

TARTU ÜLIKOOL
LOODUS- JA TÄPPISTEADUSTE VALDKOND
Arvutiteaduse instituut
Informaatika õppekava

Anastassia Fomin
Informaatikaviktoriini Kobras tulemuste
visualiseerimisvahend ja interaktiivne ülesannete kogumik
Magistritöö (30 EAP)

Juhendaja: Lidia Feklistova, MSc

Tartu 2021

Informaatikaviktoriini Kobras tulemuste visualiseerimisvahend ja interaktiivne ülesannete kogumik

Lühikokkuvõte:

Magistritöös käsitletakse 2020. aasta veebruaris toimunud informaatikaviktoriini Kobras ülesandeid. Osalejate tulemused on esitatud autori poolt loodud veebipõhise interaktiivse visualiseerimisvahendiga. Töös tutvustatakse loodud veebipõhist ülesannete kogumikku, mis lubab interaktiivselt tutvuda ülesandega, vaadata ülesande vastust, selle seletust ja seost informaatikaga. Autoril valmis 10 uut ülesannet tulevaste viktoriinide jaoks.

Võtmesõnad:

Informaatikaviktoriin, Kobras, tulemuste visualiseerimisvahend, ülesannete kogu

CERCS: P175 Informaatika, süsteemiteooria, S270 Pedagoogika ja didaktika

Informatics challenge Bebras results visualization tool and interactive collection of tasks

Abstract:

The master's thesis considers tasks that were used in the informatics challenge Bebras in February 2020. Participants results are presented with an interactive web-based results visualization tool created by the author. In the current thesis, a web-based collection of tasks is provided. It is possible in an interactive way to view a task, its answer, and explanation, also pay attention to task connection to informatics. The author created 10 new tasks for future challenges as well.

Keywords:

Informatics challenge, Bebras, results visualization tool, collection of tasks

CERCS: P175 Informatics, systems theory, S270 Pedagogy and didactics

Sisukord

Sissejuhatus.....	5
1. Informaatikaviktoriin Kobras	6
1.1 Ülevaade viktoriini arengust	6
1.2 Kobras Eestis.....	7
1.2.1 Osalejate arv Eestis aastatel 2014-2020.....	7
1.3 Kobras mujal	10
2. Tulemuste visualiseerimisvahend.....	11
2.1 Vajalikud algandmed.....	11
2.2 Visualiseerimisvahendi vajadus	11
2.3 Kasutatud tehnoloogiad.....	11
2.4 Tehniline kirjeldus.....	12
2.5 Tulemuste visualiseerimisvahendi kasutamine	14
2.5.1 Vaade: ülevaade tulemustest.....	15
2.5.2 Vaade: tulemused ülesannete tüüpide kaupa	17
2.5.3 Vaade: punktide tabel	17
2.5.4 Vaade: kirjeldav statistika.....	18
2.6 2020 a. II vooru tulemuste analüüs	18
2.6.1 Vanuserühm Benjamin.....	18
2.6.2 Vanuserühm Juunior	20
2.6.3 Vanuserühm Senior.....	21
2.7 Visualiseerimisvahendi kohandamine järgmisteks viktoriinideks (arendajale).....	23
2.7.1 Moodle keskkonnast saadud .xlsx faili muutused.....	23
2.7.2 Muutused .html failides	23

2.7.3	Muutused .js failis.....	24
2.8	Võimalik tuleviku arendus	25
3.	Interaktiivne ülesannete kogumik.....	26
3.1	Ülesannete kogumiku vajadus.....	26
3.2	Benjaminid (6.–8. klass).....	26
3.3	Juuniorid (9.–10. klass)	57
3.4	Seniorid (11.–12. klass).....	79
3.5	Veebipõhise ülesannete kogumiku kasutamine.....	98
3.6	Kogumiku kohandamine järgmisteks viktoriinideks (arendajale)	101
3.7	Võimalik tuleviku arendus	101
4.	Uued ülesanded.....	102
4.1	Järgitud head tava.....	102
4.2	Ülesanded eesti keeles.....	102
	Kokkuvõte.....	122
	Viidatud kirjandus.....	123
	Lisa 1. Uute ülesannete tekstide inglise ja venekeelsed versioonid.....	124
	Litsents	171

Sissejuhatus

Tänapäeva ühiskonnas kasutatakse aina rohkem erinevaid infotehnoloogilisi vahendeid ja teenu-seid. Nende arendamiseks on tarvis omandada programmeerimisoskust ning nende võimaluste as-jalik ja tark kasutamine eeldab haritud ja mõistlikku kasutajat. Enne kui noored jõuavad program-meerimiseni, on vaja pöörata nende tähelepanu arvutile kui töövahendile. Selleks on kasutatud erinevaid lähenemisviise. Alates 2006. aastast Eestis peetakse kooliõpilastele mõeldud informaa-tikaviktoriini Kobras, mis on osa rahvusvahelisest perest. Viktoriinis on kaks vooru ja teemade ring on lai: alates arvuti riist- ja tarkvarast, lõpetades informatsiooni mõistmisega. Iga aasta osaleb üle 3000 õpilase ning kogutakse suur hulk andmeid, mis vajavad analüüsimist. Antud töö käigus sai valmis tulemuste visualiseerimisvahend, mis annab parema ülevaate nii ülesannete kui ka tee-made lõikes. Saadud infost on huvitatud viktoriini korraldajad ja õpetajad, kes õpilasi viktoriiniks ette valmistavad.

Informaatikaviktoriini Kobras kodulehel on olemas ülesannete eesti- ja venekeelne tõlge ning õi-ged vastused. Veebruaris 2020 toimunud viktoriini kohta on lisatud eestikeelsed vastuste seletused ja info, kuidas iga ülesanne on seotud just informaatikaga. Antud töö raames analoogne ülesannete kogumik on loodud venekeelsetele õpilastele, kes iga aasta aktiivselt osalevad viktoriinis. Kokku on 45 ülesannet kolmest vanuserühmast: benjaminid, juuniorid ja seniorid. Loodud kogumik lu-bab interaktiivselt tutvuda ülesannetega, vaadata ülesannete vastust ning nende seletust ja seost informaatikaga. Valmiv kogumik on kasulik abimaterjal nii viktoriiniks ettevalmistamisel, kui ka arvutiõpetuse tundide mitmekesiseks läbiviimiseks.

Informaatikaviktoriin toimub rohkem kui 50 riigis ja iga riik panustab ülesannete panka. Käeso-leva töö autor on koostanud 10 uut ülesannet, mida korraldajad saavad lisada rahvusvahelisesse küsimuste panka.

Töö esimeses osas antakse ülevaade informaatikaviktoriinist Kobras.

Teises osas tutvustatakse tulemuste visualiseerimisvahendit.

Kolmandas osas on toodud venekeelne ülesannete kogumik.

Neljandas osas vaadeldakse uued ülesanded.

1. Informaatikaviktoriin Kobras

Käesolevas peatükis antakse ülevaade informaatikaviktoriini arengust alatest sünnist tänapäevani. Samuti tuuakse välja viktoriini korraldamine Eestis. Peatüki lõppu on lisatud ülevaade teiste riikide kodulehtedel oleva informaatikaviktoriini korraldamisega seotud infost.

1.1 Ülevaade viktoriini arengust

Informaatikaviktoriini loomise idee kuulub Leedu professorile Valentina Dagiene'le. Viktoriini idee sündis 2003. aasta reisirõõmu Soome ja professori algatus hakkas kandma nimetust "bebras", mis leedu keelest tõlgituna tähendabki "kobras". Viktoriini eesmärgiks on tekitada kooli õpilaste seas huvi informaatika vastu ja arendada arvutusliku mõtlemist [1].

Esimene viktoriin viidi läbi Leedus 2004. aasta oktoobris, millest võttis osa 3470 õpilast. Viktoriin kestis 45 minutit ja sisaldas 18 küsimust, kus iga küsimus oli erineva raskusastmega ja andis vastavalt raskusele kas vähem või rohkem punkte. Pakid viktoriini küsimustega olid kättesaadavad allalaadimiseks üks nädal enne viktoriini, kuid avada sai need vaid teatud päeval. Failid õpilaste vastustega edastati ühisesse serverisse, kus neid analüüsiti [2].

V. Dagiene soovis, et tema idee jõuaks rahvusvahelisele tasemele ja see unistus täitus juba 2006. aastal, kui viktoriiniga liitusid Eesti, Saksamaa, Holland ja Poola. Aasta hiljem korraldasid oma esimese viktoriini ka Austria, Läti ja Slovakkia ja 2008. aastal Ukraina ja Tšehhi. Igal aastal liitub informaatikaviktoriiniga aina rohkem riike ning aasta 2020 aprilli seisuga oli nende arv 54 [3].

Tänapäeval enamikes riikides toimub ainult üks voor. On olemas üksikud riigid, kus kontrollitud oludes toimub ka teine voor. Lahendatavate ülesannete hulk võib ulatuda 24-ni. Peategelasteks on traditsiooniliselt koprad. Sõltuvalt vanusest, võib õpilane kuuluda ühte kuuest vanuserühmast [4]:

- eel-algajad – 1. ja 2. klassid (õpilased vanuses 5-8);
- algajad – 3. ja 4. klassid (õpilased vanuses 8-10);
- benjaminid – 5. ja 6. klassid (õpilased vanuses 10-12);
- kadetid – 7. ja 8. klassid (õpilased vanuses 12-14);
- juuniorid – 9. ja 10. klassid (õpilased vanuses 14-16);
- seniorid – 11. ja 12. klassid (õpilased vanuses 16-19).

Antud vanuserühmad on soovituslikud ja riigiti võivad varieeruda. Samuti riigid ei ole kohustatud läbi viima viktoriini kõigis kuues vanuserühmas, see on rahvusliku korraldajate otsustada.

1.2 Kobras Eestis

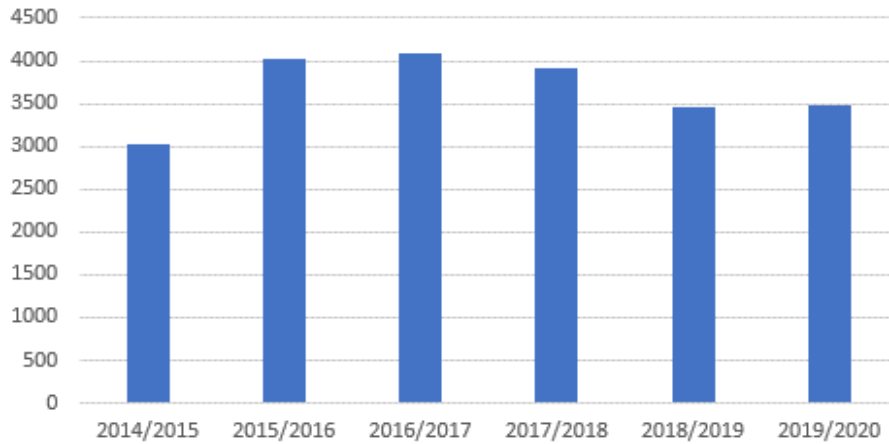
Eestis peetava informaatikaviktooriini Kobras formaat ei eelda üldhariduskooli õpilastelt programmeerimisalaseid eelteadmisi. Käsitletatavate teemade hulgas on ajalugu, eetika, turve, õigus, riistvara, algoritmid, diskreetne matemaatika, info mõistmine, loogika, geomeetria, tekstitöötlus, tabelitöötlus ja muu tarkvara [5]. Küsimuste eesmärk on arendada loogilist mõtlemist, võimekust leida lahendus probleemile, mis käsitleb informaatika aspekte. Õpilane peab olema suuteline jagama keerulisi ülesandeid osadeks ning leidma probleemis mingi mustri, algoritmi või üldistuse. Kõik see toetab arvutusliku mõtlemise arengut.

