1. Homne kuupäev

Selle ülesande lahendamiseks tuleb kuupäeva esitusest kõigepealt aasta, kuu ja päev eraldi arvmuutujatesse parsida.

Seejärel võib ülesande lahendada koolist tuntud kirjaliku liitmise sarnase algoritmiga:

- suurendame kuupäeva ühe võrra;
- kui nüüd sai kuupäev suuremaks kui päevade arv antud kuus, siis järelikult oli tegemist kuu viimase päevaga; sel juhul paneme kuupäevaks 1 ja suurendame kuu numbrit;
- kui nüüd sai kuu number suuremaks kui 12, siis oli tegemist detsembri viimase päevaga; sel juhul paneme kuu numbriks 1 ja suurendame aastanumbrit.

Nüüd on õige aasta, kuu ja päev käes. Väljastamiseks võib arvud kõigepealt õige pikkusega sõnedeks teisendada ja siis tühikud nullidega asendada (nagu on tehtud lahenduses `hommelah1.pas`) või siis teha teisendus arvudest tekstiks käsitsi (nagu on tehtud lahenduses `hommelah2.pas`).

Mõnes keeles on võimalik ka standardvahendite abil sisend kohe arvmuutujatesse lugeda ja väljund õiges vormingus saada (nagu on tehtud lahenduses `hommelah3.c`).

Testid

1. Sisend 11.10.2000. Väljund 12.10.2000. Lihtne päeva liitmine. 2 punkti.
2. Sisend 30.10.5432. Väljund 31.10.5432. Kuu lõpu lähedal. 2 punkti.
3. Sisend 30.04.8765. Väljund 01.05.8765. Kuuvahetus. Ühekohalised arvud. 2 punkti.
4. Sisend 31.12.9000. Väljund 01.01.9001. Aastavahetus. Hiliseim lubatud sisend. 2 punkti.
5. Sisend 01.01.1000. Väljund 02.01.1000. Varaseim lubatud sisend. 2 punkti.
6. Sisend 28.02.1901. Väljund 01.03.1901. Tavaline lihtaasta. 2 punkti.
7. Sisend 28.02.3004. Väljund 29.02.3004. Tavaline liigaasta. 2 punkti.
8. Sisend 28.02.2500. Väljund 01.03.2500. Lihtaasta (100 reegel). 2 punkti.
9. Sisend 28.02.2000. Väljund 29.02.2000. Liigaasta (400 reegel). 2 punkti.
10. Sisend 29.02.2000. Väljund 01.03.2000. Mis saab neist, kes arvavad, et 29.02.2000 pole olemas? 2 punkti.

Kokku 20 punkti.

2. Punktid sirgel

Kuna ülesande tingimuse kohaselt on etteantud punktide arv maksimaalselt 100, siis võime iga punktipaari jaoks üles leida kõik punktid, mis asuvad selle paariga samal sirgel. Ülesande raskus seisnebki pigem selles, kuidas kontrollida, kas mingid kolm punkti A, B ja C asuvad samal sirgel.

Üks võimalus selleks on leida vektorid $\vec{AB}$ ja $\vec{AC}$ ning kontrollida nende kollineaarsust nende vastavate koordinaatide jagatiste võrdlemise teel. Näiteks, olgu $\vec{AB} = (x_1, y_1)$ ja $\vec{AC} = (x_2, y_2)$. Siis punktid A, B ja C on ühel sirgel parajasti siis, kui $\frac{x_1}{x_2} = \frac{y_1}{y_2}$. Kui vektor $\vec{AC}$ on horisontaalne või vertikaalne, siis on üks tema koordinaatidest on 0 ja seetõttu peame neid juhte eraldi vaatlema. Lisaks tuleb ujukomaarvude võrdlemisel arvestada sellega, et nendega arvutamine on mõnikord ebatäpne.

