

Kartojimas. Šakotieji algoritmai

1 lygio užduotys

1. **Lenktynės.** Trys draugai susirungė žaisdami kompiuterinį žaidimą - mašinų lenktynes. Važiavo tris ratus ir fiksavo, per kiek laiko kiekviena mašina įveikė ratą.

Pradinių duomenų faile **lenktynes_data.txt** yra trys eilutės. Kiekvienoje eilutėje yra dalyvavusiojo žaidime vardas (naudokite duomenų tipą string) ir kiekvieno rato įveikimo laikas (sekundėmis).

Rezultatų faile **lenktynes_res.txt** turi būti įrašytos prizinės vietos nuo pirmos iki trečios su žaidėjų vardais.

Pastaba: Laikykite, kad lygiųjų niekada nebus.

lenktynes_data.txt	lenktynes_res.txt
Aidas 536 546 587	1) Aidas
Rokas 587 566 573	2) Lukas
Lukas 556 597 536	3) Rokas

2. **Skaičių karas.** Skaičių šalyje gyvena vienetai ir nuliai. Kartą vienetai susipyko su nuliais ir norėdami įrodyti, kad yra geresni už nulius, nusprendė nesutarimus išspręsti kariaudami. Žinoma, kad iš viso, sudėjus nulių ir vienetų kariuomenes, į karą buvo pasiūstas a kiekis kareivių. Kadangi nulių populiacija kiek didesnė nei vienetų, todėl jie į karą išsiuntė daugiau karių nei vienetai, tačiau visiems yra žinoma, jog vienetų kariai yra 2 kartus stipresni už nulių. Kariuomenėms susitikus mūšio lauke, tapo žinoma, kiek karių sudaro nulių kariuomenę, šis skaičius lygus nul ($nul > 0$).

Parašykite programą, skaičiuojančią:

1. Kokia yra visos vienetų kariuomenės karinė galia **vGalia**, kai yra žinoma, jog vieno nulių kario galia yra **b**.
2. Padėkite pasauliui sužinoti, kas laimės šį karą – "NULIAI", "VIENETAI" ar "NIEKAS". Raskite rezultatą **rez**.

Pradiniai duomenys **a**, **b** ir **nul** yra sveikieji skaičiai. Duomenys įvedami tokia tvarka: **a**, **b**, **nul**.

Rezultatas **vGalia** yra sveikasis skaičius, o **rez** – string tipo tekstas. Jie išvedami vienoje eilutėje, atsikiriant vieną nuo kito tarpu. Atsakymas **rez** priklausomai nuo to, kas laimės karą yra lygus "NULIAI", "VIENETAI" arba "NIEKAS". Rezultatai išvedami tokia tvarka: **vGalia**, **rez**.

karas_data.txt	karas_res.txt
8 2 5	12 VIENETAI

3. **Žalos skaičiavimas.** Žaidime yra 3 žaidėjų klasės: karys, magas ir lankininkas. Kiekviena iš šių klasių turi skirtingą sugebėjimą, nuo kurio priklauso žala priešininkui. Sugebėjimai:

- Karys. Smūgio žala: **124 + 7%** priešininko turimų gyvybių.

- Magas. Burto žala: **240 + 120 jeigu priešininkas turi 2000 gyvybių ir daugiau.**
- Lankininkas. Strėlės žala: 180; **Lankininkai vienu šūviu šauna 3 strėles iš karto.**

Apskaičiuokite, kiek žalos padarys žaidėjas, žinodami jo klasę ir priešininko gyvybių skaičių.

Pradinių duomenų failo **zala_data.txt** pirmoje eilutėje pateikiama žaidėjo klasė, antroje - priešininko gyvybių skaičius.

Rezultatų faile **zala_res.txt** reikia išvesti žaidėjo padaromą žalą 2 skaitmenų po kablelio tikslumu.

zala_data.txt	zala_res.txt
Karys 1000	194.00

- 4. Stačiakampio ketvirtoji viršūnė.** Lukas konstruoja stačiakampį, kurio kraštinės yra lygiagrečios koordinatinių ašims. Berniukas pasirinko 3 stačiakampio viršūnių koordinates. Padėkite Lukui tinkamai pasirinkti ketvirtosios viršūnės koordinates.

Pradiniai duomenys įrašyti pradinių duomenų faile **staciakampis_data.txt**: Luko pasirinktų 3 stačiakampio viršūnių koordinatės x ir y (sveikieji skaičiai ne didesni už 100).