Eestis peetakse võistlust kolmes vanuserühmas: 6.-8. klass (benjaminid), 9.-10. klass (juuniorid), 11.-12. klass (seeniorid). Eesti on üks vähestest riikidest, kus toimub kaks vooru. Esimene voor on avatud tavaliselt kaks nädalat ja sisaldab 15 erinevat ülesannet. Lahendamiseks eraldatakse üks koolitund (45 minutit). Iga korrektselt vastatud ülesanne annab maksimaalselt 3 punkti ning vale vastus -1 punkti. Kui õpilane jätab küsimusele vastamata, antakse selle eest 0 punkti. Kohapeal õpetajatest korraldajad peavad tagama, et võistlejad ei kasutaks kõrvalist abi – ei lahendaks seda rühmatööna ega otsiks vastuseid internetist. Enne 2020. aastat esimese vooru ülesanded asusid Miksikesse keskkonnas veebipõhise kontrollitööna. Alates 2020 novembrist viktooriin on kättesaadav Teaduskooli viktooriini keskkonnas.

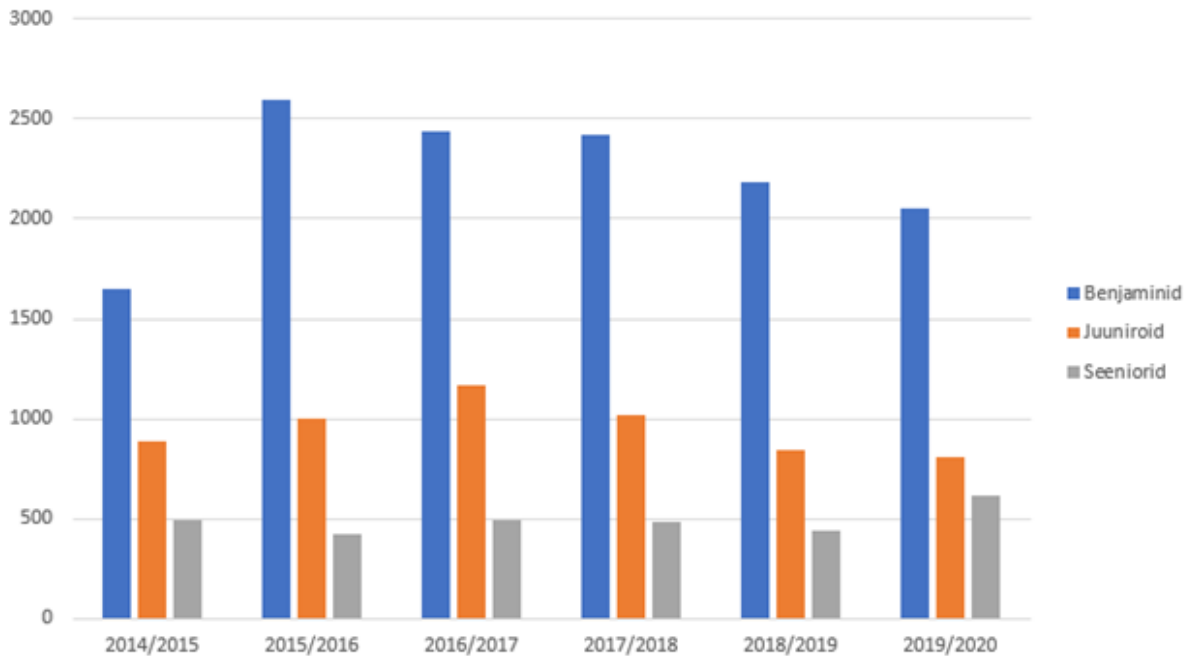
Parimad õpilased osalevad teises voorus, mis peetakse veebruaris, Tartu linnas. Viktooriini eelneval päeval on korraldatud erinevad huvitegevused: IT-ettevõtete külastamised ja informaatikat puudutavad huviloengud. Viktooriini teises voorus õppijad peavad samuti 45 minuti jooksul lahendama 15 ülesannet. Iga õige vastuse eest on võimalik saada 1 punkt, osa punkte on võimalik saada osaliselt õigete vastuste eest ning vale vastus või selle puudumine annab 0 punkti. Seega teises voorus saab õpilane maksimaalselt 15 punkti. Ülesanded on toodud testina õpikeskkonnas Moodle. Auhindade hulgas on digivahendid, silmaringi laiendav kirjandus, loogikamängud ja palju muud.

1.2.1 Osalejate arv Eestis aastatel 2014-2020

Iga aasta osaleb viktooriinis üle 3000 õpilase (vt. joonis 1). Kõige rohkem osalejaid on benjaminide vanuserühmas, kes moodustavad umbes $\frac{2}{3}$ kõikidest osalejatest (vt. joonis 2). Juuniore osaleb rohkem kui seniore.



Joonis 1. I vooru osalejate arv, kokku



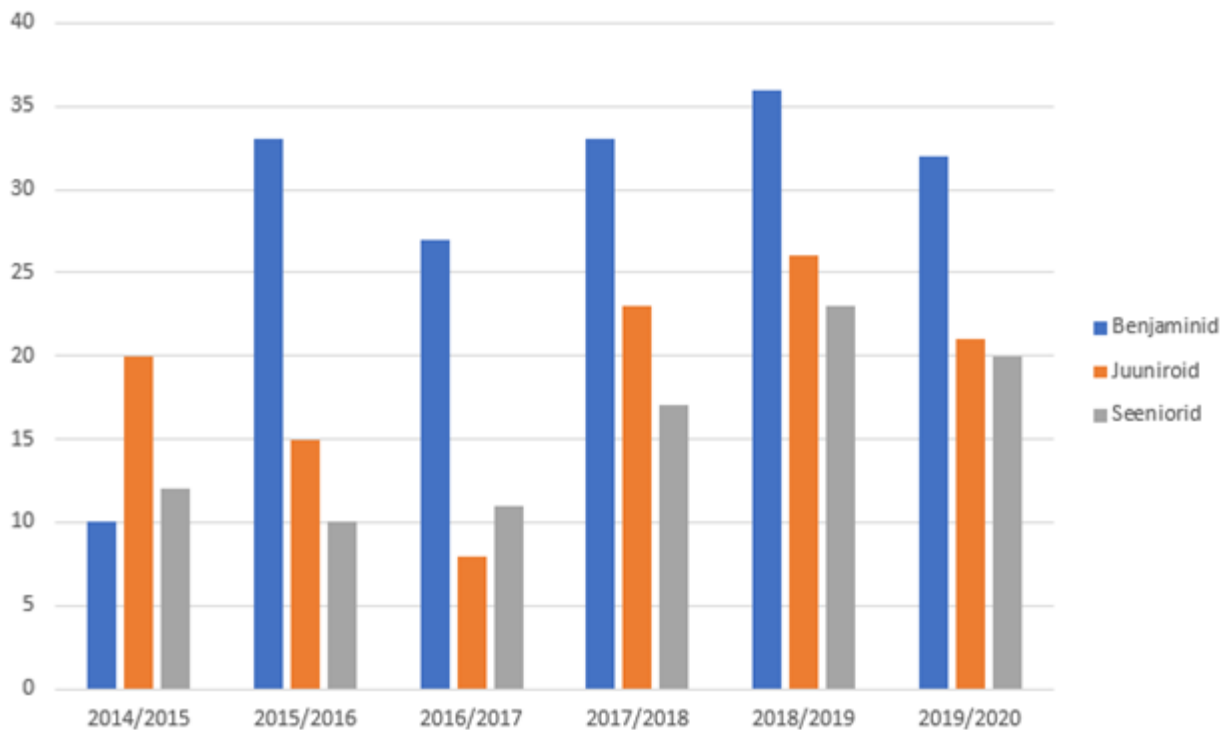
Joonis 2. I vooru osalejate arv, vanuserühmade lõikes

Arvestades Eesti rahvaarvu on viktoriinis osalejate hulk märkimisväärne. Näiteks Eestis osalejate arv 2019/2020 hooajal oli lähedal sellistele riikidele nagu Rootsi ja Jaapan, kus rahvaarv on oluliselt suurem (vt. joonis 3).



Joonis 3. Väljavõte informaatikaviktoriini osalejate arvust, rahvusvaheline koondleht november 2019 – aprill 2020 [3]

Parimate hulka jõudnute õpilaste arv varieerub, kuid Eesti keskmine jääb 63 juurde (vt. joonis 4). Näiteks 2019/2020 hooajal oli II vooru kutsutud 55 benjamini, 26 juunioidi ja 28 seeniori, kuid tuleb tõdeda, et mitte kõik õpilased võtavad kutse vastu. Veebruaris 2020 tuli kohale vaid 32 benjamini, 21 juunioidi ja 20 seeniori ehk kokku 73 õpilast.



Joonis 4. II vooru osalejate arv, vanuserühmade lõikes

1.3 Kobras mujal

Käesoleva töö autor tutvus viktoriini kodulehtedega, mis on koostatud täisliikme hulka kuuluvate riikide poolt [6]. Näiteks Egiptuse, Prantsuse ja Jaapani kodulehel on võimalik lahendada interaktiivseid ülesandeid, Iraani kodulehel on nähtav piltide galerii viimase voozu pääsenud õpilastega, Põhja-Makedoonia kodulehel on välja toodud statistika, kui palju tüdrukuid ja poisse, vastavalt tasemele, võttis viktoriinist osa. Portugali kodulehel on esitatud osalenud koolide nimekiri. Ukrainas koostatakse iga aasta kohta eraldi dokument, kus on välja toodud ülesanded koos lahendustega ja määratletud mitu protsenti õpilastest said antud ülesandega hakkama [7]. Austria kodulehel on välja toodud statistika selle kohta mitu tüdrukut või poissi võttis viktoriinist osa ja iga aasta kohta koostatakse dokument, kus on küsimused vastustega, nende seletus ja seos informaatikaga [8]. Analoogne dokument on kättesaadav ka Saksamaa veebilehelt, kus lisaks on küsimuste kogumikud ka ilma vastusteta [9]. Belgia kodulehel on leitav statistika osalenud õpilaste arvu kohta, kuid saab ka jälgida, kuidas iga aastaga muutub osalenud õpilaste arv [10]. Kanada kodulehel on arvutatud aritmeetiline keskmine tulemus iga kategooria kohta [11]. Pakistani kodulehel on välja toodud osalejate üldarv [12]. Tšehhi leheküljel on täpsem statistika küsimuste lõikes: mitu küsimust oli vastatud õigesti, valesti ning vastamata. Lisaks on toodud keskmine tulemus ning tüdrukute osakaal võrreldes poistega [13]. Statistikat kuvatakse ka Slovakkia [14] kodulehel.

2. Tulemuste visualiseerimisvahend

Antud peatükk on pühendatud tulemuste visualiseerimisvahendile. Tööprintsiibid on demonstree-ritud veebruaris 2020 toimunud II vooru tulemuste põhjal. Samuti peatükis on välja toodud ana-lüüs, kuidas õpilased said hakkama ülesannetega. Peatüki lõppu on lisatud kirjeldus sellest, kuidas loodud tulemuste visualiseerimisvahendit saab kohandada järgmistele viktoriinidele.

