendust $F$ nimetatakse **pöördeks** ümber punkti (c, d) nurga $\alpha$ võrra, kui iga punkti (x,y) korral
$F((x, y)) = ((x - c)\cos(\alpha) - (y - d)\sin(\alpha) + c, (y - d)\sin(\alpha) + (x - c)\cos(\alpha) + d)$.
Kui pööre tehakse nullpunkti (0,0) ümber, siis kehtib
$F((x, y)) = (x\cos(\alpha) - y\sin(\alpha), y\sin(\alpha) + x\cos(\alpha))$.
Kaht punkti A ja A’ nimetatakse **sümmeetrilisteks** sirge s suhtes, kui
A=A’ ja nad asuvad sirgel s või sirge AA’ on risti sirgega s, A ja A’ on teine teisel pool sirget s ja A ja A’ on võrdsetel kaugustel sirgest s.
Teisendust F nimetatakse **peegelduseks** sirge suhtes, kui ta kujutab iga punkti (x, y) temaga sirge s suhtes sümmeetriliseks punktiks.
Kujundit (punktihulka) nimetatakse **sümmeetriliseks** sirge s suhtes, kui ta sisaldab koos iga punktiga ka selle punktiga sirge s suhtes sümmeetrilist punkti. Sirget s nimetatakse antud kujundi sümmeetriateljeks.
Lüke, pööre ja peegeldus on **isomeetrilised** teisendused, st säilitavad suvaliste kahe punkti korral nendevahelise kauguse. Seetõttu teiseneb nende teisenduste korral iga kujund endaga kongruentseks kujundiks.

Ülesannetes vaatleme selliste kujundite teisendusi, mis koosnevad lõplikust arvust koordinaattasandi punktidest. On selge, et isomeetrilistel teisendustel selliste kujundite punktide arv ei muutu.

1. Lükke parameetrite leidmine
   **Antud:** naturaalarv $n > 0$ ja kaks $n$-elemendilist erinevate punktide loendit $(x_1, y_1), \dots, (x_n, y_n)$ ja $(u_1, v_1), \dots, (u_n, v_n)$.
   **Leida:** sellised arvud $a$ ja $b$, et esimese loendi punktide poolt moodustatud kujundi lükkel vektori $(a, b)$ võrra on tulemuseks teise loendi poolt moodustatud kujund.
   **Märkus.** Punktid ei tarvitse loendites olla sellises järjekorras, et $(x_i + a, y_i + b) = (u_i, v_i)$.
   **Sisend.** Sisendfaili esimesel real on naturaalarv $n$ $(0 < n \leq 10)$, teisel ja kolmandal real komadega eraldatud punktide loendid $(x_1, y_1), \dots, (x_n, y_n)$ ja $(u_1, v_1), \dots, (u_n, v_n)$, kus punktid kummaski loendis ei kordu ja punktide koordinaadid on täisarvud absoluutväärtusega mitte üle 10.
   **Väljund.**

Kui sobivaid väärtusi a ja b ei leidu, siis väljastada EI SAA.
Kui lüke leidub, siis kirjutada väljundfaili esimesele reale tühikuga eraldatuna lükke parameetrid $a$ ja $b$.

**Näide**

```
Sisend:
3
(1,10),(-5,7),(0,0)
(15,-37),(10,-30),(16,-27)
Väljund:
15 -37
```

**Kuidas lahendada?** Vaatleme lükkeid, mis viivad punkti $(x_1\ ,y_1)$ punktiks $(u_i\ ,v_i)$ (kus $1 \leq i \leq n$). Lüke sobib parajasti siis, kui iga j korral (kus $2 \leq j \leq n$) teisendatakse punkt $(x_j\ ,$ $y_j)$ mingiks punktiks $(u_k\ ,v_k)$. Algoritmi saab kiirendada ja kõigi $k$ väärtuste läbivaatuse ära jätta, kui sorteerime eelnevalt punktid mõlemas loendis (näiteks kõigepealt $x$-koordinaadi ja siis $y$-koordinaadi järgi) sest lüke säilitab sellise järjestuse.

2. Pöördenurga leidmine

**Antud:** naturaalarv $n > 0$ ja kaks $n$-elemendilist täisarvuliste koordinaatidega erinevate punktide loendit $(x_1\ ,y_1), \dots, (x_n\ ,y_n)$ ja $(u_1\ ,v_1), \dots, (u_n\ ,v_n)$.

**Leida:** selline nurk $\alpha$, et esimese punktide loendi poolt moodustatud kujund teiseneb pöördel nurga $\alpha$ võrra ümber koordinaatide alguspunkti teise punktide loendi poolt moodustatud kujundiks.

**Märkus.** Punktid ei tarvitse loendites olla sellises järjekorras, et $F((x_i\ ,y_i))= (u_i\ ,v_i)$.

**Sisend.** Sisendfaili esimesel real on naturaalarv $n$ $(0 < n \leq 10)$, teisel ja kolmandal real komadega eraldatud punktide loendid $(x_1\ ,y_1), \dots, (x_n\ ,y_n)$ ja $(u_1\ ,v_1), \dots, (u_n\ ,v_n)$, kus punktid kummaski loendis ei kordu ja punktide koordinaadid on täisarvud absoluutväärtusega mitte üle 1000.

**Väljund.**

Kui sobivat pöördenurka $\alpha$ ei leidu, siis väljastada EI SAA.
Kui pöördenurk $\alpha$ leidub, siis kirjutada väljundfaili esimesele reale üks selline nurk $\alpha$ (kraadides, täpsusega 0,01 kraadi).

**Näide.** Kui sisend on

1
(20;0)
(16;12)

Siis minimaalse nurgaga väljund on

36,9

3. Peegelduse leidmine

**Antud:** naturaalarv $n > 0$ ja kaks $n$-elemendilist täisarvuliste koordinaatidega punktide loendit $(x_1\ ,y_1), \dots, (x_n\ ,y_n)$ ja $(u_1\ ,v_1), \dots, (u_n\ ,v_n)$.

**Leida:** selline sirge $s$, et esimese punktide loendi poolt moodustatud kujund teiseneb peegeldusel sirge $s$ suhtes teise punktide loendi poolt moodustatud kujundiks.

**Märkus.** Punktid ei tarvitse loendites olla sellises järjekorras, et $F((x_i\ ,y_i))= (u_i\ ,v_i)$.

**Sisend.** Sisendfaili esimesel real on naturaalarv $n$ $(0 < n \leq 10)$, teisel ja kolmandal real komadega eraldatud punktide loendid $(x_1, y_1), \dots, (x_n, y_n)$ ja $(u_1, v_1), \dots, (u_n, v_n)$, kus punktid kummaski loendis ei kordu ja punktide koordinaadid on täisarvud absoluutväärtusega mitte üle 10.

**Väljund.**

Kui sobivat peegeldust ei leidu, siis väljastada EI SAA.

Kui sobiv peegeldus leidub, siis kirjutada väljundfaili esimesele reale sirget $s$ määravate kahe punkti koordinaadid.

**Kuidas lahendada.** Kui peegelduse korral $F((x, y)) = (u, v)$, siis peegeldus toimub punkti $(x, y)$ ja tema kujutise $(u, v)$ vahelise lõigu keskristsirge suhtes. Punkt $(x_1, y_1)$ peab kujutuma mingiks punktiks $(u_k, v_k)$. Võime otsida sellist $k$ väärtust, mille puhul iga punkti $(x_i, y_i)$ peegeldus punktide $(x_1, y_1)$ ja $(u_k, v_k)$ järgi saadud telje suhtes kuulub punktide loendisse $(u_1, v_1), \dots, (u_n, v_n)$. NB! Programmeerimisel soovitame pöörata tähelepanu ka piirjuhtudele.

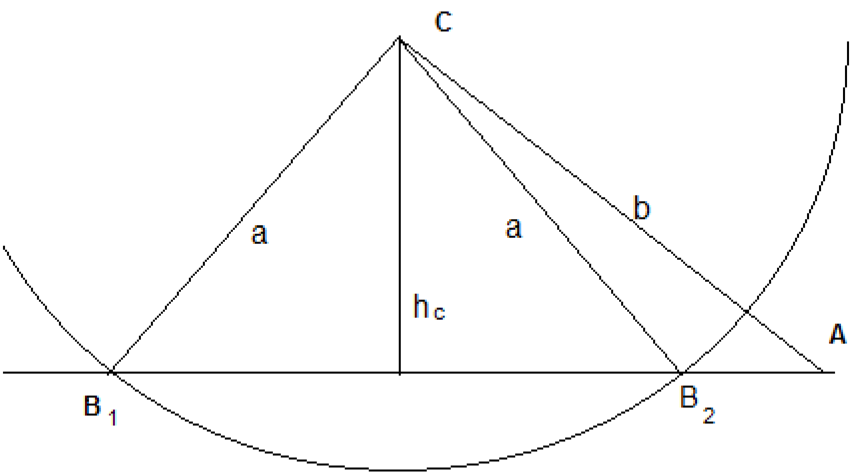
4. Kujundi telgsümmeetrilisuse kontroll
   **Antud:** naturaalarv $n > 0$ ja $n$-elemendiline punktide loend $(x_1, y_1), \dots, (x_n, y_n)$.
   **Leida:** Kas punktide poolt moodustatav kujund on telgsümmeetriline? Kui on, siis väljastada kaks punkti, mis määravad sellise sirge $s$, et punktide loendi poolt moodustatud kujund teiseneb peegeldusel sirge $s$ suhtes iseendaks. Vastasel korral väljastada EI OLE.

5. Arvu $\pi$ alumise ja ülemise hinnangu täpsus
   Jagame arvu $\pi$ alumise ja ülemise hinnangu saamiseks ühikringi kesknurga $n$ võrdseks osaks ja vaatleme võrdhaarseid kolmnurki, mille külgedeks on ringi keskpunktist lähtuvad kiired ja ringjoone kõõlud/puutujad. Alumise hinnanguna kasutame sellise kõõlhulknurga ümbermõõtu, mille tipud on ringjoonel. Ülemise hinnangu jaoks kasutame sellise hulknurga ümbermõõtu, mille servad on ringjoone puutujad.
   **Antud:** reaalarv $a > 0$.
   **Leida:** mitmeks tuleb $360°$ jagada, et $\pi$ alumine hinnang erineks ülemisest hinnangust vähem kui $a$.

6. Tingimused kolmnurga külgedele ja nurkadele
   **Antud:**
   a) kolm positiivset reaalarvu (kolmnurga külgede pikkused) $a, b$ ja $c$,
   b) kaks positiivset reaalarvu (kolmnurga kaks nurka kraadides) A ja $B$,
   c) kolm positiivset reaalarvu a, b ja $C$ (kolmnurga kaks külge ja nurk kraadides nende vahel),
   d) kolm positiivset reaalarvu a, B ja C (kolmnurga külg ja tema kaks lähisnurka kraadides).
   **Leida:** Kas leidub selline kolmnurk?