Teine võimalus on leida vektorkorrutis $\vec{AB} \times \vec{AC}$. Üldiselt on kahe vektori vektorkorrutis defineeritud 3-mõõtmelises ruumis, ja see on võrdne järgmise determinandiga:

$$\begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \end{vmatrix} = (y_1 \cdot z_2 - y_2 \cdot z_1) \cdot \vec{i} + (z_1 \cdot x_2 - x_1 \cdot z_2) \cdot \vec{j} + (x_1 \cdot y_2 - y_1 \cdot x_2) \cdot \vec{k},$$

kus $\vec{i}$, $\vec{j}$ ja $\vec{k}$ on vastavalt x, y ja z-telje suunalised ühikvektorid: $\vec{i} = (1, 0, 0)$, $\vec{j} = (0, 1, 0)$, $\vec{k} = (0, 0, 1)$). Kuna meie vektorid on tasandil, siis $z_1 = z_2 = 0$ ja determinant lihtsustub kujule $\vec{AB} \times \vec{AC} = (x_1 \cdot y_2 - y_1 \cdot x_2) \cdot \vec{k}$. Teisalt on teada, et vektorkorrutise $\vec{AB} \times \vec{AC}$ pikkus avaldub tegurite pikkuste kaudu:

$$|\vec{AB} \times \vec{AC}| = |\vec{AB}| \cdot |\vec{AC}| \cdot sin(\text{nurk } \vec{AB} \text{ ja } \vec{AC} \text{ vahel}).$$

Kuna ülesande tingimuste kohasel on punktid paarikaupa erinevad, järeldub sellest, et $|\vec{AB}| > 0$ ja $|\vec{AC}| > 0$ ja seega $|\vec{AB} \times \vec{AC}| = 0$ parajasti siis, kui nurk nende vektorite vahel on $0°$ või $180°$, ehk siis, kui need vektorid on kollineaarsed. Järelikult, kontrollimaks, kas punktid A, B ja C on ühel sirgel, peame vaatama, kas $x_1 \cdot y_2 - y_1 \cdot x_2 = 0$.

Rohkem midagi keerulist ei olegi. Punktikolmikute genereerimisel ei tohi unustada, et antud võib olla vähem kui 3 punkti, mille korral tulebki need punktid väljastada.

Testid

1. $N = 1$. $K = 1$. minimaalne test, vastus ühene. 2 punkti.
2. $N = 2$. $K = 2$. peaaegu minimaalne test, vastus ühene. 2 punkti.
3. $N = 3$. $K = 2$. suured arvud, vastus pole ühene. 2 punkti.
4. $N = 11$. $K = 4$. mitu punkti ühel sirgel, nulliga jagamise oht, vastus pole ühene. 2 punkti.
5. $N = 15$. $K = 6$. mitu punkti ühel sirgel, vastus ühene. 2 punkti.
6. $N = 50$. $K = 50$. kõik punktid ühel sirgel, vastus ühene. 2 punkti.
7. $N = 75$. $K = 6$. suur juhuslik test, vastus ühene. 2 punkti.
8. $N = 100$. $K = 100$. maksimaalne test, kõik punktid ühel sirgel, vastus ühene. 2 punkti.
9. $N = 100$. $K = 7$. maksimaalne juhuslik test, väikesed arvud, vastus ühene. 2 punkti.
10. $N = 100$. $K = 2$. maksimaalne juhuslik test, suured arvud, vastus pole ühene. 2 punkti.

Kokku 20 punkti.

3. Ratsukäik

Nagu igasuguseid lühima tee otsimise ülesandeid, tuleks ka seda lahendada "laiuti otsimise" algoritmiga.

Üks lihtsam viis selle algoritmi realiseerimiseks on teha mällu laua kaart, kus hoiame iga välja jaoks selle kaugust sihtväljast. Selleks märgime sihtvälja kauguseks 0. Edasi märgime 1 kõikidesse väljadesse, kust saab ühe ratsukäiguga sihtväljale; arvu 2 kõikidesse väljadesse, mis pole veel märgitud ja kust saab ühe ratsukäiguga mõnele 1-ga märgitud väljale; ja nii edasi, kuni jõuame lähteruuduni. Nüüd on teada vajalik käikude arv.

Käikude endi leidmiseks jätab failis `ratsulah1.pas` toodud lahendus iga välja märkimisel meelde, milliselt väljalt me sinna jõudsime. Siis on käikude leidmiseks vaja vaid järgida kauguse otsimisel maha pandud märke lähteväljast sihtväljani. Kui jõuame sihtväljale, oleme eesmärgi saavutanud ja võime lõpetada.