2.1 Vajalikud algandmed

Teise vooru ülesanded asuvad õpikeskkonnas Moodle. Nii õpikeskkonnast, kui ka registreerimis-vormist on võimalik iga osaleja kohta saada järgmist infot:

- sugu;
- vanuse kategooria;
- ajakulu ülesannete lahendamisele;
- ülesannete tulemused;
- punktide koguarv.

2.2 Visualiseerimisvahendi vajadus

Kui võitjate väljaselgitamine on kiiresti leitav kogutud punktide põhjal, siis tulemuse analüüs (näi-teks millised ülesanded õnnestusid paremini; millised ülesanded jäid vastamata; kui hästi tüdrukud said hakkama konkreetse ülesande tüübiga jne) võtab aga tavaliselt palju aega. Analüüsi hõlbusta-miseks sai arendatud veebipõhine tulemuste visualiseerimisvahend. See lubab liikuda interaktiiv-sete graafikute vahel ning valida ja kuvada just huvipakkuvaid tulemusi tabelist.

2.3 Kasutatud tehnoloogiad

Rakenduse töö põhineb pistikmoodulil Chart.js [15]. Samuti on kasutatud JavaScript (näiteks funktsioonide defineerimiseks, interaktiivsete elementide loomiseks veebilehel), HTML (näiteks konteinerite defineerimiseks) ja CSS (stiili määramiseks). Visualiseerimisvahendi arendustöö toi-mus läbi selleks loodud GitHub-i privaatse hoidla – tööfailid on kättesaadavad aadressilt https://github.com/vesta33/kobras_features.git.

Antud töö raames on kasutatud Visual Studio Code redaktorit, kuhu on lisatud Live Server XMLHttpRequest()-i tegemiseks. Live Server'i lisamiseks avatakse *Laiendused* (ingl. *Extensions*)

vajutades Ctrl-Shift-X ja otsingu reale kirjutatakse *Live Server*. Sellele järgneb *Install* nupu vajutamine. XMLHttpRequest() rakendusliides on ette nähtud andmete vahendamiseks veebisaidi ja serveri vahel [16]. Käesolevas töös antud liides on kasutatud andmete kättesaamiseks Excel'i failidest, mis paiknevad lokaalserveris, *datafiles* kaustas. Projekti lisamiseks Visual Studio Code redaktorisse on tarvis ühendada kaust, kus paikneb projekt (*File – Open folder...* abil). Selle käivitamiseks on vajalik *index.html* faili avamine ja paremas alumises nurgas *Go live* nupu vajutamine (vt. joonis 5).



Joonis 5. Live Serveri käivitamise nupp Visual Studio Code redaktoris

Valminud rakendus on testitud brauserites Mozilla Firefox (ver. 80.0.1) ja Google Chrome (ver. 86.0.4240.183). Viimase puhul testimisel ilmnis veateade (vt. joonis 6) – Console: GET http://127.0.0.1:5500/undefined 404 (Not Found). Network:



Joonis 6. Veebirakenduse testimisel Google Chrome's (ver. 86.0.4240.183) ilmnenud viga Põhjuseks on teek *chartjs-plugin-labels.js*, mis lubab lisada nimetused sektordiagrammile. Veaparandamisel teegist sai eemaldatud rida 427 (*image.src = obj.src*);

2.4 Tehniline kirjeldus

Tulemuste visualiseerimisvahend on järgneva kaustade struktuuriga:

- **css** – kaust stiili määramiseks;
- **datafiles** – kaust, kus asuvad Excel'i failid (.xlsx) tulemustega. Antud kaustas paiknevad ka alamkaustad. Igas alamkaustas on pildifailid, mis on vajalikud mallil info kuvamiseks;
- **js** – JavaScript failid;
- **libs** – vajalikud teegid graafikute näitamiseks ning Excel'ist saadud andmete kuvamiseks tabelis;
- **juurkataloog** – HTML failid:
 - *index.html* – kuvab linke, mis viivad vastava vanuserühma tulemuste lehele (vastavalt *benjamins.html*, *juniors.html*, *seniors.html*);

- *examples.html* – mall ülesannetega;
- *package-lock.json* – andmevahetusformaadis automaatselt genereeritav fail, seotud paketi halduriga (ingl. *Node Package Manager, npm*).

Graafikute joonistamiseks on kasutatud järgnevad meetodid (failid *benjamins.js*, *juniors.js*, *seniors.js*):

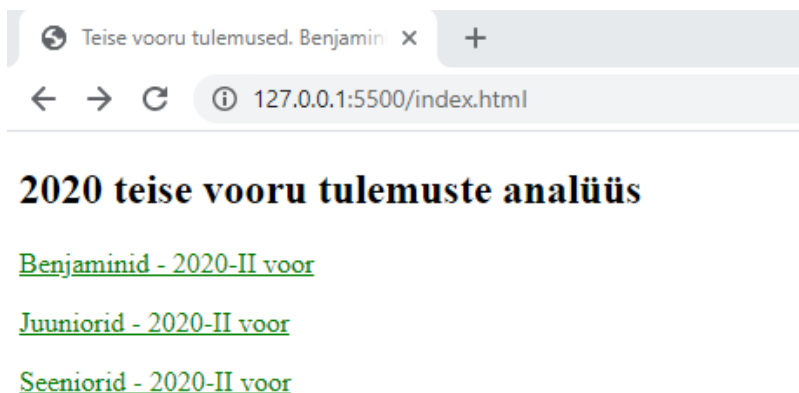
- **listParticipants()** – eesti keelt kõnelevate poiste ja tüdrukute ning vene keelt kõnelevate poiste ja tüdrukute arvu leidmiseks;
- **findGradesTotal(par)** – abimeetod tüdrukute ja poiste keskmise tulemuse leidmiseks;
- **minValueOfBoysArray(array)**, **maxValueOfBoysArray(array)** – poiste minimaalse ja maksimaalse väärtuse leidmiseks;
- **minValueOfGirlsArray(array)**, **maxValueOfGirlsArray(array)** – tüdrukute minimaalse ja maksimaalse väärtuse leidmiseks;
- **findAverage(par)**, **findAverageBoys(par)**, **findAverageGirls(par)** – üldise keskmise tulemuse, keskmise poiste tulemuse ja keskmise tüdrukute tulemuse leidmiseks;
- **countCorrectAnswers(array)**, **countCorrectAnswersBoys(array)**, **countCorrectAnswersGirls(array)** – õigesti vastatud küsimuste arvu leidmiseks sektordiagrammil (üldine, poisid, tüdrukud);
- **countWrongAnswers(array)**, **countWrongAnswersBoys(array)**, **countWrongAnswersGirls(array)** – valesti vastatud küsimuste arvu leidmiseks sektordiagrammil (üldine, poisid, tüdrukud);
- **countNoAnswers(array)**, **countNoAnswersBoys(array)**, **countNoAnswersGirls(array)** – puuduolevate vastuste arvu leidmiseks sektordiagrammil (üldine, poisid, tüdrukud);
- **countPartAnswers(array)**, **countPartAnswersBoys(array)**, **countPartAnswersGirls(array)** – osaliselt õigesti vastatud küsimuste arvu leidmiseks sektordiagrammil (üldine, poisid, tüdrukud);
- **standartDeviationBoys(array, avg)**, **standartDeviationGirls(array, avg)** – standardhälbe leidmiseks poiste ja tüdrukute lõikes;
- **findAverageForBoysGroup(arr1, arr2)**, **findAverageForGirlsGroup(arr1, arr2)** – ülesannete tüüpide lõikes keskmise leidmiseks. Poiste ja tüdrukute jaoks eraldi (näiteks eetika kohta oli kaks küsimust ning järelikult töödeldakse kahte tulemuse massiivi);

- `minValueOfBoysGroup(array1,array2)`, `maxValueOfBoysGroup(array1,array2)`, `minValueOfGirlsGroup(array1,array2)`, `maxValueOfGirlsGroup(array1,array2)` – ülesannete tüüpide lõikes minimaalse ja maksimaalse tulemuse leidmiseks (poisid ja tüdrukud eraldi);
- `standartDeviationBoysGroup(array1, array2, avg)`, `standartDeviationGirlsGroup(array1, array2, avg)` – ülesannete tüüpide lõikes standarthälbe leidmiseks (poisid ja tüdrukud eraldi);

Viimase kolme meetodi kohta on olemas ka variatsioonid: `..2Columns`, `..3Columns`, et eristada, kas ülesannete tüübis oli kaks või kolm küsimust.

2.5 Tulemuste visualiseerimisvahendi kasutamine

Loodud rakendus töötab veebibrauseris. Esialgu kasutaja satub avalehele, kus on välja toodud lingid kujul: *vanuserühm-aasta* (vt. joonis 7).



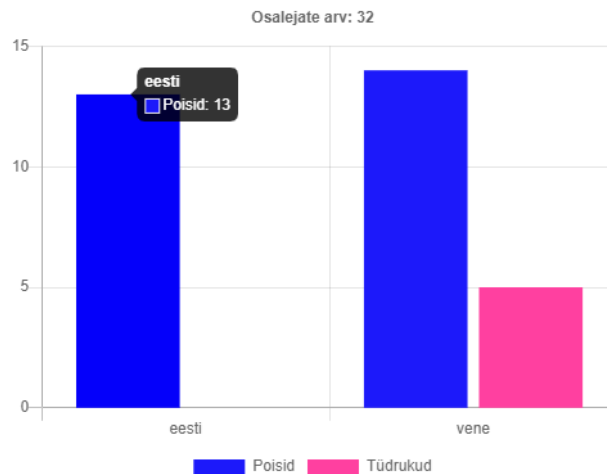
Joonis 7. Tulemuste visualiseerimisvahendi avaleht lokaalserveris

Vanuserühma valimisel kuvatakse kogu sisu, kuid vasakus ülemises nurgas on esitatud erinevad valikud. Nende abil on võimalik valida järgmiste vaadete vahel:

- ülevaade tulemustest;
- tulemused ülesannete tüüpide kaupa;
- punktide tabel;
- kirjeldav statistika.

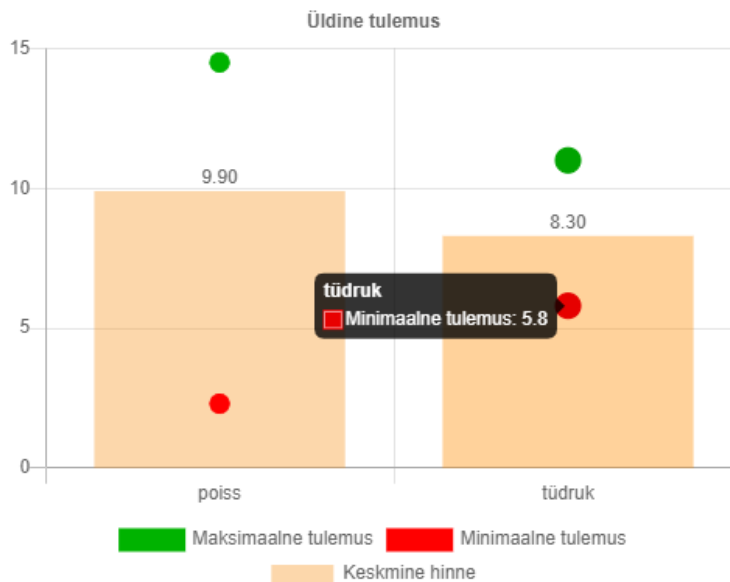
2.5.1 Vaade: ülevaade tulemustest

Vasakpoolne ülemine joonis näitab osalejate arvu antud rühmas. Parema ülevaate saamiseks on joonise pealkirjas välja toodud ka osalejate koguarv. Joonisel kuvatakse mitu eesti keelt kõnelevat poissi ja tüdrukut ning mitu vene keelt kõnelevat poissi ja tüdrukut osales viktoriinis. Graafik on interaktiivne, ehk hiire asetamisel tulba kohale kuvatakse täpsem informatsioon (vt. joonis 8).



