

### 3. Neutriinoradar (neu)

1 sek 100 punkti

Mõne aja eest leidsid astronoomid tähe  $\zeta$ -2019A, millel on  $N$  planeeti. Nüüd avastati, et üks neist planeetidest on eluks kõlbulik. Täht on aga nii kaugel, et selle planeete saab uurida ainult võimsa neutriinoradariga.

Radar töötab järgmiselt. Algul saadab radar tähesüsteemi suunas neutriinovoo, mida iseloomustavad täisarvulised parameetrid  $X$  ja  $Y$ . Tähesüsteemilt tagasi peegeldunud signaali analüüsid on võimalik sellest eraldada täisarv  $B = \gcd(X, A + Y)$ , kus  $A$  on eluks kõlbliku planeedi järjekorranumber (planeedid on nummerdatud  $1 \dots N$  alates tähele  $\zeta$ -2019A lähimast) ja  $\gcd(X, Y)$  tähistab arvude  $X$  ja  $Y$  suurimat ühistegurit, see tähendab suurimat sellist täisarvu, millega nii  $X$  kui  $Y$  jaguvad jäägita. Kuna neutriinovoo väljasaatmine on väga energiakulukas, võib seda teha maksimaalselt 40 korda.

Nüüd on astronoomidel vaja programmi, mis juhiks radarit ja tuvastaks eluks kõlbliku planeedi numbrit. Programmi tuumaks on funktsioon `int Locate(int n)`. Seda funktsiooni kutsutakse välja üks kord, andes parameetrina tähe  $\zeta$ -2019A planeetide arvu  $N$ , ja funktsioon peab tagastama eluks kõlbliku planeedi numbrit  $A$  ( $1 \leq A \leq N$ ).

Funktsioon `Locate` võib oma töö käigus kasutada funktsiooni `int Scan(int x, int y)`, mis käivitab radari parameetritega  $X$  ja  $Y$  ( $1 \leq X \leq 10^9$ ,  $0 \leq Y \leq 10^9$ ) ning tagastab radarisignaali analüüsi tulemuse  $B = \gcd(X, A + Y)$ . Funktsiooni `Scan` võib kasutada maksimaalselt 40 korda.

Testimiskeskkonnas on antud näitefailid, kus vajalikud funktsioonid on juba kirjeldatud ja vaja on lisada ainult funktsiooni `Locate` realisatsioon. Lisaks võib lahenduse faili kirjutada ka oma funktsioone. Oma lahenduse oma arvutis testimiseks on ka hindamisprogrammi näide, mille sisendi ja väljundi kirjeldus on toodud allpool (serveris on kasutusel teine hindamisprogramm, mis kontrollib ka lahenduse tagastatud vastuse õigsust). Oma lahenduse oma arvutis kompileerimiseks ja testimiseks:

Keel	Lahendus	Kompileerimine ja käivitamine
C++	<code>neu.cpp</code>	<code>g++ -o grader grader.cpp neu.cpp ./grader</code>
Java	<code>neu.java</code>	<code>javac -cp . grader.java neu.java java -cp . grader</code>
Python	<code>neu.py</code>	<code>python grader.py</code>

**Sisend.** Tekstifaili `neusis.txt` ainsal real on kaks täisarvu: tähe  $\zeta$ -2019A planeetide koguarv  $N$  ( $1 \leq N \leq 10^9$ ) ja eluks kõlbliku planeedi number  $A$  ( $1 \leq A \leq N$ ).

**Väljund.** Tekstifaili `neuval.txt` väljastatakse hindamisprogrammi ja funktsiooni `Locate` vahelise suhtluse logi.

<b>Näide.</b>	<code>neusis.txt</code>	<code>neuval.txt</code>
	<code>5 3</code>	<code>Locate(5) Scan(3, 6) = 3 Scan(4, 2) = 1 Scan(2, 1) = 2 Scan(4, 5) = 4 result = 3</code>

Tähel  $\zeta$ -2019A on 5 planeeti ja eluks kõlbulik on planeet number 3. Funktsioonikutse `Scan(3, 6)` tagastab  $\gcd(3, 3 + 6) = \gcd(3, 9) = 3$ ; `Scan(4, 2)` tagastab  $\gcd(4, 3 + 2) = \gcd(4, 5) = 1$ ; `Scan(2, 1)` tagastab  $\gcd(2, 3 + 1) = \gcd(2, 4) = 2$ ; `Scan(4, 5)` tagastab  $\gcd(4, 3 + 5) = \gcd(4, 8) = 4$ . Pärast seda tagastab funktsioon `Locate` hindamisprogrammile väärtuse 3.

**Hindamine.** Selles ülesandes on testid jagatud gruppidesse. Iga grupi eest saavad punkte ainult need lahendused, mis läbivad kõik sellesse gruppi kuuluvad testid. Gruppides kehtivad järgmised lisatingimused:

1. (11 punkti)  $N \leq 40$ ;
2. (8 punkti)  $N \leq 80$ ;
3. (12 punkti)  $N \leq 150$ ;
4. (9 punkti)  $N \leq 10^3$ ;
5. (18 punkti)  $N \leq 10^4$ ;
6. (23 punkti)  $N \leq 2 \cdot 10^5$ ;
7. (19 punkti) lisapiirangud puuduvad.