Failis `ratsulah2.pas` toodud lahendus konstrueerib käikude jada ainult kauguste kaardi abil. Esiteks väljastame lähtevälja, siis vaatleme lähteväljast ratsukäigu kaugusel olevaid. Nende hulgas peab olema vähemalt üks, mille kaugus sihtväljast on ühe võrra väiksem. Järgmisena väljastamegi selle ja jätkame otsimist tema naabrite hulgast. Jätkame, kuni jõuame sihtväljale.

Just ratsu teekonna väljatrükkimise lihtsustamiseks on mõlemal juhul kasulikum kaugusi otsida sihtvälja poolt lähtevälja poole, mitte vastupidi.

Testid

1. $d5 \to d5$. 0 käiku. Minimaalne test, vastus ühene. 2 punkti.
2. $d3 \to e5$. 1 käik. Peaaegu minimaalne test, vastus ühene. 2 punkti.
3. $d4 \to e5$. 2 käiku. Vastus pole ühene. 2 punkti.
4. $b1 \to c2$. 2 käiku. Vastus ühene. 2 punkti.
5. $a1 \to d7$. 3 käiku. Vastus ühene. 2 punkti.
6. $g6 \to a5$. 3 käiku. Vastus pole ühene. 2 punkti.
7. $a4 \to a5$. 3 käiku. Vastus pole ühene. 4 punkti.
8. $b2 \to g3$. 4 käiku. Vastus pole ühene. 4 punkti.
9. $a4 \to h6$. 5 käiku. Vastus pole ühene. 4 punkti.
10. $d8 \to b1$. 5 käiku. Vastus pole ühene. 4 punkti.
11. $h7 \to a1$. 5 käiku. Vastus pole ühene. 4 punkti.
12. $a7 \to h1$. 5 käiku. Vastus pole ühene. 4 punkti.
13. $a1 \to h8$. 6 käiku. Maksimaalne test, vastus pole ühene. 4 punkti.

Kokku 40 punkti.

4. Lototron

Esiteks tuleb ülesande tekstist korralikult aru saada. Iga ringi jaoks antud kaks arvu: periood S_i näitab, kui kaua aega kulub ühe ringi tegemiseks, algfaas P_i näitab, millal on i. ring esmakordselt tunneli positsioonis. Seega on i. ring tunneli positsioonis ajahetkedel

$$t = P_i + k \cdot S_i \quad (k = 0, 1, \dots).$$

Teine tähelepanek on, et kui kuul on parasjagu i. ringil, siis huvitab meid ainult see, millal on tunneli positsioonis kohakuti nii i. kui ka $(i+1)$. ringi märgitud sektorid. See saab toimuda ajahetkel T_i, kus T_i rahuldab tingimusi

$$\begin{aligned} T_i &= P_i + k_1 \cdot S_i, \\ T_i &= P_{i+1} + k_2 \cdot S_{i+1} \end{aligned}$$

sobivate k_1 ja k_2 korral.

Kui sellel süsteemil leidub lahend T_i, on lahendid ka kõik arvud kujul $T_i + \text{VÜK}(S_i, S_{i+1})$. Selleks, et lahend leiduks, peavad P_i ja P_{i+1} andma $\text{SÜT}(S_i, S_{i+1})$-ga jagamisel sama jäägi.

Kui on teada, et lahend leidub, võib vaadata suurusi $P_i + k_1 \cdot S_i$ ja $P_{i+1} + k_2 \cdot S_{i+1}$ juhul $k_1 = k_2 = 0$. Kui need arvud ei ole võrdsed, siis tuleb suurendada 1 võrra vastavalt kas arvu k_1 või k_2. Kui arvud on võrdsed, ongi sobiv T_i leitud.

Ülesande lahendamiseks tuleb seega vaadelda ringe numbrite järjerkorras ja igal sammul leida, kas ringid i ja $i+1$ satuvad tunnelis kohakuti. Kui ei, siis võib väljastada tulemuse (kuul jõuab ringile i, kuid mitte ringile $i+1$). Kui jah, tuleb leida esimene selline ajahetk pärast kuuli kukkumist i. ringile. Selle ajahetke võib arvutada valemiga

$$t_{i+1} = t_i + \big((T_i - t_i) \bmod \text{VÜK}(S_i, S_{i+1})\big)$$

kus $a \bmod b$ tähendab jääki, mis tekib arvu a jagamisel arvuga b.