Joonis 8. Benjaminide vanuserühma osalejate arv soo lõikes

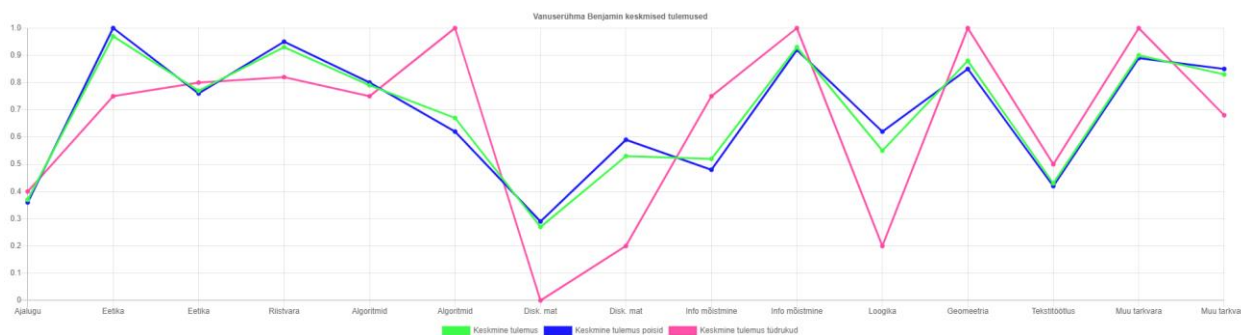
Parempoolne ülemine interaktiivne graafik näitab üldtulemust: keskmine, minimaalne ja maksimaalne väärtus poiste ja tüdrukute lõikes (vt. joonis 9).



Joonis 9. Benjaminide vanuserühma osalejate üldine tulemus soo lõikes

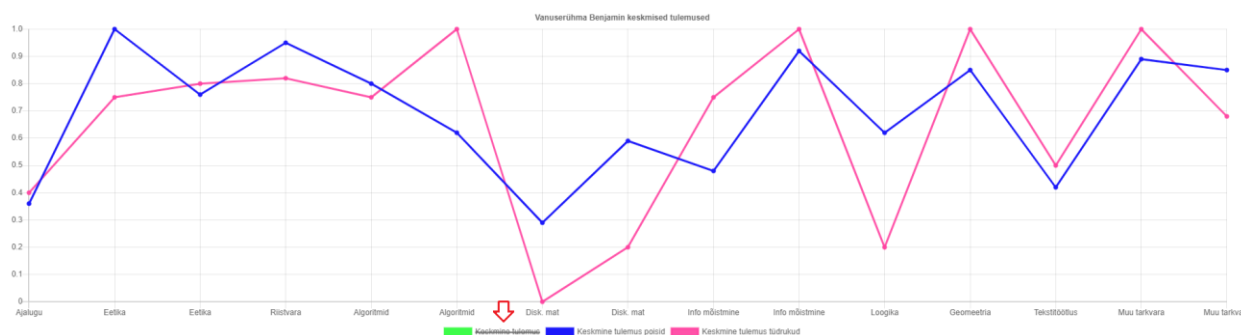
Samal lehel olev alumine interaktiivne graafik näitab keskmist tulemust küsimuste lõikes (vt. joonis 10):

- roheline joon näitab osalejate keskmist tulemust;
- sinine joon näitab poiste keskmist tulemust;
- punane joon näitab tüdrukute keskmist tulemust.



Joonis 10. Benjaminide vanuserühma osalejate keskmine tulemus küsimuste lõikes

Kõikide graafikute puhul saab valida, kas kuvada kõik andmed korruga või osa sellest. Näiteks viimase graafiku puhul saab valida, kas kuvada kolm joont korruga või näiteks ükshaaval/kahekaupa. Joone eemaldamiseks tuleb legendis vajutada valitud pealkirja – joon kaob ära ning pealkiri on maha kriipsutatud (vt. joonis 11). Analoogselt, kui kasutaja soovib joont tagasi graafikule tuua, tuleb uuesti vajutada samale pealkirjale.



Joonis 11. Benjaminide vanuserühma poiste ja tüdrukute keskmised tulemused küsimuste lõikes, osalejate ühine keskmine tulemus on peidetud

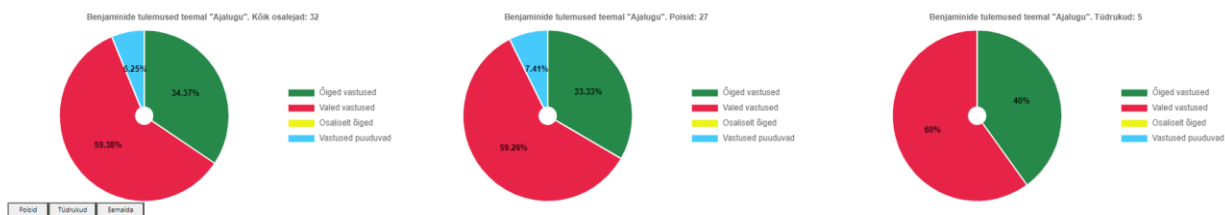
2.5.2 Vaade: tulemused ülesannete tüüpide kaupa

Viktoriinis olevad küsimused on jagatud erinevateks ülesannete tüüpideks. Igas ülesande tüübis võib olla üks või rohkem küsimust. Iga ülesande tüübi kohta näitab tulemuste visualiseerimisvahend tüübi pealkirja koos üldise osalejate arvuga ning interaktiivset üldist sektordiagrammi, kus protsentides on toodud õigete, valede, osaliselt õigete ja puuduvate vastuste osakaal selles ülesande tüübis.

Sektordiagrammi all on kolm nuppu:

- **poisid** – kuvab lisa sektordiagrammi ainult poiste tulemustega;
- **tüdrukud** – kuvab lisa sektordiagrammi ainult tüdrukute tulemustega;
- **eemalda** – eemaldab poiste ja tüdrukute sektordiagrammid ning jätab alles üldise sektordiagrammi.

Näiteks Benjaminide 2020. a. II vooru tulemused teemal “Ajalugu” näitavad, et 59,38% kõikidest osalejatest andsid vale vastuse, õiged vastused oli 34,37% ning vastused puudusid 6,25% õpilastel. Poistest 33,33% andsid õige vastuse, tüdrukutest aga 40% (vt. joonis 12).



Joonis 12. Benjaminide tulemused teemal “Ajalugu” – üldine vasakul, poisid keskel, tüdrukud paremal

2.5.3 Vaade: punktide tabel

Punktide tabelis on esitatud andmed, mis on imporditud Moodle keskkonnas genereeritud Excel'i failist. Tabelis kuvatakse osalejate koguarv, poiste arv ja tüdrukute arv. Tabeli kohal olev tähis „-“ tähistab vastuse puudumist ehk õppija ei vastanud küsimusele.

Vaadet saab kohandada vastavalt vajadusele. Saab uurida ainult poiste andmeid (linnuke on märkeruudu “Poisid” ees), ainult tüdrukute andmeid (linnuke on märkeruudu “Tüdrukud” ees, vt. joonis 13). Vaikimisi kuvatakse kõik tulemused.

Poisid

Tüdrukud

Ajalugu	Eetika, turve, õigus	Eetika, turve, õigus	Riisvara	Algoritmid	Algoritmid	Diak.mat	Diak.mat	Info mõistmine	Info mõistmine	Loogika	Geomeetria	Tekstiloetus	Muu tarkvara	Muu tarkvara
0.0	1.0	1.0	1.0	1.0	-	-	0.0	-	-	0.0	1.0	-	-	0.0
0.0	0.0	0.5	0.5	0.0	1.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0	1.0	0.0
0.0	1.0	1.0	0.8	-	-	-	1.0	1.0	0.0	1.0	0.0	0.0	1.0	0.0
1.0	-	0.5	1.0	1.0	1.0	0.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0	0.5
1.0	1.0	1.0	0.8	0.0	1.0	-	0.0	1.0	1.0	0.0	1.0	1.0	-	0.5

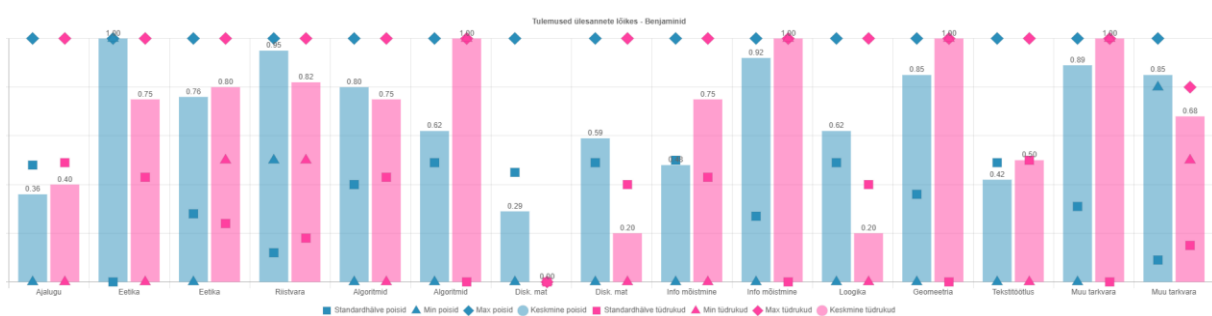
Joonis 13. Punktide tabel, benjaminide vanuserühma tüdrukute tulemused

2.5.4 Vaade: kirjeldav statistika

Kirjeldava statistika all on välja toodud kaks interaktiivset graafikut:

- tulemused ülesannete lõikes;
- tulemused ülesannete tüüpide kaupa.

Vaadet saab kohandada vastavalt vajadusele. Legendi juures saab maha kriipsutada teatud peal- kirjad, mida antud hetkel pole tarvis. Iga küsimuse ja ülesannete tüübi kohta on välja toodud eraldi tulemused poiste (sinine tulp) ja tüdrukute (punane tulp) lõikes. Lisaks igal graafikul on näidatud minimaalne, maksimaalne ja keskmine tulemus ning standardhälve (vt. joonis 14).



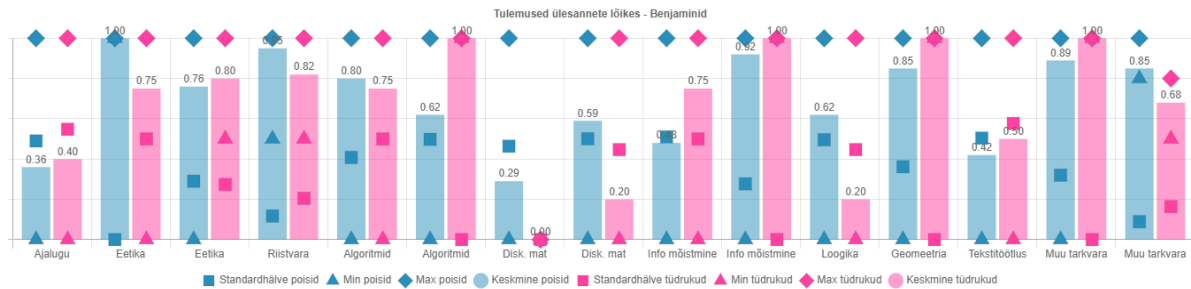
Joonis 14. Benjaminide vanuserühma tulemused ülesannete lõikes

2.6 2020 a. II voo tulemuste analüüs

2020. aasta veebruaris toimunud viktoriini II voo tulemuste analüüsimiseks on kasutatud loodud tulemuste visualiseerimisvahend.