7. Kolmnurga määramine kolme elemendiga
   Kolmnnurkade juures vaadeldakse mitmeid erinevaid suurusi: küljed $a$, $b$ ja $c$; nurgad $A$, $B$ ja $C$; külgedele $a$, $b$ ja $c$ joonistatud kõrgused $h_a$, $h_b$ ja $h_c$, pindala $S$, siseringjoone raadius $r$, ümberringjoone raadius $R$.
   Kuuenda klassi geomeetriast teame mitmeid ülaltoodud suuruste kolmikuid, mille (lubatud) väärtused määravad kolmnurga üheselt: kolm külge, külg ja tema lähisnurgad jne. Aga nende suuruste mõnede kolmikute jaoks leiduvad ka väärtused, millele ei vasta ükski kolmnurk. Näiteks ei saa kolm külge olla 1, 2 ja 5. Mõne kolmiku korral avaldub üks kolmest suurusest ülejäänute

kaudu ja siis võib antud elementidega kolmnurki leiduda lõpmata palju. Näiteks kolm nurka (summaga 180 kraadi) ei määra kolmnurga joonelementide pikkusi; elemendid a, ha ja S ei määra nurki ja ülejäänud joonelemente. Küljed a ja b ning kõrgus $h_c$ võivad aga lubada kahte erinevat kolmnurka $AB_1C$ ja $AB_2C$:



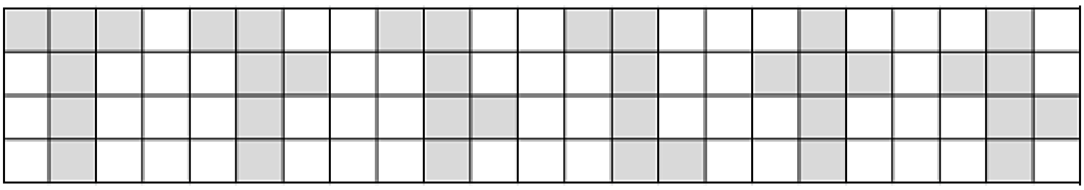

Vaatleme siin ülesandeid, milles on fikseeritud mingi eespool kirjeldatud 12-elemendilise suuruste hulga alamhulk H, selle alamhulga kolme suuruse väärtused on antud ja leida tuleb alamhulga H ülejäänud suuruste väärtused.
Antud hulka kuuluvate kõigi suuruste leidmise ülesanne võib programmide koostamisel tihti ette tulla. Programm peab olema koostatud nii, et lahendus leitaks hulka H kuuluvate suvaliste võimalike algsuuruste ja nende suvaliste väärtuste korral. Tüüpiline on ka see moment, et lahendamiseks vajalikku kõigi valemite komplekti ei anta programmeerijale ette, vaid tal tuleb ise otsustada, milliseid seoseid on vaja, ja leida vastavad valemid. Selle ülesannete kogu kasutajale on niisuguse ülesandega tegelemine kasulik ka siis, kui ei osata kõiki vajalikke trigonomeetrilisi seoseid välja otsida. Aga oleks vaja, et peetakse järge, mida Teie programm juba oskab ja mis seal veel puudub.
Põhikooli matemaatikakursuses on üsna väike osa otsitavate suuruste leidmiseks vajalikest valemitest (sisenurkade summa, pindala valem kõrguse kaudu). Suur osa otsitavate suuruste leidmiseks vajalikest valemitest kuulub gümnaasiumi matemaatikakursusesse (siinus- ja koosinusteoreem, Heroni valem jms), aga võib tarvis minna ka valemeid, mida meie gümnaasiumi matemaatikakursuses ei vaadelda. Mitmed programmi kirjutamiseks kasulikud valemid võib leida Wikipediast: [https://en.wikipedia.org/wiki/Triangle](https://en.wikipedia.org/wiki/Triangle)

Konkreetne programmeerimisülesanne on määratud vaadeldava 12-elemendilise suuruste hulga kolmeelemendilise alamhulgaga $H$.
**Sisend.** Sisendfaili kolmel real on kolm võrdust kujul <suuruse nimi>=<suuruse väärtus>, kus antud suurused kuuluvad hulka $H$ ja väärtused on reaalarvud.
**Väljund.** Kirjutada väljundfaili esimesele reale otsus antud tingimustele vastavate kolmnurkade hulga kohta: EI OLE/ÜKS/KAKS/LÕPMATA PALJU. Järgmistele ridadele kirjutada võrdused nende hulka H kuuluvate suuruste kohta, mida ei ole sisendis, aga mille väärtused on sisendiga määratud. Kui leidub kaks varianti, siis kirjutada teisele reale 'Variant 1:', millele järgnevad esimese variandi võrdused, edasi 'Variant 2:' ja selle variandi võrdused.

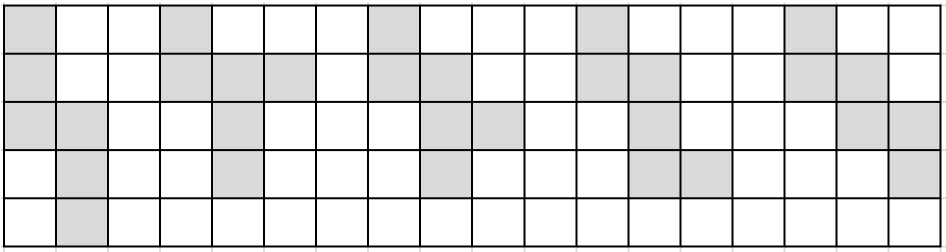
**Näide.** Olgu vaadeldav suuruste hulk $H = \{a, b, c, A, B, C, ha, hb, hc, S\}$ sisendiks:
a=4
b=5
hc=2
Siis väljundiks võivad olla järgmised read, kus suurused on esitatud arvutamise järjekorras (sulgudes on selgitus arvutamisel kasutatud seose kohta, mis ei kuulu nõutud väljundisse):

KAKS
Variant 1:
$B$ =30 (=arcsin(hc/a))
$A$ =23,58 (siinusteoreem)
$C$ =126,42 (sisenurkade summa)
c=8,05 (siinusteoreem)
S=8,05 (pindala valem)
ha=4,02 (pindala valem)
hb=3,22 (pindala valem)
Variant 2:
$B$ =150
$A$ =23,58
$C$ =6,42
c=1,12
S=1,12
ha=0,56
hb=0,45
**Variante hulga H jaoks:**
**a)** $H = \{a, b, c, \ S\}$,
b) $H = \{a, b, c, A, B, C\}$,
c) $H = \{a, b, c, ha, hb, hc, S\}$

8. Kuubi pinnalaotuse äratundmine
**Antud:** kuus sellist märgitud ruutu $5 \times 5$ ruudust, mis moodustavad sidusa kujundi.
**Leida:** Kas ruudud moodustavad kuubi pinnalaotuse?
**Kuidas lahendada.** Millised kujundid sobivad kuubi pinnalaotuseks? Siintoodud kujundid on esitatud ka Wikipedias: https://en.wikipedia.org/wiki/Cube. On selge, et pinnalaotus ei saa sisaldada $5 \times 1$ ühikruuduga rida ega ruutu, mis sisaldab $2 \times 2$ ühikruutu. Kui kujund sisaldab $4 \times 1$ ruudust koosnevat rida, siis ülejäänud kaks ruutu peavad asuma teine teisel pool seda rida ja iga selline kujund (6 erinevat kuju) on kuubi pinnalaotus.



Kui kujundi pikim rida sisaldab $3 \times 1$ ruutu, siis on kaks võimalust teise $3 \times 1$ rea jaoks, kus ei teki $4 \times 1$ rida või $2 \times 2$ ruutu ega kuubi tahu kahekordset katet. Ühe 3x1 rea puhul peab ülejäänud ruutudest üks olema ühel pool rida ja teised kaks teisel pool. Kahest ühel pool olevat ruudust üks peab ulatuma üle $3 \times 1$ rea otsa (muidu tekiks $2 \times 2$ ruut või kuubi ühe tahu kahekordne katmine). Teisel pool $3 \times 1$ rida asuv ruut võib olla ükskõik millisel 3-st positsioonist. Kui kujundis pole ka $3 \times 1$ rida, siis sellise kujundi moodustamiseks on ainult üks võimalus ja saame kontrollida, et ka see kujund sobib kuubi pinnalaotuseks.



Üks võimalik viis ülesande lahendamiseks on kontrollida, kas kujund on üks toodud 11-st (muidugi arvestades pööret ja peegeldust). Aga on võimalik ka ülesande otsene lahendus, kus etteantud kujundit püütakse laotada kuubi tahkudele.

9. Kolmnurga liik (…-nurkne, külgede järgi)
   **Antud:** kolm reaalarvu $a, b$ ja $c$.
   **Leida:** Kas ja millise kolmnurga moodustavad küljed $a, b$ ja $c$?
   (ei moodusta kolmnurka/teravnurkse/täisnurkse/nürinurkse)?

10. Kolmnurga liik (….-külgne, nurkade järgi)
    **Antud:** kaks naturaalarvu A ja $B$ (0<A, $B$ <180).
    **Leida:** Milliste külgedega on kolmnurk, millel on nurgad A kraadi ja $B$ kraadi?
    (sellist ei olegi/isekülgne/võrdhaarne mittevõrdkülgne /võrdkülgne)?

11. Kolmnurga liik (…-nurkne, nurkade järgi)
    **Antud:** kaks naturaalarvu A ja $B$ (0<A, $B$ <180).
    **Leida:** Milliste nurkadega on kolmnurk, millel leiduvad nurgad A kraadi ja $B$ kraadi?
    (sellist ei olegi/teravnurkne/täisnurkne /nürinurkne)?

12. Kolmnurgad koordinaattasandil
    **Antud:** kolme punkti koordinaadid tasandil (täisarvulised).
    **Leida:** a) kolmnurga liik külgede järgi,
    b) kolmnurga liik nurkade järgi.

13. Nelinurgad koordinaattasandil
    **Antud:** nelja punkti koordinaadid tasandil (täisarvulised).
    **Leida:** kas need punktid moodustavad (mingis järjekorras vaadelduna)
    a) mittekõdunud nelinurga,
    b) trapetsi,
    c) rööpküliku,
    d) ristküliku,
    e) rombi,
    f) ruudu,
    g) romboidi,
    h) kumera nelinurga?
    Näide. Kujund tippudega (0; 4), (1; 0), (4; 5), (5; 1) on mittekõdunud nelinurk, rööpkülik, ristkülik,

romb, romboid, kumer nelinurk.
**Soovitus.** Vaadata üle vastavate kujundite definitsioonid (leidub ebaregulaarne definitsioon).

14. Ruutude arv
**Antud:** naturaalarv $n$ $(4 \leq n \geq 1000)$ ja $n$ erineva tasandipunkti koordinaadid (täisarvud).
**Leida:** mitu erinevat ruutu saab moodustada, mille tippudeks on mingisugused neli sisendis antud punkti?
**Soovitus.** Püüdke vältida neljakordset tsüklit, millega vaadatakse läbi kõik punktide nelikud.

15. Ristkülikute arv
**Antud:** naturaalarv $n$ $(4 \leq n \geq 100)$ ja $n$ erineva tasandipunkti koordinaadid (täisarvud).
**Leida:** mitu erinevat ristkülikut saab moodustada, mille tippudeks on mingisugused neli sisendis antud punkti?
**Soovitus.** Püüdke vältida neljakordset tsüklit, millega vaadatakse läbi kõik punktide nelikud.

16. Rombide arv
**Antud:** naturaalarv $n$ $(4 \leq n \geq 100)$ ja $n$ erineva tasandipunkti koordinaadid (täisarvud).
**Leida:** mitu erinevat rombi saab moodustada, mille tippudeks on mingisugused neli sisendis antud punkti?
**Soovitus.** Püüdke vältida neljakordset tsüklit, millega vaadatakse läbi kõik punktide nelikud.

17. Rööpkülikute arv
**Antud:** naturaalarv $n$ $(4 \leq n \geq 100)$ ja $n$ erineva tasandipunkti koordinaadid (täisarvud).
**Leida:** mitu erinevat rööpkülikut saab moodustada, mille tippudeks on mingisugused neli sisendis antud punkti?
**Soovitus.** Püüdke vältida neljakordset tsüklit, millega vaadatakse läbi kõik punktide nelikud.

18. Trapetsite arv
**Antud:** naturaalarv $n$ $(4 \leq n \geq 100)$ ja $n$ erineva tasandipunkti koordinaadid (täisarvud).
**Leida:** mitu erinevat trapetsit saab moodustada, mille tippudeks on mingisugused neli sisendis antud punkti?
**Soovitus.** Püüdke vältida neljakordset tsüklit, millega vaadatakse läbi kõik punktide nelikud.

