

Informatikos valstybinio brandos egzamino antros dalies vertintojų mokymai, III dalis

Duomenų tyryba

Mokymų tikslas ir uždaviniai

Tobulinti informatikos mokytojų, kaip informatikos valstybinio brandos egzamino antros dalies vertintojų, kompetencijas

- Išanalizuoti VBE užduoties reikalavimus ir vertinimo kriterijus.
- Pateikti VBE vertintojams skirtingų pasiekimų lygių užduoties kandidatų atlikčių.
- Supažindinti vertintojus su elektroninėmis VBE vertinimo sistemomis.
- Sudaryti galimybes vertintojams praktiškai išbandyti vertinimui reikalingas aplinkas.

Informatikos valstybinio brandos egzamino antros dalies užduočių aprašas



Bendrai finansuoja
Europos Sąjunga

Informatikos mokymo(si) turinio ir pasiekimų sritys procentais informatikos VBE antros dalies užduotyje

Mokymo(si) turinio sritys	Pasiekimų sritys						Užduoties taškai procentais
	Skaitmeninio turinio kūrimas (A)	Algoritmai ir programavimas (B)	Duomenų tyryba ir informacija (C)	Technologinių problemų sprendimas (D)	Virtualioji komunikacija ir bendradarbiavimas (E)	Saugus elgesys (F)	
Skaitmeninio turinio kūrimo mokymo(si) turinys							8
Algoritmų ir programavimo mokymo(si) turinys							54
Duomenų tyrybos ir informacijos mokymo(si) turinys							30
Technologinių problemų sprendimo mokymo(si) turinys							-
Virtualiosios komunikacijos ir bendradarbiavimo mokymo(si) turinys							-
Saugaus elgesio mokymo(si) turinys							8
Iš viso taškų procentais	8	54	30	-	-	8	100

Bendra informacija apie informatikos egzamino antrą dalį

- 60 taškų.
- Trukmė: 180 minučių.
- Priemonės: užduoties sąsiuvinis, kompiuteris, praktinių užduočių pradiniai failai.
Reikalavimai kompiuteriui ir programinei įrangai nustatyti Informatikos valstybinio brandos egzamino antros dalies vykdymo instrukcijoje.
- Vertinimas: centralizuotas. Vertina vertintojai elektroninėje vertinimo sistemoje.

Informatikos egzamino antros dalies užduoties pobūdis

I dalis. Klausimai ir struktūriniai klausimai

- Pateikiama ir atliekama elektroninėje užduoties atlikimo (testavimo) sistemoje. Klausimo vertė taškais pateikiama prie kiekvieno klausimo.
- Sudaro 5-8 įvairaus tipo klausimai.
- 8-12 taškų.

II dalis. Duomenų tyrybos užduotis

- Atliekama kompiuteriu naudojant skaičiuoklę.
- Atlikti veiksmus su pateiktais duomenimis (skaičiuoti, rikiuoti, atrinkti, grupuoti, pavaizduoti, analizuoti ir pan.) ir pateikti išvadas.
- 16-20 taškų.

III dalis. Programavimo praktinė užduotis

- 1-2 programavimo užduotys, atliekamos kompiuteriu.
- *C++* arba *Python* programavimo kalba.
- 30-35 taškai.

Informatikos VBE
vertinimas.

Duomenų tyryba

INFORMATIKOS VBE UŽDUOTIS

Užduotis rengiama remiantis Programos *IV gimnazijos klasės mokymo(si) turiniu. Pagrindinio ugdymo* programos ir *III gimnazijos klasės* mokymo(si) turinys į užduotį įtraukiamas tik tiek, kiek būtina užduotims, parengtoms pagal IV gimnazijos klasės mokymo(si) turinį, atlikti.

31. Mokymo(si) turinys. IV gimnazijos klasė.

31.3. Duomenų tyrybos ir informacijos mokymo(si) turinys.

31.3.1. Informacijos (rezultatų) pateikimas. Mokomasi pateikti duomenis suvestinėmis lentelėmis, juos rikiuoti ir atrinkti pagal kelis kriterijus, grupuoti, skaičiuoti tarpines sumas, formuoti ataskaitas. Prisimenamos anksčiau išmoktos duomenų tyrybai naudotos funkcijos, diagramų braižymas (linijinės, stulpelinės, skritulinės, juostinės, histogramos, stačiakampės, sklaidos, kombinuotosios). Mokomasi apdoroti tekstinę informaciją naudojant darbui su tekstu skirtas funkcijas. Aptariami galimi sprendimai, formuluojamos išvados.

INFORMATIKOS BENDROJI PROGRAMA

Užduotis rengiama remiantis Programos *IV gimnazijos klasės mokymo(si) turiniu. Pagrindinio ugdymo* programos ir *III gimnazijos klasės* mokymo(si) turinys į užduotį įtraukiamas tik tiek, kiek būtina užduotims, parengtoms pagal IV gimnazijos klasės mokymo(si) turinį, atlikti.

30. Mokymo(si) turinys. III gimnazijos klasė.

30.3. Duomenų tyrybos ir informacijos mokymo(si) turinys.

.....

30.3.2. Duomenys apdorojami skaičiuokle naudojant kai kurias statistines, matematinės, logines, datos ir laiko, peržvalgų ir nuorodų funkcijas (pavyzdžiui, apvalinimo, apvalinimo iki mažesniojo, apvalinimo iki didesniojo, netuščių langelių skaičiavimo, sąlyginio sumavimo, sąlyginio langelių skaičiavimo, sąlyginio vidurkio, variantinės sąlygos, loginio IR, loginio ARBA, variantinės sąlygos sumavimo, variantinės sąlygos vidurkio variantinės sąlygos langelių skaičiavimo, variantinės sąlygos minimumo (maksimumo), datos (metų, mėnesio, dienos, šiandienos), vertikaliosios paieškos), sujungimo operatorių &.

Pasiruošimas vertinimui

- ❑ Vertinimo komisijos pirmininkas, pavaduotojas ir vyresnieji vertintojai atlieka kontrolinį žvalgomąjį vertinimą.
- ❑ Išskiriami vertinimo instrukcijoje neaprašyti, nenumatyti vertinimo niuansai, koreguojama vertinimo instrukcija, aptariamas išskirtinių atvejų vertinimas.
- ❑ Vyresnieji vertintojai pasiruošia vertinimo standartizavimui.
- ❑ Pirmojo susitikimo metu vyksta vertinimo standartizavimas, aptariami organizaciniai vertinimo proceso aspektai, komunikavimo organizavimas.
- ❑ Aptariamas darbas su elektronine vertinimo sistema.

INFORMATIKA

Valstybinio brandos egzamino antrosios dalies užduotis

Bandomasis egzaminas

2026 m. kovo 6 d.

Trukmė – 3 val. (180 min.)

II. DUOMENŲ TYRYBOS PRAKTINĖ UŽDUOTIS

Maksimalus vertinimas – 20 taškų

Visa užduotis atliekama skaičiuokle.

Faile *Pasaulio futbolo čempionatas 2022.xlsx* (*Pasaulio futbolo čempionatas 2022.ods*), kurį sudaro 9 darbo lakštai, pateikti 2022 m. pasaulio futbolo čempionato statistiniai duomenys paimti iš internetinės enciklopedijos „Vikipedija“ (https://lt.wikipedia.org/wiki/XXII_pasaulio_futbolo_%C4%8Dempionatas).

Darbo lakštuose *1 užduotis*, *2 užduotis*, *3 užduotis*, *4 užduotis*, *5 užduotis*, *6 užduotis*, *Atranka* ir *Rikiavimas* pateikti visų 2022 m. pasaulio futbolo čempionate įvykusių rungtynių tie patys duomenys: rungtynių data, rungtynėse dalyvavusios šalys, jų komandų įmuštų įvarčių skaičiai, miestai, kuriuose vyko rungtynės, ir jas stebėjusių žiūrovų skaičiai. Kiekviena komanda gali būti pateikta ir kaip *1 komanda*, ir kaip *2 komanda*.

Darbo lakšte *Diagrama* pateiktos čempionato rungtynių datos, dienos varžybų skaičius bei vidutinis žiūrovų skaičius.

Kiekviename darbo lakšte yra pateikiama ir užduotis.

Naudodamiesi pateiktais duomenimis ir laikydamiesi nurodymų, atlikite užduotis. Darbo lakštuose *1 užduotis*, *2 užduotis*, *3 užduotis*, *4 užduotis*, *5 užduotis* ir *6 užduotis* užduotims atlikti naudokite **tinkamas formules ir funkcijas**. Šiuose lakštuose galima atlikti ir tarpinius skaičiavimus.

	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	1 komanda	2 komanda	1 komanda įmušė	2 komanda įmušė	Miestas	Žiūrovų skaičius								
2	Kataras	Ekvadoras	0	2	Chauras	67372								
3	Senegalas	Nyderlandai	0	2	Doha	41721								
4	Kataras	Senegalas	1	3	Doha	41797								
5	Nyderlandai	Ekvadoras	1	1	Rajanas	44833								
6	Ekvadoras	Senegalas	1	2	Rajanas	44569								
7	Nyderlandai	Kataras	2	0	Chauras	66784								
8	Anglija	Iranas	6	2	Rajanas	45334								
9	JAV	Velsas	1	1	Rajanas	43418								
10	Velsas	Iranas	0	2	Rajanas	40875								
11	Anglija	JAV	0	0	Chauras	68463								
12	Velsas	Anglija	0	3	Rajanas	44297								
13	Iranas	JAV	0	1	Doha	42127								
14	Argentina	Saudo Arabija	1	2	Lusailas	88012								
15	Meksika	Lenkija	0	0	Doha	39369								
16	Lenkija	Saudo Arabija	2	0	Rajanas	44259								
17	Argentina	Meksika	2	0	Rajanas	88966								
18	Lenkija	Argentina	0	2	Doha	44089								
19	Saudo Arabija	Meksika	1	2	Lusailas	84985								
20	Danija	Tunisas	0	0	Rajanas	42925								
21	Prancūzija	Australija	4	4	Vakra	40875								
22	Tunisas	Australija	0	1	Vakra	41823								
23	Prancūzija	Danija	2	1	Doha	42860								
24	Australija	Danija	1	0	Vakra	41232								
25	Tunisas	Prancūzija	1	0	Rajanas	43627								

1. Darbo lakšte 1 užduotis langelyje P3 pateikite formulę, kuri apskaičiuotų, **kiek dienų** truko pasaulio futbolo čempionatas (dienų skaičius skaičiuojamas nuo pirmų iki paskutinių čempionato rungtynių **imtinai**).

(1 taškas)

1.1. Darbo lakšte 1 užduotis pateikta teisinga formulė, pavyzdžiui,
=DAYS(A65;A2)+1

1

Formulė	Taškai	Komentaras	Rezultatas
=MAX(A2:A65)-MIN(A2:A65)+1	1	Teisinga formulė	29
=DAYS(A65;A2)+1	1	Nepridėta diena	29
=A65-A2+1	1	Nepridėta diena	29
=DAYS(MAX(A2:A65);MIN(A2:A65))	0	Nepridėta diena	28
=Diagrama!A24-Diagrama!A2+1	1	Galimas toks duomenų paėmimas	29

29

	A	B	C	D	E	F	G
1	Rungtynių data	1 komanda	2 komanda	1 komanda įmušė	2 komanda įmušė	Miestas	Žiūrovų skaičius
2	2022-11-20	Kataras	Ekvadoras	0	2	Chauras	67372
3	2022-11-21	Senegalas	Nyderlandai	0	2	Doha	41721
4	2022-11-25	Kataras	Senegalas	1	3	Doha	41797
5	2022-11-25	Nyderlandai	Ekvadoras	1	1	Rajanas	44833
6	2022-11-29	Ekvadoras	Senegalas	1	2	Rajanas	44569
7	2022-11-29	Nyderlandai	Kataras	2	0	Chauras	66784
8	2022-11-21	Anglija	Iranas	6	2	Rajanas	45334
9	2022-11-21	JAV	Velsas	1	1	Rajanas	43418
10	2022-11-25	Velsas	Iranas	0	2	Rajanas	40875
11	2022-11-25	Anglija	JAV	0	0	Chauras	68463
12	2022-11-29	Velsas	Anglija	0	3	Rajanas	44297
13	2022-11-29	Iranas	JAV	0	1	Doha	42127
14	2022-11-22	Argentina	Saudo Arabija	1	2	Lusailas	88012
15	2022-11-22	Meksika	Lenkija	0	0	Doha	39369
16	2022-11-26	Lenkija	Saudo Arabija	2	0	Rajanas	44259
17	2022-11-26	Argentina	Meksika	2	0	Rajanas	88966
18	2022-11-30	Lenkija	Argentina	0	2	Doha	44089
19	2022-11-30	Saudo Arabija	Meksika	1	2	Lusailas	84985
20	2022-11-22	Danija	Tunisas	0	0	Rajanas	42925
21	2022-11-22	Prancūzija	Australija	4	4	Vakra	40875
22	2022-11-26	Tunisas	Australija	0	1	Vakra	41823
23	2022-11-26	Prancūzija	Danija	2	1	Doha	42860
24	2022-11-30	Australija	Danija	1	0	Vakra	41232
25	2022-11-30	Tunisas	Prancūzija	1	0	Rajanas	43627
26	2022-11-23	Vokietija	Japonija	1	2	Rajanas	42608
27	2022-11-23	Ispanija	Kosta Rika	7	0	Doha	40013
28	2022-11-27	Japonija	Kosta Rika	0	1	Rajanas	41479
29	2022-11-27	Ispanija	Vokietija	1	1	Chauras	68895
30	2022-12-01	Japonija	Ispanija	2	1	Rajanas	44851

2. Darbo lakšte 2 užduotis langelyje N2 pateikite formulę, kuri apskaičiuotų, kiek **mažiausiai** žiūrovų apsilankė vienoje rungtynėse, o langelyje N4 – **miestą**, kuriame vyko šios rungtynės.

(2 taškai)

2.1. Darbo lakšte 2 užduotis pateikta teisinga formulė, pavyzdžiui,
=MIN(G2:G65)

(2)
1

39089

Formulė	Taškai	Komentaras	Rezultatas
=MIN(G2:G65)	1	Pavyzdinė teisinga formulė	39089
Surikiuota ir =G2	0	Turi tikti įvairiems pradiniais duomenims	39089
=G38	0	Turi tikti įvairiems pradiniais duomenims	39089

2.2. Darbo lakšte 2 užduotis pateikta teisinga formulė, pavyzdžiui,
=INDEX(F2:F65;MATCH(N2;G2:G65;0))

1

Vakra

Formulė	Taškai	Komentaras	Rezultatas
=INDEX(F2:F65;MATCH(N2;G2:G65;0))	1	Pavyzdinė teisinga formulė	Vakra
=XLOOKUP(N2;G2:G65;F2:F65)	1	Teisinga formulė, veikianti su vėlesniais MS Office	Vakra
Surikiuota ir =F2	0	Turi tikti įvairiems pradiniais duomenims	Vakra
=F38	0	Turi tikti įvairiems pradiniais duomenims	Vakra
=LOOKUP(N2;G2:G65;F2:F65)	0	Netinkama funkcija	#N/A

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Rungtynių data	1 komanda	2 komanda	1 komanda įmušė	2 komanda įmušė	Miestas	Žiūrovų skaičius								
2	2022-11-20	Kataras	Ekvadoras	0	2	Chauras	67372								
3	2022-11-21	Senegalas	Nyderlandai	0	2	Doha	41721								
4	2022-11-25	Kataras	Senegalas	1	3	Doha	41797								
5	2022-11-25	Nyderlandai	Ekvadoras	1	1	Rajanas	44833								
6	2022-11-29	Ekvadoras	Senegalas	1	2	Rajanas	44569								
7	2022-11-29	Nyderlandai	Kataras	2	0	Chauras	66784								
8	2022-11-21	Anglija	Iranas	6	2	Rajanas	45334								
9	2022-11-21	JAV	Velsas	1	1	Rajanas	43418								
10	2022-11-25	Velsas	Iranas	0	2	Rajanas	40875								
11	2022-11-25	Anglija	JAV	0	0	Chauras	68463								
12	2022-11-29	Velsas	Anglija	0	3	Rajanas	44297								
13	2022-11-29	Iranas	JAV	0	1	Doha	42127								
14	2022-11-22	Argentina	Saudo Arabija	1	2	Lusailas	88012								
15	2022-11-22	Meksika	Lenkija	0	0	Doha	39369								
16	2022-11-26	Lenkija	Saudo Arabija	2	0	Rajanas	44259								
17	2022-11-26	Argentina	Meksika	2	0	Rajanas	88966								
18	2022-11-30	Lenkija	Argentina	0	2	Doha	44089								
19	2022-11-30	Saudo Arabija	Meksika	1	2	Lusailas	84985								
20	2022-11-22	Danija	Tunisas	0	0	Rajanas	42925								
21	2022-11-22	Prancūzija	Australija	4	4	Vakra	40875								
22	2022-11-26	Tunisas	Australija	0	1	Vakra	41823								
23	2022-11-26	Prancūzija	Danija	2	1	Doha	42860								

3. Darbo lakšte 3 užduotis langelyje O2 pateikite formulę, kuri apskaičiuotų, koks buvo **dažniausiai pasikartojantis** žiūrovų skaičius rungtynėse.

(1 taškas)

3.