2.6.1 Vanuserühm Benjamin

Benjaminide vanuserühma tulemuste analüüsimiseks on kasutatud vaade *Kirjeldav statistika*, kus on välja toodud tulemused ülesannete lõikes (vt. joonis 15).



Joonis 15. Benjaminide vanuserühma tulemused ülesannete lõikes

Lohistades arvutihiirega huvipakkuvale tulbale või punktile, saab teada keskmise, minimaalse ja maksimaalse väärtuse ning standardhälve. Tulemuste koondvaade on toodud tabelis 1.

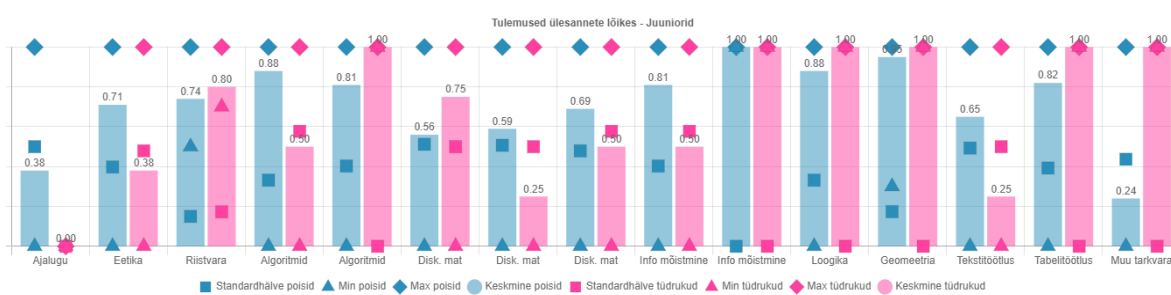
Tabel 1. Vanuserühma Benjamin tulemused ülesannete lõikes

Ülesanne	Poisid			Tüdrukud		
	Keskmine (SD)	MIN	MAX	Keskmine (SD)	MIN	MAX
Ajalugu	0.36 (SD = 0.490)	0	1	0.40 (SD = 0.548)	0	1
Eetika, turve, õigus 1	1.00 (SD = 0.000)	1	1	0.75 (SD = 0.500)	0	1
Eetika, turve, õigus 2	0.76 (SD = 0.290)	0	1	0.80 (SD = 0.274)	0.5	1
Riistvara	0.95 (SD = 0.117)	0.5	1	0.82 (SD = 0.205)	0.5	1
Algoritmid 1	0.80 (SD = 0.408)	0	1	0.75 (SD = 0.500)	0	1
Algoritmid 2	0.62 (SD = 0.498)	0	1	1.00 (SD = 0.000)	1	1
Diskreetne matemaatika 1	0.29 (SD = 0.464)	0	1	0.00 (SD = 0.000)	0	0
Diskreetne matemaatika 2	0.59 (SD = 0.501)	0	1	0.20 (SD = 0.447)	0	1
Info mõistmine 1	0.48 (SD = 0.510)	0	1	0.75 (SD = 0.500)	0	1
Info mõistmine 2	0.92 (SD = 0.277)	0	1	1.00 (SD = 0.000)	1	1
Loogika	0.62 (SD = 0.496)	0	1	0.20 (SD = 0.447)	0	1
Geomeetria	0.85 (SD = 0.362)	0	1	1.00 (SD = 0.000)	1	1
Tekstitöötlus	0.42 (SD = 0.504)	0	1	0.50 (SD = 0.577)	0	1
Muu tarkvara 1	0.89 (SD = 0.320)	0	1	1.00 (SD = 0.000)	1	1
Muu tarkvara 2	0.85 (SD = 0.089)	0.8	1	0.68 (SD = 0.164)	0.5	0.8

2020. aastal toimunud II voo poiste keskmine tulemus oli 9.90 (SD = 2.68), tüdrukute aga 8.30 (SD = 2.44). Kuna andmed ei olnud normaaljaotusega (kontrollitud Kolmogorov-Smirnovi testiga, $p < 0.05$), rakendati mitteparameetiline Mann-Whitney U test. Tulemus näitas, et poiste ja tüdrukute tulemused ei olnud statistiliselt oluliselt erinevad ($U = 40.0$, $p > 0.05$). Uurides osalejate edukust ülesannete lõikes, selgus, et poisid said paremini hakkama ülesandega Eetika, turve, õigus 1 ja Muu tarkvara 2 ($p < 0.05$). Kõikides teistes ülesannetes erinevus ei olnud statistiliselt oluline ($p > 0.05$).

2.6.2 Vanuserühm Juunior

Juunioride vanuserühma tulemuste analüüsimiseks on samuti kasutatud vaade *Kirjeldav statistika*, kus on toodud tulemused ülesannete lõikes (vt. joonis 16).



Joonis 16. Juunioride vanuserühma tulemused ülesannete lõikes

Kasutades eespool toodud joonist 16 sai tehtud koondtabel 2.

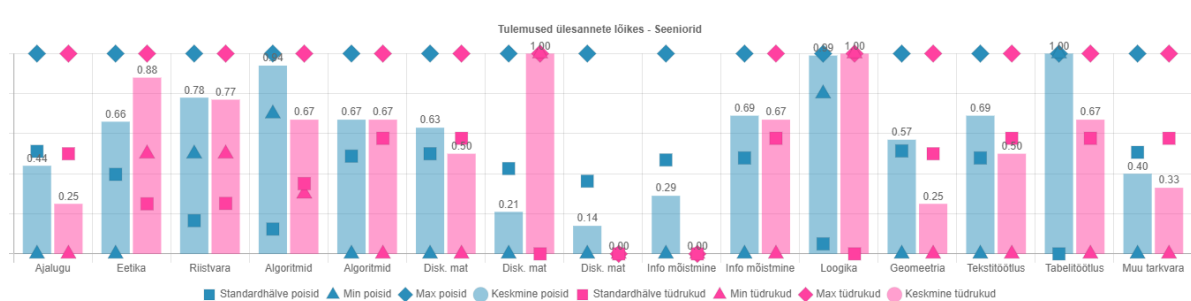
Juunioride vanuserühmas oli poiste keskmine tulemus 10.46 (SD = 2.83), tüdrukutel aga 8.72 (SD = 2.88). Ka selles vanuserühmas poiste ja tüdrukute tulemused ei olnud statistiliselt oluliselt erinevad ($U = 18.5$, $p > 0.05$). Kuid tüdrukud said paremini hakkama ülesandega Muu tarkvara ($p < 0.05$). Kõikides teistes ülesannetes erinevus ei olnud statistiliselt oluline ($p > 0.05$).

Tabel 2. Vanuserühma Juunior tulemused ülesannete lõikes

Ülesanne	Poisid			Tüdrukud		
	Keskmine (SD)	MIN	MAX	Keskmine (SD)	MIN	MAX
Ajalugu	0.38 (SD = 0.500)	0	1	0.00 (SD = 0.000)	0	0
Eetika, turve, õigus	0.71 (SD = 0.398)	0	1	0.38 (SD = 0.479)	0	1
Riistvara	0.74 (SD = 0.150)	0.5	1	0.80 (SD = 0.173)	0.7	1
Algoritmid 1	0.88 (SD = 0.332)	0	1	0.50 (SD = 0.577)	0	1
Algoritmid 2	0.81 (SD = 0.403)	0	1	1.00 (SD = 0.000)	1	1
Diskreetne matemaatika 1	0.56 (SD = 0.512)	0	1	0.75 (SD = 0.500)	0	1
Diskreetne matemaatika 2	0.59 (SD = 0.507)	0	1	0.25 (SD = 0.500)	0	1
Diskreetne matemaatika 3	0.69 (SD = 0.479)	0	1	0.50 (SD = 0.577)	0	1
Info mõistmine 1	0.81 (SD = 0.403)	0	1	0.50 (SD = 0.577)	0	1
Info mõistmine 2	1.00 (SD = 0.000)	1	1	1.00 (SD = 0.000)	1	1
Loogika	0.88 (SD = 0.332)	0	1	1.00 (SD = 0.000)	1	1
Geomeetria	0.95 (SD = 0.174)	0.3	1	1.00 (SD = 0.000)	1	1
Tekstitöötlus	0.65 (SD = 0.493)	0	1	0.25 (SD = 0.500)	0	1
Tabelitöötlus	0.82 (SD = 0.393)	0	1	1.00 (SD = 0.000)	1	1
Muu tarkvara	0.24 (SD = 0.437)	0	1	1.00 (SD = 0.000)	1	1

2.6.3 Vanuserühm Senior

Senioride vanuserühma tulemuste analüüsimiseks on kasutatud vaade *Kirjeldav statistika*, kus on välja toodud tulemused ülesannete lõikes (vt. joonis 17).



Joonis 17. Senioride vanuserühma tulemused ülesannete lõikes

Kasutades joonist 17 on tehtud koondtabel 3.

Tabel 3. Vanuserühma Senior tulemused ülesannete lõikes

Ülesanne	Poisid			Tüdrukud		
	Keskmine (SD)	MIN	MAX	Keskmine (SD)	MIN	MAX
Ajalugu	0.44 (SD = 0.512)	0	1	0.25 (SD = 0.500)	0	1
Eetika, turve, õigus	0.66 (SD = 0.397)	0	1	0.88 (SD = 0.250)	0.5	1
Riistvara	0.78 (SD = 0.166)	0.5	1	0.77 (SD = 0.252)	0.5	1
Algoritmid 1	0.94 (SD = 0.124)	0.7	1	0.67 (SD = 0.351)	0.3	1
Algoritmid 2	0.67 (SD = 0.488)	0	1	0.67 (SD = 0.577)	0	1
Diskreetne matemaatika 1	0.63 (SD = 0.500)	0	1	0.50 (SD = 0.577)	0	1
Diskreetne matemaatika 2	0.21 (SD = 0.426)	0	1	1.00 (SD = 0.000)	1	1
Diskreetne matemaatika 3	0.14 (SD = 0.363)	0	1	0.00 (SD = 0.000)	0	0
Info mõistmine 1	0.29 (SD = 0.469)	0	1	0.00 (SD = 0.000)	0	0
Info mõistmine 2	0.69 (SD = 0.479)	0	1	0.67 (SD = 0.577)	0	1
Loogika	0.99 (SD = 0.050)	0.8	1	1.00 (SD = 0.000)	1	1
Geomeetria	0.57 (SD = 0.514)	0	1	0.25 (SD = 0.500)	0	1
Tekstitöötlus	0.69 (SD = 0.479)	0	1	0.50 (SD = 0.577)	0	1
Tabelitöötlus	1.00 (SD = 0.000)	1	1	0.67 (SD = 0.577)	0	1
Muu tarkvara	0.40 (SD = 0.507)	0	1	0.33 (SD = 0.577)	0	1

Senioride vanuserühmas oli poiste keskmine tulemus 8.63 (SD = 2.63), tüdrukutel aga 6.45 (SD = 2.78). Antud vanuserühmas samuti poiste ja tüdrukute tulemused ei olnud statistiliselt oluliselt erinevad ($U = 18.0$, $p > 0.05$). Käsitletavas vanuserühmas tüdrukud said paremini hakkama ülesandega Diskreetne matemaatika 2 ($p < 0.05$), aga poisid Tabelitöötluse ülesandega ($p < 0.05$). Kõikides teistes ülesannetes erinevus ei olnud statistiliselt oluline ($p > 0.05$).