## 3.7 VII klass

### 3.7.1 Lineaarsed võrrandid ja võrratused. Võrrandisüsteemid

**Ühe tundmatuga lineaarvõrrandi lahendamiseks** annavad õpikud ja eestikeelsed juhendid veebis tavaliselt järgmise **algoritmi**:

1) kui võrrand sisaldab harilikke murde, siis vabaneme nendest, korrutades võrrandi mõlemaid pooli kõigi murdude ühise nimetajaga;
2) lihtsustame võrrandi mõlemaid pooli (sulgude avamine, sarnaste liidetavate koondamine);
3) viime tundmatuga liikmed võrrandi ühele poolele ja vabaliikmed teisele poolele, muutes kõigi üleviidavate liikmete märgid vastupidiseks;
4) koondame sarnased liidetavad;
5) leiame lahendi, jagades võrrandi mõlemat poolt tundmatu kordajaga (kui kordaja on null, siis võib lahendite hulk olla tühi või sisaldada kõiki reaalarve).

**Märkus.** Punkti 1 sõnastus ei kõlba üldjuhu jaoks, kus murrud võivad olla ka üksteisega korrutatud. Näiteks võrrandis ¼ $\left(\frac{1}{6}x+1\right)=1$ tuleks murdudest vabanemise korrutada võrrand mitte ühise nimetaja 12-ga, vaid nimetajate korrutise 24-ga. Enne punkti 1 täitmist võib tegelikult olla otstarbekohane ka murdude taandamine (aga sealjuures nii, et järgneval sammul 1 ei tuleks „tagasi laiendada“).

**Soovitus.** Võrrandite ja võrratuste lahendamisalgoritmide ning lahendusetappide kohta on soovitav uurida programmi T-algebra tööd. Seda programmi saab kasutada ka testülesannete õige vastuse teadasaamiseks.

**Võrrandisüsteemid.** Ülesannetes võrrandisüsteemide kohta eeldame, et sisendis on võrrandid kujul $a_1x_1+a_2x_2+\cdots+a_nx_n=a_0$.

1. Võrrandi $ax=b$ lahendamine
   **Antud:** lineaarvõrrand $ax=b$, kus $a$ ja $b$ on reaalarvud, $a\neq 0$, $x$ tundmatu.
   **Leida:** võrrandi lahend.

2. Tundmatuga liikmed vasakule ja vabaliikmed paremale
   **Antud:** lineaarvõrrand $A(x)=B(x)$ , kus $A(x)$ ja $B(x)$ on reaalarvudest ja liikmetest kujul $ax$ (kus $a$ on reaalarv) plusside ja miinuste abil moodustatud avaldised.
   **Viia** võrrandis tundmatuga liikmed vasakule ja vabaliikmed paremale.

3. Sarnaste liikmete koondamine
   **Antud:** lineaarvõrrand kujul $A(x)=B$, kus $A(x)$ on liikmetest kujul $ax$ (kus $a$ on reaalarv) plusside ja miinuste abil moodustatud avaldis, ja $B$ on reaalarvudest plusside ja miinuste abil moodustatud avaldis.
   **Koondada** kummalgi pool sarnased liikmed.

4. Vabanemine murdudest lineaarvõrrandis
   **Antud:** lineaarvõrrand, mis sisaldab harilikke murde.
   **Vabaneda harilikest murdudest:** korrutada võrrandi pooli sobiva kordajaga ja lihtsustada, nii et tundmatu kordajad ei sisalda murde (vabaliikmetes võib neid olla).

5. Sulgude avamine
**Antud:** lineaarvõrrand $A(x) = B(x)$, mille pooled sisaldavad sulge.
**Avada** sulud ja koondada sarnased liikmed kummalgi poolel.

6. Lineaarvõrrandi lahendamine standardalgoritmiga
**Antud:** lineaarvõrrand $A(x) = B(x)$, millel on üks lahend.
**Leida:** võrrandi lahend, kasutades standardalgoritmi ja väljastades etappide tulemused.

7. Lineaarvõrrandid: samasus ja vastuoluline võrrand
**Antud:** lineaarvõrrand (üldjuht).
**Lahendada** võrrand, tehes kindlaks, kas tegemist on peajuhu, samasuse või vastuolulise võrrandiga.

8. Absoluutväärtusega lineaarvõrrand
**Antud:** lineaarvõrrand, mis sisaldab
a) mingi ühe avaldise absoluutväärtust (võimalik, et mitmes kohas),
b) mitme avaldise absoluutväärtusi.
**Lahendada** võrrand.

9. Võrratuse tõeväärtuse leidmine
**Antud:** lineaarne võrratus $A(x) \lesseqgtr B(x)$, kus märk $\lesseqgtr$ tähistab ranget või mitteranget võrratuse märki, ja reaalarv $x$ (vaadeldav tundmatu $x$ väärtus).
**Leida:** võrratuse tõeväärtus tundmatu antud väärtuse korral .

10. Tehted võrratuse ja arvuga
**Antud:** Lineaarne võrratus $A(x) \lesseqgtr B(x)$, kus märk $\lesseqgtr$ tähistab ranget või mitteranget võrratuse märki, võrratusega tehtava tehte märk $\odot$ (+, -, * või :) ja tehte teine argument $a$ (reaalarv).
**Leida:** Tehte tulemusena saadav võrratus (tehet ei pea $A(x)$ ja $B(x)$ osaavaldistega läbi tegema, vastus võib olla kujul $A(x)\odot a \lesseqgtr B(x)\odot a$).

11. Lineaarse võrratuse lahendamine
Õpikud tavaliselt ei anna üldist algoritmi lineaarse võrratuse lahendamiseks, vaid esitavad rea näiteid.
**Antud:** lineaarne võrratus $A(x) \lesseqgtr B(x)$, kus märk $\lesseqgtr$ tähistab ranget või mitteranget võrratuse märki.
a) võrratusest saadud võrrandil on üks lahend,
b) võrrandi lahendamine võib anda ka erijuhu.
**Lahendada** võrratus. Esitada teisendus sammude kaupa.

12. Absoluutväärtus(t)ega lineaarvõrratuse lahendamine
**Antud:** lineaarvõrratus, mis sisaldab
a) mingi ühe avaldise absoluutväärtust (võimalik, et mitmes kohas),
b) mitme avaldise absoluutväärtusi,
c) mingi ühe avaldise absoluutväärtust (võimalik, et mitmes kohas), kusjuures võrrandi lahendamine võib viia ka erijuhule.

**Lahendada** võrratus.

13. Kahe tundmatuga lineaarne süsteem liitmisvõttega
    **Antud:** kahe tundmatuga lineaarne võrrandisüsteem , millel on üks lahend.
    **Leida** süsteemi lahend, kasutades liitmisvõtet.

14. Kahe tundmatuga lineaarne süsteem asendusvõttega
    **Antud:** kahe tundmatuga lineaarne võrrandisüsteem , millel on üks lahend.
    **Leida** süsteemi lahend, kasutades asendusvõtet.

15. Kahe tundmatuga lineaarvõrrandisüsteemi uurimine
    **Antud:** kahe tundmatuga lineaarne võrrandisüsteem.
    **Leida:** kas süsteemil on
    `üks lahend/pole lahendit/lõpmata palju lahendeid`.

16. Kolme tundmatuga lineaarne süsteem liitmisvõttega
    **Antud:** kolme tundmatuga lineaarne võrrandisüsteem, millel on üks lahend.
    **Leida** süsteemi lahend, kasutades liitmisvõtet.

17. Kolme tundmatuga lineaarne süsteem asendusvõttega
    **Antud:** kolme tundmatuga lineaarne võrrandisüsteem, millel on üks lahend.
    **Leida** süsteemi lahend, kasutades asendusvõtet.

18. Nelja tundmatuga lineaarne süsteem liitmisvõttega
    **Antud:** nelja tundmatuga lineaarne võrrandisüsteem, millel on üks lahend.
    **Leida** süsteemi lahend, kasutades liitmisvõtet.

19. Nelja tundmatuga lineaarne süsteem asendusvõttega
    **Antud:** nelja tundmatuga lineaarne võrrandisüsteem, millel on üks lahend.
    **Leida** süsteemi lahend, kasutades asendusvõtet.

## 3.8 VIII klass

### 3.8.1 Üksliikmed

**Üksliikmeks** nimetatakse arvude ja mittenegatiivses täisarvulises astmes muutujate korrutist (milles on üks või rohkem liiget). Muutujad astmes 0 jäetakse tavaliselt kirjutamata . Üksliige on **normaalkujul**, kui ta algab arvulise kordajaga, millele järgnevad tähestikulises järjekorras erinevate muutujate positiivsed astmed. Muutujad astmes 0 jäetakse kirjutamata, samuti kõik muutujad kordaja 0 korral. Kordaja 1 kirjutatakse siis, kui muutujaid ei ole. Üksliikmeid, mille normaalkuju sisaldab samu muutujaid samades astmetes, nimetatakse **sarnasteks**. Näiteks $3 \cdot x \cdot x \cdot 5 \cdot y$ ja $6 \cdot y \cdot x^2$ on sarnased üksliikmed.

Programmide sisendis ja väljundis võiks astendamist tähistada kahe tärniga (näiteks x**2), pannes tehte operandid vajadusel sulgudesse.

1. Üksliikmete korrutamine
   **Antud:** avaldis $A_1 \cdot \ldots \cdot A_k$, milles $k \geq 2$ ja tegurid on üksliikmed (võivad olla sulgudes).
   **Leida:** nende üksliikmete korrutise normaalkuju.

2. Üksliikmete astendamine
   **Antud:** avaldis kujul $(A)^n$, kus $A$ on üksliige ja $n$ naturaalarv.
   **Leida:** üksliikme aste (normaalkujul).

3. Korrutamise ja astendamisega avaldised
   **Antud:** avaldis kujul $(A_1)^{n_1} \cdot \ldots \cdot (A_k)^{n_k}$ , kus $A_1, \ldots, A_k$ on üksliikmed ($k \geq 1$) ja $n_1, \ldots, n_k$on naturaalarvud.
   **Leida:** avaldise (kui üksliikme) normaalkuju

4. Üksliikmete jagamine
   **Üksliikmete jagamisel** tuleb jagada kordajad ja iga muutuja astmed (puuduva muutuja astmeks tuleb lugeda null). Kui jagatavas on mõne muutuja astendaja väiksem kui jagaja vastava muutuja aste, siis tekib jagamisel negatiivne aste, st tulemus ei ole enam üksliige. **Näiteks**
   $ab^3c^5 : 4a^2bd^4 = \frac{1}{4}a^{-1}b^2c^5d^{-4}$
   Üksliikmete jagamist võib vaadelda ka kui selliste murdude taandamist, kus lugeja ja nimetaja on üksliikmed: $ab^3c^5 : 4a^2bd^4 = \frac{ab^3c^5}{4a^2bd^4} = \frac{b^2c^5}{4ad^4}$.

   **Antud:** kaks üksliiget.
   **Leida:** nende jagatis sellisel normaalkuju sarnasel kujul, kus mõni astendaja võib olla ka negatiivne. Sellisel juhul lisada väljundile ’ei ole üksliige’.

5. Vabanemine negatiivsetest astendajatest
   Mõnedel juhtudel (ka teisenduste lõpptulemuse esitamisel) loetakse sobivaks **vabaneda negatiivsetest astendajatest**. Selleks võime teisendada avaldise murruks (kui ta seda veel ei ole) ja asendada murru lugeja korrutamise mingi muutuja negatiivse astmega murru nimetaja korrutamisega selle muutuja sama suure positiivse astmega ja vastupidi. Näiteks
   $a^{-2}b^3c^{-3} = \frac{b^3}{a^2c^3}$ ja $\frac{2a^{-2}b^3c^{-3}}{bc^{-5}d^{-1}} = \frac{2b^3c^5d}{a^2bc^3} = \frac{2b^2c^2d}{a^2}$.