Testid

1. $N = 1$. $I = 1$, $T = 0$. Minimaalne test. 1 punkt.
2. $N = 5$. $I = 3$, $T = 38$. Ringidel 3 ja 4 on perioodid võrdsed, faas erinev. 1 punkt.
3. $N = 4$. $I = 3$, $T = 11$. Ringi 4 periood on ringi 3 perioodi kordne. 1 punkt.
4. $N = 6$. $I = 6$, $T = 396$. Kuul läheb lõpuni, esimesel ja viimasel ringil ainult 1 sektor. 1 punkt.
5. $N = 5$. $I = 4$, $T = 225$. Ringi 5 periood on ringi 4 perioodi kordne. 1 punkt.
6. $N = 5$. $I = 5$, $T = 26$. Kuul läheb lõpuni. 1 punkt.
7. $N = 5$. $I = 5$, $T = 32\,011$. Natuke suurem test. 1 punkt.
8. $N = 20$. $I = 15$, $T = 18\,014$. Juhuslik test. 1 punkt.
9. $N = 100$. $I = 97$, $T = 4850$. Vaheldumisi `99 98`, `2 0` ja `2 1` (välja arvatud 98. ring). 1 punkt.
10. $N = 200$. $I = 1$, $T = 0$. Kuul peatub kohe esimesel ringil. 2 punkti.
11. $N = 995$. $I = 995$, $T = 0$. Kuul läheb kohe lõppu välja. 2 punkti.
12. $N = 200$. $I = 199$, $T = 169\,820$. Keskmine test. 2 punkti.
13. $N = 100$. $I = 100$, $T = 497\,183$. Järjestikuste ringide perioodid ühistegurita, sammhaaval modelleerimisega lahendus jääb ajahätta. 2 punkti.
14. $N = 100$. $I = 100$, $T = 970\,348$. Vaheldumisi `100` k ja `99` m. 2 punkti.
15. $N = 1\,000$, $I = 1\,000$, $T = 2\,534\,705$. Üsna suur test. 3 punkti.
16. $N = 1\,000$. $I = 1\,000$, $T = 499\,500$. Vaheldumisi `1000` k ja `1 0`. 4 punkti.
17. $N = 1\,000$. $I = 998$, $T = 164\,325\,327$. Suur juhuslik test. 7 punkti.
18. $N = 1\,000$. $I = 1\,000$, $T = 997\,003\,498$. Vaheldumisi `1000` k ja `999` m. 7 punkti.

Kokku 40 punkti.

5. XOR-šifreerimine

Lihtne viis selle ülesande lahendamiseks on teha N erinevat oletust: esmalt oletame, et teksti T šifreerimine võtmega K_1 andis salateksti E_1, siis, et see andis E_2, E_3, ..., E_N. Iga hüpoteesi $T \oplus K_1 = E_i$ kontrollimiseks saame kõigepealt vastavalt ülesande tekstis antud reeglile avaldada $T = E_i \oplus K_1$ ja seejärel leida T oletatavale väärtusele T^i vastava krüptogrammide komplekti $E_1^i = T^i \oplus K_1$, $E_2^i = T^i \oplus K_2$, ..., $E_N^i = T^i \oplus K_N$. Nii saame iga hüpoteesi $T = T^i$ jaoks sellele vastava krüptogrammide hulga $\mathcal{E}^i = \{E_1^i, E_2^i, \ldots, E_N^i\}$. Edasi jääb veel üle kontrollida, milline hulkadest $\mathcal{E}^i$ on võrdne tegeliku krüptogrammide hulgaga $\mathcal{E} = \{E_1, E_2, \ldots, E_N\}$.

Eelnevast on näha, et kontrollida tuleb kokku maksimaalselt N hüpoteesi. Ühe hüpoteesi kontrollimiseks tuleb kõigepealt leida N krüptogrammi (igaühe leidmiseks kulub $8L$ tehet, kokku seega $8NL$ tehet), ja seejärel kontrollida kahe N-elemendilise hulga võrdsust, milleks on kõige lihtsam elemente paarikaupa võrrelda (iga võrdlus kulutab ülimalt $8L$ tehet, kokku seega $8N^2L$ tehet). Kokku kulub maksimaalselt $N(8NL + 8N^2L) = O(N^3L)$ tehet.