2.7 Visualiseerimisvahendi kohandamine järgmisteks viktoriinideks (arendajale)

Ülesannete tüübid ja iga tüübi kohta valitud küsimuste arv võib aastast aastasse varieeruda. See sõltub sellest, mitu küsimust Eestis peetava informaatikaviktoriini korraldajad valivad iga ülesande tüübi jaoks. Arvestades seda, on välja toodud sammud, mida arendaja peab ette võtma visualiseerimisvahendi kohandamisel järgmisteks viktoriinideks. Uue analüüsi lehe lisamise ajakulu on orienteeruvalt 30-60 min.

2.7.1 Moodle keskkonnast saadud .xlsx faili muutused

Iga küsimuse tulba (kokku 15) nimi peab olema toodud kujul K1, K2, ..., K15.

2.7.2 Muutused .html failides

Järgmistel aastatel visualiseerimisvahendi lehtedel tuleb teha järgmised muudatused (Benjamini vanuserühma näitel):

- 1) täiendada *index.html* leht uue lingiga lehele *benjamins2021.html* (siin ja edasi toodud faili nimetus on illustratiivne ja ei ole rangelt määratud);
- 2) tekitada uus fail nimega *benjamins2021.html*;
- 3) kopeerida *benjamins.html* faili sisu *benjamins2021.html* faili;
- 4) `<table id="tableData"></table>` – määrab vaadet *punktide tabel*, kus pealkirjas tuleb määrata ülesande tüüp ja tüübile vastav värv. Värvide määramisel tuleb lähtuda sellest, mitu tulpa sama värviga kõrvuti asub. Näiteks küsimusi ajaloo kohta oli 1, siis *span="1"* jääb samaks. Kui aga oleks 2, siis tuleks määrata *span="2"*, jne.;
- 5) `<div id="chart-container-sel2">` – see on element, mis määrab konteinerite arvu sektor-diagrammide jaoks. Kuna viimane sõltub ülesannete tüüpide arvust, siis tuleb kontrollida, et konteinerite arvusid ei oleks üle või et neid ei jääks puudu. Viimasel juhul tuleb lisada:
 - `<canvas id="doughnut-chartXX"></canvas>` elementi igasse konteinerisse, kus *XX* määrab vaba numbrit;
 - kolm nuppu esimese konteineri juurest (Poisid, Tüdrukud, Eemalda);
- 6) `<div id="chart-container-sel3">` – kaks konteinerit *kirjeldava statistika* vaates graafikute jaoks.

2.7.3 Muutused .js failis

JavaScripti failis peavad olema tehtud järgmised muudatused (Benjamini vanuserühma näitel):

1. *var url* – omistada uus .xlsx faili asukoht;
2. *line-chart* – kontrollida *labels* väärtuste vastavust küsimustele;
3. iga *doughnut-chart* kohta kontrollida:
 - *labels* väärtus, näiteks *label*: "Ajalugu kõik osalejad";
 - *text* väärtus, näiteks *text*: 'Benjaminide tulemused "Ajalugu". Kõik osalejad: ' + *osalejadKokku*;
 - *data* väärtus – tuleb arvestada, kas andmed on võetud ühest või mitmest punktide tabeli tulpadest. *Data* defineeritakse enne sektordiagrammi defineerimist, ehk tuleb kõik väärtused üle vaadata;
4. *bar-chart-stat* kohta kontrollida:
 - *labels* väärtuse vastavust küsimustele;
 - *data* väärtuse – defineeritakse enne tulpdiaagrammi;
5. *bar-chart-stat-group* kohta kontrollida:
 - *labels* väärtuse vastavust küsimuste grupile;
 - *data* väärtus – uurida, kas andmed on võetud ühest või mitmest punktide tabeli tulpadest. Vajadusel kohandada andmete asukoht x-teljel (näiteks: standardhälve poiste jaoks tulbas „Ajalugu“ määratakse järgnevalt {y:stdDevk1B, x:0.5} ja tüdrukute jaoks {y:stdDevk1G, x:1.1}). Viimast arvvaartust nimetatakse sammu pikkuseks (ingl. *Step size*) x-teljel. Kui ülesannete arv tüüpide kaupa suureneb või vähe- neb, nihkuvad punktid teljel (minimaalse, maksimaalse ja standardhälbe punktid);

Andmete määramisel alumisele graafikule tuleb kontrollida, et *kirjeldav statistika* vaates on ar- vesse võetud erinevate ülesannete tüüpide hulk ja mitu küsimust on seotud iga tüübiga. Üheks kontrollimise võimaluseks on ülemise ja alumise graafiku tulemuste võrdlemine (*bar-chart-stat* vs *bar-chart-stat-group*). Näiteks kui ülemisel graafikul on ainult üks *Tekstitöötuse* ülesanne ja selle juures tüdrukute keskmine tulemus on 0.50 ja poiste oma on 0.42, siis peab see jääma samaks ka alumisel graafikul, välja arvatud juhul, kui *Tekstitöötuse* ülesannete tüübis on rohkem kui üks üle- sanne ja keskmine on saadud mitme arvujada pealt.

6. *var list* – kontrollida, et ridade arv vastaks osalejate arvule (vajadusel lisada /eemaldada read).

2.8 Võimalik tuleviku arendus

Ühe ja sama ülesande tüübiga võib olla seotud erinev arv küsimusi. Kui see oleks püsiv, siis oleks võimalik automatiseerida visualiseerimislehe lisamist järgmise aasta tulemustega. Sel juhul ajakulu oleks seotud ainult uue Excel'i faili üleslaadimisega *datafiles* kausta. Ilma selle protsessi automatiseerimiseta nõutakse arendaja poolset jälgimist, et õiged andmed jõuaksid õigesse kohta.

Rakenduse edasiarenduse üheks võimaluseks on erinevate kontrollimisvõimaluste lisamine Excel'ist andmete importimisel. Sel juhul rakendus hakkaks manuaalset sisestatud osa automaatselt kontrollima. Sellise võimaluse implementeerimine põhjustaks lisa ajakulu tulemuste kuvamisel *.html* lehel. Samas kuna Excel'i andmed ei muutu, siis tulemuste kuvamine võtaks aega vaid esimesel avamisel ja edasi saaks andmeid hoida brauseri küpsistes (ingl. *HTTP cookie*).

Praegu kehtiva viktoriini korralduse kohaselt ükski küsimus ei anna rohkem, kui ühe punkti. Kui tulevikus hindamismudel muutub, siis tuleb kontrollida graafikute kuvamist lehel. Võimalik, et tuleb kohandada skaala väärtused ja/või konteinerite suurused.

3. Interaktiivne ülesannete kogumik

Antud peatükis käsitletakse interaktiivset ülesannete kogumikku, mille sisuks on 2020. aasta teise voo ülesannete sõnastused ja õiged vastused koos seletustega. Samuti on lisatud info selle kohta, kuidas iga küsimus on seotud informaatikaga. Kokku on toodud 45 küsimust kolmele vanuserühmale. Peatüki lõpus on kirjeldatud interaktiivse ülesannete kogumiku tehniline pool ning selle koostamisvõimalused järgmistele viktoriinidele.

3.1 Ülesannete kogumiku vajadus

Eestis informaatikaviktoriini korraldatakse juba peaaegu 15 aastat. Kodulehel on alati olnud ülesannete sõnastused ja õiged vastused. Kuid alles eelmisel aastal ilmus kodulehele esimene täisväärtuslik kogumik, kus olid toodud ka ülesannete lahenduskäigud ning iga ülesande seos informaatikaga. Eestikeelse versiooni koostas Veronika Lehesaar oma lõputöö raames. Arvestades, et vene keelt kõnelevate osalejate arv (eriti benjaminide vanuserühmas) on märkimisväärne, on korraldajate arvamusel venekeelse versiooni loomine tänuväärne ja vajalik tegevus. Käesolevas interaktiivses ülesannete kogumikus pealkirjaga „Задания II тура викторины по информатике КОBRAS“ on ülesanded, mida kasutati viktoriini 2020. aasta II vooorus. Loodud kogumik on suunatud eelkõige vene keelt kõnelevatele informaatikaviktoriini osalejatele. Lahenduskäikude koostamise ja informaatika seoste välja toomise aluseks on võetud failid, mis on kättesaadavad viktoriini rahvusvahelisest andmebaasist. Materjal, mis on esitatud interaktiivses kogumikus on atraktiivseks õppevahendiks. Kasutajad saavad liikuda ülesannete vahel, vaadata kõik ülesanded korraga või üksikhaaval ning kuvada või peita õiget vastust.

3.2 Benjaminid (6.–8. klass)

1. Юбилей WWW

Автор: Ахто Труу (Эстония)

В 2019 году у всеизвестного веба (WWW, World Wide Web) был юбилей.

Вопрос

Сколько лет исполнилось вебу?

- A. 15 лет
- B. 20 лет
- C. 25 лет
- D. 30 лет

Ответ

D. 30 лет

Объяснение решения

Веб зародился в Швейцарии в 1989 году, когда научные документы стали объединять в гипертексты, которые, в дальнейшем, размещали в распределительные сервера. В оборот взяли и другие документы, так появилась всемирная паутина.

Это информатика, потому что

Большой вклад в развитие интернета внёс инженер Тим Бернерс-Ли, который в конце 1980-ых годов работал над созданием большой базы данных со ссылками. Она была необходима для передачи цифровой информации между компьютерами в организации по ядерным разработкам CERN. Информацию хранили в виде гипертекстов в соответствии с правилами HTML. Помимо HTML использовалось приложение HTTP и WorldWideWeb. HTML и веб играют очень важную роль в науке о вычислительных машинах.

2. Ложные новости

Автор: Виктор Шмидт (Голландия)



Национальное агентство новостей Бобруйска выпускает программу с вечерними новостями. Большинство новостей содержит правдивую информацию, хотя встречаются и ложные новости.

Каждый день четыре бобра – Алиса, Боря, Валера и Галя, смотрят все вместе выпуск новостей.

- Алиса всегда точно знает, является ли новость правдивой или же это ложная новость.
- Боря считает, что все новости являются ложными новостями.
- Валера считает, что все правдивые новости — это ложные новости, а все ложные новости — это правдивые новости.
- Галя считает, что все новости являются правдивыми.

Бобры договорились между собой о том, что новость будет являться правдивой, если по меньшей мере трое из них будут считать (или знать) эту новость правдивой.

Вопрос

Когда бобры будут считать новость правдивой?

- A. Когда новость на самом деле будет правдивой
- B. Когда новость на самом деле будет ложной новостью
- C. Всегда
- D. Никогда

Ответ

- D. Никогда

Объяснение решения

Рисуем таблицу:

Условие	Алиса считает, что это	Боря считает, что это	Валера считает, что это	Галя считает, что это
Новость является правдивой	Правда	Ложь	Ложь	Правда
Новость является ложной	Ложь	Ложь	Правда	Правда

В обоих случаях только два бобра утверждают, что новость правдивая. Так как по меньшей мере трое бобров должны считать новость правдивой, они никогда не придут к общему мнению.