**Antud:** arvudest ja muutujate täisarvulistest astmetest moodustatud korrutis või murd, mille lugeja ja nimetaja on sellised korrutised.
**Leida:** antud avaldisega samaväärne normaalkujuline üksliige või murd, mille lugeja ja nimetaja on normaalkujul ning mille lugejal ja nimetajal ei ole ühiseid muutujaid.
**Märkus.** Arvude paigutus ja esitus vastuses pole ühene ja vajab täiendavat kirjeldamist väljundtingimustes.

### 3.8.2 Hulkliikmed

**Hulkliikmeks** nimetatakse avaldist, mis on üksliikmete summa. Üksliikmeid, mille liitmisel hulkliige moodustub, nimetatakse **hulkliikme liikmeteks**. Tavaliselt tegeldakse **korrastatud hulkliikmetega**, milles üksliikmed on normaalkujul ja sarnased liikmed on koondatud. Mõnikord eeldatakse ka, et liikmed on järjestatud muutujate astmete summade mittekasvavas järjekorras.

1. Sarnaste üksliikmete koondamine
   **Koondada** saab üksliikmeid, mille normaalkujus on muutujate osad identsed. Koondamise tulemuses on üksliikme kordajaks koondatud üksliikmete kordajate summa.
   **Antud:** avaldis, mis on moodustatud üksliikmetest pluss- ja miinusmärke kasutades (võib olla rohkem kui üks rühm sarnaseid üksliikmeid).
   **Koondada** sarnased üksliikmed.

2. Hulkliikmete liitmine ja lahutamine
   **Antud:** avaldis $(B_1) \oplus \ldots \oplus (B_n)$, kus märgid $\oplus$ tähistavad plusse ja miinuseid ning $B_1, \ldots, B_n$ on hulkliikmed.
   **Leida:** liitmis- ja lahutamistehete tulemus (avada sulud, koondada).

3. Hulkliikme korrutamine üksliikmega
   **Antud:** hulkliige $B$ ja üksliige $A$.
   **Leida** $A$ ja $B$ korrutis ja korrastada tulemus.

4. Hulkliikme jagamine üksliikmega
   Üksliikmega tuleb jagada kõik hulkliikme liikmed eraldi ja tulemused liita. Tulemus on hulkliige, kui kõigis üksliikmetes on muutujate astmed vähemalt sama suured kui jagajas.
   **Näited:** $(10a^3b^4 - 15a^2b): 3ab = 3\frac{1}{3}a^3b^3 - 5a$ on hulkliige, aga
   $(a^4 + 2b): a^3 = a + 2a^{-3}b$ ei ole hulkliige.
   **Antud:** hulkliige $B$ ja üksliige $A$.
   **Leida** jagatis $B: A$ ja esitada tulemus hulkliikmena või (vajadusel) summana, mis sisaldab ka liikmeid, milles on negatiivseid astendajaid.

5. Üksliikmete SÜT toomine sulgude ette
   Sulgude ette saab tuua üksliikmete kordajate SÜT ja muutujate astmete korrutise, milles ühegi muutuja aste pole suurem, kui selle muutuja minimaalne aste üksliikmetes.
   **Antud:** täisarvuliste kordajatega hulkliige.
   **Esitada** hulkliige üksliikme ja hulkliikme korrutisena, kus üksliikme kordaja on hulkliikme

liikmete kordajate SÜT ja muutujatel on võimalikult suured astmed.

6. Kaksliikmete korrutamine
**Antud:** kaksliikmed $B$ ja $C$.
**Leida:** $B$ ja $C$ korrutis hulkliikmena, milles sarnased liikmed on koondatud.

7. Hulkliikmete korrutamine
**Antud:** hulkliikmed $B$ ja $C$.
**Leida:** $B$ ja $C$ korrutis hulkliikmena, milles sarnased liikmed on koondatud.

8. Rühmitamisvõte hulkliikme tegurdamiseks
**Tegurdamine (**korrutisena esitamine) on hulkliikmete jaoks oluliselt raskem ülesanne kui korrutamine (summana esitamine). Reaalarvuliste kordajatega hulkliikmete seas saab ühe muutuja hulkliikmeid esitada lineaarsete ja ruuthulkliikmete korrutistena, aga mitme muutuja korral leidub palju keerulisemaid hulkliikmeid, mida ei saa tegurdada.
**Rühmitamisvõtet** kasutades jaotatakse hulkliige rühmadeks nii, et pärast neist ühise teguri ette toomist jääb igas rühmas sulgudesse üks ja seesama avaldis. Lihtsamatel juhtudel saab rühmadesse jagada neidsamu üksliikmeid, mis on olemas esialgses hulkliikmes:
$ax + bx + ay + by = x(a + b) + y(a + b) = (x + y)(a + b)$. Õpikud mainivad ka võimalust, et mõni üksliige võidakse rühmitamiseks esitada summana:
$x^2 + 2xy + y^2 = x^2 + xy + xy + y^2 = x(x + y) + y(x + y) = (x + y)^2$. Aga mingit algoritmi sobivate rühmade otsimiseks õpikud ei esita. Olukord on muidugi ka keeruline, sest üldjuhul ei tarvitse hulkliikme tegurdamine üldse võimalik olla ja selle kindlakstegemine pole lihtne. Arvutialgebra jaoks on tegurdamise algoritmid välja töötatud, aga need nõuavad käsitsi rakendamise jaoks liiga mahukaid arvutusi.

**Antud:** kuni neljast üksliikmest koosnev mitme muutuja hulkliige
a) rühmadesse saab jagada olemasolevaid üksliikmeid,
b) saab jagada rühmadeks, jaotades mõne(d) üksliikme(d) summadeks.
**Leida:** tegurdamisülesande lahendus rühmitamisvõtte abil.

9. Abivalemite kasutamine tegurdamiseks
**Antud:** hulkliige $B$.
**Tegurdada** hulkliige, kasutades mingit abivalemit ja vajadusel ühise teguri ettetoomist, kui abivalemi kasutamine on võimalik. Vastasel juhul anda negatiivne vastus.

10. Ühe muutuja hulkliikmete jagamine
Ühe muutuja hulkliiget $a_n x^n + \cdots + a_1 x + a_0$ saab hulkliikmega $b_m x^m + \cdots + b_1 x + b_0$ **jagada** sama algoritmiga, mille abil jagame täisarve. Arvutus koosneb etappidest, mis vatavad kümne astmete asemel $x$ astmetele. Hulkliige võib teise hulkliikmega jaguda täpselt või jäägiga (jäägi aste peab olema väiksem kui jagajal, jagatises ei pea kordajad olema täisarvud).
**Näide:**
$(3x^4 - 7x^3 + x - 8) : (x^2 + 2) = 3x^2 - 7x - 6, \;\;$ jääk $15x + 4$
$3x^4 + 6x^2$

$$
\begin{array}{l}
-------- \\
\quad -7x^3 - 6x^2 + x \\
\quad -7x^3 \qquad\quad - 14x \\
-------------- \\
\qquad\qquad -6x^2 + 15x - 8 \\
\qquad\qquad -6x^2 \qquad\quad - 12 \\
----------------------------- \\
\qquad\qquad\qquad\qquad 15x + 4
\end{array}
$$

**Sisend:** kaks täisarvuliste kordajatega ühe muutuja hulkliiget $B(x)$ ja $C(x)$.
**Väljund:** jagamistehte esitus, jagatis $B(x)$: $C(x)$ ja jääk .

# 4 Nõuandeid konkreetsete lahenduste testimiseks

Selles peatükis on antud soovitusi ülesannete kogu konkreetseid ülesandeid lahendavate programmide testimiseks. Tegemist on muidugi ülesande põhjal „musta kasti“ meetodil välja pakutud testimisideedega ja tihti on vaja lisada programmis kasutatud algoritmidele ja lähenemistele vastavaid „valge kasti“ teste. Mõne ülesande jaoks on antud ka soovitusi valge kasti meetodite rakendamiseks.

Programmi valmimisel käivitatakse ta tavaliselt kõigepealt ühe või rohkema ülesandest tuleneva „tavaliste“ sisendandmete komplektiga. Tihti on programmis olemas nendele peajuhtudele vastavad harud ja programm võidakse käivitada ka juba mingi üksiku haru valmimisel. Alljärgnevad nõuanded kirjeldavad peamiselt „tavalistele“ järgnevaid vähem või rohkem silmatorkavaid andmete ekvivalentsiklasse, mis samuti vajavad testimist. Nõuandeid ei ole antud kõikide ülesannete jaoks. Näiteks võib testimise nõuanne puududa eriti lihtsate ülesannete puhul ja samuti ülesannetel, kus programm on mõeldud ainult ühe sisendandmete komplekti jaoks. Kui tegemist on mitmete üksteisega sarnaste ülesannetega, siis võivad nõuanded olla esitatud neist esimese kohta. Mõnikord võib see esimene paikneda tükk maad eespool ja isegi mõne eelmise klassi materjalis, kus juba sarnase teemaga tegeldakse. Nõuannete numbrid viitavad klassile, peatükile (kui see on olemas) ja ülesande numbrile ülesannete kogu osas 3.

Enne testi käivitamist on meil vaja teada oma testandmete jaoks õiget vastust. Isegi sisuliselt lihtsate ülesannete korral võib selleks vaja olla mingit arvutusvahendit. Testitavast programmist sõltumatul viisil õige vastuse leidmiseks võib arvutusülesannete puhul kasutada tabelarvutuse vahendeid (Excel vms). Avaldiste teisendamise ülesannete puhul võib abiks olla arvutialgebra süsteem (Geogebra vastav komponent, Maxima), samuti Tartus programmeeritud T-algebra [10]. Paljude ülesannete õige vastuse leidmiseks saab kasutada WolframAlfa abi. Kasutada saab ka Pythoni matemaatikavahendeid alates standardfunktsioonidest eval ja sort.

Soovitame iga ülesande korral kõigepealt programmeerida oma lahendus, testida seda enda poolt välja mõeldud testidega ja parandada vead ning alles siis lugeda testimisnõuandeid selle ülesande kohta. Mõnikord võivad nõuanded juba enne vastavate testide koostamist viia mõttele, et programmile on vaja lisada mingi juhu käsitlemine.

Muidugi peab ülesannete kogu autor lugejale ka ütlema, et kindlasti on ligi 200-le ülesandele testimissoovitusi kirjutades mõne ülesande juures mingid testimist vajavad aspektid kahe silma vahele jäänud. Autor on tänulik igasuguste märkuste eest ja lubab saadud täiendused võimalikult kiiresti ülesannete kogu veebiversioonile lisada.

## I klass

**1.1** Arvude võrdlemine
(Ideed ekvivalentsiklasside jaoks)**.** Juhud: $a < b, a = b, a > b$. Arvud $a$ ja $b$ võiksid olla mitmekohalised, mõned negatiivsed ja mõned positiivsed.

**1.2** Liitmistehte tulemuse leidmine
Arvud $a$ ja $b$: nii ühe- kui mitmekohalised, samuti 0.

**1.3** Võrduse $a + b = c$ puuduva liikme leidmine
Tundmatu $x$ nii esimesel, teisel kui kolmandal kohal. Nii ühe- kui mitmekohalised arvud igal kolmest kohast sõnes. Negatiivne $x$ väärtus igal kolmest kohast.

**1.4** Lahutamistehte tulemuse leidmine

Võrduse $a + b = c$ puuduva liikme leidmine. Arvud $a$ ja $b$ nii ühe- kui mitmekohalised, samuti 0. Vastus positiivne, negatiivne ja 0.