3.1. Darbo lakšte 3 užduotis pateikta teisinga formulė, pavyzdžiui,
=MODE(G2:G65)

1

88966

Formulė	Taškai	Komentaras	Rezultatas
=MODE(G2:G65)	1	Pavyzdinė teisinga formulė	88966
=MODE.SNGL(G2:G65)	1	Teisinga pažangesnė funkcija	88966
=MODE.MULT(G2:G65)	1	Teisinga pažangesnė funkcija	88966
=INDEX(G2:G65;MATCH(MAX(COUNTIF(G2:G65;G2:G65));COUNTIF(G2:G65;G2:G65);0))	1		88966
=COUNTIF(\$G\$2:\$G\$65;LOOKUP(G2;\$G\$2:\$G\$65))		L2 ir kopijuota	
=INDEX(G2:G65;MATCH(MAX(L2:L65);L2:L65;0))	0		88966
=COUNTIF(\$G\$2:\$G\$65;G2)		O3 ir kopijuota	1
=INDEX(G2:G65;MATCH(MAX(O3:O65);O3:O65;0))	1		88966

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Rungtynių data	1 komanda	2 komanda	1 komanda įmušė	2 komanda įmušė	Miestas	Žiūrovų skaičius								
2	2022-11-20	Kataras	Ekvadoras	0	2	Chauras	67372								
3	2022-11-21	Senegalas	Nyderlandai	0	2	Doha	41721								
4	2022-11-25	Kataras	Senegalas	1	3	Doha	41797								
5	2022-11-25	Nyderlandai	Ekvadoras	1	1	Rajanas	44833								
6	2022-11-29	Ekvadoras	Senegalas	1	2	Rajanas	44569								
7	2022-11-29	Nyderlandai	Kataras	2	0	Chauras	66784								
8	2022-11-21	Anglija	Iranas	6	2	Rajanas	45334								
9	2022-11-21	JAV	Velsas	1	1	Rajanas	43418								
10	2022-11-25	Velsas	Iranas	0	2	Rajanas	40875								
11	2022-11-25	Anglija	JAV	0	0	Chauras	68463								
12	2022-11-29	Velsas	Anglija	0	3	Rajanas	44297								
13	2022-11-29	Iranas	JAV	0	1	Doha	42127								
14	2022-11-22	Argentina	Saudo Arabija	1	2	Lusailas	88012								
15	2022-11-26	Meksika	Lenkija	0	0	Doha	39369								
16	2022-11-26	Lenkija	Saudo Arabija	2	0	Rajanas	44259								
17	2022-11-26	Argentina	Meksika	2	0	Rajanas	88966								
18	2022-11-30	Lenkija	Argentina	0	2	Doha	44089								
19	2022-11-30	Saudo Arabija	Meksika	1	2	Lusailas	84985								
20	2022-11-22	Danija	Tunisas	0	0	Rajanas	42925								
21	2022-11-22	Prancūzija	Australija	4	4	Vakra	40875								
22	2022-11-26	Tunisas	Australija	0	1	Vakra	41823								
23	2022-11-26	Prancūzija	Danija	2	1	Doha	42860								
24	2022-11-30	Australija	Danija	1	0	Vakra	41232								

4. Darbo lakšte 4 užduotis langelyje O2 pateikite formulę, kuri apskaičiuotų, kiek rungtynių laimėjo čempione tapusi Argentinos komanda.

(2 taškai)

4. 4.1. Darbo lakšte 4 užduotis pateiktos teisingos formulės (-ė).
4.1.1. Teisingai nustatoma laimėjo 1 ar 2 komanda, pavyzdžiui,
=IFS(D2>E2;"+";D2<E2;"-";D2=E2;"=")

Formulė	Taškai	Komentaras	Rezultatas
=IFS(D2>E2;"+";D2<E2;"-";D2=E2;"=")	1	Pavyzdinė teisinga formulė	-
=IF(D2>E2;"+";IF(E2>D2;"-";"="))	1	Teisinga formulė	-
=IF(B2="Argentina";IF(D2>E2;COUNT(D2);0);0)	0		0
=SUMPRODUCT((B2:B65="Argentina")*(D2:D65>E2:E65))+SUMPRODUCT((C2:C65="Argentina")*(E2:E65>D2:D65))	1	Viena formulė	4
COUNTIFS(B1:B65;"="&B14;D1:D65;">"&E1:E65)+COUNTIFS(C1:C65;"="&B14;D1:D65;"<"&E1:E65)	0	Viena formulė	3
=IF(D2>E2;B2;C2)	0		Ekvadoras
=SUMIF(B2:B65;"Argentina";D2:D65)+SUMIF(C2:C65;"Argentina";E2:E65)	0	Viena formulė	15
=COUNTIFS(B2:B65;"Argentina";D2:D65;">E2:E65")+COUNTIFS(C2:C65;"Argentina";D2:D65;"<E2:E65")	0	Viena formulė	0
=SUM(COUNTIFS(B2:B65;"Argentina";D2:E65;D2:D65>E2:E65);COUNTIFS(C2:C65;"Argentina";D2:E65;E2:E65>D2:D65))	0	Viena formulė	#VALUE!
=COUNTIFS(B2:B65;"Argentina"&D2:D65<E2:E65;C2:C65;"Argentina"	0	Viena formulė	0
=COUNTIFS(B2:C65;B65;D2:E65;">1")	0	Viena formulė	6

29	2022-11-27	Ispanija	Vokietija	1	1	Chauras	68895
	2022-12-01	Japonija	Ispanija	2	1	Rajanas	44851
30							
31	2022-12-01	Kosta Rika	Vokietija	2	4	Chauras	67054
32	2022-11-23	Marokas	Kroatija	0	0	Chauras	59407
33	2022-11-23	Belgija	Kanada	1	0	Rajanas	40432
34	2022-11-27	Belgija	Marokas	0	2	Doha	43738
	2022-11-27	Kroatija	Kanada	4	1	Rajanas	44374
35							
36	2022-12-01	Kroatija	Belgija	0	0	Rajanas	43984
	2022-12-01	Kanada	Marokas	1	2	Doha	43102
37							
38	2022-11-24	Šveicarija	Kamerūnas	1	0	Vakra	39089
39	2022-11-24	Brazilija	Serbija	2	0	Lusailas	88103
	2022-11-28	Kamerūnas	Serbija	3	3	Vakra	39789
40							
41	2022-11-28	Brazilija	Šveicarija	1	0	Doha	43649
	2022-12-02	Serbija	Šveicarija	2	3	Doha	41378
42							
43	2022-12-02	Kamerūnas	Brazilija	1	0	Lusailas	85986
44	2022-11-24	Urugvajus	Pietų Korėja	0	0	Rajanas	41663
45	2022-11-24	Portugalija	Gana	3	2	Doha	42662
	2022-11-28						

4.1.2. Teisingai suskaičiuojama kiek pergalių iškovojo Argentina, pavyzdžiui,
=COUNTIFS(B2:B65;"Argentina";H2:H65;"+") + COUNTIFS(C2:C65;"Argentina";H2:H65;"-")

1

Formulė	Taškai	Komentaras	Rezultatas
=COUNTIFS(B2:B65;"Argentina";H2:H65;"+") + COUNTIFS(C2:C65;"Argentina";H2:H65;"-")	1	Pavyzdinė formulė	4
=COUNTIF(M2:M65;">0")	1		3
=SUMPRODUCT((B2:B65="Argentina")*(D2:D65>E2:E65))+SUMPRODUCT((C2:C65="Argentina")*(E2:E65>D2:D65))	1	Viena formulė	4
COUNTIFS(B1:B65;"="&B14;D1:D65;">"&E1:E65)+COUNTIFS(C1:C65;"="&B14;D1:D65;"<"&E1:E65)	0	Viena formulė	3
=COUNTIF(H2:H65;"argentina")	1		5
=SUMIF(B2:B65;"Argentina";D2:D65) + SUMIF(C2:C65;"Argentina";E2:E65)	0	Viena formulė	15
=COUNTIFS(B2:B65;"Argentina";D2:D65;">E2:E65")+COUNTIFS(C2:C65;"Argentina";D2:D65;"<E2:E65")	1	Viena formulė	0
=SUM(COUNTIFS(B2:B65;"Argentina";D2:E65;D2:D65>E2:E65);COUNTIFS(C2:C65;"Argentina";D2:E65;E2:E65>D2:D65))	1	Viena formulė	#VALUE!
=COUNTIFS(B2:B65;"Argentina"&D2:D65<E2:E65;C2:C65;"Argentina")	0	Viena formulė	0
=COUNTIFS(B2:C65;B65;D2:E65;">1")	0	Viena formulė	6

4

	B	C	D	E	F	G	H
1	1 komanda	2 komanda	1 komanda įmušė	2 komanda įmušė	Miestas	Žiūrovų skaičius	
2	Kataras	Ekvadoras	0	2	Chauras	67372	
3	Senegalas	Nyderlandai	0	2	Doha	41721	
4	Kataras	Senegalas	1	3	Doha	41797	
5	Nyderlandai	Ekvadoras	1	1	Rajanas	44833	
6	Ekvadoras	Senegalas	1	2	Rajanas	44569	
7	Nyderlandai	Kataras	2	0	Chauras	66784	
8	Anglija	Iranas	6	2	Rajanas	45334	
9	JAV	Velsas	1	1	Rajanas	43418	
10	Velsas	Iranas	0	2	Rajanas	40875	
11	Anglija	JAV	0	0	Chauras	68463	
12	Velsas	Anglija	0	3	Rajanas	44297	
13	Iranas	JAV	0	1	Doha	42127	
14	Argentina	Saudo Arabija	1	2	Lusailas	88012	
15	Meksika	Lenkija	0	0	Doha	39369	
16	Lenkija	Saudo Arabija	2	0	Rajanas	44259	
17	Argentina	Meksika	2	0	Rajanas	88966	
18	Lenkija	Argentina	0	2	Doha	44089	
19	Saudo Arabija	Meksika	1	2	Lusailas	84985	
20	Danija	Tunisas	0	0	Rajanas	42925	
21	Prancūzija	Australija	4	4	Vakra	40875	
22	Tunisas	Australija	0	1	Vakra	41823	

5. Darbo laiške 5 užduotis langelyje O5 pateikite formulę, kuri apskaičiuotų **didžiausiu skirtumu** pasibaigusias rungtynes. Pateikite šių rungtynių rezultatą, užrašydami komandų pavadinimus trumpinant pirmosiomis 3 didžiosiomis raidėmis atskiriant brūkšneliu, o po tarpo abiejų komandų įmuštų įvarčių skaičius atskiriant dvitaškiu: pvz., **ABC-KLM 0:0**.

(3 taškai)

5. 5.1. Darbo laiške **5 užduotis** pateiktos teisingos formulės (-ė).
 5.1.1. Teisingai randamas kiekvienų rungtynių įvarčių skirtumas, pavyzdžiui,
=ABS(D2-E2) (3)
 1

Formulė	Taškai	Komentaras	Rezultatas	H
=ABS(D2-E2)	1	Pavyzdinė formulė	2	
=D2+E2	0		2	2
=IF(D2-E2>0;D2-E2;-(D2-E2))	1		2	2
=MAX(D2-E2;E2-D2)	1		2	0
=MAX(ABS(D2-E2))	1		2	1
				2
				4
				0

5.1.2. Teisingai nustatomi didžiausių skirtumu pasibaigusių rungtynių komandų pavadinimai ir abiejų komandų įmuštų įvarčių skaičiai, pavyzdžiui,
=INDEX(B2:B65;MATCH(MAX(\$H2:\$H65);\$H2:\$H65;0)) – randama pirma komanda ir formulė nukopijuota į tris dešinėje esančiu langeliu.
 (pvz., stulpelyje H įvarčių skirtumas)

1

Formulė	Taškai	Komentaras	Rezultatas
=INDEX(B2:B65;MATCH(MAX(\$H2:\$H65);\$H2:\$H65;0))	1	Pavyzdinė formulė	Ispanija
=XLOOKUP(M5;G2:G65;B2:B65)	1	M5-Max sumos	Anglija
=UPPER(LEFT(B2;3))&"-"&UPPER(LEFT(C2;3))&" "&D2&" ":"&E2	1		KAT-EKV 0:2

Ispanija	Kosta Rika	7	0
----------	------------	---	---

5.1.3. Naudojama tinkama formulė rungtynių rezultatui pateikti, pavyzdžiui,
=UPPER(LEFT(K10;3))&"-"&UPPER(LEFT(L10;3))&" "&M10&" ":"&N10)

1

Formulė	Taškai	Komentaras	Rezultatas
=UPPER(LEFT(K10;3))&"-"&UPPER(LEFT(L10;3))&" "&M10&" ":"&N10	1	Pavyzdinė formulė	ISP-KOS 7:0
=UPPER(CONCAT(LEFT(M6;3);"-" ;LEFT(M7;3);" ";N6;" ":" ;N7))	1		ANG-IRA 6:2
=XLOOKUP(MAX(H2:H65);H2:H65;I2:I65)	1		ISP-KOS 7:0

ISP-KOS 7:0

6.1.2. Teisingai suskaičiuojama kiek įvarčių įmušė kiekviena komanda, pavyzdžiui,
=SUMIF(B\$2:B\$65;P4;D\$2:D\$65)+SUMIF(C\$2:C\$65;P4;E\$2:E\$65)
 (pavyzdžiui, stulpelyje P yra unikalus komandų sąrašas)

1

Formulė	Taškai	Komentaras
=SUMIF(\$B\$2:\$B\$65;P3;\$D\$2:\$D\$65)+SUMIF(\$C\$2:\$C\$65;P3;\$E\$2:\$E\$65)	1	Vertinimo instrukcija

Kataras	1
Senegalas	5
Nyderlandai	10
Ekvadoras	4
Anglija	13
JAV	3
Velsas	1
Iranas	4
Argentina	15

6.1.3. Teisingai suskaičiuojamas ir suapvalinamas iki sveikojo skaičiaus įmuštų įvarčių vidurkis, pavyzdžiui,
=ROUND(AVERAGE(R4:R35);0)
 (pavyzdžiui, stulpelyje R yra kiekvienos komandos pelnytų įvarčių skaičius)

1

Formulė	Taškai	Komentaras
=ROUND(AVERAGE(R3:R34);0)	1	Vertinimo instrukcija
=ROUND(AVERAGE(D2:E65);0)	1	1 formulė
=TRUNC(AVERAGE(D2:E65);0)	0	
=AVERAGE(D2:E65)	0	Suapvalinta formatavimu

5

	A	B	C	D	E	F	G
1	Rungtynių data ir laikas	1 komanda	2 komanda	1 komanda jmu	2 komanda jmu	Miestas	Žiūrovų skaičius
4	2022-11-25 16:00	Kataras	Senegalas	1	3	Doha	41797
5	2022-11-25 19:00	Nyderlandai	Ekvadoras	1	1	Rajanas	44833
6	2022-11-29 18:00	Ekvadoras	Senegalas	1	2	Rajanas	44569
7	2022-11-29 18:00						

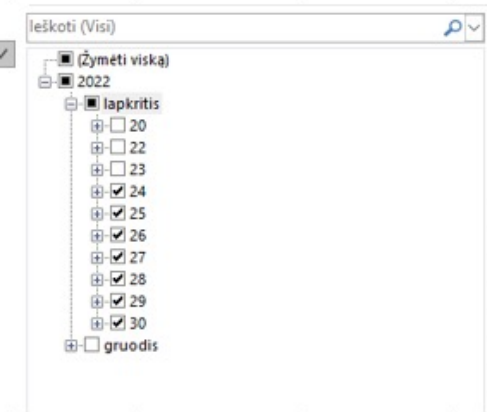
7. Darbo laške *Atranka* atrinkite paskutinės lapkričio savaitės (nuo lapkričio 24 d. iki lapkričio 30 d. imtinai) rungtynes, kuriose apsilankė daugiau kaip 50 000 žiūrovų. (2 taškai)

7.1. Atrinkami paskutinės lapkričio savaitės (nuo lapkričio 24 d. iki lapkričio 30 d. imtinai) rungtynių duomenys.

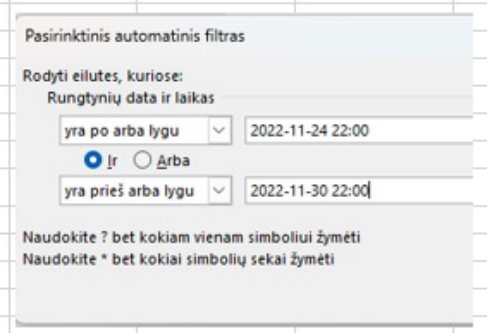
Taškai

7.2. Atrinkami rungtynių, kuriose apsilankė virš 50000 žiūrovų, duomenys.

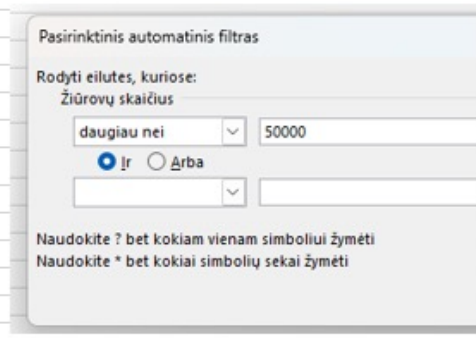
Taškai



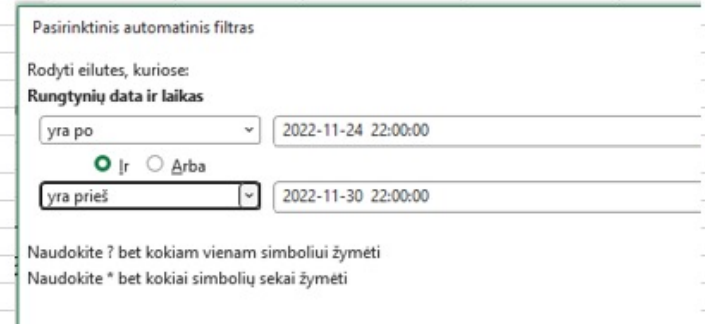
1



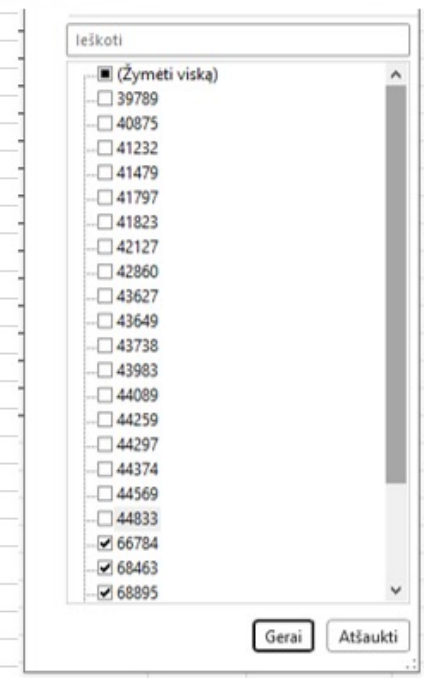
0



1



0



0

Pasirinktinis automatinis filtras

Rodyti eilutes, kuriose:

Rungtynių data ir laikas

yra po

2022-11-23

Ir Arba

yra prieš

2022-12-01

Naudokite ? bet kokiam vienam simboliui žymėti

Naudokite * bet kokiai simbolių sekai žymėti

0

Pasirinktinis automatinis filtras

Rodyti eilutes, kuriose:

Rungtynių data ir laikas

yra po arba lygu

2022-11-24

Ir Arba

yra prieš

2022-12-01

1

Pasirinktinis automatinis filtras

Rodyti eilutes, kuriose:

Rungtynių data ir laikas

yra po

2022-11-24 13:00:00

Ir Arba

yra prieš

2022-11-30 22:00:00

1

	A	B	C	D	E	F	G
1	Rungtynių data	1 komanda	2 komanda	1 komanda įmušė	2 komanda įmušė	Miestas	Žiūrovų skaičius
2	2022-11-26	Argentina	Meksika	2	0	Rajanas	88966
3	2022-11-28	Portugalija	Urugvajus	2	0	Lusailas	88668
4	2022-11-24	Brazilija	Serbija	2	0	Lusailas	88103
5	2022-11-22	Argentina	Saudo Arabija	1	2	Lusailas	88012
6	2022-11-30	Saudo Arabija	Meksika	1	2	Lusailas	84985
7	2022-11-27	Ispanija	Vokietija	1	1	Chauras	68895
8	2022-11-25	Anglija	JAV	0	0	Chauras	68463
9	2022-11-20	Kataras	Ekvadoras	0	2	Chauras	67372
10	2022-11-29	Nyderlandai	Kataras	2	0	Chauras	66784
11	2022-11-23	Marokas	Kroatija	0	0	Chauras	59407
12	2022-11-21	Anglija	Iranas	6	2	Rajanas	45334
13	2022-11-25	Nyderlandai	Ekvadoras	1	1	Rajanas	44833
14	2022-11-29	Ekvadoras	Senegalas	1	2	Rajanas	44569
15	2022-11-27	Kroatija	Kanada	4	1	Rajanas	44374
16	2022-11-29	Velsas	Anglija	0	3	Rajanas	44297
17	2022-11-26	Lenkija	Saudo Arabija	2	0	Rajanas	44259
18	2022-11-30	Lenkija	Argentina	0	2	Doha	44089
19	2022-11-28	Pietų Korėja	Gana	2	3	Rajanas	43983
20	2022-11-27	Belgija	Marokas	0	2	Doha	43738
21	2022-11-28	Brazilija	Šveicarija	1	0	Doha	43649
22	2022-11-30	Tunisas	Prancūzija	1	0	Rajanas	43627
23	2022-11-21	JAV	Velsas	1	1	Rajanas	43418
24	2022-11-22	Danija	Tunisas	0	0	Rajanas	42925
25	2022-11-26	Prancūzija	Danija	2	1	Doha	42860
26	2022-11-24	Portugalija	Gana	3	2	Doha	42662
27	2022-11-23	Vokietija	Japonija	1	2	Rajanas	42608
28	2022-11-29	Iranas	JAV	0	1	Doha	42127
29	2022-11-26	Tunisas	Australija	0	1	Vakra	41823
30	2022-11-25	Kataras	Senegalas	1	3	Doha	41797
31	2022-11-21	Senegalas	Nyderlandai	0	2	Doha	41721
32	2022-11-24	Urugvajus	Pietų Korėja	0	0	Rajanas	41663
33	2022-11-27	Japonija	Kosta Rika	0	1	Rajanas	41479
34	2022-11-30	Australija	Danija	1	0	Vakra	41232
35	2022-11-22	Prancūzija	Australija	4	4	Vakra	40875
36	2022-11-25	Velsas	Iranas	0	2	Rajanas	40875

8. Darbo lakšte *Rikiavimas* surikiuokite įrašus taip, kad sąrašo viršuje būtų pateikiamos lapkričio mėnesio rungtynės, išdėstytos pagal žiūrovų skaičių mažėjimo tvarka, o sąrašas užbaigiamas gruodžio mėnesio rungtynėmis – žiūrovų skaičių didėjimo tvarka.