Это информатика, потому что

Компьютеры всегда точно выполняют команды, данные им. Данные должны быть представлены в понятной да/нет форме. Однако бывают ситуации, когда информация интерпретируется не совсем правильно. Например, световой датчик загрязнился и не может распознать, что солнце уже встало, но нам не хотелось бы, чтобы свет в комнате горел целый день.

Во многих приложениях компьютерные программы должны правильно распознавать противоречивые данные. Например, у нас три световых датчика и два из них говорят, что на улице светло, а один, что всё-таки темно. Что мы пожелаем, чтобы сделал компьютер в

данной ситуации? (А что, если мы говорим не о лампочках, а, например, о самоуправляемом автомобиле, который должен распознавать, есть ли человек на его пути или нет?)

Один из способов решить данную проблему – доверять большинству источников данных, или, как приведено в данной задаче, доверять только в том случае, если достаточное количество источников согласится. (Но, как показано в задании, даже это не всегда работает).

Интересный факт: на Space Shuttle (американском многоразовом транспортном космическом корабле) было 4 одинаковых компьютера с одинаковыми программами, чтобы принимать решение. Более 6 раз в секунду проверялось, согласны ли все 4 компьютера и делают ли они абсолютно тоже самое. Если один из них делал что-то отличное от других (три раза подряд), срабатывал сигнал тревоги, и компьютер выводился из строя и заменялся резервной копией.

3. Фотографирование в школе

Автор: Ахто Труу (Эстония)

Все чаще задают вопросы о сделанной в школе фото- и видеосъемке. Когда это можно делать без ограничений, а когда на съемку надо спрашивать разрешения?

Вопрос

Если фотография сделана в школе, то в каких случаях надо спрашивать разрешения на ее использование у запечатленных на фотографии людей? Выбери все правильные ответы!

- A. Фотографирование себе на память
- B. Опубликование фотографии в социальных сетях
- C. Опубликование фотографии в печати
- D. Фотографирование в качестве доказательства правоохранительным органам

Ответ

B. Опубликование фотографии в социальных сетях, C. Опубликование фотографии в печати

Объяснение решения

В законе сказано, что человек может сохранить свою анонимность, если он того желает. Это значит, что его данные, в том числе фотографии, нельзя опубликовать без его разрешения, за исключением публичных мероприятий или если это разрешено законом. Что касается публичных мероприятий, то камера не должна быть скрытой, чтобы у человека была

возможность отойти от прицела камеры. Фотографирование в качестве доказательства правоохранительным органам разрешено без отдельного разрешения запечатленных на фотографии людей. Регулировкой выше приведённых законов занимается [Инспекция по защите данных](#). Часто задаваемые вопросы касательно фотографирования собраны ими в рубрике [Полезно знать](#).

Это информатика, потому что

Этика и право играют важную роль в науке о вычислительных машинах. Подобно тому, что нельзя фотографировать людей, проходящих на улице или даже своего друга без разрешения, надо соблюдать установленные в виртуальном мире правила.

4. Дополнительные устройства

Авторы: Лидия Феклистова, Реймо Пальм (Эстония)

У компьютера может быть много разных дополнительных устройств. Некоторые из них являются устройствами ввода (они предназначены для ввода информации в компьютер), другие – устройствами вывода (они предназначены для вывода информации из компьютера), но есть и носители информации (они предназначены для хранения данных, которыми потом можно вновь воспользоваться).

Вопрос

К какому типу относятся приведенные ниже устройства?

- A. Сканер
- B. Колонки
- C. SD-карта
- D. Стилус
- E. USB-флеш-накопитель
- F. Проектор

Ответ

- A. Сканер – устройство ввода
- B. Колонки – устройство вывода
- C. SD-карта – носитель информации
- D. Стилус – устройство ввода
- E. USB-флеш-накопитель – носитель информации
- F. Проектор – устройство вывода

Объяснение решения

- A. Сканер – картинка/документ сканируется в компьютер с бумаги
- B. Колонки – вывод звука
- C. SD-карта – карта памяти с флеш-памятью
- D. Стилус – устройство похожее на карандаш, используется для ввода данных на сенсорном экране
- E. USB-флеш-накопитель – носитель информации с флеш-памятью
- F. Проектор – картинка проецируется на большой экран

Это информатика, потому что

Все эти устройства используются каждодневно в науке о вычислительных машинах для упрощения выполняемых задач. Например, используя сканер, не придётся рисовать картинку на компьютере или вновь вносить документ на компьютер.

5. Переработка стекла

Автор: Лаура Унгуреану (Румыния)

Школьный директор уделяет особое внимание переработке стекла. При переработке стекла имеет большое значение, является ли оно белым или цветным. Директор приобрёл магические машины, которые перерабатывают стекло. Каждая из двух машин берёт по два кусочка стекла и перерабатывает их. Третья машина берёт один кусочек стекла и перерабатывает его.



Эта машина выдаёт белое стекло только в том случае, если в машину было положено два кусочка белого стекла. Результатом любой другой комбинации будет цветное стекло.

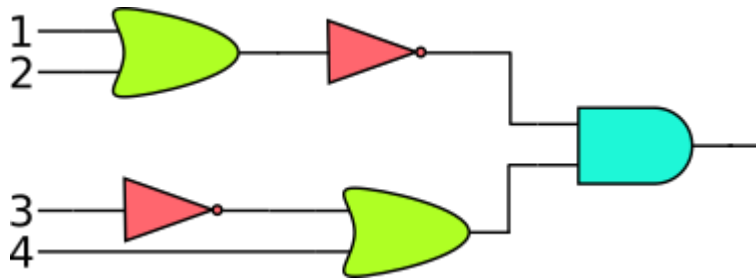


Эта машина выдаёт цветное стекло только в том случае, если в машину было положено два кусочка цветного стекла. Результатом любой другой комбинации будет белое стекло.



Эта машина преобразовывает цветное стекло в белое, а белое стекло – в цветное.

Директор составил из машин следующую систему:



Вопрос

Какой тип стекла следует положить на входы 1, 2, 3 и 4, чтобы результатом было белое стекло?

- A. 1 = белое, 2 = белое, 3 = цветное, 4 = белое
- B. 1 = цветное, 2 = цветное, 3 = цветное, 4 = белое
- C. 1 = белое, 2 = цветное, 3 = цветное, 4 = белое
- D. 1 = цветное, 2 = цветное, 3 = белое, 4 = цветное

Ответ

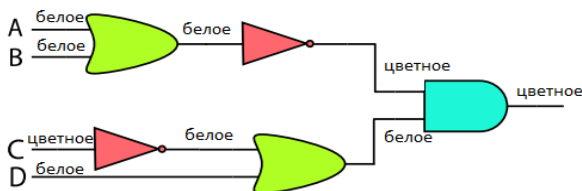
B. 1 = цветное, 2 = цветное, 3 = цветное, 4 = белое

Объяснение решения

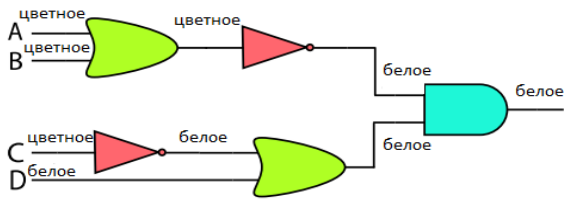
Решение 1

Мы можем нарисовать картинку для каждого варианта и подписать на линиях, какое стекло идёт на переработку и какое получается в итоге. Так, проверив все возможные комбинации, мы придём к правильному ответу. В данном случае, единственный вариант, при котором в результате получается белое стекло, приведён на картинке под буквой В.

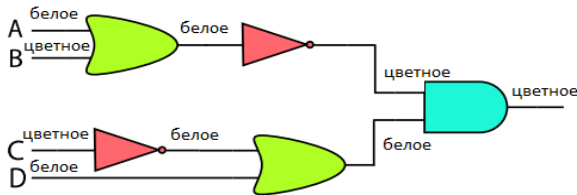
Картинка А



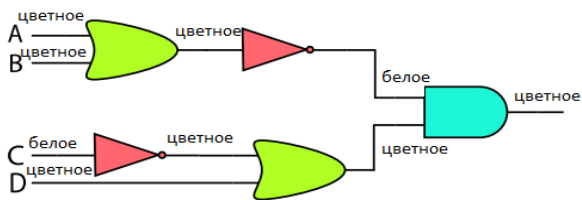
Картинка В



Картинка С



Картинка D



В сумме существует три возможных комбинации, при которых на выходе получится белое стекло:

- 1 = цветное, 2 = цветное, 3 = цветное, 4 = цветное
- 1 = цветное, 2 = цветное, 3 = цветное, 4 = белое (ответ к данной задаче)
- 1 = цветное, 2 = цветное, 3 = белое, 4 = белое

Решение 2

Для более эффективного решения данной задачи мы можем начать с конца, двигаясь в начало. Чтобы в итоге получилось белое стекло, в последнюю машину должно быть положено два белых стекла.

Чтобы на верхний вход получить белое стекло, надо в среднюю машину положить цветное. Последнее подразумевает, что на входе в первую машину должны быть цветные стёкла.

Чтобы получить белое стекло на последний нижний вход, хоть одно стекло на входах в предыдущую машину должно быть белым. На верхнем входе для получения белого стекла нам нужно положить в машину цветное. Если на самом нижнем входе белое стекло, то мы

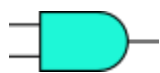
получим цветное и белое на входе в среднюю нижнюю машину и белое на выходе из машины. Либо, как вариант, мы можем на вход 3 положить цветное и на вход 4 цветное и получить на входе в среднюю нижнюю машину белое и цветное стёкла.

Таким образом, на вход 1 и 2 надо положить цветное стекло, на вход 3 подходит белое, либо цветное, если на входе 4 белое. На 4 вход подходит цветное стекло только в том случае, если на входе 3 цветное. Как видно из последнего, единственный, подходящий нам вариант ответа это В.

Это информатика, потому что

Все компьютеры содержат схемы, которые состоят из различных типов небольших элементов, называемых логическими воротами или логическими элементами. Некоторые из самых популярных логических ворот (логические операторы): НЕ, ИЛИ, И, ИСКЛЮЧАЮЩЕЕ ИЛИ (англ. XOR).

В данной задаче есть операторы И, ИЛИ и НЕ (их графическое представление такое же, как и в инженерии на электрических схемах). В технике элементы работают с использованием электрических сигналов. Мы отмечаем их как 1, если есть сигнал, и 0, если нет сигнала. В логике представлены как: 1 – ИСТИНА и 0 – ЛОЖЬ. В этой задаче мы рассматриваем белое стекло как ИСТИНА (или 1) и цветное стекло как ЛОЖЬ (или 0).



Оператор **И** требует, чтобы оба входа были = ИСТИНА (белое стекло), чтобы получить ИСТИНА в качестве выхода.



Оператор **ИЛИ** требует, чтобы хотя бы один из входов был = ИСТИНА, чтобы получить ИСТИНА в качестве выхода. Для сравнения ИСКЛЮЧАЮЩЕЕ ИЛИ (англ. XOR). ИСКЛЮЧАЮЩЕЕ ИЛИ даёт ИСТИНУ только в том случае, если входы разные, то есть ИСТИНА (белое стекло) получится только если один вход = ИСТИНА (белое стекло), а другой вход = ЛОЖЬ (цветное стекло). В обиходе мы чаще используем слово ИЛИ скорее для обозначения ИСКЛЮЧАЮЩЕГО ИЛИ, в официальных же документах неисключающее значение подчёркивают отдельно и обычно записывают следующим образом: А или В, или оба.