**1.5** Võrduse $a - b = c$ puuduva liikme leidmine (koos mittenegatiivsuse kontrolliga)

Tundmatu $x$ nii esimesel, teisel kui kolmandal kohal. Antud arvud $a, b$ ja c nii ühe- kui mitmekohalised, samuti 0. Antud arvud positiivsed, negatiivsed, 0. Vastus positiivne, negatiivne ja 0.

**1.6** Võrdlusmärgi panemine tehte tulemuse ja arvu vahele

Antud arvud igal kohal nii ühe- kui mitmekohalised, samuti 0. Märk pluss/miinus. Vastus </=/>.

**1.7** Tehtemärgi panemine $\oplus$ asemele

Antud arvud igal kohal nii ühe- kui mitmekohalised, samuti 0. Kõik võimalikud võrdlusmärgid. Õige vastus: pole/-/+/mõlemad.

**1.8** Tehtemärgi ja võrdlusmärgi valik

Testid peavad sisaldama kõiki võimalikke juhte kummagi küsimärgi jaoks ja negatiivset juhtu.

**1.9** Lõikude summa pikkus (ühikutega)

Andmetes peaks esinema juhud, kus iga komponent on olemas/puudub. Nii $b, c$ kui $d$ jaoks peaks esinema juhud, kus summa on kümnest väiksem, kümme ja suurem kui

**1.10** Summa saamine arvutusrahadega

On vaja teste, kus lõppsumma saab/ei saa moodustada. Peaks olema teste erinevate arvutusrahade väärtuste arvudega, näiteks 1, 2,3, ...,6 erinevat arvutusraha väärtust. Vastuse poolest peaks olema teste , kus lõppsummaks on vaja näiteks 1, 2, 3, … , 7 arvutusraha (vastavalt programmi töökiirusele võivad arvudel olla piirangud).

**1.11** Leida summade tabel 10-ni.

Sellel ülesandel ei ole erinevaid sisendandmeid ja testimiseks tuleb meil programm lihtsalt käivitada. Programmis kasutatud ideede kontrolliks võib programmi modifitseerida, et arvutataks summad mitte 10-ni, vaid mõne veidi väiksema/suurema liidetavate maksimaalväärtuse jaoks.

**1.13** Nädalapäevade liitmine ja lahutamine

Peaks olema teste, kus summa/vahe jääb (täisnädalaid arvust $a$ maha lahutades) samasse nädalasse/läheb üle nädala piiri.

**1.14** Veehulga mõõtmine antud mahuga anumaid kasutades

Testida tuleb nii 2 kui 3 mõõduanumaga juhtu. On vajalikud ka testid, kus lahendit pole. Testide hulgas peaks olema ka selliseid, kus mingit vahetulemust on vaja tekitada mitu korda.

**1.15** Kangkaaludega kaalumine

On vaja juhte, kus ühele kaalukausile tuleb panna 0, 1, 2, … vihti, sealhulgas ka võrdseid. Testida tuleb ka negatiivse vastusega võimalusi.

**1.16** Järeldamine kahest võrratusest

Vaadelda tuleb juhte, kus sisendseoste hulgas on:

a) kaks võrdust,

b) üks võrdus,

c) võrratused, mis paigutavad kaks muutujat kolmandast ühele poole,

d) võrratused, mis paigutavad kaks muutujat kolmandast kahte erinevasse suunda (saab rakendada transitiivsust).

Programmi korrektsus sõltub väga tugevasti seal kasutatud järelduste tegemise meetodist ja tingimuste kirjapanemise viisist. On võimalik programm, mis teeb erinevate muutujate puhul sisuliselt samadest eeldustest lähtudes erinevad järeldused. Testide valikul tuleb oluliselt kasutada valge kasti meetodit ja vajalik testide arv võib olla suur (kümnetes).

**1.17** Võrduste ja võrratuste tabel

Vastuolu tekitavaid andmeid peaks testides esinema nii vastuoluna refleksiivsusega, vastuoluna kahe sisendseose vahel kui ka vastuoluna, mis tekib transitiivsuse rakendamise järel (1 ja enam korda). Kooskõlalise lõpptulemuse jaoks peaks testid samuti sisaldama juhte, kus lahtri lõplik sisu tekib refleksiivsusest, otseselt sisendist, sümmeetria rakendamisest, transitiivsuse rakendamisest (erinev arv kordi). Tuleb uurida ka seda, et programm teeb järeldusi korrektselt (ei asenda tugevamat seost nõrgemaga, vaatab läbi järelduste tegemise järel tekkivad uued harud jms). Programmi korrektsus sõltub väga tugevasti seal kasutatud järelduste tegemise meetodist ja tingimuste kirjapanemise viisist. On võimalik programm, mis teeb erinevate muutujate puhul sisuliselt samadest eeldustest lähtudes erinevad järeldused. Testide valikul tuleb oluliselt kasutada valge kasti meetodit ja vajalik testide arv võib olla suur (kümnetes).

**1.18** Kivimite jagamine kaalult võrdseteks hulkadeks

On vaja teste, kus otsitav jaotus leidub/ei leidu; erinevate kivimite arvudega teste; teste, kus on mitmeid ühesuguse kaaluga kivimeid.

## II klass

**2.1** Summa maksmine antud müntidega

On vaja teste, kus vajaliku summa saab/ei saa moodustada. Peaks olema teste erinevate müntide väärtuste arvudega, näiteks 1, 2,3, ...,6 erinevat münti. Vastuse poolest peaks olema teste , kus lõppsummaks on vaja näiteks 1, 2, 3, … , 7 arvutusraha (vastavalt programmi töökiirusele võivad arvudel olla piirangud). On vaja nii teste, kus piirangud ühesuguste müntide arvudele mõjutavad/ ei mõjuta vastust, kui ka juhtu, kus lõppsummat ei saa moodustada just mingite müntide arvu tõttu.

**2.2** Järjestamine pikkuse järgi

Peale „tavaliste andmete“ on vaja ka teste, kus on erineva nimega, aga võrdse pikkusega lapsi. Ja vastupidi. Võiks olla ka test, kus kõik pikkused on võrdsed. Veel: ainult üks laps, juba õiges järjestuses nimekiri.

**2.3** Kellaajale minutite liitmine

On vaja teste, kus tund ei muutu/suureneb 1 võrra/suureneb rohkem/kuulub järgmisse päeva.

**2.4** Võrdsete liikmetega summa väljendamine korrutisena

Testid erineva rühmade arvuga 1, 2, 3, … . Testid, kus rühmas on liikmeid üks/rohkem.

## III klass

**3.2** Tehted detsimeetrite ja sentimeetritega

Peale peajuhu testide veel: testid, kus lõigu sentimeetrite/detsimeetrite arv on null; testid, kus sama tekib vastuses; kaks võrdset lõiku.

**3.3** Mitmel erineval viisil saab ostu eest tasuda?

On vaja teste erineva ostjal olevate müntide arvuga, võimalusega moodustada sama summa erinevatest kahest/kolmest mündist, teste erinevate vastustega.

**3.5** Võluruudu kontroll

Vaja on nii positiivse kui negatiivse vastusega teste. Negatiivne vastus võiks tulla erinevate ridade/veergude/diagonaalide tõttu. Variandi b) jaoks on vaja ka teste, kus variandi a) tingimus on täidetud.

**3.6** Võluruudu ehitamine antud arvudest

Programmilt saadud lahendite kontrolliks tuleks omakorda arvuti abi kasutada, sest peast arvutades võime kergesti mõne eksliku summa tähele panemata jätta. Aga kindlasti ei sobi siin lahendamisprogrammist kopeeritud summade arvutamine. Antud ülesande puhul võiks näiteks kasutada Exceli tabelit (koos sobiva tehnoloogiaga väljundfaili kiireks tõstmiseks tabelisse).

**3.7** Teisendamine: meetrid, detsimeetrid, sentimeetrid, millimeetrid

On vaja teste, kus esinevad/võrduvad nulliga/puuduvad sisendi erinevad komponendid.

**3.9** Jäägiga jagamine

On vaja teste, kus jääk tekib/ei teki. Ka juht, kus jagatis on 0.

**3.10** Homne kuupäev

Vaja on teste iga kuu kohta, kus tänane päev on
kuu keskel/kuu viimane päev/päev, mis on mõnes kuus viimane,
lisaks teste veebruari erineva pikkuse kohta.

## IV klass

**4.1** Kas kõiki numbreid on kasutatud?

On vaja teste, kus kõik numbrid on kasutatud/üks number kasutamata/rohkem numbreid kasutamata. Peaks esinema erineva pikkusega naturaalarve.

**4.2** Suuruselt teine arv

Testid peaksid sisaldama juhte, kus
- jada suuremad liikmed on kõik erinevad,
- mõned suuremad liikmed (erinevates kombinatsioonides) on omavahel võrdsed,
- vastus tekib juba tsükli invariandi algväärtustamisel,
- tsükli invariandi muutujate väärtusi tuleb muuta mõned korrad või palju kordi
- jadas on liikmeid 1/2/rohkem.

**4.3** Tingimustele vastavate arvude moodustamine

Näiteks teise alamülesande jaoks oleks vaja teste, kus
1) on vähem kui 5 kaarti,
2) pole ühtegi paarisnumbriga kaarti/üks paarisnumbriga kaart/kaks erinevate paarisnumbritega kaarti/rohkem paarisnumbreid,
3) sorteerimise tulemuse viimane koht on paarisnumber/tuleb asendada vähima vaba paarisnumbriga/tuleb asendada juba arvu paigutatud paarisnumbriga.

**4.4** Värvitud kuupidest koosneva kujundi vaade

Testimist võiks alustada väikese kuupide arvuga (1, 2, 3, …), lisades kuupe ja tõstes kasutatud kihtide arvu.

**4.5** Ajavahemike summa

Testid peaks kontrollima korrektset liitmist juhtudel, kus minutie/sekundite arvude summad on alla/üle 60.

**4.6** Üldistatud 15-mäng

Eraldi saab testida otsustamist, kas saab/ei saa liikuda lõppseisu. Laiuti otsingut võiks esialgu testida $3 \times 2$ ruudustikus, kasvatades vajalikku sammude arvu: 0, 1, 2, … . Edasi võime uurida, kui suure ruudustiku korral töötab optimaalset lahendust leidev programm vastuvõetava ajaga.

Programmi poolt leitava lahenduse optimaalsust me üldiselt testida ei suuda, kui meil pole sõltumatut vahendit optimaalse lahenduse leidmiseks. Suuremate ruudustike korral nuppe ühekaupa paigutades tuleks eraldi testida nupu paigutamist erinevatesse positsioonidesse (esimene/teine/keskmised/eelviimane/viimane) parajasti täidetavas reas.

## V klass

### 5.1 Naturaalarvude liitmine ja lahutamine

**5.1.1** Naturaalarvu esitamine järkarvude summana
Testid peaksid sisaldama erineva suurusega arve (kindlasti 1-kohalisi, 2-kohalisi, 6-kohalisi ja 7-kohalisi), erineva arvu järkarve, erineva arvu nulliga võrduvaid kohti.

**5.1.2** Naturaalarvu kirjutamine sõnadega
Testides peavad iga numbri 1, 2, … jaoks esinema kõik sõnad, mida kasutatakse arvude kirjutamisel: kaks, kaksteist, kakskümmend, kakssada. Regulaarsel viisil moodustatud fraase (näiteks … tuhat) tuleb katta vastavalt nende moodustamise meetodile (ühekaupa välja kirjutatud tegevusi tuleb ka ühekaupa testida). Testid peavad sisaldama nulli kõigis võimalikes positsioonides, ühe kirjutamise/mittekirjutamise kontrolli kõigis positsioonides.