(2 taškai)

8.	8.1. Sąrašo viršuje pateiktos lapkričio mėnesio rungtynės išdėstytos pagal žiūrovų skaičių mažėjimo tvarka. Pastaba <i>Taškas skiriamas, jei surikiuota tik duomenų lentelė, be antraščių.</i>	1	2
	8.2. Sąrašas užbaigiamas gruodžio mėnesio rungtynėmis, kurio pateikiamos surikiuotos žiūrovų skaičių didėjimo tvarka. Pastaba <i>Taškas skiriamas, jei surikiuota tik duomenų lentelė, be antraščių.</i>	1	

Rūšiavimas ? X

Mano duomenyse yra antraščiai

Stulpelis	Rūšiuoti	Tvarka
Rūšiuoti pagal	Žiūrovų skaičius	Langelio reikšmės
		Nuo mažiausios iki didžiausios

Taškai

1 1

41	2022-11-24	Šveicarija	Kamerūnas	1	0	Vakra	39089
42	2022-12-04	Prancūzija	Lenkija	3	1	Doha	40989

	A	B	C	D	E	F	G
1	Rungtynių data	1 komanda	2 komanda	1 komanda jmu	2 komanda jmu	Miestas	Žiūrovų skaiči
2	2022-11-26	Argentina	Meksika	2	0	Rajanas	88966
3	2022-11-28	Portugalija	Urugvajus	2	0	Lusailas	88668
4	2022-11-24	Brazilija	Serbija	2	0	Lusailas	88103
5	2022-11-22	Argentina	Saudo Arabija	1	2	Lusailas	88012
6	2022-11-30	Saudo Arabija	Meksika	1	2	Lusailas	84985
7	2022-11-27	Ispanija	Vokietija	1	1	Chauras	68895
8	2022-11-25	Anglija	JAV	0	0	Chauras	68463
9	2022-11-20	Kataras	Ekvadoras	0	2	Chauras	67372
10	2022-11-29	Nyderlandai	Kataras	2	0	Chauras	66784
11	2022-11-23	Marokas	Kroatija	0	0	Chauras	59407
12	2022-11-21	Anglija	Iranas	6	2	Rajanas	45334
13	2022-11-25	Nyderlandai	Ekvadoras	1	1	Rajanas	44833
14	2022-11-29	Ekvadoras	Senegalas	1	2	Rajanas	44569
15	2022-11-27	Kroatija	Kanada	4	1	Rajanas	44374
16	2022-11-29	Velsas	Anglija	0	3	Rajanas	44297
17	2022-11-26	Lenkija	Saudo Arabija	2	0	Rajanas	44259
18	2022-11-30	Lenkija	Argentina	0	2	Doha	44089
19	2022-11-28	Pietų Korėja	Gana	2	3	Rajanas	43983
20	2022-11-27	Belgija	Marokas	0	2	Doha	43738
21	2022-11-28	Brazilija	Šveicarija	1	0	Doha	43649
22	2022-11-30	Tunisas	Prancūzija	1	0	Rajanas	43627
23	2022-11-21	JAV	Velsas	1	1	Rajanas	43418
24	2022-11-22	Danija	Tunisas	0	0	Rajanas	42925
25	2022-11-26	Prancūzija	Danija	2	1	Doha	42860
26	2022-11-24	Portugalija	Gana	3	2	Doha	42662
27	2022-11-23	Vokietija	Japonija	1	2	Rajanas	42608
28	2022-11-29	Iranas	JAV	0	1	Doha	42127
29	2022-11-26	Tunisas	Australija	0	1	Vakra	41823
32	2022-11-25	Kataras	Senegalas	1	3	Doha	41797
33	2022-11-21	Senegalas	Nyderlandai	0	2	Doha	41721
34	2022-11-24	Urugvajus	Pietų Korėja	0	0	Rajanas	41663
35	2022-11-27	Japonija	Kosta Rika	0	1	Rajanas	41479
38	2022-11-30	Australija	Danija	1	0	Vakra	41232
39	2022-11-25	Velsas	Iranas	0	2	Rajanas	40875
40	2022-11-22	Prancūzija	Australija	4	4	Vakra	40875
41	2022-11-23	Belgija	Kanada	1	0	Rajanas	40432
44	2022-11-23	Ispanija	Kosta Rika	7	0	Doha	40013
45	2022-11-28	Kamerūnas	Serbija	3	3	Vakra	39789
46	2022-11-22	Meksika	Lenkija	0	0	Doha	39369
47	2022-11-24	Šveicarija	Kamerūnas	1	0	Vakra	39089

10

Rūšiavimas

Mano duomenyse yra antraščių

Stulpelis: Rūšiuoti pagal **Rungtynių data** | Rūšiuoti: **Langelio reikšmės** | Tvarka: **Nuo seniausio iki naujausio**

Po to pagal: **Žiūrovų skaičius** | **Langelio reikšmės** | **Nuo didžiausio iki mažiausio**

00

Rūšiavimas

Mano duomenyse yra antraščių

Stulpelis: Rūšiuoti pagal **Rungtynių data** | Rūšiuoti: **Langelio reikšmės** | Tvarka: **Nuo seniausio iki naujausio**

Po to pagal: **Žiūrovų skaičius** | **Langelio reikšmės** | **Nuo mažiausio iki didžiausio**

00

37	2022-11-29	Nyderlandai	Kataras	2	0	Chauras	66784
38	2022-11-30	Australija	Danija	1	0	Vakra	41232
39	2022-11-30	Tunisas	Prancūzija	1	0	Rajanas	43627
40	2022-11-30	Lenkija	Argentina	0	2	Doha	44089
41	2022-11-30	Saudo Arabija	Meksika	1	2	Lusailas	84985
42	2022-12-13	Argentina	Kroatija	3	0	Lusailas	88966
43	2022-12-18	Argentina	Prancūzija	3	3	Lusailas	88966
44	2022-12-09	Nyderlandai	Argentina	2	2	Lusailas	88235
45	2022-12-02	Kamerūnas	Brazilija	1	0	Lusailas	85986
46	2022-12-06	Portugalija	Šveicarija	6	1	Lusailas	83720
							68895
							68294
							67054
							65985
							45032
							44851
							44846
							44667
55	2022-12-10	Marokas	Portugalija	1	0	Doha	44198
56	2022-12-17	Kroatija	Marokas	2	1	Rajanas	44137
57	2022-12-02	Prancūzija	Prancūzija	2	1	Rajanas	44007

00

Rūšiavimas

Mano duomenyse yra antraščių

Stulpelis: Rūšiuoti pagal **Žiūrovų skaičius** | Rūšiuoti: **Langelio reikšmės** | Tvarka: **Nuo didžiausio iki mažiausio**

38	2022-11-30	Saudo Arabija	Meksika	1	2	Lusailas	84985
39	2022-11-30	Lenkija	Argentina	0	2	Doha	44089
40	2022-11-30	Tunisas	Prancūzija	1	0	Rajanas	43627
41	2022-11-30	Australija	Danija	1	0	Vakra	41232
42	2022-12-01	Japonija	Ispanija	2	1	Rajanas	44851
43	2022-12-04	Prancūzija	Lenkija	3	1	Doha	40989
44	2022-12-02	Serbija	Šveicarija	2	3	Doha	41378
45	2022-12-05	Japonija	Kroatija	1	1	Vakra	42523
46	2022-12-01	Kanada	Marokas	1	2	Doha	43102
47							43443
48							43847
49							43893
50							43984
51							44097
52							44137
53							44198
54							44667
55							44846
56							45032
57							65985
58							67054
59							68294
60	2022-12-10	Anglija	Prancūzija	1	2	Chauras	68895
61	2022-12-06	Portugalija	Šveicarija	6	1	Lusailas	83720
62	2022-12-02	Kamerūnas	Brazilija	1	0	Lusailas	85986
63	2022-12-09	Nyderlandai	Argentina	2	2	Lusailas	88235
64	2022-12-13	Argentina	Kroatija	3	0	Lusailas	88966
65	2022-12-18	Argentina	Prancūzija	3	3	Lusailas	88966

Rūšiavimas ? X

Mano duomenyse yra antraščių

Stulpelis: Rūšiuoti pagal
 Rūšiuoti:
 Tvarka:

00

39	2022-11-30	Lenkija	Argentina	0	2	Doha	39789
40	2022-11-30	Tunisas	Prancūzija	1	0	Rajanas	39369
41	2022-11-30	Australija	Danija	1	0	Vakra	39089
42	2022-12-01	Kanada	Marokas	1	2	Doha	40989
43	2022-12-01	Kroatija	Belgija	0	0	Rajanas	41378
44	2022-12-01	Japonija	Ispanija	2	1	Rajanas	42523
45							43102
46							43443
47							43847
48							43893
49							43984
50							44097
51							44137
52							44198
53							44667
54							44846
55							44851
56							45032
57							65985
58	2022-12-05	Kroatija	Brazilija	1	1	Rajanas	67054
59	2022-12-09	Nyderlandai	Argentina	2	2	Lusailas	68294

Rūšiavimas ? X

Mano duomenyse yra antraščių

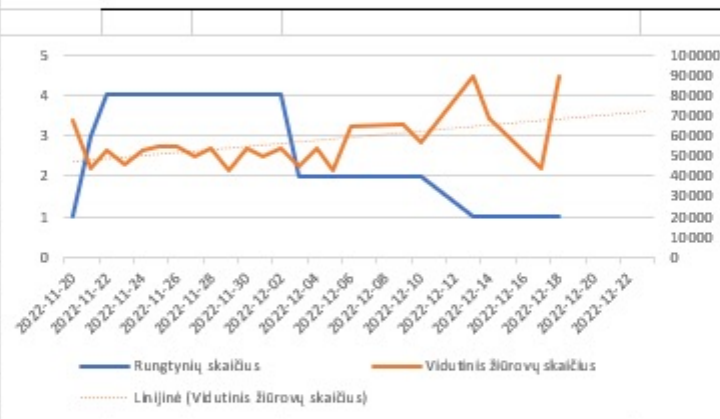
Stulpelis: Rūšiuoti pagal
 Rūšiuoti:
 Tvarka:

00

	A	B	C	D	E	F	G	H	I	J
1	Rungtynių	Rungtynių	Vidutinis žiūrovų skaičius							
2	2022-11-20	1	67372							
3	2022-11-21	3	43491							
4	2022-11-22	4	52795							
5	2022-11-23	4	45615							
6	2022-11-24	4	52879							
7	2022-11-25	4	54669							
8	2022-11-26	4	54477							
9	2022-11-27	4	49622							
10	2022-11-28	4	54022							
11	2022-11-29	4	43212							
12	2022-11-30	4	53483							
13	2022-12-01	4	49748							
14	2022-12-02	4	53726							
15	2022-12-03	2	44939							
16	2022-12-04	2	53487							
17	2022-12-05	2	43185							
18	2022-12-06	2	64194							
19	2022-12-09	2	66064							
20	2022-12-10	2	56547							
21	2022-12-13	1	88966							
22	2022-12-14	1	68294							
23	2022-12-17	1	44137							
24	2022-12-18	1	88966							

9. Darbo lakšte *Diagrama* atlikite šiuos veiksmus:
- 9.1. Įterpkite linijinę diagramą atvaizduoti kiekvienos dienos rungtynių skaičių ir vidutinį žiūrovų skaičių. Diagramą pateikite be pavadinimo;
 - Pastaba. Diagramos viršutinis kairysis kampas turi būti E16 langelyje.
 - 9.2. Sekai *Vidutinis žiūrovų skaičius* parinkite antrinę ašį. Nustatykite, kad sekos *Rungtynių skaičius* ašies skaitinės reikšmės būtų rodomos kas 1;
 - 9.3. Į diagramą įterpkite linijinę krypties liniją sekai *Vidutinis žiūrovų skaičius*, kuri prognozuotų žiūrovų skaičių po 5 dienų;
 - 9.4. remdamiesi esančiais duomenimis ir (ar) nubrėžta diagrama, langelyje F10 pateikite išvadą apie vidutinio žiūrovų skaičiaus priklausomybę nuo varžybų per dieną skaičiaus.

(4 taškai)



9. 9.1. Įterpta linijinė diagrama, atvaizduojanti kiekvienos dienos rungtynių skaičių ir vidutinį žiūrovų skaičių. Diagrama pateikiama be pavadinimo.

Pastaba

Taškas skiriamas, jei teisingai įvykdyti abu nurodymai.

- 9.2. Sekai *Vidutinis žiūrovų skaičius* parinkta antrinė ašis. Sekos *Rungtynių skaičius* ašies skaitinės reikšmės rodomos kas 1.

Pastaba

Taškas skiriamas, jei teisingai įvykdyti abu nurodymai.

- 9.3. Į diagramą įterpta linijinė krypties linija sekai *Vidutinis žiūrovų skaičius*. Prognozuojamas žiūrovų skaičius po 5 dienų.

Pastaba

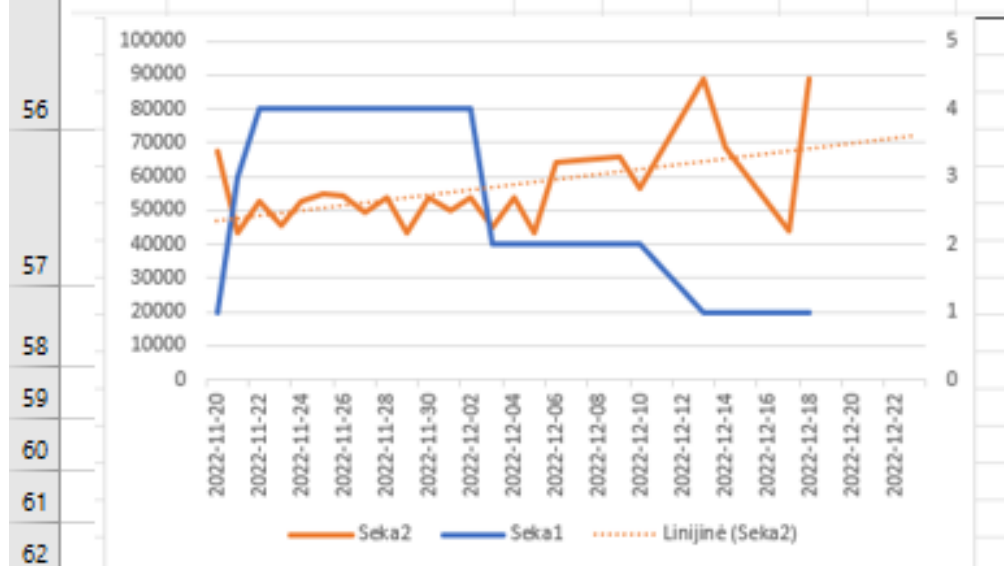
Taškas skiriamas, jei teisingai įvykdyti abu nurodymai.

1

1

1

4



Taškai	Atsakymas	Taškai
	kuo mažiau rungtynių, tuo labiau didėja žiūrovų skaičius.	1
	Didėjant varžybų skaičiui per dieną žiūrovų skaičius irgi linksta didėti.	0
	Bendra tendencija(krypties linija) rodo, kad žiūrovų skaičius kiekvieną dieną kilo, tačiau tam kiek turėjo įtakos ir rungtynių skaičius tą dieną. Dažnesnis kiekis rungtynių tą dieną, reiškia kiek mažesnę vidutinę žiūrovų skaičių tai dienai, nei jei vyktų tik vienos varžybos.	1
0	Kad Pasaulio čempionato atidaryme buvo daug susirunkusių žiūrovų. Kai buvo mažiau varžybų daugiau žiūrovų žiūrėjo jas. Dėl trečios vietos rungtynes	1
0	kad po 5 dienu pakils daugiau žiūrovų skaičius	0
1	Pagal pateiktus duomenis, vidutinis žiūrovų skaičius nepriklauso nuo varžybų kiekio, nes iki maždaug gruodžio 6d. varžybų vyko daugiau negu po gruodžio 6d., tačiau vidutinis žiūrovų prieš gruodžio 6d. Yra mažesnis negu po gruodžio 6d.	0
	Kuo mažesnis rungtynių skaičius, tuo didesnis vidutinis žiūrovų skaičius, tačiau stipresni kitimai. Kuo didesnis rungtynių skaičius, tuo mažesnis vidutinis žiūrovų skaičius, tačiau stabilesnis vidutinis žiūrovų skaičius(mažiau kinta)	1
1	Jei tik 1-os varžybos vyks per dieną, tai daugiau žiūrovų susirinks negu, kai daugiau negu 1-os.	0
0	Žiūrovų skaičius didėja artėjant Čempionato pabaigai.	0
1		



Programavimo praktinė užduotis

Atliekant programavimo užduotį kompiuteriu, reikia (jei nenurodyta kitaip) pagal pateiktą sąlygą laikantis nurodymų sukurti programą.