Arvestades, et maksimaalses testis $N = 1\,000$ ja $L = 28$, tundub seda olevat üsna palju, aga tegelikult võib suurema osa hüpoteeside kontrollimise lõpetada juba paari esimese krüptogrammi koostamise järel ja seetõttu kulutab failis `xorlahx.pas` toodud programm selle testi lahendamiseks 100 MHz protsessoril kõigest paar sekundit.

Veel võiks märkida, et paaritu N korral on sellel ülesandel alati ühene lahendus kujul

$$T = K_1 \oplus K_2 \oplus \cdots \oplus K_N \oplus E_1 \oplus E_2 \oplus \cdots \oplus E_N.$$

Tõepoolest, kuna iga E_j jaoks leidub selline K_i, et $E_j = T \oplus K_i$, võime selle summa ümber kirjutada kujule

$$T = K_1 \oplus K_2 \oplus \cdots \oplus K_N \oplus (T \oplus K_1) \oplus (T \oplus K_2) \oplus \cdots \oplus (T \oplus K_N).$$

Pannes tähele, et mistahes arvu x korral kehtib $x \oplus x = 0$ ja $x \oplus 0 = x$, saame, et kõik liidetavad, mis esinevad summas paarisarv kordi, kaovad lihtsalt ära. Kuna iga võti K_i esineb eeltoodud summas täpselt kaks korda (üks kord ise ja teine kord mingi E_j avaldises), kaovad nad puhtalt ära. Kuna T esineb summas N korda, siis paaritu N korral moodustavad tema $N - 1$ esinemist paarid ja kokkuvõttes jääbki summa lihtsustamisel järele täpselt üks T. (Paaris N korral kaovad ära kõik T esinemised, mistõttu paaris N korral on eeltoodud summa alati null.)

Testid

1. $N = 1$, $K = 1$. Minimaalne test. Vastus ühene. 4 punkti.
2. $N = 2$, $K = 2$. Vastus pole ühene. 4 punkti.
3. $N = 3$, $K = 8$. Vastus ühene. 4 punkti.
4. $N = 11$, $K = 8$. Vastus ühene. 4 punkti.
5. $N = 12$, $K = 12$. Vastus ühene. 4 punkti.
6. $N = 99$, $K = 12$. Vastus ühene. 4 punkti.
7. $N = 100$, $K = 12$. Vastus ühene. 4 punkti.
8. $N = 199$, $K = 24$. Vastus ühene. 4 punkti.
9. $N = 200$, $K = 24$. Vastus ühene. 4 punkti.
10. $N = 1000$, $K = 28$. Maksimaalsed read, mis mahuvad 256-baidistesse sõnedesse. Vastus ühene. 4 punkti.

Kokku 40 punkti.

6. Bitinihked

Uurime, kuidas sõltub 4-kohalisest esialgsest arvust A sellest saadav 7-kohaline tulemus B. Olgu meil kahendarv $A = abcd$, kus a, b, c, d on kahendnumbrid (0 või 1). Kirjutame üksteise alla arvu A nihked s_0, s_1, s_2, s_3 ning lõppsumma B (esialgu küll tundmatul kujul):

$$\begin{array}{rccccccc}
s_0 = & & & & a & b & c & d \\
s_1 = & & & a & b & c & d & 0 \\
s_2 = & & a & b & c & d & 0 & 0 \\
s_3 = & a & b & c & d & 0 & 0 & 0 \\
B = & o & p & q & r & s & t & u
\end{array}$$

Ülesandes on summad x_i defineeritud nii, et $x_0 = s_0$, $x_1 = s_0 \oplus s_1$, $x_2 = (s_0 \oplus s_1) \oplus s_2$ ja $s_3 = ((s_0 \oplus s_1) \oplus s_2) \oplus s_3$. Antud neljakohalise juhu jaoks $B = x_3$. Lisaks on veel teada, et $\oplus$-tehe ei tekita bitiülekandeid, mistõttu võime oma näites summa $B = ((s_0 \oplus s_1) \oplus s_2) \oplus s_3$ asendada iga kahendkoha jaoks eraldi summana:

$$\begin{aligned}
u &= ((d \oplus 0) \oplus 0) \oplus 0 \\
t &= ((c \oplus d) \oplus 0) \oplus 0 \\
s &= ((b \oplus c) \oplus d) \oplus 0 \\
r &= ((a \oplus b) \oplus c) \oplus d \\
q &= (a \oplus b) \oplus c) \\
p &= a \oplus b \\
o &= a
\end{aligned}$$