**5.1.3** Kirjaliku liitmise vormistamine
Testimist vajavad aspektid: õige liitmine, lühikeste ja pikkade liidetavate esinemine erinevatel kohtadel, liidetavatega sama pikk/pikem summa.

**5.1.4** Pluss- ja miinusmärke sisaldava avaldise väärtuse leidmine
Avaldised peaks sisaldama 1/2/3/rohkem liiget. Peaks olema arve pikkusega 1/2/3/4, mõlema märgiga. Vastusena peaks esinema nii positiivne ja negatiivne arv kui null.

**5.1.5** Naturaalarvudest pluss- ja miinusmärkide ning sulgude abil moodustatud avaldise väärtuse leidmine
Avaldised peaks sisaldama 1/2/3/rohkem liiget. Peaks olema arve pikkusega 1/2/3/4, mõlema märgiga. Vastusena peaks esinema nii positiivne ja negatiivne arv kui null. Peaks olema teste, kus on üksteise sees 1, 2, … taset sulge. Sulud peaks asuma avaldise alguses/plussi järel/miinuse järel.

**5.1.6** Ristküliku pindala ja ümbermõõdu leidmine
Testid peavad sisaldama kõiki ühikuid.

**5.1.7** Ümardamine
Testimist vajavad aspektid: kõik võimalikud täpsused, arvu $a$ erinevad suurused (ka täpsusest väiksem), erinevad esimese ümardamisel väljajäetava kümnendkoha väärtused.

**5.1.8** Ühikuga suuruse teisendamine vastavalt etteantud ühikule
Testimist vajavad aspektid: väärtuse erinev muutumine pikkuse, pindala ja ruumala puhul, väärtuse suurenemine/vähenemine/säilimine.

**5.1.9** Araabia ja rooma numbrid
Testid peavad katma: kõiki rooma numbrites kasutatavaid tähti, kõiki järjestikku ja suurema väärtusega tähe järele/ette kirjutatavate väiksema väärtusega tähtede arve

**5.1.10** Järjestamine (2-dimensionaalne)
Testid peavad sisaldama nii erineva $x$ väärtusega, võrdse $x$ väärtusega kui päris võrdseid paare. Veel: $n = 1/2/suurem$ ja juhud, kus sama kehtib väljundi kohta.

**5.1.11** Muutujate asendamine väärtustega
Et vaja on ainult muutujad asendada, siis avaldised peavad sisaldama muutujaid

alguses/plussi järel/miinuse järel ja sulgude panemise kontrolliks tuleb kasutada nii negatiivseid kui mittenegatiivseid väärtusi.

**5.1.12** Muutujatega avaldise väärtuse leidmine
Lisaks asenduse testimisele on vaja testida ka tehete tegemist. Seda võiks teha tehete arvu järk-järgult kasvatades.

**5.1.13** Lineaarse funktsiooni väärtuste tabel
On vaja teste, kus kordajate A ja $B$ väärtus on negatiivne/null/positiivne, lõigu pikkus jagub/ei jagu sammuga.

**5.1.14** Üldkujulise funktsiooni väärtuste tabel
Lisaks eelmise ülesande probleemidele on siin vaja tõsisemalt testida ka funktsiooni väärtuste arvutamist. Mitte kõikjal määratud funktsioonide puhul peavad testid sisaldama vastavaid argumendi väärtusi.

**5.1.15** Tehete järjekorra lubatavuse kontroll
Testid peaksid sisaldama nii õiget kui ka ekslikku  järjekorra otsustamist kõigi võimalike tehtepaaride vahel ja sulgude arvestamist. Peaks leiduma ka teste erinevas järjekorras tehtavate sõltumatute tehetega sama avaldise juures (näiteks korrutamistehted avaldises $2 \cdot 3 + 4 \cdot 5$).

**5.1.17** Aritmeetilise progressiooni summa tsükliga
Liikmete arv: 1/2/suurem, vahe: negatiivne/null/positiivne.

**5.1.18** Võrrand $x \oplus a = b$
Testides võiks olla a ja b nii negatiivne, null kui positiivne, samuti erineva pikkusega ja täisarv/kümnendmurd.

**5.1.19** Aditiivne võrrand
Testida võiks $k$ väärtustel 1/2/…, erinevates testides võiks olla 1/rohkem $x$ esinemist, lahendite arv 0/1/lõpmatu, lahend täisarv/kümnendmurd.

**5.1.21** Nulliga võrduva avaldise moodustamine
On vaja teste, kus avaldisele saab/ei saa anda väärtust 0, arve on 1/2/3/…/20, arvud on erineva pikkusega.

## 5.2 Naturaalarvude korrutamine ja jagamine. Tegurid, algarvud

**5.2.1** Korrutamistehete tulemuse arvutamine
Testides võiks esineda korrutised 1/2/3/4 liikmega, arvude pikkused 1/2/3, tegurid 0/1/suuremad, maksimaalne võimalik vastus.

**5.2.3** Kirjalik korrutamine
Testid võiksid sisaldada olukordi, kus: kumbki tegur on pikkusega 1, 2, …, 5; teine tegur sisaldab nulle erinevatel kohtadel; lõpptulemus ulatub viimasest liidetavast ettepoole; esimene/teine tegur on väärtusega 0.

**5.2.4** Kirjalik jagamine
Testid võiksid sisaldada olukordi, kus: Jagaja on jagatavaga võrreldes väiksem/võrdne/suurem; jagatav on null; jagatis sisaldab mingitel kümnendkohtadel nulle; vähendatav ulatub lahutatavast ettepoole.

**5.2.6** Nelja tehtega arvavaldised
Testid peaksid sisaldama järjekorra otsustamist kõigi võimalike tehtepaaride vahel ja sulgude arvestamist. Peaks leiduma ka teste erinevas järjekorras tehtavate sõltumatute tehetega sama avaldise juures (näiteks korrutamistehted avaldises $2 \cdot 3 + 4 \cdot 5$).

**5.2.8** Koondamine lineaarses hulkliikmes

Testide sisendhulkliikmes võiks olla: 0/1/2/… lineaarliiget/vabaliiget; kordajad võiks olla 1/2/3-kohalised; vastuses võiks esineda negatiivne/0/positiivne lineaarliikme/vabaliikme/mõlema kordaja; miinus esimese sümbolina sisendis.

**5.2.9** Lineaarse hulkliikme korrutamine arvuga

Testide sisendhulkliikmes võiks olla: 1/2/… lineaarliiget; kordajad võiks olla 1/2/3-kohalised; vastuses võiks esineda negatiivne/positiivne lineaarliikme/vabaliikme kordaja; miinus esimese sümbolina sisendis.

**5.2.10** Tärnide/tähtede asendamine kirjaliku korrutamise tehtes vm tuntud struktuuris

Selle ülesande igas konkreetses variandis on tegemist fikseeritud sisendandmetega, mistõttu muud testid otsest mõtet ei oma.

**5.2.11** Tehtemärkide panemine arvude vahele

Avaldiste käsitlemise testimiseks võiks võrdust tekitavaid avaldisi valida analoogiliselt ülesandele 5.2.6. Peaks olema ka teste, kus vastus on negatiivne. Testida tuleks avaldiste keerukust niimoodi tõstes, et saadaks ettekujutus programmi töökiirusest.

**5.2.12** Tegurite leidmine

Testid võiksid sisaldada: arve igasuguse tüvikohtade arvuga 1,…,10; naturaalarve nii väikese kui suure erinevate tegurite arvuga.

**5.2.13** Algarvulisuse kontroll

Teste võiks esitada arvude suurust järk-järgult kasvatades, esitades igas suuruses nii algarve kui kordarve, nii väikese kui suure tegurite arvuga.

**5.2.14** Algtegurite ja nende astmete leidmine (kanooniline kuju)

Testid võiks sisaldada/tekitada: arve erineva algtegurite arvuga 1, 2, 3, … ; algtegurite astmeid 1, 2, … ; algtegurite vahel peaks olema ka algarve, mis ei ole tegurid.

**5.2.15** Eratosthenese sõel

Sisendina saame ette anda ainult arvu $M$. Korralikult tuleks läbi vaadata väikesed $M$ väärtused. Testimisel võiks väljastada iga etapi järel arvud $2, \dots, M$ koos juba tehtud märgetega.

**5.2.16** Eukleidese algoritm

Testid peaks sisaldama: nii väikesi kui suuri $a$ ja $b$ väärtusi, nii väikesi kui suuri SÜT väärtusi; juhte, kus SÜT on 1 või üks arvudest $a$ ja $b$.

**5.2.17** SÜT, VÜK

Testima peaks juhte, kus: $n = 2, 3, 4, \dots$ ; SÜT on 1/üks arvudest $a_i$/võrdne mingi osahulga SÜT-ga. Testide valik sõltub SÜT leidmise meetodist (Eukleidese meetod/algtegurite leidmine/…).

## 5.3 Geomeetria konstruktsioonide algoritmid

Geomeetria konstruktsioonide korral annavad sisendandmed võrdlemisi vähe erinevaid variante konstruktsiooni kulgemiseks. Alljärgnevas on pakutud ka teste programmi sellise variandi (taseme) jaoks, mis hoolitseb joonise paigutumise eest joonistamiseks valitud ekraaniaknas. Mitmetel ülesannetel võimaldavad sisendandmed joonise erinevasse suunda paigutamist, erineva raadiusega ringjooni jms, kus paremad valikud paigutavad konstruktsiooni joonistamisakna suhtes paremini.

Selle peatüki programmide testimisel võib osutuda otstarbekaks alamprogrammidena kasutatavate lahenduste testide salvestamine, et neid hilisemate ülesannete juures uuesti kasutada.

**5.3.1** Lõigu joonestamine punktide järgi
Peale peajuhu töötamises veendumise tuleks testida ka lõigu kummagi otspunkti paiknemist igas suunas väljaspool joonisele eraldatud akent.

**5.3.2** Kiire joonestamine punktide järgi
Testid peaks sisaldama juhte, kus kiir ulatub igaüheni joonistamisakna servadest.

**5.3.3** Sirge joonestamine punktide järgi
Testid peaks sisaldama kõiki võimalikke sirge lõikumisi joonistamisakna servadega ja juhtu, kus sirge asub väljaspool akent.

**5.3.4** Nurga joonestamine
Testid peaks sisaldama:
nurki kõigis tasandi veerandites, samuti üle 360 kraadi;
juhte, kus punkt $A/B$ on väljaspool joonistamisakent.

**5.3.5** Ringjoone joonestamine keskpunkti ja ringjoonel oleva punkti/raadiuse järgi
Testid peaks sisaldama juhte, kus
keskpunkt/etteantud punkt asub joonistamisaknas/on erinevates suundades väljaspool;
ringjoon asub joonistamisaknas/lõikub akna 1…4 servaga/jääb väljapoole akent.

**5.3.6** Kahe sirge lõikepunkti leidmine
Üldjuhu testid peaksid sisaldama juhte, kus:
etteantud punktid asuvad joonistamisaknas/osaliselt või täielikult väljaspool;
joonistamisaknas on nähtavad 2/1/0 sirget;
sirged lõikuvad joonistamisaknas/väljaspool;
sirged on lõikuvad/paralleelsed/ühtivad.

**5.3.7** Sirge ja ringjoone lõikepunktide/puutepunkti leidmine
Enne konstruktsiooni tuleks testida etteantud sirge ja ringjoone käsitlemist (vt üles. 3 ja 5). Kui varem testitud lahendusi kasutatakse alamprogrammidena, ei tarvitse uuesti käivitada kõiki teste, vaid ainult veenduda pöördumiste korrektsuses. Testid peaks sisaldama ka juhte, kus:
sirgel ja ringjoonel on 0/1/2 ühist punkti;
ühised punktid asuvad joonistamisaknas/väljaspool.