Kiekviena teisingai suprogramuota užduoties dalis vertinama 1–5 taškais.

Mokymosi turinys (1)

Duomenų struktūrų naudojimas. III–IV gimnazijos klasių koncentras.

III gimnazijos klasė

Apibrėžiama duomenų tipo sąvoka. Mokomasi naudoti vienmačiu masyvu (sąrašu), tekstine eilute. Mokomasi rasti simbolių eilutės ilgį, priskirti, palyginti, sujungti simbolių eilutes. Kuriamos programos nesudėtingiems skaičiavimams, taikomosioms užduotims spręsti. Taikomi įvairūs duomenų tipai, akcentuojamas tekstinių eilučių ir masyvų naudojimas.

Duomenų struktūrų naudojimas. III–IV gimnazijos klasių koncentras.

IV gimnazijos klasė

Mokomasi naudoti standartinėmis pasirinktos programavimo kalbos duomenų struktūromis (klasėmis). Aptariami abstrakčiųjų duomenų struktūrų pavyzdžiai (dinaminis masyvas ir pan.). Aiškinamasi, kokius veiksmus galima atlikti su aptartomis duomenų struktūromis pasirinktoje programavimo kalboje. Kuriamos sudėtinės duomenų struktūros, pavyzdžiui, struktūrų (klasių) masyvas (sąrašas), masyvas (sąrašas) struktūroje (klasėje).

Mokymosi turinys (2)

Darbas su tekstinių duomenų srautais. III–IV gimnazijos klasių koncentras.

III gimnazijos klasė

Mokomasi skaityti iš tekstinių failų (.txt, .csv ir kt.) ir rašyti rezultatus į failus, papildyti rezultatų failus.

Mokomasi skaityti duomenis iš išorinių įrenginių (pavyzdžiui, *Arduino* nuoseklaus prievado) analizuojant tekstą ir (ar) taikyti paprogramių bibliotekas.

Algoritmai. III–IV gimnazijos klasių koncentras.

III gimnazijos klasė

Prisimenami sumos, sandaugos, kiekio, vidurkio skaičiavimo, mažiausios (didžiausios) reikšmės radimo, tiesinės paieškos algoritmai. Mokomasi rikiavimo, reikšmės paieškos nerikiuotame ir rikiuotame masyve, reikšmių šalinimo iš masyvo, masyvo papildymo naujomis reikšmėmis algoritmų.

Algoritmai. III–IV gimnazijos klasių koncentras.

IV gimnazijos klasė

Mokomasi užrašyti loginius reiškinius ir juos panaudoti masyvo (ar struktūrų (klasių) masyvo (sąrašo) rikiavimo, reikšmių paieškos, reikšmių šalinimo, reikšmių papildymo veiksmuose. Konstruojamos programos iš standartinių kalbos ir mokinių parašytų rikiavimo, paieškos algoritmų; akcentuojama jų nauda kuriamai programai.

Mokymosi turinys (3)



Mokymo(si) turinys

Paprogramės. III–IV gimnazijos klasių koncentras.

III gimnazijos klasė

Prisimenama paprogramės, faktinių ir formalių parametrų sąvokos, kaip rašomos paprogramės, kurios grąžina vieną reikšmę per funkcijos vardą, rašomi kreipiniai į jas. Mokomasi rašyti paprogrames, kai parametrais perduodamos reikšmės ir kai parametrais perduodami kintamųjų adresai, tinkamai užrašyti kreipinius į jas.

Programų testavimas ir taisymas. III–IV gimnazijos klasių koncentras.

IV gimnazijos klasė

Diskutuojama apie testavimo strategijas, aiškinamasi, kuo skiriasi įprasti ir ekstremalūs duomenų rinkiniai. Konstruojami pradiniai duomenų rinkiniai, mokomasi naudojantis šiais rinkiniais ieškoti klaidų ir jas taisyti.

Pagrindiniai darbai rengiantis programavimo užduoties vertinimui

Standartizavimas

- parenkami atsitiktinių kandidatų darbai vertinimo instrukcijos tikslinimui;
- tikslinama vertinimo instrukcija.

Bandomasis vertinimas

- Naudojami 2026 m. bandomojo egzamino mokinių atliktų užduočių pavyzdžiai.

Bandomojo egzamino programavimo praktinė užduotis (1)

Bitlandijos karalystės mokykloje artėja egzaminų metas. Siekiant sklandžiai organizuoti egzaminų pasirinkimus, buvo parengta mokinių apklausa. Kiekvienam mokiniui buvo suteiktas asmeninis identifikatorius (ID), o kiekvieno dalyko egzaminui – atskiras kodas. Apklausos metu mokiniai turėjo nurodyti, kurių dalykų egzaminus pasirinko laikyti. Apklausa buvo sukurta taip, kad mokinys negalėtų jos užpildyti daugiau nei vieną kartą ir neturėtų galimybės pasirinkti to paties dalyko egzamino kelis kartus.

Pradiniai duomenys

Pradiniai duomenys saugomi faile *Duomenys.txt*.

- Pirmoje failo eilutėje pateiktas mokinių skaičius n ($1 \leq n \leq 500$).
- Tolesnėse n eilutėse pateikiamas mokinių sąrašas, kuriame nurodoma:
 - mokinio vardas ir pavardė;
 - mokinio ID, sudarytas iš 10 simbolių: pirmųjų trijų abituriento vardo raidžių, pirmųjų trijų pavardės raidžių ir keturių atsitiktinių skaitmenų, kurių suma turi būti lygi 15;
 - mokinio klasė (12a, 12b arba 12c).
- Tolesnėse failo eilutėse pateikiami apklausos duomenys: apklausą užpildžiusių mokinių ID ir jų pasirinktų dalykų egzaminų kodai. Dalyko egzamino kodas – simbolių seka, sudaryta iš didžiųjų lotyniškų raidžių.

Duomenų laukai faile atskirti kabliataškiu (:). Failo ilgis – ne daugiau kaip 1001 eilutė. Daroma prielaida, kad faile nebus raidžių su diakritiniais ženklais (lietuvių ar kitų kalbų), naudojama vieno baido koduotė.

Bandomojo egzamino programavimo praktinė užduotis (2)

Užduotis

Parašykite programą, kuri nuskaitytų pradinį duomenį, atliktų mokinių duomenų analizę ir išveda rezultatus į failą *Rezultatai.txt* pagal toliau pateiktus nurodymus.

1. Programos stilius ir komentarai:
 - 1.1. rašydami programą laikykitės vientiso kodo stiliaus;
 - 1.2. naudokite prasmingus kintamųjų pavadinimus;
 - 1.3. komentuokite pagrindines programos dalis.

(1 taškas)

1. Programos stilius ir komentarai 1.1. Rašydami programą laikykitės vientiso kodo stiliaus. 1.2. Naudokite prasmingus kintamųjų pavadinimus. 1.3. Komentuokite pagrindines programos dalis.	1	Taškas skiriamas, kai įvykdyti visi trys nurodymai. Ar tiksliname šią vertinimo instrukcijos dalį? Jei kintamųjų vardai ir funkcijų pavadinimai yra prasmingi (pilni žodžiai) taškas už komentarus skiriamas.
---	---	---

Bandomojo egzamino programavimo praktinė užduotis: stilius ir komentarai (3)

```
#include<bits/stdc++.h>
using namespace std;
const char CD [] = "Duomenys.txt";
const char CR [] = "Rezultatai.txt";

struct Mokinyas { string vardas;
int pavarde;
int ID;
int n;

void Skaityti (Mokinyas M[], int n, int m);
void Rikiuoti (Mokinyas M[], int n, int m);

int main(){
void Skaityti (Mokinyas M[50], int n, int m);
ifstream fd (Duomenys.txt);
ofstream fr (Rezultatai.txt);

return 0;
}

void Skaityti (Mokinyas M[], int n, int m){
ifstream fd (Duomenys.txt);
char vardas [50];
fd(getline(eilute, char k);
M[i]= vardas;
fd>>k;
for (int i=0;i<n;i++){

}

void Rikiuoti (Mokinyas M[], int n, int m){
for (int j=0;j<i-1;j++){
ios::app;
for (int i=0;i<n;i++){
```

```
#include<bits/stdc++.h>
using namespace std;
const char CD[] = "Duomenys.txt";
const char CR[] = "Rezultatai.txt";

Struct Mokinyas {
string eilute;
int vardas;
int pavarde;
double ID;
int klase;
}

void Skaityti (vector<Mokinyas>M; string eilute);
int kiek (vector<Mokinyas>M; int klase);
void ArteisingasID (double ID; string eilute);
void Rasyti (vector<Mokinyas>M; int kiek );

int main() {
Skaityti (M);
Rasyti (M, ID);

return 0;
}

void Skaityti (vector<Mokinyas>M; string eilute){
string eilute;
ifstream fd (CD);
fd>>n;
for (int i=0; i<M.size(); i++){
while (getline(fd, eilute);
fd>>vardas>>pavarde>>ID>>klase>>endl;
}

int kiek (vector<Mokinyas>M; int klase){
int j=aklase;
int z=bklase;
int k=cklase;
for(int i=0; i<M.size; i++){
if (i=="a")j++;
else if (i=="b")z++;
else k++;
}

}

void ArteisingasID (double ID; string eilute){
int l=neteisingiID;
for(int i=0; i<M.size(); i++){
if (ID ) true;
else false;
return l;
}

}

void Rasyti (vector<Mokinyas>M; int kiek ){
ofstream fr (CR);
fr << j << z << k << endl;
for (int i=0; i<M.size(); i++){
fr << l << endl;
}

}

}
```

```
#include<bits/stdc++.h>
using namespace std;
const char CD[] = "Duomenys.txt";
const char CR[] = "Rezultatai.txt";

struct Mokinyas{
string v, pav, id, klase;
};

void Skaityti(Mokinyas M[], int &n);

int main() {

Mokinyas M[500];

return 0;
}

//-----
void Skaityti(Mokinyas M[], int &n){
int n;
char k;
ifstream fd(CD);
fd >> n;
string eilute;
getline(fd, eilute);
stringstream (getline, ss);
for(int i=0; i<=n; i++){
ss >> M[i].v >> k >> M[i].pav >> k >> M[i].id >> k >> M[i].klase;
}
ss >> M[] .id;
fd.close();
}

//-----
```

Bandomojo egzamino programavimo praktinė užduotis (4)

2. Programos testavimas ir taisymas: rašydami programą, nuolat ją testuokite ir taisykite atsiradusias sintaksės bei logikos klaidas.

(1 taškas)

2. Programos testavimas ir taisymas
2.1. Rašydami programą, nuolat ją testuokite ir taisykite atsiradusias sintaksės bei logikos klaidas.

- 1 | Taškas skiriamas, kai yra parašyta programa (veikia, ar beveik veikia nors viena skaičiavimo užduotis) bei programoje nėra nei vienos sintaksės klaidos.

Ar tiksliname šią vertinimo instrukcijos dalį?
Kaip vertinti, jei yra smulki sintaksės klaida?

Bandomojo egzamino programavimo praktinė užduotis: pavyzdys (5)

Skirti tašką, ar neskirti?

```
#include<bits/stdc++.h>
using namespace std;

struct Mokinyas{string vardasp; string mokinioID; string klase;};

struct Blogimokiniai{string vardasp; string mokinioID; int suma;};

struct Apklausa{string mokinioID; string E[1000]; int kiek;};

void Skaityti(Mokinyas M[], vector<Apklausa>&A, int&n);

void Kiekiai(Mokinyas M[], int&n, int&A, int&B, int&C);

void IDtikrinimas(Mokinyas M[], int&n, Blogimokiniai Z[], int&z);

void Pasirinkimas(vector<Apklausa>&A, Mokinyas M[], string W[], int&w, int&n);

void Rasyti(int&X, int&B, int&C, int&z, Blogimokiniai Z[], string W[], int&w);

int main(){
Mokinyas M[500];
Blogimokiniai Z[500];
vector<Apklausa>A;
string W[500];
int n;
int w = 0;
int z = 0;
Skaityti(M, A, n);

int X = 0;
int B = 0;
int C = 0;
Kiekiai(M, n, X, B, C);
IDtikrinimas(M, n, Z, z);
Pasirinkimas(A, M, W, w, n);
Rasyti(X, B, C, z, Z, W, w);

}
```

```
void Skaityti(Mokinyas M[], vector<Apklausa>&A, int&n){
ifstream fd("Duomenys.txt");
fd>>n;
fd.ignore();

for(int i = 0; i<n; i++){
string eilute1;
if(i==0){
getline(fd, eilute1);
}
getline(fd, eilute1);
stringstream ss(eilute1);

getline(ss, M[i].vardasp, ' ');
getline(ss, M[i].mokinioID, ' ');
ss>>M[i].klase;
}
// OK

string eilute;
int K = 0;

while(getline(fd, eilute)){
Apklausa a;
stringstream ss(eilute);

getline(ss, a.mokinioID, ' ');

while(getline(ss, a.E[K], ' ')){
getline(ss, a.E[K], ' ');
K++;
}
a.kiek = K;
A.push_back(a);
}
}
```

```
void IDtikrinimas(Mokinyas M[], int&n, Blogimokiniai Z[], int&z){

for(int i = 0; i<n; i++){
int suma = 0;
string eilute = M[i].mokinioID;
for(int j = 0; j<eilute.length(); j++){
if(isdigit(eilute.at(j))) {suma+=stoi(eilute.at(j));} //problema cia
}

if(suma!=15){

Z[z].vardasp = M[i].vardasp;
Z[z].mokinioID = M[i].mokinioID;
Z[z].suma = suma;
z++;

void Pasirinkimas(vector<Apklausa>&A, Mokinyas M[], string W[], int&w, int&n){
for(int i = 0; i<A.size(); i++){
int Q = 0;
int R = 0;
for(int j = 0; j<A[i].kiek; j++){
if(A[i].E[j]=="BL") Q++;
if(A[i].E[j]=="MAT") R++;
}
}if(Q==0||R==0){
for(int c = 0; c<n; c++){
if(M[c].mokinioID==A[i].mokinioID){
W[w] = M[c].vardasp;
w++;
}
}
}

void Rasyti(int&X, int&B, int&C, int&z, Blogimokiniai Z[], string W[], int&w){
ofstream fr("Rezultatai.txt");
fr<<X<<" "<<B<<" "<<C<<" "<<endl;
fr<<z<<endl;
for(int i = 0; i<z; i++){
fr<<Z[i].vardasp<<" "<<Z[i].mokinioID<<" "<<Z[i].suma<<endl;
}
for(int i = 0; i<w; i++){
fr<<W[i]<<endl;
}
}
```

Bandomojo egzamino programavimo praktinė užduotis (6)

3. Duomenų struktūros pasirinkimas: aprašykite pasirinktą duomenų struktūrą mokinio duomenims saugoti.

(1 taškas)

3. Duomenų struktūros pasirinkimas 3.1. Aprašykite pasirinktą duomenų struktūrą mokinio duomenims saugoti.	1	Taškas skiriamas, kai teisingai aprašyta pasirinkta duomenų struktūra, kiti kintamieji ir jie naudojami veiksmuose.
---	---	---

Ar tiksliname šią vertinimo instrukcijos dalį?
Turi būti būtini teisingi laukeliai (pagal sąlygą),
gali būti papildomų, jei juose saugomi
skaičiavimų rezultatai.

Bandomojo egzamino programavimo praktinė užduotis: duomenų struktūros (7)

```
#include<bits/stdc++.h>
using namespace std;
const char CD[] = "Duomenys.txt";
const char CR[] = "Rezultatai.txt";

struct Mokinys{
    string v, pav, id, klase;
};

void Skaityti(Mokinys M[], int &n);

int main(){
    Mokinys M[500];

    return 0;
}
//-----
void Skaityti(Mokinys M[], int &n){
    int n;
    char k;
    ifstream fd(CD);
    fd >> n;
    string eilute;
    getline(fd, eilute);
    stringstream (getline, ss);
    for(int i=0; i<=n; i++){
        ss >> M[i].v >> k >> M[i].pav >> k >> M[i].id >> k >> M[i].klase;
    }
    ss >> M[0].id;
    fd.close();
}
//-----
```

```
#include<bits/stdc++.h>
using namespace std;

struct Mokinys{
    string v, pav, id, klase, kodas;
};

void Skaityti(vector<Mokinys>&M);
void Skaic(vector<Mokinys>&M);

int main(){
    vector<Mokinys>M;

    Skaityti(M);
    Skaic(M);

    return 0;
}

void Skaityti(vector<Mokinys>&M){
    ifstream fd(CD);
    int n;
    fd>>n;
    char kabl;
    Mokinys m;
    string eilute;
    while(getline(fd, eilute)){
        stringstream ss(eilute);
        getline(ss, m.v, ' ');
        getline(ss, m.pav, ',');
        getline(ss, m.id, ',');
        getline(ss, m.klase, ' ');
        ss >> m.id >> kabl >> m.kodas;
        M.push_back(m);
    }
    fd.close();
}

void Skaic(vector<Mokinys>&M){
```

Bandomojo egzamino programavimo praktinė užduotis (8)

4. Mokinių skaičius klasėse:

4.1. Nuskaitykite mokinių sąrašą.

(2 taškai)

4.2. Suskaičiuokite ir į rezultatų failą *Rezultatai.txt* įrašykite, kiek kiekvienos klasės (12a, 12b ir 12c) mokinių yra duotame sąraše.

(3 taškai)

4. Mokinių skaičius klasėse	(5)	Du taškai skiriami už teisingai nuskaitytą ir išsisaugotą pradinių duomenų sąrašą.
4.1. Nuskaitykite mokinių sąrašą.	2	
4.2. Suskaičiuokite ir rezultatų faile <i>Rezultatai.txt</i> pateikite, kiek mokinių mokosi 12a, 12b ir 12c klasėse.	2	Du taškai skiriami už teisingai apskaičiuotus mokinių skaičius 12a, 12b ir 12c klasėse.
	1	Vienas taškas skiriamas už teisingai į rezultatų failą atspausdintus mokinių skaičius (3 skaičiai, atskirti tarpu) klasėse.

Reikia susitarti už ką skiriama po 1 tašką.

1 taškas už n reikšmės nuskaitymą ir ciklo antraštę.

1 taškas už sąrašo formavimą.

1 taškas už dalinai teisingai skaičiuojamus mokinius skaičius.

2 taškai už teisingus mokinių skaičius.