Rezultatų faile **staciakampis_res.txt** vienoje eilutėje turi būti rodomos 4-osios viršūnės koordinatės x ir y atskirtos tarpu.

staciakampis_data.txt	staciakampis_res.txt
5 5	7 7
5 7	
7 5	

- 5. Triliemenių slibinų kovos.** Pasakose slibinas dažniausiai vieną liemenį ir daug galvų, o Vytukas sugalvojo sukurti žaidimą, kuriame kovoja du triliemeniai slibinai. Vytukas kurs žaidimo dizainą ir ieško programuotojo, kuris suprogramuotų slibinų kovos baigtį. Vytuko sumanymas toks. Slibinai turi po 3 liemenis: kairinį, vidurinį ir dešinį. Kiekvienas liemuo turi tam tikrą galvų skaičių (ant vieno liemens ne daugiau kaip 1000). Kai slibinai pradeda kovoti, pirmiausia kovoja abiejų slibinų kairiojo liemens galvos. Jei abiejų slibinų kairiojo liemens galvų skaičiai yra vienodi, tuomet kova baigiasi lygiosiomis. Jei ne – į kovą įsitraukia viduriniojo liemens galvos ir kovoja pagal tas pačias taisykles, kaip ir kairiojo liemens galvos. Jei viduriniojo liemens galvų skaičius vienodas, tuomet kova irgi baigiasi lygiosiomis, jei ne – į kovą įsitraukia dešiniojo liemens galvos. Jei dešiniųjų galvų skaičius yra vienodas, tuomet abiejų slibinų jėgos lygios. Jei ne – laimi tas slibinas, kuris turi daugiau dešiniojo liemens galvų.

Parašykite programą, kuri nustatytų slibiną laimėtoją.

Pradiniai duomenys skaitomi iš failo **slibinas_data.txt** tokia tvarka: pirmojo slibino kairiojo, viduriniojo ir dešiniojo liemens galvos, po to antrojo slibino kairiojo, dešiniojo ir viduriniojo liemens galvos.

Rezultatai išvedami į failą **slibinas_res.txt**, nurodant slibiną laimėtoją. Jei laimėjo pirmasis slibinas turi būti išvedamas 1, jei antrasis turi būti išvedamas 2, jei abu – turi būti išvedama 1 2.

slibinas_data.txt	slibinas_res.txt
3 5 7 2 5 8	1

2 lygio užduotys

- 6. Kompasas.** Jonas gavo užduotį sukurti programą, kuri apskaičiuotų, koku mažiausiu kampu turi pasisukti kompasas rodyklė, kad iš dabartinės pozicijos pasiektų norimą poziciją.

Pradiniai duomenys: dabartinė rodyklės pozicija laipsniais ir norima rodyklės pozicija laipsniais (sveikieji skaičiai). Apskaičiuokite trumpiausią pokytį laipsniais tarp pradinės pozicijos ir norimos. Teigiamas skaičius parodo, jog kompasas rodyklė suksis pagal laikrodžio rodyklę, neigiama – prieš laikrodžio rodyklę. Jei pradinė ir galutinė kompasas rodyklės pozicijos sudaro 180 laipsnių kampą, tuomet kompasas rodyklė turėtų sukis pagal laikrodžio rodyklę.

kompasas_data.txt	kompasas_res.txt
315 45	90
180 270	90

- 7. Kriptografija.** Programuotojas užkodavo dviejų raidžių r1, r2 kodą. Tos dvi raidės gali būti "X", "Y" arba "Z". "X" kodas yra 0, "Y" – 1 0, "Z" – 1 1.

Pradinių duomenų faile **kriptografija_data.txt** įrašytas pirmosios raidės **r1** kodas, tada antrosios **r2** (atkreipkite dėmesį, kad įvedamų skaičių kiekis gali kisti priklausomai nuo raidžių).

Rezultatų faile **kriptografija_res.txt** išveskite atkurtą kodą. **r1, r2** – char (simbolių) tipo kintamieji, jie išvedami vienas nuo kito atskirti tarpu.

kriptografija_data.txt	kriptografija_res.txt
0 1 0	XY

- 8. Mikroprocesoriaus funkcija.** Kuriant naują mikroprocesorių, komanda sugalvojo pridėti naują funkciją, kuri galėtų konvertuoti bet kurio 8 bitų registro reikšmę į kitą skaičiavimo sistemą.