**5.3.8** Antud pikkusega lõigu joonestamine kiirele
Analoogiliselt eelmise ülesandega tuleks testida kiire käsitlemist. Edasised testid peaks sisaldama juhte, kus:
lõik asub aknas/ulatub aknast välja (erinevad suunad).

**5.3.9** Kahe ringjoone lõikepunktide/puutepunkti leidmine
Analoogiliselt eelmiste ülesannetega tuleks testida ringjoone käsitlemist. Edasised testid peaks sisaldama juhte, kus:
ringjoontel on 2/1/0 ühist punkti ja sealjuures kummagi keskpunkt asub väljaspool teist ringjoont/vähemalt ühe keskpunkt on teise sees;
ringjooned ühtivad.

**5.3.10** Ristsirge joonestamine
Analoogiliselt eelmiste ülesannetega tuleks testida sirge ja ringjoone käsitlemist. Edasised testid peaks sisaldama juhte, kus:
punkt $C$ asub joonistamisaknas/on erinevates suundades väljaspool;
punkt $C$ asub sirgel $s$/väljaspool.

**5.3.11** Paralleelsirge joonestamine
Analoogiliselt eelmiste ülesannetega tuleks testida sirge ja ringjoone käsitlemist. Edasised testid peaks sisaldama juhte, kus:

punkt $C$ asub joonistamisaknas/on erinevates suundades väljaspool;
punkt $C$ asub sirgel $s$/väljaspool;
programm annab teate, kui konstruktsioon jääb väljapoole joonistamisakent.

**5.3.12** Nurga teisaldamine
Selle ülesande testid peaksid sisaldama juhte, kus:
etteantud punktid asuvad joonistamisaknas/väljaspool;
teisaldatav nurk on (orientatsiooni arvestades) positiivne/0/negatiivne;
orientatsiooni säilitades teisaldatud nurga punkt Z asub joonistamisaknas/väljaspool.

**5.3.13** Kolmnurga joonestamine kolme külje järgi
Testid peaksid sisaldama juhte, kus:
kolmnurk leidub/$a > b + c/b > a + c/c > a + b$;
kolmnurk ulatub aluse/kõrguse poolest väljapoole joonistamisakent.

**5.3.14** Kolmnurga joonestamine kahe külje ja nurga järgi
Testid peaksid sisaldama juhte, kus:
kolmnurk leidub/$a > b + c/b > a + c/c > a + b$;
kolmnurk ulatub aluse/kõrguse poolest väljapoole joonistamisakent.

**5.3.15** Kolmnurga joonestamine külje ja kahe nurga järgi
Testid peaksid sisaldama juhte, kus:
kolmnurk leidub/ei leidu;
kolmnurk ulatub aluse/kõrguse poolest väljapoole joonistamisakent.

**5.3.16** Ruudu joonestamine
Testid peaksid sisaldama juhte, kus:
punktid $A$ ja $B$ asuvad joonistamisakna sees/väljas;
joonistamisakna sees asub 2/1/0 tekkiva ruudu varianti.

**5.3.19** Lõigu poolitamine
Testid peaksid sisaldama juhte, kus:
punktid $A$ ja $B$ asuvad joonistamisakna sees/väljas;
lõigu keskristsirge saamiseks konstrueeritavad punktid on joonistamisakna sees/konstruktsioon vajab korrigeerimist.

**5.3.20** Nurga poolitamine
Testid peaksid sisaldama juhte, kus:
punktid $A, B$ ja $C$ asuvad joonistamisakna sees/väljas.

**5.3.21** Ringi puutuja joonestamine
Testid peaksid sisaldama juhte, kus:
punkt $A$ ja konstruktsiooni jaoks vajalik osa ringist asuvad joonistamisakna sees/väljas;
punkt $A$ asub ringjoone sees/ringjoonel/väljaspool.

**5.3.22** Kolmnurga ümberringjoone joonestamine
Testid peaksid sisaldama juhte, kus:
kolmnurga tipud $A, B$ ja $C$ asuvad joonistamisakna sees/väljas;
standardne konstruktsioon mahub joonistamisaknasse/vajab demonstreerimise jaoks korrigeerimist.

## 5.4 Nurgad kella osutite vahel

**5.4.1** Minutiosuti liikumine ja kellaaeg
Testida on vaja:

minuti muutumist, kus kasvab ainult minutinäit/kasvab ka tunninäit/muutub päev;
tunni muutumist, kus ainult tund kasvab/muutub päev.

**5.4.3** Nurk osutite vahel
On vaja teste, kus minutiosuti jaoks arvutatud nurk on suurem/võrdne/väiksem kui tunniosuti nurk.

## 5.5 Statistika mõisted

**5.5.2** Osahulkade aritmeetilised keskmised
Testida tuleks ka olukorda, kus poiste/tüdrukute arv on 0.

**5.5.4** Moodi arvutamine
Testid peaks sisaldama juhte, kus kõige sagedamini esinevaid väärtusi on 1/2/rohkem.

**5.5.5** Mediaani arvutamine
Testid peaks sisaldama juhte, kus õpilaste arv on paaris/paaritu.

# VI klass

## 6.1 Harilikud murrud

**6.1.1** Liht- ja liigmurrud
Testid peaks sisaldama juhte, kus lugeja on väiksem/võrdne/suurem kui nimetaja.

**6.1.2** Murru laiendamine
Selles ülesandes ei ole peajuhust erinevaid võimalusi.

**6.1.5** Taandamine suurima ühisteguriga
Testida tuleb ka juhte, kus SÜT on 1/SÜT on üks arvudest $x$ ja $y$/ $x$=$y$.

**6.1.6** Osa meetrist, ruutmeetrist, kuupmeetrist
Testid peavad sisaldama antud suurusi kõigi vaadeldavate ühikutega; juhte, kus suurus on 1 ühikuga võrreldes väiksem/võrdne/suurem, suhtarv on murdarv/täisarv.

**6.1.7** Tundmatu $x$ leidmine võrdest
Testid peaksid sisaldama tundmatu täisarvulisi ja murdarvulisi väärtusi <1/1/>1.

**6.1.8** Ühenimeliseks teisendamine
Testida võiks vähemalt murdude arvudega 2,3,4;
juhtudega, kus VÜK on korrutis/ üks nimetajatest/nendest erinev;
mitmesuguste kombinatsioonidega võrdustest nimetajate vahel.

**6.1.9** Millised murdude võrdused on tõesed?
Testid peaks sisaldama juhte, kus võrdus kehtib/ei kehti; murrud on ühenimelised/erinimelised;

**6.1.11** Murru teisendamine segaarvuks
Testid võiks sisaldada murde, kus väärtus on lihtmurd/täisarv/segaarv.

**6.1.12** Segaarvu teisendamine liigmurruks
Testides võiks esineda täisarv/murdosa lugejaga 0/murdosa nimetajast väiksema lugejaga.

**6.1.13** Ühenimeliste murdude liitmine ja lahutamine
Vaatleme lahutamist. Testides võiks esineda juhud, kus a<b/a=b/a>b; a=0;b=0.

**6.1.14** Ühenimeliste segaarvude liitmine
Testides võiks esineda juhud, kus

summa täisosa on liidetavate täisosade summa/summa+1/summa+2;
liidetava täisosa=0/liidetava murdosa=0;
summa murdosa=0.

**6.1.15** Ühenimeliste segaarvude lahutamine
Vt üles 13 ja 14.

**6.1.16** Erinimeliste murdude liitmine ja lahutamine
Vt üles. 8, 11, 13.

**6.1.17** Lahendada murdudega aditiivne võrrand
Testides peaks esinema:
tundmatu on 1./2./3. kohal; märk on +/-; arvuline liige null 1./2./3. kohal;
negatiivne/null/positiivne vastus; vastus on lihtmurd/täisarv/segaarv.

**6.1.18** Kümnendmurd harilikuks murruks
Testides võiks esineda juhud, kus:
vastus on täisarv/lihtmurd/segaarv;
vastuse nimetaja on kümne aste/saadud taandamisel;
sisendi murdosa lõpus on nulle.

**6.1.19** Harilik murd kümnendmurruks (ka perioodilised kümnendmurrud)
Testides võiks esineda juhud, kus:
vastus on täisarv/lihtmurd/segaarv;
sisendi murdosa nimetaja on kümne aste/kahe ja viie astmete korrutis/tekitab perioodilise kümnendmurru;
tekkiva kümnendmurru perioodile eelneb 0/1/2/… kümnendkohta ja perioodis on 1/2/3/… kohta.

**6.1.20** Hariliku murru kümnendlähend
Testides võiks esineda juhud, kus:
sisend on lihtmurd/liigmurd;
sisend avaldub täisarvuna/lõpliku kümnendmurruna/perioodilise kümnendmurruna;
lõplikku kümnendmurdu on/ei ole vaja ümardada;
on erinevad esimese ümardamisel väljajäetava kümnendkoha väärtused.

**6.1.21** Avaldise väärtuse arvutamine (kümnendmurd ja harilik murd)
Testides võiks esineda juhud, kus harilik murd on täisarv/avaldub lõpliku kümnendmurruna/ei avaldu lõpliku kümnendmurruna. Vt ka üles. 18 soovitusi.

**6.1.23** Harilike murdude korrutamine
Testides võiks esineda juhud, kus
sisendi komponentides on 1/2/… kümnendkohta;
pärast lugejate ja nimetajate korrutamist on/ei ole vaja taandamist;
tulemus on null/täisarv/lihtmurd/segaarv.

**6.1.25** Osa leidmine arvust
Testides võiks esineda juhud, kus
arv $x$ on positiivne/0/negatiivne;
arv $x$ on täisarv/harilik murd/segaarv/kümnendmurd;
osamäär on 0.

**6.1.26** Pöördarvu leidmine
Testides võiks esineda juhud, kus
arv $x$ on positiivne/0/negatiivne;
arv $x$ on täisarv/harilik murd/segaarv/kümnendmurd;
$x$ pöördarv on täisarv/lihtmurd/segaarv.

**6.1.27** Harilike murdude jagamine
Vt. üles 23, lisaks juht $c = 0$.

**6.1.29** Murdudega avaldise väärtus
Avaldise käsitlemise testimise kohta vt üles. 5.1.5, 5.2.6, tehete kohta 6.1.16, 6.1.23, 6.1.27.

## 6.2 Tasandi teisendused. Geomeetrilised kujundid

**6.2.1** Lükke parameetrite leidmine
Testida võiks punktide arvu kasvatades 1, 2, 3, … . Peaks olema teste, kus leidub/ei leidu lüke.

**6.2.2** Pöördenurga leidmine
Testida võiks punktide arvu kasvatades 1, 2, 3, … . Peaks olema teste, kus leidub/ei leidu pööre. Täisarvulisi näiteid saaks leida Pythagorase arve kasutades.

**6.2.3** Peegelduse leidmine
Testida võiks punktide arvu kasvatades 1, 2, 3, … . Peaks olema teste, kus peegeldusi on 0/1/2/3/…/lõpmata palju. Testida tuleks ka näidetega, kus on vaja peegeldust horisontaalse/vertikaalse sirge suhtes; kujundite ainsad punktid ühtivad.

**6.2.4** Kujundi telgsümmeetrilisuse kontroll
Testida võiks punktide arvu kasvatades 1, 2, 3, … . Peaks olema teste, kus:
kujund on/ei ole telgsümmeetriline. Vt ka üles 3.
leidub 1/2/3 … sümmeetriatelge;

**6.2.5** Arvu $\pi$ alumise ja ülemise hinnangu täpsus
Testimisel võiks programm väljastada ka $\pi$ alumise ja ülemise hinnangu.

**6.2.6** Tingimused kolmnurga külgedele ja nurkadele
Testides peaks esinema juhud, kus
kolmnurk leidub/ei leidu (sealjuures nurgad üle 180 kraadi).