1 taškas už tinkamą spausdinimą faile.

Bandomojo egzamino programavimo praktinė užduotis: mokinių skaičius klasėse (9)

```
#include<bits/stdc++.h>
using namespace std;

const char FD[]="Duomenys.txt";
const char CD[]="Rezultatai.txt";

struct Mokinyas{string vardas, pavarde, id, klase;};

void Skaityti(vector<Mokinyas>&M);
int Artikras(vector<Mokinyas>&M);

int main(){
    Mokinyas M;
    Skaityti(M);
    ofstream fr();
    fr<<dva<<" "<<dvb<<" "<<dvc<<endl;

return 0;
};

void Skaityti(vector<Mokinyas>&M){
    ifstream fd();
    Mokinyas m;
    int n, dva, dvb, dvc;
    char k;
    string eilute;
    fd>>n;
    while(getline(fd, eilute)){
        stringstream ss(eilute);
        for(int i=0; i<n; i++){
            getline(ss, m.vardas, ',');
            getline(ss, m.pavarde, ',');
            getline(ss, m.id, ',');
            getline(ss, m.klase, ',');
            if(m[i].pavarde=="12a"){
                dva++;
            }
            else if(m[i].pavarde=="12b"){
                dvb++;
            }
            else if(m[i].pavarde=="12c"){
                dvc++;
            }
        }
    }

M.push_back(m);
return dva, dvb, dvc;
}

fd.close();
};

int Artikras(vector<Mokinyas>&M){
    for(int i=0; i<M.size(); i++){
        if()
    }
}
```

```
include <iostream>
include <fstream>

using namespace std;

struct Mokynis{
    int var;
    int pav;
    int v;
    int p;
    int e;
    int f;
    int g;
    int h;
    int kl;
};

int Mokynis M[1000];

int{
    int Mokynis M[1000];
    ifstream fd("Duomenys.txt");
    ofstream fr("Rezultatai.txt");
    int n, k=';', a=0, b=0, c=0, l=0, n=0;
    fd>>n;
    for (int i=0; i<=n; i++){
        fd>>M[i].var>>" ">>M[i].pav>>k>>M[i].v>>M[i].p>>M[i].e>>M[i].f>>M[i].g>>M[i].h>>x>>M[i].kl>>endl;
        if (M[i].kl=='12a') a++;
        if (M[i].kl=='12b') b++;
        if (M[i].kl=='12c') c++;
        if (M[i].e+M[i].f+M[i].g+M[i].h!=15) {
            l++;
            n=i;
        }
    }

    for (int i=0; i<=n; i++){

fr<<a<<" "<<b<<" "<<c<<endl;
fr<<l<<endl;
fr<<M[i].var<<" "<<M[i].pav<<" "<<M[i].v<<M[i].p<<M[i].e<<M[i].f<<M[i].g<<M[i].h<<" "<<M[i].e+M[i].f+M[i].g+M[i].h<<endl;
fr<<a<<" "<<b<<" "<<c<<endl;
fr<<a<<" "<<b<<" "<<c<<endl;
}
```

Bandomojo egzamino programavimo praktinė užduotis (10)

5. Neteisingi mokinio ID:

5.1. Mokinio ID sudarytas iš 10 simbolių: pirmi trys – vardo pirmosios raidės, kiti trys – pavardės pirmosios raidės, o paskutiniai keturi – atsitiktiniai skaitmenys, kurių suma turi būti lygi 15.

Parašykite funkciją, kuri patikrina, ar mokinio ID skaitmenų suma lygi 15.

(3 taškai)

5.2. Apskaičiuokite, keliems mokiniams neteisingai suformuoti ID sudarantys skaitmenys, t. y. skaitmenų suma nėra lygi 15.

(1 taškas)

5.3. Papildykite rezultatų failą. Antroje eilutėje pateikite mokinių, kurių ID skaitmenų suma nėra 15, skaičių. Toliau pateikite šių mokinių sąrašą. Kiekvienoje eilutėje pateikite mokinio vardą ir pavardę, mokinio ID bei skaitmenų sumą. Sąrašą pateikite tokia eilės tvarka, kokia jis pateiktas pradinio duomenų faile. Rezultatų laukus atskirkite vienu tarpo simboliu.

(2 taškai)

Reikia susitarti už ką skiriama po 1 tašką.

1 taškas už dalinai teisingai apskaičiuotą sumą, 2 taškai už teisingai apskaičiuotą sumą.

1 taškas, jei tinkama tvarka spausdinami vieno mokinio duomenys, 2 taškai skiriami, jei teisingai spausdinamas sąrašas.

5. Neteisingi mokinio ID

5.1. Mokinio ID sudarytas iš 10 simbolių: pirmi trys – mokinio vardo pirmosios raidės, kiti trys – pavardės pirmosios raidės, o paskutiniai keturi – atsitiktiniai skaitmenys, kurių suma turi būti lygi 15.

Parašykite funkciją, kuri patikrina, ar mokinio ID paskutinių keturių skaitmenų suma lygi 15.

5.2. Apskaičiuokite, keliems mokiniams neteisingai priskirti paskutiniai keturi ID simboliai.

5.3. Papildykite rezultatų failą. Antroje eilutėje spausdinkite mokinių, kuriems neteisingai priskirti paskutiniai keturi ID skaitmenys, skaičių. Toliau pateikite šių mokinių sąrašą. Kiekvienoje eilutėje spausdinkite mokinio vardą ir pavardę, mokinio ID bei paskutinių keturių skaitmenų sumą. Sąrašą pateikite tokia eilės tvarka, kokia jis pateiktas pradinio duomenų faile. Rezultatų laukus atskirkite vienu tarpo simboliu.

(6)

1

Vienas taškas skiriamas už sukurtą ir panaudotą funkciją, kuri patikrina, ar mokinio ID paskutiniai keturi skaitmenys sudaro sumą, lygią 15.

2

Du taškai skiriami už teisingai apskaičiuotą sumą iš mokinio ID.

1

Vienas taškas skiriamas už teisingai suskaičiuotą keliems mokiniams neteisingai priskirti paskutiniai keturi ID simboliai.

2

Du taškai skiriami už teisingai papildytą rezultatų failą, kuriame teisinga tvarka spausdinami mokinių vardai, pavardės ir ID bei rezultatai atskirti vienu tarpo simboliu.

Bandomojo egzamino programavimo praktinė užduotis: mokinio ID (11)

```
struct NetinkantisMokinys{ //Mokinys kurio ID suma nelygi 15
    string Vardas;
    string Pavarde;
    string ID;
    int IDsuma;
};

bool ArSuma15(Mokinys mokinys)
{
    int IDsuma = 0;

    //Suskaiciuojame ID skaiciu suma
    for(int j = 6; j < 10; j++)
    {
        //Is ID char gauname skaiciu kuri pridedame sumai
        mokinys.ID[j] - '0';

        IDsuma += mokinys.ID[j] - '0';
    }

    if(IDsuma == 15) return true;

    return false;
}

int KokiaIDsuma(Mokinys mokinys)
{
    int IDsuma = 0;

    //Suskaiciuojame ID skaiciu suma
    for(int j = 6; j < 10; j++)
    {
        //Is ID char gauname skaiciu kuri pridedame sumai
        mokinys.ID[j] - '0';

        IDsuma += mokinys.ID[j] - '0';
    }

    return IDsuma;
}

int neteisingsum = 0;
for (int i = 0; i < n; i++){ //mokinių kodo sumos tikrinimas
    int sum = 0;
    for (int g = 10; g > 5; g--){
        if (mok[i].kod[g] == '0') sum+=0;
        if (mok[i].kod[g] == '1') sum+=1;
        if (mok[i].kod[g] == '2') sum+=2;
        if (mok[i].kod[g] == '3') sum+=3;
        if (mok[i].kod[g] == '4') sum+=4;
        if (mok[i].kod[g] == '5') sum+=5;
        if (mok[i].kod[g] == '6') sum+=6;
        if (mok[i].kod[g] == '7') sum+=7;
        if (mok[i].kod[g] == '8') sum+=8;
        if (mok[i].kod[g] == '9') sum+=9;
    }
    if (sum != 15) {
        neteisingsum++;
        mok[i].sum = sum;
    }
}

out << neteisingsum << endl; //klaidingų sumų išvedimas
//-----

int Konvertuoti(char s){
    int k;
    if(s=='0') k=0;
    if(s=='1') k=1;
    if(s=='2') k=2;
    if(s=='3') k=3;
    if(s=='4') k=4;
    if(s=='5') k=5;
    if(s=='6') k=6;
    if(s=='7') k=7;
    if(s=='8') k=8;
    if(s=='9') k=9;
    return k;
}

bool Tikrinti(string id){
    return(Konvertuoti(id[9])+Konvertuoti(id[8])+Konvertuoti(id[7])+Konvertuoti(id[6])!=15);
}
//-----

void SimboliuSuma(Mokinys A[], int n, int&kiekBlogaiId){
    // randame ar simboliu suma yra 15
    for(int i=0;i<n;i++){
        // gauname paskutinius 4 skaicius ir suskaiciuojame ju suma
        string galas = A[i].id.erase(0, 6);
        int skaicius = stoi(galas);
        A[i].idSuma = (skaicius / 1000) + (skaicius%1000/100) + (skaicius%100/10) + (skaicius%10);
        if(A[i].idSuma != 15) kiekBlogaiId++;
    }
}
```

Bandomojo egzamino programavimo praktinė užduotis (12)

6. Neteisingi pasirinkimai:

6.1. Nuskaitykite apklausos duomenis.

(4 taškai)

6.2. Raskite mokinius, kurie neteisingai pasirinko egzaminų dalykus. Mokinys gali laikyti nuo trijų iki septynių dalykų egzaminus ir privalo pasirinkti matematikos (dalyko kodas – MAT) bei bitlandų kalbos (dalyko kodas – BL) egzaminus – toks pasirinkimas laikomas teisingu.

(2 taškai)

6.3. Papildykite rezultatų failą. Pateikite neteisingai dalykų egzaminus pasirinkusių mokinių vardus ir pavardes. Sąrašą pateikite ta pačia tvarka, kokia jis pateiktas pradinių duomenų faile, mokinių sąrašė.

(2 taškai)

Reikia susitarti už ką skiriama po 1 tašką

Apklausos duomenys: 1 taškas – išorinis while, 1 taškas – eilutės skaidymas, 1 taškas – vidinis ciklas, 1 taškas – tinkamai formuojamas sąrašas.

Mokinių pasirinkimai: 1 taškas – dalinai teisingai, 2 taškai – teisingai atlikta užduotis.

Neteisingai egzaminus pasirinkę mokiniai: 1 taškas – vieno mokinio duomenys teisingai, 2 taškai – teisingas mokinių sąrašas.

6. Neteisingi pasirinkimai

6.1. Nuskaitykite apklausos duomenis.

6.2. Raskite mokinius, kurie neteisingai pasirinko egzaminus. Mokinys gali laikyti nuo trijų iki septynių egzaminų ir privalo pasirinkti matematikos (dalyko kodas – MAT) bei bitlandų kalbos egzaminą (dalyko kodas – BL) – toks pasirinkimas laikomas teisingu.

6.3. Papildykite rezultatų failą. Spausdinkite neteisingai egzaminus pasirinkusių mokinių vardus ir pavardes. Sąrašą pateikite ta pačia tvarka, kokia jis pateiktas pradinių duomenų faile, mokinių sąrašė.

8

4

2

2

4 taškai skiriami už teisingai pabaigtus nuskaityti duomenis.

Du taškai skiriami už teisingai surastus mokinių, kurie neteisingai pasirinko egzaminus.

Du taškai skiriami už teisingai papildytą rezultatų failą, kuriame teisinga tvarka pateikiami mokinių, kurie neteisingai pasirinko egzaminą, vardai ir pavardės.

Bandomojo egzamino programavimo praktinė užduotis (13)

7. Dalykų sąrašas:
- 7.1. Raskite, kiek mokinių pasirinko kiekvieną dalyko egzaminą. *(4 taškai)*
- 7.2. Sąrašą surikiuokite pagal mokinių skaičių mažėjančiai, o jei skaičiai vienodi – pagal dalyko egzamino kodą didėjančiai. *(2 taškai)*
- 7.3. Papildykite rezultatų failą gautais rezultatais. Kiekvienoje eilutėje pateikite dalyko egzamino kodą ir jį pasirinkusių mokinių skaičių. *(2 taškai)*

Reikia susitarti už ką skiriama po 1 tašką

7.1. – 1 taškas už dalinai teisingai suformuotą dalykų sąrašą, 2 taškai – už teisingai suformuotą dalykų sąrašą.

1 taškas už dalinai teisingus skaičiavimus, 2 taškai – už teisingus skaičiavimus.

7.2. – 1 taškas – sukuria (taiko) rikiavimo algoritmą, 2 taškai – tinkamai rikiuoja sąrašą.

7.3. – 1 taškas – tinkamai spausdina vieno dalyko duomenis, 2 taškai – tinkamai spausdina visų dalykų duomenis.

7. Dalykų sąrašas	(8)	
7.1. Raskite, kiek mokinių pasirinko kiekvieną egzamino dalyką.	2	Du taškai skiriami už teisingai suformuotą egzaminų dalykų kodų sąrašą (sąrašas unikalus).
	2	Du taškai skiriami už teisingą suskaičiavimą, kiek mokinių pasirinko kiekvieną dalyką.
7.2. Sąrašą surikiuokite pagal mokinių skaičių mažėjančia tvarka, o jei skaičiai vienodi – pagal egzamino dalyko kodą didėjančiai.	2	Du taškai skiriami už teisingą rikiavimą (rikiuojama pagal abu raktus).
7.3. Papildykite rezultatų failą gautais rezultatais. Kiekvienoje eilutėje spausdinkite egzamino dalyko kodą ir jį pasirinkusių mokinių skaičių.	2	Du taškai skiriami už teisingai papildytą rezultatų failą, kuriame teisinga tvarka pateikiami dalyko kodai ir jį pasirinkusių mokinių skaičius.

Bandomojo egzamino programavimo praktinė užduotis: dalykai (14)

```
while(getline(fd,eilute)){
    stringstream ss(eilute);
    getline(ss,p.id,',');
    for(int j=0; j<eilute.size(); j++){
        getline(ss,dal,',');
        if(dal=="BL") p.bl=1;
        else if(dal=="MAT") p.mat=1;
        else if(dal=="GEO") p.geo=1;
        else if(dal=="HIS") p.his=1;
        else if(dal=="BIO") p.bio=1;
        else if(dal=="ENG") p.eng=1;
        else if(dal=="INF") p.inf=1;
        else if(dal=="POL") p.pol=1;
        else if(dal=="CHE") p.che=1;
        else p.phy=1;
    }
    P[i].push_back(p);
}

int MAT =0;int ENG =0;
int GEO =0;int INF =0;
int HIS =0;int POL=0;
int BIO =0;int CHE =0;
int PHY =0; int BL=0;
int l = 9;

for (int i=0;i<n;i++){
    for (int j=0; j<20;j++){
        if (M[i].P[j].kodas == "MAT") MAT++;
        if (M[i].P[j].kodas == "ENG") ENG++;
        if (M[i].P[j].kodas == "GEO") GEO++;
        if (M[i].P[j].kodas == "INF") INF++;
        if (M[i].P[j].kodas == "HIS") HIS++;
        if (M[i].P[j].kodas == "POL") POL++;
        if (M[i].P[j].kodas == "BIO") BIO++;
        if (M[i].P[j].kodas == "CHE") CHE++;
        if (M[i].P[j].kodas == "PHY") PHY++;
        if (M[i].P[j].kodas == "BL") BL++;
    }
}

vector<Dalykas>D;
for (int i = 0;i<15;i++){
    Dalykas a; a.pav = "MAT"; a.kiek = MAT; D.push_back(a);
    Dalykas b; b.pav = "ENG"; b.kiek = ENG;D.push_back(b);
    Dalykas c; c.pav = "GEO"; c.kiek = GEO;
    Dalykas d; d.pav = "INF"; d.kiek = INF;
    Dalykas e; e.pav = "HIS"; e.kiek = HIS;
    Dalykas t; t.pav = "POL"; t.kiek = POL;
    Dalykas u; u.pav = "BIO"; u.kiek = BIO;
    Dalykas h; h.pav = "CHE"; h.kiek = CHE;
    Dalykas p; p.pav = "PHY"; p.kiek = PHY;
    Dalykas l; l.pav = "BL"; l.kiek = BL;
}
```

```
egz[0].pav = "BL";
egz[1].pav = "MAT";
egz[2].pav = "GEO";
egz[3].pav = "HIS";
egz[4].pav = "BIO";
egz[5].pav = "ENG";
egz[6].pav = "INF";
egz[7].pav = "POL";
egz[8].pav = "CHE";
egz[9].pav = "PHY";
```

```
for (int i = 0; i < n; i++){ //egzaminų laikymo patvirtinimas, egzaminų skaičiavimas, BL ir MAT egzaminų laikymo patvirtinimas
    string eil;
    string kod;
    getline(in, eil);
    char a[11] = {0};
    for (int g = 0; g < 10; g++){
        a[g] = eil[g];
    }
    for (int g = 0; g < n; g++){
        if (a == mok[g].kod)
            mok[g].laik++;
    }
    for (int g = 10; g < eil.size(); g++){
        if (eil[g] == 'B') {
            if (eil[g+1] == 'L') {
                egz[0].sk++;
                mok[i].bl++;
            }
            if (eil[g+1] == 'I') egz[4].sk++;
        }
        if (eil[g] == 'M') {
            mok[i].mat++;
            egz[1].sk++;
        }
        if (eil[g] == 'G') {
            if (eil[g+1] == 'E') egz[2].sk++;
        }
        if (eil[g] == 'H') {
            if (eil[g-1] != 'C' && eil[g-1] != 'P') egz[3].sk++;
        }
        if (eil[g] == 'E') {
            if (eil[g+1] == 'N') egz[5].sk++;
        }
        if (eil[g] == 'I') {
            if (eil[g+1] == 'N') egz[6].sk++;
        }
        if (eil[g] == 'P') {
            if (eil[g+1] == 'O') egz[7].sk++;
        }
        if (eil[g] == 'C') egz[8].sk++;
        if (eil[g] == 'P') {
            if (eil[g+1] == 'H') egz[9].sk++;
        }
        if (eil[g] == ';') mok[i].egzek++;
    }
}
```



Klausimų ir struktūrinių klausimų dalies vertinimo ypatybės

Bandomojo Informatikos VBE II dalies klausimai ir struktūriniai klausimai

5 klausimai / 10 taškų / 6 proc. visų taškų

1.Skaitmeninio turinio kūrimas

1.1.–1.3. Animuotų kompiuterinės grafikos objektų kūrimas. (4 taškai)

2.Skaitmeninio turinio kūrimas

Elektroninių leidinių rengimas. (1 taškas)

3.Saugus elgesys

Asmens duomenų teisėtas naudojimas. (1 taškas)

4.Saugus elgesys

Elektroninis parašas ir duomenų šifravimas. (1 taškas)

5.Saugus elgesys

5.1.–5.3. Asmens duomenų teisėtas naudojimas. (3 taškai)

1.1. klausimas

1. Kuriate svetainę jaunesniems mokiniams apie saugų elgesį internete. Svetainėje numatyta naudoti trumpą animaciją.

1.1. Nurodykite galimą animacijos paskirtį svetainėje.

(1 taškas)

Galimi atsakymai:

- Animacijos paskirtis – aiškiai ir vaizdžiai parodyti procesą, kaip saugiai elgtis internete, kad būtų lengviau suprasti ir įsiminti informaciją.
- Animacijos paskirtis – judančiais vaizdais perteikti saugaus elgesio taisykles internete.