Komanda pasirinko parašyti funkciją, konvertuojančią 8 bitų registre saugomą dvejetainį kodą į dešimtainę sistemą.

Pradiniai duomenys bus bet kurio 8 bitų registro duomenys, tai yra bitai **b7, b6, b5, b4, b3, b2, b1, b0**. b7 bitas yra ženklo skiltis, kuri nurodo kokio ženklo skaičius saugojamas registre. Jeigu b7 lygus 0 tai reiškia kad registre saugomas teigiamas skaičius **tiesioginiame** kode. Jeigu b7 bitas lygus 1, tai registre saugoma neigiama reikšmė ir ji saugoma **papildomame** kode.

Dvejetainio kodo konvertavimas į dešimtainį vyks pagal tokį algoritimą:

1) jei reikia, registro duomenys iš papildomo kodo paverčiami į tiesioginį kodą. Bitai b6, b5, b4, b3, b2, b1 ir b0 invertuojami (jei b6 buvo 1, invertavę jį gausime 0 ir su kitais bitais tas pats mechanizmas). Gauti bitai padauginami iš reikšmės, kurią saugo: b6 * 64, b5 * 32, b4 * 16, b3 * 8, b2 * 4, b1 * 2, b0 * 1. Gautos reikšmės sudedamos ir prie galutinės reikšmės pridedamas 1;

2) tiesioginio kodo bitai padauginami iš reikšmės, kurią saugo. Gautos reikšmės sudedamos.

Rezultatas – dešimtainis skaičius (int tipo).

mikroprocesorius_data.txt	mikroprocesorius_res.txt
0 1 1 1 1 0 0 1	121

9. Simbolis? Raidė? Skaičius? Iš tekstinio failo **simbolis_data.txt** perskaitomas simbolis. Reikia patikrinti, ar įvestas simbolis yra lotyniškos abėcėlės raidė, ar skaitmuo, ar kitas simbolis. Rezultatai atitinkamai turi būti: "skaitmuo", "raide", "kitas simbolis". Rezultatai išvedami į failą **simbolis_res.txt**.

Iššūkis: atlikti užduotį sąlyginiame sakinyje netikrinant kiekvienos lotyniškos raidės ir skaitmenų atskirai. Papildomi simboliai įvedami laikant paspaustą mygtuką ALT ir rašant skaičius. Pvz.: *ALT+43*

Kodas	Simbolis
251	√
174	«
175	»
168	ı
236	∞
241	±

simbolis_data.txt	simbolis_res.txt
!	kitas simbolis
_	kitas simbolis
Æ	kitas simbolis
∞	kitas simbolis
2	skaitmuo
8	skaitmuo
i	raide
N	raide

10. Šviesoforas. Šviesoforas veikia pagal tokį algoritmą: kiekvienos valandos pirmąsias z minutes dega žalia šviesa, po to r minutes – raudona, po to vėl tris minutes žalia ir t. t. Žinoma, kiek minučių t (t – sveikasis skaičius) praėjo nuo valandos pradžios. Parašykite programą, kuri nustatytų, kokia šviesa dega.

Pradinių duomenų faile **sviesoforas_data.txt** įrašytos t , z ir r reikšmės. Rezultatų faile **sviesoforas_res.txt** turi būti įrašyta raidė **Z**, jei dega žalia šviesa, **R** – jei dega raudona, **Z-R** – žalia keičiasi į raudoną, **R-Z** – raudona keičiasi į žalią.

sviesoforas_data.txt	sviesoforas_res.txt
12 3 2	Z

3 lygio uždutys

11. Nutrinti skaičiai. Ant popieriaus lapo užrašyti keturi natūralieji skaičiai: a , b , s , d . Po to du iš jų buvo nutrinti (juos žymėsime nuliais). Reikia atkurti nutrintuosius skaičius, jeigu žinoma, kad yra likęs bent vienas iš skaičių a ir b ir kad skaičiai tenkino šitokias lygybes:

$$s = a + b;$$

$$d = a * b.$$

Pradinių duomenų faile **skaiciai_data.txt** įrašyti skaičiai a , b , s ir d . Rezultatų faile **skaiciai_res.txt** turi būti įrašyti skaičiai a , b , s , d , vienas nuo kito atskirti tarpo simboliu.