Mistahes kahendbiti a puhul kehtib võrdus $a \oplus 0 = a$. Selles on lihtne veenduda, sest definitsiooni kohaselt $1 \oplus 0 = 1$ ja $0 \oplus 0 = 0$. Seega lihtsustuvad arvu B numbrite avaldised:

$$\begin{aligned}
u &= d \\
t &= c \oplus d \\
s &= (b \oplus c) \oplus d \\
r &= ((a \oplus b) \oplus c) \oplus d \\
q &= (a \oplus b) \oplus c) \\
p &= a \oplus b \\
o &= a
\end{aligned}$$

Hakkame nüüd mõtlema, kuidas arvu B numbrite abil taastata arv A. Märkame, et arvude A ja B viimased kahendnumbrid peavad ühtima ($u = d$). Võrrandis $t = c \oplus d$ on number t arvu B eelviimane number ning number d on meil juba käes, mistõttu peaksime saama sellest võrrandist määrata numbri c.

Võtame lähema vaatluse alla võrrandi $m = x \oplus n$ ning üritame sellest avaldada x väärtuse. Kui $n = 0$, siis ilmselt $x = m$. Kui aga $n = 1$, siis $\oplus$-tehte definitsiooni tõttu x ja m ei saa olla võrdsed, seega peavad olema erinevad, ehk $x = 1 - m$. Edasi on lihtne veenduda, et iga a puhul kehtib võrdus $a \oplus 1 = 1 - a$ (tõepoolest, $1 \oplus 1 = 0$ ja $0 \oplus 1 = 1$). Seega võime võrrandi $m = x \oplus n$ lahendi kirjutada kujul $x = m \oplus n$.

Kasutades $\oplus$-tehte äsja leitud omadust, võime eelpool kirja pandud võrrandid teisendada kujule:

$$\begin{aligned} u &= d \\ t \oplus d &= c \\ (s \oplus d) \oplus c &= b \\ ((r \oplus d) \oplus c) \oplus b &= a \\ (q \oplus c) \oplus b &= a \\ p \oplus b &= a \\ o &= a \end{aligned}$$

Kui nüüd hakata arvu A numbreid tagantpoolt leidma, siis sõltub iga parasjagu arvutatav number vaid juba leitud numbritest. Esimesed neli võrrandit määravad üheselt arvu A, järgneva kolme võrrandi paikapidavuses tuleks aga kindlasti veenduda, sest mitte igale arvule B pole võimalik vastavusse panna arvu A.

On kasulik teada, et selles ülesandes vaadeldav $\oplus$-tehe ehk "välistav või" ehk XOR on paljudesse keeltesse juba sisse ehitatud. Kui me hoiame arvuti mälus täisarve `x` ja `y` (näiteks C's `int`-tüüpi või Pascalis `integer`-tüüpi muutujad), siis on nende arvude $\oplus$-liitmine üksainus operatsioon (näiteks C's `z = x ^ y;` või Pascalis `z := x xor y;`). Siiski peame selle ülesande lahenduses $\oplus$-arvutused tegema "käsitsi", sest ülesandes lubatud arvude pikkused ületavad standardsete täisarvutüüpide pikkusi.

Testid

1. $n = 1$. Vastus JAH, 0. Minimaalne positiivne test. 4 punkti.
2. $n = 5$. Vastus JAH, 11111. Väike positiivne test. 4 punkti.
3. $n = 6$. Vastus JAH, 111000. Väike positiivne test. 4 punkti.
4. $n = 10$. Vastus JAH, 0000111001. Nii A kui ka B algavad nullidega. 4 punkti.
5. $n = 96$. Vastus JAH. Pikk juhuslik positiivne test. 4 punkti.
6. $n = 100$. Vastus JAH. Maksimaalne positiivne test, kõik bitid nullid. 4 punkti.
7. $n = 2$. Vastus EI. Minimaalne negatiivne test. 4 punkti.
8. $n = 10$. Vastus EI. Vastuoluline vaid B vasakpoolseim number. 4 punkti.
9. $n = 13$. Vastus EI. Vastuoluline vaid B parempoolseim number. 4 punkti.
10. $n = 60$. Vastus EI. Keskmise suurusega juhuslik negatiivne test. 4 punkti.

Kokku 40 punkti.