Kiti panašūs atsakymai, kuriuose minimi animacijos elementai – judėjimas, vaizdai, kadrai, objektams „suteikiama gyvybė“ ir pan.

*1 taškas skiriamas už **animacijos paskirties** įvardijimą.*

Jei kandidatas pateikė daugiau nei vieną atsakymą, vertinamas tik pirmas.

1.1. klausimas (2)

Teisingi atsakymai	Klaidingi atsakymai
<p>1) Jei atsakyme minima, kaip animacija bus susieta su sąlygoje aprašyta svetaine, nurodant jos turinį ir / arba tikslinę auditoriją.</p> <p>2) Jei atsakyme minima, kad animacija bus skirta sudominti</p> <p>3) Jei atsakyme minima, kad animacija bus skirta sąlygoje nurodytai tikslinei grupei</p> <p>4) Jei atsakyme nurodoma, kad animacija bus skirta mokymo tikslams, nedetalizuojant nei turinio, nei tikslinės auditorijos</p>	<p>1) Jei atsakyme minima, kad animacija padės gražiau / patraukliau / įdomiau / lengviau suprasti / perteikti informaciją arba nurodoma, kas bus anumuojama, tačiau nenurodyta, kaip animacija bus susieta su sąlygoje aprašyta svetaine ar auditorija;</p> <p>2) nurodoma, kas ar/ir kaip bus animuojama;</p> <p>3) atsakymas yra ne į pateiktą klausimą</p>
Pritraukti jaunesnių mokinių dėmesį ir perteikti jiems žinutę kaip saugiai elgtis internete.	Pagauti dėmesį lankytojo, pateikti informacija idomesniu būdu.
Galimas nesaugaus elgesio internete pasekmes. Tai turi būti trumpa animacija, kuri pritrauktų dėmesį.	Animacijos paskirtis - perteikti duomenis taip, kad jie butu suprantami, lengviau įskaitomi.
Parodyti kodėl yra svarbu naršant internetu nespausti ant reklamų.	grazus vaizdas uzeinant i svetaine
Animacijos paskirtis gali būti informacinė apie slaptažodžių saugumą, tai yra informuoti jaunesnius mokinius apie saugų elgesį internete.	Animuoti paveikslėlius, kad būtų gražus perėjimas į kitą langą/puslapį.
Sudominti skaitytoją. Aiškiau perteikti mintį.	Gali būti hiper mygtukas nuvedantis į kita svetainės dalį
Pritraukti mokinių dėmesį.	Galma padaryti Tekso ar raidziu judejima.
Labiau intriguoti ir sudominti jaunesnius mokinius, sukuriant trumpą animaciją svetainėje.	Animacija gali būti naudojama reklamai, ar informacijai perteikti.
Padaryti svetainę "smagesne" jaunesniems mokiniams.	parduodamų produktų interaktyviam demonstravimui.
Animacija gali būti skirta pamokinti mokinius arba tiesiog dėl grožio.	
Estetinė, mokomoji paskirtis.	

1.2. klausimas (2)

Teisingi atsakymai (2 taškai)	Iš dalies teisingi (1 taškas)	Klaidingi atsakymai (0 taškų)
Aiškiai įvardytas animacijos elementas ir jo paskirtis	Įvardytas elementas, bet neįvardyta jo paskirtis	Atsakymai ne į pateiktą klausimą
Personažas, nes jis būtų pagrindinis veikėjas aiškinantis kaip, ką ir kodėl žmonės turėtų daryti internete, ko turėtų vengti ir kaip saugiai elgtis.	Spalvingumas	Reikia nurodyti nesaugu elgesį internete.
Pasirodantys paveikslėliai su padariniais, potencialiomis grėsmėmis (kas būna jei nesaugiai elgies internete, potencialias grėsmes)	Dubliavimas - garso dubliavimas animaciją	Animacija gali būti sukurta apie viską, taip pat ir veiksmus, kurie galetu pamokyti apie saugumą, kuris negaletu būti atkurtas reliai.
tekstas, kalbėjimas per animaciją gali perteikti naudingą informciją apie saugų elgesį internete.	balso sinchronizavimas ar net lupu sinchronizavimas	Kompiuterį su atidaryta naršykle.
Ryškios spalvos, kurios pritraukia dėmesį paaugliams ir leidžia lengviau suprasti mintį bei labiau isitelkti į nesauguma internete ir kad bet kas gali apsimesti geru žmogumi ir apgauti, pavogti privačia informacija.		Saugi nuoroda, nes apsaugotu nu visokiu sukčiu.
Vienas iš animacijos elementų - įgarsinimas, kuri leidžia lengviau suprasti turinį.		animacija padeda sutelkti jaunimo dėmesį
Elementas; Personazas, kuris parodo emocija (pvz., baime susidurus su virusu), taip padedamas vaikui geriau isiminti pavojū.		mokinys isijungia gerai zinoma VPN programa
Animacijos veido išraiškos, labiau suprantama kaip jaučiasi po tokios situacijos, kokios emocijos.		Vaizdavimas pasekmių

1.3. klausimas (2)

Teisingi atsakymai	Klaidingi atsakymai
Atsakymai, kuriuose teisingai įvardijami animacijos techniniai / etiniai / turinio aspektai	Neįvardyta ypatybė / atsakymas nekonkretus / ne į pateiktą klausimą
Ar atitinka failo tipas.	Kad nemokintų vaikų kaip kurti virusus ir apgauti kitus žmones.
Reikia patikrinti ar nėra pateiktų asmeninių duomenų.	Realistiškumas
Failo formatas ir dydis.	Svetainės apkrova.
Garso takelio kokybę.	Judesiai
ar animacija ryškiai juda	Ar animacija kartojasi.
kad veiktų su skirtingu tipu įrenginiais.	animacijos struktūra
Ar spalvos išlieka pastovios.	Ar šypsosi
Patikrinti, ar ji nepažeidžia jokių autorių teisių, ar tinkamas plėtinys.	Jos veiksnumą.
Ar ji yra lengvai suprantama, nes svetainė buvo kuriama jaunesniesiems.	Ar nėra rašybos klaidų ir ar pati animacija yra kokybiška / ar nėra klaidų /gramatines klaidos
Reikia patikrinti ar animacija yra lengvai suprantama jaunesniems interneto naudotojams ir ,kad būtų paruošta tokiu būdu, kad jaustųsi patikima.	Animacijos ilgį.
animacijoje būtina patikrinti klaidas tiek iš animacijos pusės (ar joje nėra klaidų susijusių su personažų ar fono piešimu), tiek iš faktinės pusės (ar pateikta informacija atitinka realybę).	ar ji veikia taip, kaip nori ir tikiesi.

2. klausimas (2)

Teisingi atsakymai	Klaidingi atsakymai
Atsakymai, artimi vertinimo instrukcijai; tekstai, iliustracijos; interaktyvūs elementai (pvz., tekstai)	Neteisingai įvardyti daugialypės terpės abu elementai / teisingai įvardytas tik vienas elementas / kalbant apie tekstą, nėra akcentuojama, kad jis yra interaktyvus / Atsakymai ne į pateiktą klausimą
Vaizdo įrašas, garsas.	Šriftai, animacijos
animacijos vaizdo irasai	Paveikslukai, atrinkta informacija.
Vaizdo įrašai su garsu bei grafiniai paveikslukai, kurie neperkrauna informacijos, bet yra lengvai suprantami.	Piešiniai, fontas, šifras
Vaizdo įrašai , interaktyvus elementai pvz testai, žaidimai.	Paveiksleliai ir uzduotys
1. Vaizdo įrašai 2. Interaktyvūs elementai.	duomenų apsauga, tik naudoti šablonai.
Paryškintas tekstas, išsamūs vaizdai	Saugi nuoroda ir mažai reklamu.
Turinį atitinkantis vaizdai, ir idomiajai suformuluotas tekstas.	Spalva ir forma
QR kodai ir ryškūs, aukštos kokybės paveikslėliai.	Duomenų vizualizavimas Dėstomos medžiagos iliustravimas
Tekstas, iliustracijos	Viršelis ir turinys.
	Failo dydis, formatas.
	Statiniai elementai ir dinaminiai elementai
	photoshop. adobe

3. klausimas

3. 15-metis Tomas žaidimų svetainėje pasidalijo asmeninio telefono numeriu, namų adresu ir mokyklos pavadinimu. Po metų jis nutarė nesinaudoti šios svetainės paslaugomis.

Kokia Bendrojo duomenų apsaugos reglamento (GDAR) teise Tomas turėtų pasinaudoti?

(1 taškas)

Galimas atsakymas:

- teisė į asmens duomenų ištrynimą;
- teisė būti pamirštam (*angl. right to be forgotten*).

3. klausimas (2)

Teisingi atsakymai	Klaidingi atsakymai
Atsakymai, artimi vertinimo instrukcijai; teisė atšaukti sutikimą tvarkyti savo duomenis	Jei atsakymas netikslus, prieštarauja BDAR ar yra ne į pateiktą klausimą
Teise nenaudojamų asmeninių duomenų pašalinimui. Ty. paprašyti svetainės valdytojų ištrinti jo duomenis.	Skaitmeniniu parašu.
Duomenų salinimo teise	Tomas turėtų prašyti BDAR, kad Tomo asmeninius duomenis išimtu iš sistemos.
Teisė į duomenų sunaikinimą.	Asmens autorinėmis teisėmis.
Tomas turi teisę atsiimti savo sutikimą duoti ir saugoti jo duomenis.	Kad svetainė nebeturi teisės naudotis naudotojo duomenimis, jei jis svetainėje nebesilanko
Teisė atšaukti naudojimą savo asmeninėmis duomenėmis	Duomenų apsaugos teisė
	Duomenų nutekimo apsaugos teise
	Kad savo duota informacija gali bet kada nutraukti dalinimasi
	Duomenų nutraukimo teise.

4. klausimas

4. Įvardykite **du** šifravimo raktų tipus, naudojamus asimetriškai šifruojant duomenis.

(1 taškas)

Galimas atsakymas:

- viešasis ir privatusis

Pastaba

*1 taškas skiriamas už įvardintus **abiejų** šifravimo raktų tipus.*

4. klausimas (2)

Teisingi atsakymai	Klaidingi atsakymai
Teisingai įvardyti abu raktai: viešasis (angl. public key) ir privatusis (angl. private key)	Atsakymas neteisingas arba „ne į temą“
Viešas raktas, privatus raktas.	Ceaser ir Roman / Cezario, morse
Viešasis ir asmeninis raktai.	SHA-256 RSH-256
Teisė į duomenų sunaikinimą.	HTTPS, STML.
Tomas turi teisę atsiimti savo sutikimą duoti ir saugoti jo duomenis.	simetrinis raktas asimetrinis raktas
Teisė atšaukti naudojimą savo asmeninėmis duomenėmis	Raidėmis, simboliais.
	Asimiliacinis ir simetrinis
	0 ir 1 sistema. Morzės kodas
	iššifruojantys, neiššifruojantys.
	https ir www
	Sinchroninis ir nesinchroninis.
	Laidusis; Vortinklis
	Slapukavimas ir kodavimas.
	koduotas šifravimas ir laisvo būdo šrifavimas

5.1 klausimas

5. Įmonės IT sistemą paveikė duomenis šifruojantis virusas (*ransomware*). Buvo užblokuoti visi klientų asmens duomenys, sustojo elektroninės parduotuvės veikla. Kibernetiniai nusikaltėliai reikalauja išpirkos už duomenų atblokavimą.

Remdamiesi aprašyta situacija ir Lietuvos Respublikos kibernetinio saugumo įstatymo nuostatomis, įvardykite:

5.1. vieną instituciją Lietuvoje, kuriai pranešama apie tokį kibernetinį incidentą;

(1 taškas)

Galimi atsakymai:

- Valstybinė duomenų apsaugos inspekcija (VDAI) – nes incidentas susijęs su asmens duomenų pažeidimu.
- Nacionalinis kibernetinio saugumo centras.
- Lietuvos policija.

Pastaba

*1 taškas skiriamas už **vienos** įstaigos įvardijimą.*

Jei kandidatas pateikė daugiau nei vieną instituciją, vertinama tik pirma pateikta.

5.1. klausimas (2)

Teisingi atsakymai	Klaidingi atsakymai
Atsakymai, artimi vertinimo instrukcijai	Dalykiškai neteisingi arba ne į pateiktą klausimą
Lietuvos Respublikos kibernetinio saugumo institucija.	https://nordvpn.lt
epolicijai	BDAR
VDAI	Lietuvos Respublikos Kibernetinės Apsaugos Tarnyba
Nacionalinio Kibernetinio Saugumo Centras	Lietuvos Respublikos kibernetinis saugumuas.
Lietuvos duomenų apsaugos tarnybai	Kibernetinės saugos atstovai, darbuotojai
Policija	Valstybės saugumo departamentas (VSD)
Kibernetinio saugumo institucija	Duomenų departamentas.
Viešųjų duomenų apsaugos institucija (VDAI)	Lietuvos kibernetiniui saugumui
Iš pradžių skambinčiau į 112, o jie duos kontaktus kur reikia kreiptis dėl šios problemos.	Lietuvos kibernetinio saugumo ministerija.
	STT

5.2. klausimas

5.2. vieną veiksmą, kurį turi atlikti įmonė po incidento;

(1 taškas)

Galimi atsakymai:

- Nedelsiant pranešti paveiktiems klientams apie duomenų saugumo pažeidimą, jeigu klientų duomenys yra išsaugoti atsarginėse duomenų kopijose.
- Užregistruoti incidentą.
- Atkurti duomenis iš atsarginių kopijų.

Kiti panašūs atsakymai, kuriuose minima atsakomybė ir žalingų padarinių sumažinimas, viruso pakliuvimo į sistemą analizė, pažeidžiamų komponentų paieška, programinės įrangos naujinimai, paveiktų sistemų identifikavimas, darbuotojų mokymai.

*1 taškas skiriamas už **vieno** veiksmo įvardijimą.*

Jei kandidatas pateikė daugiau nei vieną veiksmą, vertinamas tik pirmas pateiktas.

5.2. klausimas (2)

Teisingi atsakymai	Klaidingi atsakymai
Atsakymai, artimi vertinimo instrukcijai; imtis techninių priemonių panašioms incidentams užkirsti kelią ateityje	Atsakymas dalykiškai neteisingas arba ne į pateiktą klausimą
Nedelsiant informuoti apie duomenų nutekėjimą Lietuvos Respublikos kibernetinio saugumo įstaigai.	Viską ištrinti
Turi pranešti savo klientams apie duomenų praradimą.	Derėtis su nusikaltėliais dėl duomenų atšifravimo.
Pranešti ir toliau vadovautis saugumo protokolais.	Visi klientų asmeniniai duomenys turi būti iš karto pakeisti ir apie tai pranešta jiems.
Atkurti visus duomenis iš atsarginių kopijų.	Atlikti tyrimus, susijusius su nusikaltimu.
Patikrinti saugos sistemas ar nėra kitų silpnų vietų pro kur gali pakliūti virusai.	Nes šią problemą įmanoma išspręsti kitais būdais.
Sustiprinti savo sistemas nuo kibernetinių išpuolių.	
Veiksmas: Sistemų izoliavimas nuo tinklo arba duomenų atkurimas iš atsarginių kopijų	
Uždėti naują "firewall", kuris geriau apsaugotų nuo tokių virusų.	

5.3. klausimas

5.3. vieną priežastį, kodėl įmonė neturėtų vykdyti kibernetinių nusikaltėlių reikalavimų.

(1 taškas)

Galimi atsakymai:

- Išpirkos mokėjimas gali būti laikomas nusikaltimo finansavimu ir gali sukelti baudžiamąją atsakomybę pagal Baudžiamąjį kodeksą.
- Nėra garantijų, kad nusikaltėliai iš tiesų atblokuos duomenis arba neatliks papildomos žalos (pvz., duomenų išviešinimas).
- Nusikaltėliai gali nesilaikyti susitarimo.
- Išpirkos mokėjimas gali trukdyti teisėsaugos tyrimui ir apsunkinti nusikaltėlių identifikavimą.
- Išpirkos mokėjimas skatina kibernetinių nusikaltėlių veiklą ir padidina riziką, kad bus atakuoti kiti subjektai.

Kiti panašūs atsakymai, kuriuose minima išpirkos mokėjimas kaip nusikalstama veikla, nepasitikėjimas nusikaltėliais, trukdymas tyrimui, papildomų rizikų sukėlimas.

*1 taškas skiriamas už **vienos** priežasties įvardijimą.*

Jei kandidatas pateikė daugiau nei vieną priežastį, vertinama tik pirma pateikta.

5.3. klausimas (2)

Teisingi atsakymai	Klaidingi atsakymai
Atsakymai, artimi vertinimo instrukcijai	Jei atsakymas ne pagal vertinimo instrukciją arba ne į pateiktą klausimą
Nes nėra užtikrinta, kad įvykdžius nusikaltėlių reikalavimus, bus atblokuoti duomenys.	Jie privalo vykdyti kibernetinių nusikaltėlių reikalavimą, nes čia juos užvaldė virusas, o ne kitus.
jeigu įmonė vykdys nusikaltėlių nurodymus, įmonė praras daug reputacijos, praras visus duomenis bei nusikaltėliai gali tuo naudotis ir gali reikalauti vis daugiau ir daugiau išpirkos	Nes tai galima išvengti, galima patiems atblokuoti ir pašalinti virusą ir apsisaugoti, kad tai neįvyktų vėl.
Tuomet nusikaltėliai gali įmonę apgauti ir paprašyti dar didesnės išpirkos.	Duomenys nėra pavogti, o tik užšifruoti, taigi nusikaltėliai negali jais pasinaudoti.
Nes jie vistiek gali paviešinti viską, kai gaus ko norėjo.	nes klientu duomenys yra svarbesni nei ju imone
Kad nusikaltėliai negautų to ko nori.	Nes šią problemą įmanomą išspręsti kitais būdais.