skaiciai_data.txt	skaiciai_res.txt
0 12 0 48	4 12 16 48

12. Japonų kalendorius. Senovės japonų kalendorių sudarė 60 metų ciklas. Visi metai cikle buvo sunumeruoti nuo 1 iki 60 ir suskirstyti poromis, kurių kiekviena turėjo savo spalvą (žalią, raudoną, geltoną, baltą ar juodą). Ciklo metų spalvos buvo paskirstytos taip:

- 1, 2, 11, 12, 21, 22, ..., 51, 52 metai – žalia spalva;
- 3, 4, 13, 14, 23, 24, ..., 53, 54 metai – raudona spalva;
- 5, 6, 15, 16, 25, 26, ..., 55, 56 metai – geltona spalva;
- 7, 8, 17, 18, 27, 28, ..., 57, 58 metai – balta spalva;
- 9, 10, 19, 20, 29, 30, ..., 59, 60 metai – juoda spalva.

Žinoma, kad naujasis 60 metų ciklas prasidėjo 1984-aisiais ir baigsis 2043-iaisiais metais; 1984-ieji ir 1985-ieji buvo žalios spalvos metai, 1986-ieji ir 1987-ieji buvo raudonos spalvos metai, 2043-ieji bus juodos spalvos metai.

Pradinių duomenų faile **kalendorius_data.txt** įrašyti metai m . Rezultatų faile **kalendorius_res.txt** turi būti išvedama metų spalva: ZALIA, JUODA, RAUDONA, GELTONA, BALTA.

kalendorius_data.txt	kalendorius_res.txt
1984	ZALIA

13. Didmeninis pirkimas. Žinoma, kad perkant daugiau prekių jų vienetas kainuoja pigiau. Pundelyje yra 12 porų kojinių, dėžėje – 12 pundelių. Pavyzdžiui, kojinių dėžė kainuoja 247 eurus, pundelis – 21 eurą, pora – 2 eurus. Įdomu tai, kad jei mums reikėtų 11 porų kojinių, tai geriau pirkti pundelį ir vienerias kojines kam nors atiduoti.

Pirkėjas nori įsigyti n porų kojinių. Parašykite programą, vadovaudamiesi kuria pigiausiai nupirktume kojines. Jei už tą pačią kainą galima nupirkti didesnę ir mažesnę kiekį kojinių, tai perkamas didesnis kiekis. Raskite perkamų dėžių, pundelių ir porų skaičių.

Pradinių duomenų faile **pirkimas_data.txt** įrašyta: kojinių porų skaičius n , vienos dėžės, vieno pundelio bei vienos poros kaina eurai.

Rezultatų faile **pirkimas_res.txt** turi būti įrašyta: dėžių skaičius, ryšulių skaičius, porų skaičius.

pirkimas_data.txt	pirkimas_res.txt
720 303 32 3	5 0 0

14. Atostogos. Osvaldas nori savaitę slidinėti viename iš trijų kurortų. Kurorte A slidinėjimo sezonas prasideda lapkričio, o baigiasi balandžio mėnesį, bet dėl lavinų pavojaus visą sausio mėnesį slidinėti negalima. Kurorte B slidinėti galima nuo gruodžio pradžios iki kovo pabaigos, tačiau vasario 1–15 dienomis čia vyksta varžybos. Kurorte C slidininkai laukiami nuo lapkričio pradžios iki gegužės pabaigos. Poilsio kaina kiekviename kurorte, įtraukus ir kelionės išlaidas, atitinkamai yra **k1**, **k2**, **k3** eurų (sveikasis skaičius). Žinodami atostogų pradžios datą (mėnesį m ir dieną d), nustatykite, ar Osvaldas galės atstogauti bent viename kurorte. Jeigu taip, tai kurį kurortą jam rinktis, kad išleistų mažiausiai pinigų?

Pradinių duomenų faile **atostogos_data.txt** įrašytos m , d , $k1$, $k2$, $k3$ reikšmės.

Rezultatų faile **atostogos_res.txt** turi būti išvestas kurorto pavadinimas (A, B, C) ir kelionės kaina, atskirti tarpais. Jei galimybės slidinėti nė viename kurorte nėra, rezultatų faile turi būti išvestas žodis NE.

atostogos_data.txt	atostogos_res.txt
2 5 500 520 499	C 499