**6.2.7** Kolmnurga määramine kolme elemendiga
Seda ülesannet lahendatakse tõenäoliselt alustades mingi väiksema hulgaga $H$ ja suurendades teda siis järgmiste suuruste lisamise ja vastavate arvutuskäikude realiseerimisega. Ka testimine koos silumisega võiks toimuda sama teed mööda järk-järgult.
Testid peaks sisaldama:
sisendandmetena kõiki hulga $H$ kolmeelemendilisi alamhulki (vastavad sisendid tuleks failides säilitada, et neid hiljem hulka $H$ laiendades uuesti kasutada);
antud sisendsuuruste väärtusi, kus saadakse kõik võimalikud otsuse variandid
EI OLE/ÜKS/KAKS/LÕPMATA PALJU (ja kasutatakse sealjuures kõiki programmis olevaid arvutusteid);
kontrolli, et mitme ühte tüüpi sisendsuuruse (külg, nurk, kõrgus) puhul kolmnurga tippe ümber nimetades tulemus ei muutuks.

**6.2.8** Kuubi pinnalaotuse äratundmine
Positiivse vastusega testid peaks sisaldama kõiki 11 võimalikku kujundit, mis on erinevatel viisidel nihutatuna, pööratuna ja peegeldatuna paigutatud $5 \times 5$. Kõiki võimalusi testida oleks väga raske. Kui programm kasutab kujundite erinevate paiknemiste uurimiseks mingeid süstemaatilisi viise, siis annab see võimaluse testida eraldi neid viise ja 11 kujundit. Näiteks esimesel joonisel olevaid kujundeid saab $5 \times 5$ ruudustikus nihutada horisontaalselt 2 viisil ja vertikaalselt 1 viisil, seega nihe annab meile kokku 6 viisi. Pöörata saab 4 viisil ja peegeldada 2 viisil. Seega saame igale esimese rea kujundile natuke vähem kui 48 paigutust $5 \times 5$ ruudustikus (tegelikult natuke vähem, sest mõned teisenduste kombinatsioonid annavad

ühesuguse tulemuse).
Negatiivse vastusega testid võiks eelkõige saada positiivsetest väikeste moonutuste teel, aga võiks olla ka mingi hulk hoopis teistsuguseid testkujundeid.

**6.2.9** Kolmnurga liik (…-nurkne, külgede järgi)
Testides peaks esinema juhud, kus sisendis antud küljed
moodustavad teravnurkse/täisnurkse/nürinurkse kolmnurga/ ei moodusta kolmnurka.
Testida tuleb ka stabiilsust külgede/nurkade ümbernimetamise suhtes.

**6.2.12** Kolmnurgad koordinaattasandil
Testid peavad sisaldama vastuse kõiki juhte (ka kõdunud juhtu) ja testima ka stabiilsust sisendi ümberjärjestamise suhtes.

**6.2.13** Nelinurgad koordinaattasandil
Iga alamülesande korral peaks testid sisaldama sisendeid, mis
annavad positiivse/negatiivse vastuse;
kui programm kontrollib kujundit tingimuste kaupa (näiteks ruudu korral külgede pikkuste võrdsust ja nurkade võrdsust), siis on iga tingimuse jaoks vajalik test, mis seda rikub ainult seda.
Positiivset vastust andvatel nelinurkadel peaksid leiduma nii horisontaalseid, vertikaalseid kui ka mitmesuguse nurga all kaldu olevaid külgi.
Testida tuleb ka stabiilsust sisendi ümberjärjestamise suhtes. Trapetsi ülesandes peab esinema ka sisend, kus kujund on rööpkülik.

**6.2.14** Ruutude arv
Ruuduks olemist kontrolliva osa testimise kohta vt eelmise ülesande soovitusi.
Testida tuleks punktide arvu kasvatades: $n = 4, 5, 6, \ldots$ , kuni Teie programmi võimaluste piirini. Teatud piirini on võimalik õige vastus ka ilma programmi kasutamata välja arvutada.
Suuremate testide saamiseks võib punktid genereerida juhuslikult, aga ka väiksemate testide punktikobaraid tasandile erinevatesse kohtadesse paljundades.
Kui Teie lahendus Teeb lisaks punktineliku poolt antud kujundi moodustamisele ja peamisele loendamistsüklile veel muid teisendusi (näiteks järjestab punktid), siis vajavad ka programmi vastavad osad testimist.

# VII klass

## 7.1 Lineaarsed võrrandid ja võrratused

**7.1.1** Võrrandi $ax = b$ lahendamine
Testid võiksid sisaldada juhte, kus $a$ ja $b$ on:
täisarvud/harilikudmurrud/segaarvud/kümnendmurrud (sealjuures ka erinevat sorti);
arvudes $a$ ja $b$ esinevad täisarvulised osad on pikkusega 1, 2, 3,… ;
vastus vajab taandamist/segaarvuks teisendamist.

**7.1.2** Tundmatuga liikmed vasakule ja vabaliikmed paremale
Testid võiksid sisaldada juhte, kus $a$ ja $b$ on:
Lineaarliikmed/vabaliikmed on juba õigel pool/ vajavad üleviimist;
lineaarliikmete/vabaliikmete arv on 0/1/2/suurem.

**7.1.3** Sarnaste liikmete koondamine
Testid võiksid sisaldada juhte, kus
erinevates liikmetes $ax$ on kordajad $a$ erineva kujuga:

täisarvud/harilikudmurrud/segaarvud/kümnendmurrud);
vastus vajab taandamist/segaarvuks teisendamist.

**7.1.4** Vabanemine murdudest lineaarvõrrandis
Testid peaksid sisaldama erineva kujuga võrrandeid (erinevaid kujusid võiks saada näiteks õpikutest, muutes mingid osad murruliseks).

**7.1.5** Sulgude avamine
Testide võrrandid peaksid sisaldama 1/ 2/… taset sulge. Vt ka üles 3 testimissoovitusi.

**7.1.6** Lineaarvõrrandi lahendamine standardalgoritmiga
Testid peaks sisaldama kõigi õpikutes esinevate kujudega võrrandeid.

**7.1.7** Lineaarvõrrandid: samasus ja vastuoluline võrrand
Testid peavad sisaldama nii peajuhu, vastuolulisi kui samasuseks lihtsustuvaid võrrandeid.

**7.1.8** Absoluutväärtusega lineaarvõrrand
Testiida võiks absoluutväärtuse esinemiste arvuga 1, 2, 3, … , variandi b) korral samuti 1,2,3,… erineva absoluutväärtuse all oleva avaldisega. Varieerida tuleks ka võrrandi kuju.

**7.1.9** Võrratuse tõeväärtuse leidmine
Testid peaksid sisaldama võrratusi kõigi võrratusmärkidega. Kui muutuja väärtuse asendamiseks ja võrratuse poolte väärtuste arvutamiseks ei kasutata juba varem testitud alamprogramme, siis tuleks programmi nende jaoks kasutada vastavate ülesannete testimissoovitusi.

**7.1.10** Tehted võrratuse ja arvuga
Testid peaksid sisaldama kõiki tehtemärke koos negatiivse ja positiivse $a$ väärtusega.

**7.1.11** Lineaarse võrratuse lahendamine
Võrratuste lahendamine koosneb tavaliselt kahest etapist:
1) asendatakse võrratuse märk võrdusmärgiga ja leitakse kohad, kus võrratuse asemel kehtib võrdus,
2) leitakse, millistes leitud punktide vahelistes intervallides kehtib nõutud võrratus.
Esimese etapi töö testimiseks saab kasutada lineaarvõrrandite lahendamise programmide testimise meetodeid. Teise etapi korrektsuse kontroll on suhteliselt lihtne, sest lineaarse võrratuse puhul valitakse seal üks kahest poolteljest ja tehakse otsus punkti kohta, kus kehtib võrdus.

**7.1.13** Kahe tundmatuga lineaarne süsteem liitmisvõttega

# VIII klass

## 8.1 Üksliikmed

**8.1.1** Üksliikmete korrutamine
Testid võiks sisaldada juhte, kus:
muutujaid on 1/2/3/… , sealhulgas erinevates tegurites samad/erinevad;
muutuja astmenäitaja on 1/2/3/…/puudub;
tegureid on 2/3/… ;
esineb miinus korrutise ees/sulgudes;
kordaja on täisarv/harilik murd/kümnendmurd.

**8.1.2** Üksliikmete astendamine
Testid võiks sisaldada juhte, kus
muutujaid on 1/2/3/… ;

muutuja astmenäitaja on 1/2/3/…/puudub;
$n = 0, 1, 2, \dots$ .

**8.1.4** Üksliikmete jagamine
Testid võiks sisaldada juhte, kus jagatavas/jagajas/jagatises
muutujaid on 1/2/3/… ;
muutuja astmenäitaja on 1/2/3/…/puudub;
kordaja puudub/on positiivne/negatiivne/täisarv/harilik murd/kümnendmurd;
jagatises esineb/ei esine negatiivne astmenäitaja.

## 8.2 Hulkliikmed

Hulkliikmetes olevate üksliikmete töötlemiseks võiks kasutada jaotise 8.1 ülesannete lahendusi ja testimiseks seal antud soovitusi.

**8.2.5** Üksliikmete SÜT toomine sulgude ette
Testid võiks sisaldada juhte, kus mingis üksliikmes
kordaja puudub/on positiivne/negatiivne;
muutujad üksliikmetes on samad/erinevad osaliselt/ühisosata
astmenäitaja on 0/1/suurem/puudub
kordajate SÜT kohta vt 5.2.17;
muutujate osa SÜT – analoogilised juhud.

**8.2.8** Rühmitamisvõte hulkliikme tegurdamiseks
Testimine oleneb siin sellest, kui keeruline rühmitamine on programmis realiseeritud.
Testid võiksid sisaldada ka juhte, kus
ühesugused muutujate astmed on erinevalt esitatud (aste 0, 1);
liikme jaotamist summaks on/ei ole vaja.

**8.2.9** Abivalemite kasutamine tegurdamiseks
Testid peaks sisaldama näiteid kõigi programmis realiseeritud abivalemite kohta ja ka juhte, kus ükski abivalem ei sobi.

**8.2.10** Ühe muutuja hulkliikmete jagamine
Testid peaks sisaldama juhte, kus
jagatav jagub täpselt/jäägiga;
jagatava/jagaja pealiikme kordaja on positiivne/negatiivne;
jagatavas/jagajas/jagatises puudub mingi astmega liige
jagatise kordajad on täisarvud/esineb murde.

# 5 Kirjandus


1. *MathWorld.* http://mathworld.wolfram.com/Circle-LineIntersection.html
2. G. J. Myers. *The Art of Software testing.* John Wiley & Sons, 1979.
3. R.Palm. *Diskreetse matemaatika elemendid.* Tartu, 2003. http://kodu.ut.ee/~reimo_p/teosed/dme/dme.pdf
4. A. Saha. *Doing math with Python.* No starch press, 2015.
5. *Stackexchange.* https://math.stackexchange.com/
6. T.Tennisberg, K. Gabrel. *Võistlusprogrammeerimise õpik, 1. osa.* Tartu Ülikool, 2017. https://www.teaduskool.ut.ee/sites/default/files/teaduskool/oppetoo/voistlusprogrammeerimine_i_osa.pdf
7. T.Tennisberg, K. Gabrel. Võistlusprogrammeerimise õpik, 2. osa. Tartu Ülikool, 2018. https://www.teaduskool.ut.ee/sites/default/files/teaduskool/oppetoo/voistlusprogrammeerimine_ii.pdf
8. Aplusix. https://aplusix.com/en/
9. MathXpert. https://www.helpwithmath.com/
10. T-algebra. http://math.ut.ee/T-algebra/