Programų pavyzdžiai

1 pavyzdys, C++	3
2 pavyzdys, C++	7
3 pavyzdys, C++	11
4 pavyzdys, C++	14
5 pavyzdys, Python	17
6 pavyzdys, Python	22

1 pavyzdys, C++

```
//Vieta programos vardui
#include <bits/stdc++.h>
using namespace std;

const char CD[] = "Duomenys.txt";
const char CR[] = "Rezultatai.txt";
struct Mokinys{
    string vardpav;
    string id;
    string klase;
};
struct Apklausas{
    string id;
    int k;
    string D[20];
};
struct Dalykas{
    string pav;
    int k;
};
//-----
void Skaityti(Mokinys M[], int&n, Apklausas A[], int&a);
int KiekKlaseje(Mokinys M[], int n, string kuri);
int Konvertuoti(char s);
bool Tikrinti(string id);
int Netinkamu(Mokinys M[], int n);
bool ArTeisingasDalykas (Apklausas a,string dalykas);
bool ArTeisingaiPasirinko (Apklausas a);
string VardasPavarde(Mokinys M[], int n, string id);
int Yra(Dalykas D[], int d, string koks);
void Formuoti(Apklausas A[], int a, Dalykas D[], int&d);
bool Rikiuoti(Dalykas a, Dalykas b);
//-----
int main(){
    Mokinys M[600];
    Apklausas A[600];
    Dalykas D[20];
    int n, a, d;
    Skaityti(M,n,A,a);
    ofstream fr(CR);
    fr<<KiekKlaseje(M,n,"12a")<<" "<<KiekKlaseje(M,n,"12b")<<"
"<<KiekKlaseje(M,n,"12c")<<endl;
    fr<<Netinkamu(M,n)<<endl;
    for(int i=0; i<n; i++){
        if(!Tikrinti(M[i].id)){
            fr<<M[i].vardpav<<" "<<M[i].id<<"
"<<Konvertuoti(M[i].id[9])+Konvertuoti(M[i].id[8])+Konvertuoti(M[i].id[7])+
Konvertuoti(M[i].id[6])<<endl;
        }
    }
    for(int i=0; i<n; i++){
        for(int j=0; j<a; j++){
            if(M[i].id==A[j].id &&!ArTeisingaiPasirinko(A[j]))
                fr<<M[i].vardpav<<endl;
        }
    }
    Formuoti(A,a,D,d);
    sort(D,D+d,Rikiuoti);
}
```

```

        for(int i=0; i<d; i++){
            fr<<D[i].pav<<" "<<D[i].k<<endl;
        }
        fr.close();
        return 0;
    }
}
//-----
void Skaityti(Mokinys M[], int&n, Apklausa A[], int&a){
    ifstream fd(CD);
    fd>>n>>ws;
    for(int i=0; i<n; i++){
        string eilute;
        getline(fd,eilute);
        stringstream ss(eilute);
        getline(ss,M[i].vardpav,',' );
        getline(ss,M[i].id,',' );
        getline(ss,M[i].klase,',' );
        fd>>ws;
    }
    a=0;
    string eilute;
    while(getline(fd,eilute)){
        stringstream ss(eilute);
        getline(ss,A[a].id,',' );
        int k=0;
        while(getline(ss,A[a].D[k],',' )){
            k++;
        }
        A[a].k=k;
        a++;
    }
    fd.close();
}
//-----
int KiekKlaseje(Mokinys M[], int n, string kuri){
    int k=0;
    for(int i=0; i<n; i++){
        if(M[i].klase==kuri) k++;
    }
    return k;
}
//-----
int Konvertuoti(char s){
    int k;
    if(s=='0') k=0;
    if(s=='1') k=1;
    if(s=='2') k=2;
    if(s=='3') k=3;
    if(s=='4') k=4;
    if(s=='5') k=5;
    if(s=='6') k=6;
    if(s=='7') k=7;
    if(s=='8') k=8;
    if(s=='9') k=9;
    return k;
}
bool Tikrinti(string id){
    return(Konvertuoti(id[9])+Konvertuoti(id[8])+Konvertuoti(id[7])+Konvertuoti
(id[6])==15);
}
}

```

```

//-----
int Netinkamu(Mokinys M[], int n){
    int k=0;
    for(int i=0; i<n; i++){
        if(!Tikrinti(M[i].id)) k++;
    }
    return k;
}
//-----
bool ArTeisingasDalykas(Apklausa a,string dalykas){
    bool t=false;
    for(int i=0; i<a.k;i++){
        if(a.D[i]==dalykas){
            t=true;
            break;
        }
    }
    return t;
}
//-----
bool ArTeisingaiPasirinko(Apklausa a){
    return(a.k>=3 && a.k<=7 && ArTeisingasDalykas(a,"MAT") &&
ArTeisingasDalykas(a,"BL"));
}
//-----
string VardasPavarde(Mokinys M[], int n, string id){
    string vardpav="";
    for(int i=0; i<n; i++){
        if(M[i].id==id){
            vardpav=M[i].vardpav;
            break;
        }
    }
    return vardpav;
}
//-----
int Yra(Dalykas D[], int d, string koks){
    int k=-1;
    for(int i=0; i<d; i++){
        if(D[i].pav==koks){
            k=i;
            break;
        }
    }
    return k;
}
//-----
void Formuoti(Apklausa A[], int a, Dalykas D[], int&d){
    d=0;
    for(int i=0; i<a; i++){
        for(int j=0; j<A[i].k; j++){
            int k=Yra(D,d,A[i].D[j]);
            if(k==-1) {D[d].pav=A[i].D[j];
D[d].k=1;
d++;
}
            else{
                D[k].k++;
            }
        }
    }
}
}

```

```
//-----  
bool Rikiuoti(Dalykas a, Dalykas b){  
    return (a.k>b.k || (a.k==b.k && a.pav<b.pav));  
}
```

2 pavyzdys, C++

```
#include <iostream>
#include <fstream>
#include <string>
#include <sstream>
#include <vector>

using namespace std;

struct Mokinys{
    string vardas;
    string vartotojoVardas;
    int skaiciukuSuma = 0;        //vartotojo varde esanciu skaitmenu suma
    string klase;
    int pasirinko = 0;          //kiek egzaminu vartotojas pasirinko
    bool apklaustas = false;    //ar vartotojas uzpilde apklausa
    int BL = 0;                 //ar vartotojas pasirinko BL egzamina
    int MAT = 0;                //ar vartotojas pasirinko MAT egzamina
};

int ArPenkiolika(string vartotojoVardas, int& nePenkiolika);

int main(){
    ifstream fd("Duomenys.txt");
    ofstream fr("Rezultatai.txt");

    //struktura mokiniu duomenims saugoti
    Mokinys M[500];

    //mokiniu skaicius, klasems priklausanciu mokiniu skaicius, vienas is
    vartotojo varde esanciu skaitmenu
    int n, skA = 0, skB = 0, skC = 0, nePenkiolika = 0;
    //mokinio klase, nes kitaip nesugebu padaryti...
    char klase[5];

    fd >> n >> ws;

    for(int i = 0; i < n; i++){
        getline(fd, M[i].vardas, ';');
        getline(fd, M[i].vartotojoVardas, ';');

        M[i].skaiciukuSuma = ArPenkiolika(M[i].vartotojoVardas,
        nePenkiolika);

        fd >> ws;
        fd.get(klase, 4);

        M[i].klase = klase;

        //pasizymim, kuriai klasei moksleivis priklauso
        if(M[i].klase == "12a")
            skA++;
        else if(M[i].klase == "12b")
            skB++;
        else if(M[i].klase == "12c")
            skC++;

        fd >> ws;
    }
```

```

fr << skA << " " << skB << " " << skC << endl;
fr << nePenkiolika << endl;

for(int i = 0; i < n; i++){
    if(M[i].skaiciukuSuma != 15){
        fr << M[i].vardas << " " << M[i].vartotojoVardas << " " <<
M[i].skaiciukuSuma << endl;
    }
}

//vektoriai egzaminu rusims ir jas pasirinkusiems vartotojams zymeti.
vector<string> egzaminai;
vector<int> pasirinko;

//vartotojo vardas palyginimui, likusi eilutes dalis, egzamino rusis
string vartotojoVardas, eilute, egzaminas;

//nuskaitome apklausos duomenis tol, kol galime
while(getline(fd, vartotojoVardas, ';')){
    fd >> ws;

    for(int i = 0; i < n; i++){
        //susirandame vartotoja, kuris uzpilde apklausa
        if(M[i].vartotojoVardas == vartotojoVardas){
            //pasizymim, kad sis vartotojas buvo apklaustas
            M[i].apklaustas = true;

            //nuskaitome vartotojo pasirinkimus, pasiverciam juos
investies srautu
            getline(fd, eilute);
            fd >> ws;

            stringstream srautas(eilute);

            //kol galim, nuskaitom atskirus pasirinkimus
            while(getline(srautas, egzaminas, ';')){

                //skaiciuojam, kiek egzaminu vartotojas pasirinko
                M[i].pasirinko++;

                //pasizymim, ar pasirinko BL ir MAT egzaminus
                if(egzaminas == "BL")
                    M[i].BL++;
                else if(egzaminas == "MAT")
                    M[i].MAT++;

                //susitrumpinam nuskaitytus duomenis, nes kitaip nieko
doro nebus
                if(egzaminas == "BL")
                    egzaminas.resize(2);
                else
                    egzaminas.resize(3);

                //pirma egzamina i vektoriu itraukiam nedvejodami
                if(egzaminai.size() == 0){
                    egzaminai.push_back(egzaminas);
                    pasirinko.push_back(1);
                }else{
                    //kitu atveju, tikrinam, ar egzaminas yra
vektoriuje
                    bool arVektoriuje = false;

```

```

        for(int j = 0; j < egzaminai.size(); j++){
            //jeigu randam egzamina vektoriuje, tiesiog
padidynam pasirinkusiuju skaiciu
            if(egzaminas == egzaminai.at(j)){
                arVektoriuje = true;
                pasirinko.at(j)++;
                break;
            }
        }

        //jeigu nerandam, itraukiam egzamina i vektoriu
        if(!arVektoriuje){
            egzaminai.push_back(egzaminas);
            pasirinko.push_back(1);
        }
    }

}

}

}

}

//bubble sort vektoriams
for(int i = 0; i < pasirinko.size()-1; i++){

    for(int j = 0; j < pasirinko.size()-1-i; j++){

        if(pasirinko.at(j) < pasirinko.at(j+1)){

            swap(pasirinko.at(j), pasirinko.at(j+1));
            swap(egzaminai.at(j), egzaminai.at(j+1));

        }else if(pasirinko.at(j) == pasirinko.at(j+1)){

            if(egzaminai.at(j) > egzaminai.at(j+1)){

                swap(pasirinko.at(j), pasirinko.at(j+1));
                swap(egzaminai.at(j), egzaminai.at(j+1));

            }

        }

    }

}

for(int i = 0; i < n; i++){

    //jei vartotojas apklaustas, tikriname, ar jis nepazeidzia
apklausos taisykliu
    if(M[i].apklaustas){
        if(M[i].pasirinko > 7 || M[i].pasirinko < 3 || M[i].BL == 0 ||
M[i].MAT == 0){
            fr << M[i].vardas << endl;
        }
    }

}

}

```

```

for(int i = 0; i < egzaminai.size(); i++){
    fr << egzaminai.at(i) << " " << pasirinko.at(i) << endl;
}

fd.close();
fr.close();
return 0;
}

int ArPenkiolika(string vartotojoVardas, int& nePenkiolika){
    int skaiciukuSuma = 0;

    for(int j = 6; j < 10; j++){
        skaiciukuSuma += vartotojoVardas[j]-'0';
    }

    //jeigu suma != 15, pasizymim tai
    if(skaiciukuSuma != 15)
        nePenkiolika++;

    return skaiciukuSuma;
}

```

3 pavyzdys, C++

```
#include <bits/stdc++.h>
using namespace std;

ofstream out("Rezultatai.txt");

struct mokinys{
    string vardas;
    string pavarde;
    string kodas;
    string klase;
    vector<string> pasirinkimai;
    int id_suma;
    bool ar_rinkosi = false;
};

struct daznumas{
    string pavadinimas;
    int kiek;
};

//funkciju deklaracijos
void Nuskaitymas(vector<mokinys>& mokiniai, int& n);
void Klasese_mokiniu_skaicius(int n, vector<mokinys> mokiniai);
void id_patikrinimas(int n, vector<mokinys> mokiniai);
void Neteisingai_pasirinkusieji(int n, vector<mokinys> mokiniai);
void Egzaminu_daznis(int n, vector<mokinys> mokiniai);

int main(){
    vector<mokinys> mokiniai;
    int n;
    Nuskaitymas(mokiniai, n);
    Klasese_mokiniu_skaicius(n, mokiniai);
    id_patikrinimas(n, mokiniai);
    Neteisingai_pasirinkusieji(n, mokiniai);
    Egzaminu_daznis(n, mokiniai);
    return 0;
}

//visiskas duomenu nuskaitymas is pradiniu duomenu failo
void Nuskaitymas(vector<mokinys>& mokiniai, int& n){
    ifstream in("Duomenys.txt");
    in >> n >> ws;
    for(int i = 0; i < n; i++){
        mokinys temp;
        getline(in, temp.vardas, ' ');
        getline(in, temp.pavarde, ';');
        getline(in, temp.kodas, ';');
        getline(in, temp.klase, '\n');
        temp.klase = temp.klase.substr(0, 3);
        mokiniai.push_back(temp);
    }
    //out << mokiniai[9].klase;
    string temp_kodas;
    while(getline(in, temp_kodas, ';')){
        for(int i = 0; i < n; i++){
            if(temp_kodas == mokiniai[i].kodas){
                mokiniai[i].ar_rinkosi = true;
                string dalykai;
                string vienas_dalykas;
                getline(in, dalykai, '\n');
            }
        }
    }
}
```

```

        stringstream ss(dalykai);
        while(getline(ss, vienas_dalykas, ';')){
            if(vienas_dalykas[0] == 'B' && vienas_dalykas[1] ==
'L'){
                vienas_dalykas = vienas_dalykas.substr(0, 2);
            } else{
                vienas_dalykas = vienas_dalykas.substr(0, 3);
            }
            mokiniai[i].pasirinkimai.push_back(vienas_dalykas);
        }
        /*if(i + 1 != n){

mokiniai[i].pasirinkimai[mokiniai[i].pasirinkimai.size()-1] =
mokiniai[i].pasirinkimai[mokiniai[i].pasirinkimai.size()-1].substr(0,
mokiniai[i].pasirinkimai.size());

}//////////////////////*/
        }
    }
}
//funkcija apskaiciuojanti mokiniu kieki kiekvienoje klaseje
void Klasese_mokiniu_skaicius(int n, vector<mokinys> mokiniai){
    int a_kl = 0;
    int b_kl = 0;
    int c_kl = 0;
    for(int i = 0; i < n; i++){
        if(mokiniai[i].klase == "12a"){
            a_kl++;
        }
        if(mokiniai[i].klase == "12b"){
            b_kl++;
        }
        if(mokiniai[i].klase == "12c"){
            c_kl++;
        }
    }
    out << a_kl << ' ' << b_kl << ' ' << c_kl << '\n';
}
//funkcija tikrinanti ar mokinio id skaitmenu suma lygi 15
void id_patikrinimas(int n, vector<mokinys> mokiniai){
    int kiek_blogu = 0;
    for(int i = 0; i < n; i++){
        int skaiciai = stoi(mokiniai[i].kodas.substr(6, 4));
        mokiniai[i].id_suma = skaiciai / 1000 + skaiciai / 100 % 10 +
skaiciai / 10 % 10 + skaiciai % 10;
        if(mokiniai[i].id_suma != 15){
            kiek_blogu++;
        }
    }
    out << kiek_blogu << '\n';
    for(int i = 0; i < n; i++){
        if(mokiniai[i].id_suma != 15){
            out << mokiniai[i].vardas << ' ' << mokiniai[i].pavarde << ' '
<< mokiniai[i].kodas << ' ' << mokiniai[i].id_suma << '\n';
        }
    }
}
}

```

```

//funkcija randanti neteisingai egzaminus pasirinkusius mokinius
void Neteisingai_pasirinkusieji(int n, vector<mokinys> mokiniai){
    for(int i = 0; i < n; i++){
        if(mokiniai[i].ar_rinkosi){
            bool ar_tinkamai = false;
            if(mokiniai[i].pasirinkimai.size() >= 3 &&
mokiniai[i].pasirinkimai.size() <= 7){
                bool turi_mat = false;
                bool turi_bl = false;
                for(int j = 0; j < mokiniai[i].pasirinkimai.size(); j++){
                    if(mokiniai[i].pasirinkimai[j] == "MAT"){
                        turi_mat = true;
                    }
                    if(mokiniai[i].pasirinkimai[j] == "BL"){
                        turi_bl = true;
                    }
                }
                if(turi_mat && turi_bl){
                    ar_tinkamai = true;
                }
            }
            if(!ar_tinkamai){
                out << mokiniai[i].vardas << ' ' << mokiniai[i].pavarde <<
'\n';
            }
        }
    }
}

//funkcija apskaiciuojanti egzaminu daznumus
void Egzaminu_daznis(int n, vector<mokinys> mokiniai){
    vector<daznumas> daznumai;
    for(int i = 0; i < n; i++){
        for(int j = 0; j < mokiniai[i].pasirinkimai.size(); j++){
            bool ar_jau_yra = false;
            for(int k = 0; k < daznumai.size(); k++){
                if(daznumai[k].pavadinimas == mokiniai[i].pasirinkimai[j]){
                    ar_jau_yra = true;
                    daznumai[k].kiek++;
                }
            }
            if(!ar_jau_yra){
                daznumas temp;
                temp.pavadinimas = mokiniai[i].pasirinkimai[j];
                temp.kiek = 1;
                daznumai.push_back(temp);
            }
        }
    }
    //rusiavimas
    for(int i = 0; i < daznumai.size() - 1; i++){
        for(int j = i + 1; j < daznumai.size(); j++){
            if(daznumai[i].kiek < daznumai[j].kiek || daznumai[i].kiek ==
daznumai[j].kiek && daznumai[i].pavadinimas > daznumai[j].pavadinimas){
                swap(daznumai[i], daznumai[j]);
            }
        }
    }
    //spausdinimas
    for(int i = 0; i < daznumai.size(); i++){
        out << daznumai[i].pavadinimas << ' ' << daznumai[i].kiek << '\n';
    }
}

```

4 pavyzdys, C++

```
#include <bits/stdc++.h>

using namespace std;

struct Mokinys{
    string pilnasVardas {};
    string id {};
    char klase {};
};

struct Pasirinkimas{
    string id {};
    vector<string> egzaminai {};
};

int sumosTikrintojas(string id){
    int suma {};
    suma = char( id[6] ) - 48 + char( id[7] ) - 48 + char( id[8] ) - 48 +
char( id[9] ) - 48;
    return suma;
}

int main(){
    ifstream fin("Duomenys.txt");
    ofstream fout("Rezultatai.txt");

    int n {};
    fin >> n;
    vector<Mokinys> mokiniai {};
    map<char, int> kiekisKlaseje {};
    string eilute {};
    getline(fin, eilute);
    for(size_t i{}; i < n; ++i){
        Mokinys temp {};
        getline(fin, eilute);
        stringstream ss {eilute};
        getline(ss, temp.pilnasVardas, ';');
        getline(ss, temp.id, ';');
        string klase {};
        getline(ss, klase);
        kiekisKlaseje[klase[2]] += 1;
        temp.klase = klase[2];
        mokiniai.push_back(temp);
    }

    fout << kiekisKlaseje['a'] << ' ' << kiekisKlaseje['b'] << ' ' <<
kiekisKlaseje['c'] << '\n';

    int neatitikeKiekis {};
    vector<Mokinys> mokNeatitike {};
    for(auto i : mokiniai){
        if(sumosTikrintojas(i.id) != 15){
            neatitikeKiekis += 1;
            mokNeatitike.push_back(i);
        }
    }
}
```

```

fout << neatitikeKiekis << '\n';

for(auto i : mokNeatitike){
    fout << i.pilnasVardas << ' ' << i.id << ' ' <<
sumosTikrintojas(i.id) << '\n';
}

vector<Pasirinkimas> pasirinkimai;
while(getline(fin, eilute)){
    Pasirinkimas temp {};
    stringstream ss {eilute};
    getline(ss, temp.id, ';');
    string egzaminas {};
    while(getline(ss, egzaminas, ';')){
        string grazusEgzaminas {};
        int dydis = egzaminas.size();
        if(dydis == 4){
            dydis = 3;
        }
        for(size_t i {}; i < dydis; ++i){
            grazusEgzaminas += egzaminas[i];
        }
        temp.egzaminai.push_back(grazusEgzaminas);
    }
    pasirinkimai.push_back(temp);
}

vector<string> sarasasIsvedimui {};
for(auto i : pasirinkimai){
    if(i.egzaminai.size() > 7 || i.egzaminai.size() < 3){
        for(auto j : mokiniai){
            if(j.id == i.id){
                sarasasIsvedimui.push_back(j.pilnasVardas);
            }
        }
        continue;
    }

    bool matematika = false, kalba = false;
    for(auto j : i.egzaminai){
        if(j == "MAT"){
            matematika = true;
        }
        if(j == "BL"){
            kalba = true;
        }
    }
    if(matematika == false || kalba == false){
        for(auto j : mokiniai){
            if(j.id == i.id){
                sarasasIsvedimui.push_back(j.pilnasVardas);
            }
        }
        continue;
    }
}
}

```

```

for(auto i : mokiniai){
    for(auto j : sarasasIsvedimui){
        if(i.pilnasVardas == j){
            fout << j << '\n';
        }
    }
}

map<string, int> dalykai {};
for(auto i : pasirinkimai){
    for(auto j : i.egzaminai){
        dalykai[j] += 1;
    }
}
vector<pair<string, int>> rikiuotas {};
for(auto i : dalykai){
    rikiuotas.push_back(i);
}
sort(rikiuotas.begin(), rikiuotas.end(), [](const auto a, const auto
b){
    if(a.second == b.second){
        return a.first < b.first;
    }
    return a.second > b.second;
});
for(auto i : rikiuotas){
    fout << i.first << ' ' << i.second << '\n';
}

return 0;
}

```

5 pavyzdys, Python

```
# konstanta, nurodanti, kokia turi buti mokinio id skaitmenu suma
MOK_ID_SKAITMENU_SUMA = 15

# klasiu pavadinimu konstantos
KLASE_12A = "12a"
KLASE_12B = "12b"
KLASE_12C = "12c"

# egzaminu pasirinkimo konstantos
MIN_EGZAMINU_SK = 3
MAX_EGZAMINU_SK = 7
PRIVALOMI_EGZAMINAI = ["MAT", "BL"]

# klase vieno mokinio duomenims kaupiti, kaupiant vardas, pavarde, mokinio ID
ir klase
class Mokinys:
    def __init__(self, vardas, pavarde, mok_id, klase):
        self.vardas = vardas
        self.pavarde = pavarde
        self.mok_id = mok_id
        self.klase = klase

    # gauti mokinio id skaitmenu suma
    def gauti_id_skaitmenu_suma(self):
        skaitmenu_eilute = self.mok_id[6:]
        # surenkame individualius skaitmenis ir paverciame is teksto tipo
        (str) i sveikaji skaiciu (int)
        skaitmenys = [skaitmenu_eilute[0], skaitmenu_eilute[1],
skaitmenu_eilute[2], skaitmenu_eilute[3]]
        skaitmenys = [int(skaitmuo) for skaitmuo in skaitmenys]
        # gauname suma
        skaitmenu_suma = sum(skaitmenys)
        return skaitmenu_suma

# klase, skirta laikyti vieno mokinio egzaminu pasirinkimo duomenis
# Patebekime, kad cia laikome ne mok id, bet nurooda i visa mokinio
objekta
# Taigi sukuriant klases objekta mes pagal id gauname mokini ir ji ikeliame
kuriant sios klases objekta
class MokinioPasirinkimai:
    def __init__(self, mokinys, egzaminu_kodai):
        self.mokinys = mokinys
        self.egzaminu_kodai = egzaminu_kodai

# funkcija nuskaityti visiems duomeniu failo duomenis ir juos grazinti
# Grazina mokiniu sarasa ir sarasa mokiniu egzaminu pasirinkimu
def duomeniu_skaitymas(failo_pav):
    with open(failo_pav, "r") as failas:
        # gauname mokiniu skaiciu n
        n = int(failas.readline().strip())

        # nuskaityti mokiniu sarasa ir irasome mokiniu duomenis i sarasa
        mokiniai = []
        for _ in range(n):
            duomenys_txt = failas.readline().strip().split(";")
            vardas, pavarde = duomenys_txt[0].split()
```

```

        mok_id = duomenys_txt[1]
        klase = duomenys_txt[2]

        # sukuriame mokinio klases objekta ir irasome i sarasa
        mokinys = Mokinys(vardas, pavarde, mok_id, klase)
        mokiniai.append(mokinys)

sarasa # nuskaitome apklausos duomenis ir irasome i egzaminu pasirinkimu
        # skaitome visas likusias failo eilutes
        apklausos_duomenys = []
        pasirinkusiu_egzaminus_mokiniu_kodai = []
        for eilute in failas.readlines():
            eilute = eilute.strip().split(";")
            mok_id = eilute[0]
            egzaminu_kodai = eilute[1:]

            # gauname mokini pagal jo mokinio id
            mokinys = [mok for mok in mokiniai if mok.mok_id == mok_id][0]

            # sukuriame mokinio pasirinkimu klase ir irasome i sarasa
            mokinio_pasirinkimai = MokinioPasirinkimai(mokinys,
egzaminu_kodai)
            apklausos_duomenys.append(mokinio_pasirinkimai)
            pasirinkusiu_egzaminus_mokiniu_kodai.append(mokinys.mok_id)

        # surikiuojame apklausos duomenis pagal pradini mokiniu sarasa
        surikiuoti_apklauso_duomenys = []
        for mokinys in mokiniai:
            if mokinys.mok_id in pasirinkusiu_egzaminus_mokiniu_kodai:
                mokinio_pasirinkimai = [
                    pasirinkimai
                    for pasirinkimai in apklausos_duomenys
                    if pasirinkimai.mokinys.mok_id == mokinys.mok_id
                ][0]
                surikiuoti_apklauso_duomenys.append(mokinio_pasirinkimai)

        return mokiniai, surikiuoti_apklauso_duomenys

# suskaiciuoja, kiek kiekvienos klases mokiniu yra duotame sarase,
# grazina kaip zodyna (dict), kur raktas - klase, o reiksme - mokiniu
kiekis toje klaseje
def suskaiciuoti_mokinius_klasese(mokiniai):
    mokiniu_kiekiai = {KLASE_12A: 0, KLASE_12B: 0, KLASE_12C: 0}

    # einame per kiekviena mokini, gauname jo klase ir atnaujiname tos
klases mokiniu kieki
    for mokinys in mokiniai:
        klase = mokinys.klase
        mokiniu_kiekiai[klase] += 1

    return mokiniu_kiekiai

# funkcija patikrinti, ar mokinio ID skaitmenu suma lygi 15, ir grazinanti
bool reiksme, ar mokinio id teisingas
# si funkcija naudoja Mokinio klases metoda, suskaiciuojanti mokinio id
skaitmenu suma
# (Mokinys.gauti_id_skaitmenu_suma)
def patikrinti_mok_id_suma(mokinys):
    return mokinys.gauti_id_skaitmenu_suma() == MOK_ID_SKAITMENU_SUMA

```

```

# funkcija grazinanti mokiniu kieki, kuriu id neteisinga skaitmenu suma, ir
tu mokiniu sarasa
def gauti_neteisingu_mok_id_duomenis(mokiniai):
    neteisingu_mok_id_sk = 0
    mokiniai_su_neteisingu_mok_id = []

    for mokinys in mokiniai:
        if not patikrinti_mok_id_suma(mokinys):
            neteisingu_mok_id_sk += 1
            mokiniai_su_neteisingu_mok_id.append(mokinys)

    return neteisingu_mok_id_sk, mokiniai_su_neteisingu_mok_id

# funkcija, grazinanti sarasa mokiniu, kurie neteisingai pasirinko
egzaminus
def gauti_mokinius_pasirinkusius_neteisingai(mokiniu_pasirinkimai):
    mokiniai_neteisingai_pasirinke = []

    for mok_pasirinkimas in mokiniu_pasirinkimai:
        mokinys = mok_pasirinkimas.mokinys
        egzaminai = mok_pasirinkimas.egzaminu_kodai
        pasirinkimai_teisingi = True

        # tikriname, ar pasirinktu egzaminu skaicius neteisingas
        egzaminu_kiekis = len(egzaminai)
        if egzaminu_kiekis < MIN_EGZAMINU_SK or egzaminu_kiekis >
MAX_EGZAMINU_SK:
            pasirinkimai_teisingi = False

        # tikriname, ar pasirinkti visi privalomi egzaminai
        for privalomas_egzaminas in PRIVALOMI_EGZAMINAI:
            if privalomas_egzaminas not in egzaminai:
                pasirinkimai_teisingi = False
                break

        if not pasirinkimai_teisingi:
            mokiniai_neteisingai_pasirinke.append(mokinys)

    return mokiniai_neteisingai_pasirinke

# funkcija, zodyna (dict) kuriame raktai egzaminu kodai, reiksmes -
egzaminu pasirinku sk
# Verta pastebeti, jog salygoje ir pavyzdziuose nepateikta, jog renkame
skaicius pasirinktu egzaminu
# tik is tu mokiniu, kurie juos pasirinko teisingai
# Taigi pasirinkimu skaiciai renkami bendrai visu mokiniu, iskaiciuojant ir
neteisingus pasirinkimus
def gauti_egzaminu_pasirinkimu_sk(mokiniu_pasirinkimai):
    egzaminu_pasirinkimu_sk = {}

    # einame per kiekvieno mokinio pasirinkimu sarasa, pridedame
atitinkamus skaicius
    for pasirinkimas in mokiniu_pasirinkimai:
        mokinys = pasirinkimas.mokinys
        egzaminai = pasirinkimas.egzaminu_kodai

        for egzamino_kodas in egzaminai:
            if egzamino_kodas in egzaminu_pasirinkimu_sk:

```

```

        egzaminu_pasirinkimu_sk[egzamino_kodas] += 1
    else:
        egzaminu_pasirinkimu_sk[egzamino_kodas] = 1

    return egzaminu_pasirinkimu_sk

# funkcija, surikiuojanti egzaminu pasirinkimu skaicius ir grazinanti kaip
list[tuple],
# kur yra egzamino kodo ir pasirinkimu sk poros
def rikiuoti_egzaminu_pasirinkimo_sk(egzaminu_pasirinkimu_sk):
    surikiuoti_egzaminu_pasirinkimu_sk = []

    for egzamino_kodas, pasirinkimu_sk in egzaminu_pasirinkimu_sk.items():
        surikiuoti_egzaminu_pasirinkimu_sk.append((egzamino_kodas,
            pasirinkimu_sk))

    # rikiuojame
    surikiuoti_egzaminu_pasirinkimu_sk.sort(key=lambda x: (-x[1], x[0]))

    return surikiuoti_egzaminu_pasirinkimu_sk

# funkcija, irasanti visus reikiamus i rezultatu faila
def duomeni_isvedimas(
    failo_pav,
    mokiniu_kiekiai,
    neteisingu_mok_id_sk,
    mokiniai_su_neteisingu_mok_id,
    neteisingai_pasirinke_mokiniai,
    surikiuoti_egzaminu_pasirinkimu_sk
):
    with open(failo_pav, "w") as failas:
        # irasome mokiniu kiekius klasese
        failas.write(f"{mokiniu_kiekiai[KLASE_12A]}
{mokiniu_kiekiai[KLASE_12B]} {mokiniu_kiekiai[KLASE_12C]}\n")

        # irasome, kiek mokiniu ivesta su neteisingu mokinio id ir isvedame
tu mokiniu duomenis
        failas.write(f"{neteisingu_mok_id_sk}\n")
        for mokinys in mokiniai_su_neteisingu_mok_id:
            mokinio_id_skaitmenu_suma = mokinys.gauti_id_skaitmenu_suma()
            failas.write(f"{mokinys vardas} {mokinys.pavarde}
{mokinys.mok_id} {mokinio_id_skaitmenu_suma}\n")

        # irasome mokinius, neteisingai pasirinkusius egzaminus
        for mokinys in neteisingai_pasirinke_mokiniai:
            failas.write(f"{mokinys vardas} {mokinys.pavarde}\n")

        # irasome pasirinktu egzaminu skaicius:
        for pasirinkimo_pora in surikiuoti_egzaminu_pasirinkimu_sk:
            failas.write(f"{pasirinkimo_pora[0]} {pasirinkimo_pora[1]}\n")

def main():
    # nuskaityme visus duomenis
    mokiniai, mokiniu_pasirinkimai = duomeni_skaitymas("Duomenys.txt")
    # analizuojame duomenis ir gauname reikiamu formatu irasymui i
rezultatu faila

    mokiniu_kiekiai = suskaiciuoti_mokinius_klasese(mokiniai)
    neteisingu_mok_id_sk, mokiniai_su_neteisingu_mok_id =
gauti_neteisingu_mok_id_duomenis(mokiniai)

```

```
    neteisingai_pasirinke_mokiniai =
gauti_mokinius_pasirinkusius_neteisingai(mokiniu_pasirinkimai)
    egzaminu_pasirinkimu_sk =
gauti_egzaminu_pasirinkimu_sk(mokiniu_pasirinkimai)
    surikiuoti_egzaminu_pasirinkimu_sk =
rikiuoti_egzaminu_pasirinkimo_sk(egzaminu_pasirinkimu_sk)

# irasome visus duomenis
duomenu_isvedimas(
    "Rezultatai.txt",
    mokiniu_kiekiai,
    neteisingu_mok_id_sk,
    mokiniai_su_neteisingu_mok_id,
    neteisingai_pasirinke_mokiniai,
    surikiuoti_egzaminu_pasirinkimu_sk
)

if __name__ == "__main__":
    main()
```

6 pavyzdys, Python

```
failas = open("Duomenys.txt", "r", encoding="utf-8")

n = int(failas.readline())

klase_a, klase_b, klase_c = 0, 0, 0
duomenys = []

for i in range(n):
    vardas, mokinio_id, klase = failas.readline().strip().split(";")
    duomenys.append([vardas, mokinio_id, klase])
    if klase == "12a":
        klase_a += 1
    elif klase == "12b":
        klase_b += 1
    else:
        klase_c += 1

failas2 = open("Rezultatai.txt", "w", encoding="utf-8")
failas2.write(f"{klase_a} {klase_b} {klase_c}\n")

kas = []

def id_patikrinimas(vardas, mokinio_id):
    skaicius = mokinio_id[6:]
    pirmas_skaitymuo = skaicius[0]
    antras_skaitymuo = skaicius[1]
    trecias_skaitymuo = skaicius[2]
    ketvirtas_skaitymuo = skaicius[3]

    suma = int(pirmas_skaitymuo) + int(antras_skaitymuo) +
int(trecias_skaitymuo) + int(ketvirtas_skaitymuo)

    if suma != 15:
        return [vardas, mokinio_id, suma]
    else:
        return False

for i in range(len(duomenys)):
    if id_patikrinimas(duomenys[i][0], duomenys[i][1]) == False:
        pass
    else:
        kas.append(id_patikrinimas(duomenys[i][0], duomenys[i][1]))

failas2.write(f"{len(kas)}\n")

for i in range(len(kas)):
    failas2.write(f"{kas[i][0]} {kas[i][1]} {kas[i][2]}\n")

apklausa = []
kas_blogai = []
ka_pasirinko = {}

for i in range(n):
    try:
        t_duom = failas.readline().strip().split(";")
        if t_duom == ['']:
            break
        else:
```

```

        apklausa.append(t_duom)
except:
    break

for duom in apklausa:
    mokinio_id = duom[0]
    kiek = 0
    mat = False
    bl = False
    for i in range(1, len(duom)):
        try:
            ka_pasirinko[duom[i]] += 1
        except KeyError:
            ka_pasirinko[duom[i]] = 1

        if duom[i] == "BL":
            bl = True
            kiek += 1
        elif duom[i] == "MAT":
            mat = True
            kiek += 1
        else:
            kiek += 1

    if kiek > 7 or kiek < 3:
        kas_blogai.append(mokinio_id)
    elif mat == False:
        kas_blogai.append(mokinio_id)
    elif bl == False:
        kas_blogai.append(mokinio_id)

for duom in duomenys:
    mokinio_id = duom[1]
    for i in range(len(kas_blogai)):
        if mokinio_id == kas_blogai[i]:
            failas2.write(f"{duom[0]}\n")

ka_pasirinko_t = []

for egz in ka_pasirinko:
    ka_pasirinko_t.append([egz, ka_pasirinko[egz]])

for i in range(len(ka_pasirinko_t)):
    for j in range(i+1, len(ka_pasirinko_t)):
        if ka_pasirinko_t[i][1] < ka_pasirinko_t[j][1]:
            ka_pasirinko_t[i], ka_pasirinko_t[j] = ka_pasirinko_t[j],
ka_pasirinko_t[i]
            elif ka_pasirinko_t[i][1] == ka_pasirinko_t[j][1]:
                if ka_pasirinko_t[i][0] > ka_pasirinko_t[j][0]:
                    ka_pasirinko_t[i], ka_pasirinko_t[j] = ka_pasirinko_t[j],
ka_pasirinko_t[i]

for egz in ka_pasirinko_t:
    failas2.write(f"{egz[0]} {egz[1]}\n")

```