Оператор **НЕ** изменяет значения, поэтому, если вход ИСТИНА (белое стекло), выход ЛОЖЬ, а если вход ЛОЖЬ, то выход ИСТИНА (белое стекло).

Логические элементы используются, помимо прочего, в архитектуре микропроцессора для выполнения арифметических и логических операций.

6. Перестановки

Автор: Ростислав Шпакович (Украина)

Игровое поле состоит из девяти ячеек. В первоначальном состоянии три красных и три зелёных карты располагаются в ячейках в данной последовательности:



Перестановка означает, что определённые карты были перемещены из одной ячейки в другую в одно и то же время. Допустимы два вида перестановок (карты считаем слева направо).

Первая перестановка: Карты под номерами 1 и 4 одновременно перемещаем на три клетки вправо. Таким образом, четвертую карту перемещаем в пустую ячейку, расположенную правее шестой карты; затем перемещаем первую карту в ячейку, где ранее находилась четвёртая карта. Внимание! Пустые ячейки нельзя принимать за карты.

Вторая перестановка: Карты под номерами 1, 3 и 5 одновременно перемещаем на две клетки вправо. Таким образом, пятую карту перемещаем в пустую ячейку, расположенную правее шестой карты; затем перемещаем третью карту в ячейку, где ранее находилась пятая карта, а первую карту перемещаем в ячейку, где ранее находилась третья карта.

Перестановки можно применять последовательно друг за другом. Например, применяя к первоначальному состоянию последовательно "первую перестановку; вторую перестановку", мы можем получить следующий результат:

После первой перестановки:



и после второй перестановки:



Вопрос

Какая последовательность перестановок приведёт к следующему конечному состоянию, если ее применить к первоначальному состоянию?



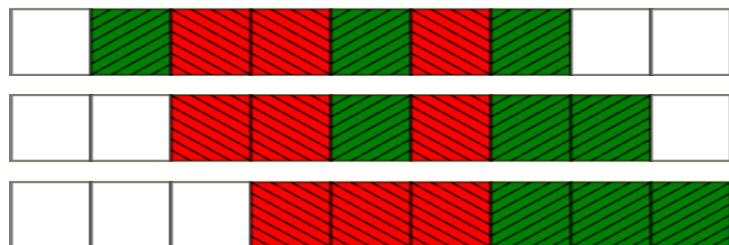
- A. Первая перестановка; вторая перестановка; первая перестановка
- B. Вторая перестановка; первая перестановка; первая перестановка
- C. Первая перестановка; первая перестановка; вторая перестановка
- D. Первая перестановка; первая перестановка; первая перестановка

Ответ

C. Первая перестановка; первая перестановка; вторая перестановка

Объяснение решения

Расположение карт после каждой перестановки:



Тестирование каждой последовательности перестановок занимает время. Один из способов ускорить процесс – найти что-то, что дисквалифицирует последовательность без необходимости применения каких-либо дополнительных перестановок. Например, любая перестановка, которая помещает красное поле в позиции 7, 8 или 9, не может привести к желаемому порядку. Это легко увидеть, заметив, что ни одна перестановка не может переместить карту влево, а в конечном порядке красные карты находятся в позициях 4, 5 и 6, которые находятся слева от позиции 7.



В приведённом в задании примере показано, что происходит после первой перестановки, а затем после второй перестановки. На картинке видно, что это приводит к появлению красной карты в позиции 8. Это означает, что любая последовательность, которая

начинается с «Первая перестановка; вторая перестановка;» не может быть правильной, и поэтому ответ А не может быть верным.

Применение второй перестановки сначала помещает красные карты в позиции 5 и 6. Применение другой перестановки поместит красные карты в позиции 7 или 8, используя первую перестановку или вторую соответственно. Это означает, что ответ В не может быть верным.

Разница между ответом С и D заключается в последнем этапе, поэтому мы не можем избежать тестирования последней перестановки.

Это информатика, потому что

Перестановки – это упорядоченные наборы, в которых некоторые элементы могут быть (пере)упорядочены процессом, называемым перестановкой. Эти упорядоченные элементы образуют кортежи. Например, существует шесть перестановок набора $\{1,2,3\}$, а именно кортежи (1,2,3), (1,3,2), (2,1,3), (2,3,1), (3,1,2) и (3,2,1).

Перестановки очень важны в математике, а также в информатике. Например, они используются для анализа алгоритмов сортировки, которые чрезвычайно важны в программировании.

Эта задача связана с сортировкой или перестановкой элементов, что развивает навыки проектирования и написания алгоритма для точного выполнения поставленной задачи. Здесь также показано, как изменение порядка выполнения команд может существенно повлиять на конечный результат.

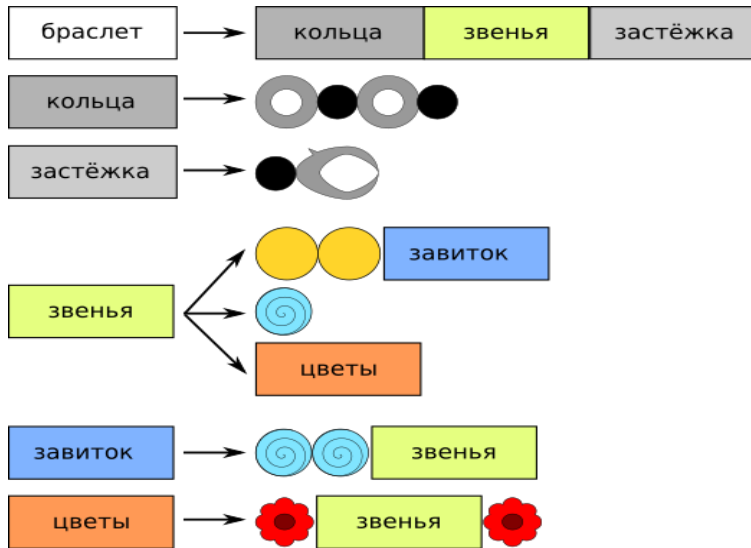
7. Браслет дружбы

Автор: Эллиен Ванхове (Бельгия)

Саша изготавливает браслеты дружбы. Он начинает с элемента

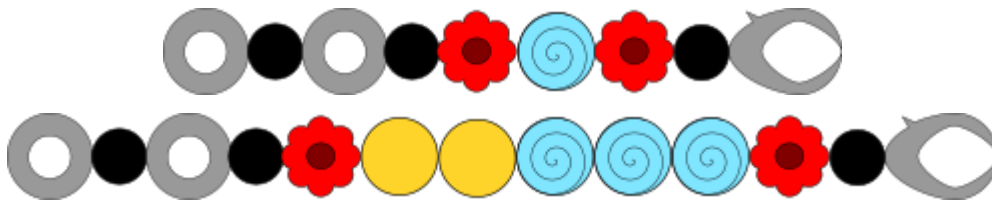
браслет

 и далее следует нижеприведённым правилам:



Каждое правило означает, что элемент слева замещается на элементы, которые располагаются после одной из стрелок.

Неоднократно используя эти правила, Саша может составить, например, два таких браслета:



Для своих четырёх друзей Саша изготовил четыре браслета. К сожалению, один из друзей случайно порвал свой браслет. Он попытался его починить, но допустил ошибку.

Вопрос

Какой из приведённых ниже браслетов содержит допущенную ошибку?

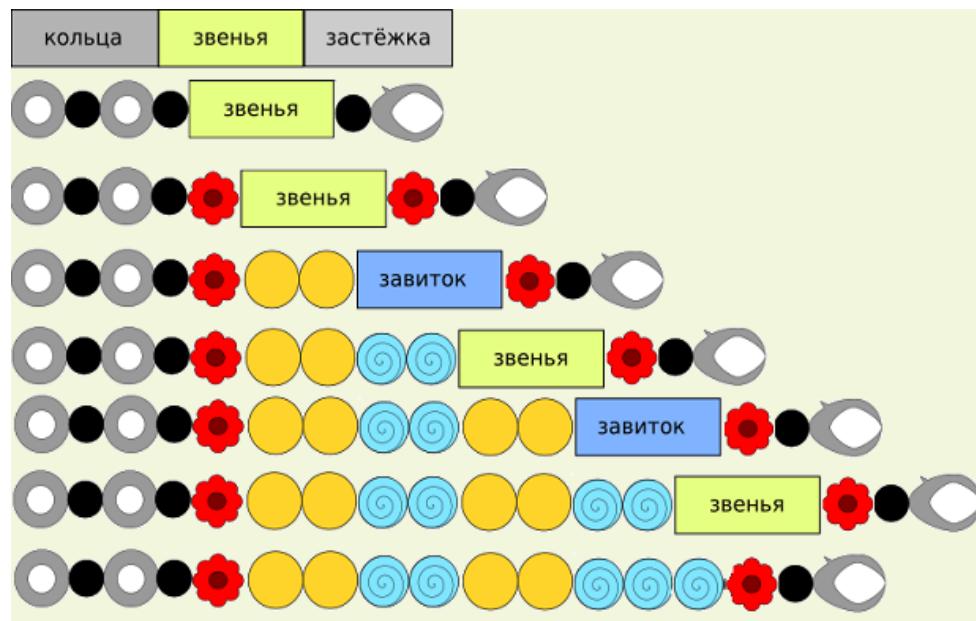
- A.
- B.
- C.
- D.

Ответ

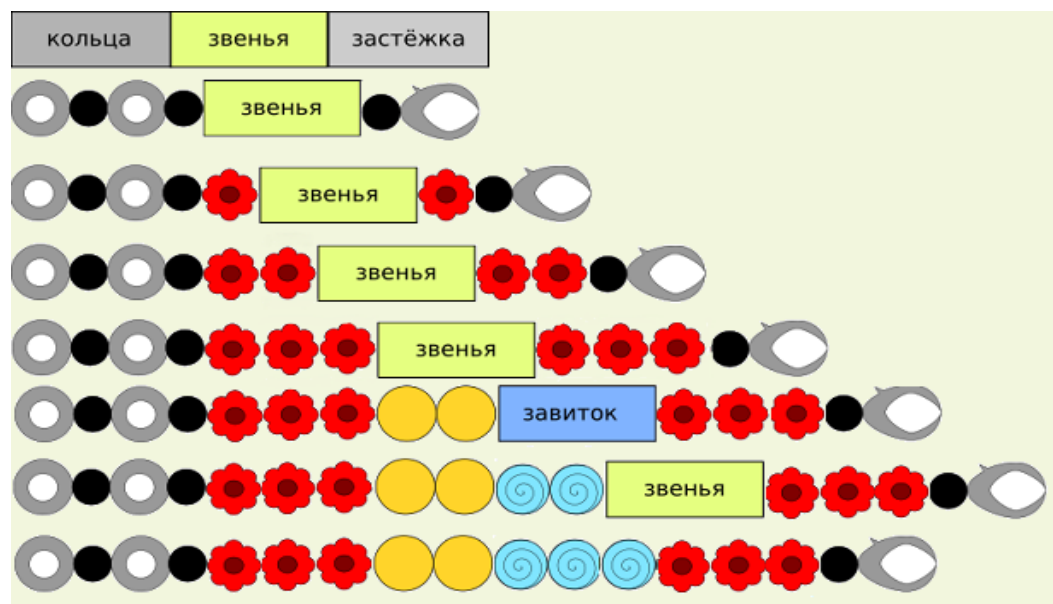
Ошибка в браслете С.

Объяснение решения

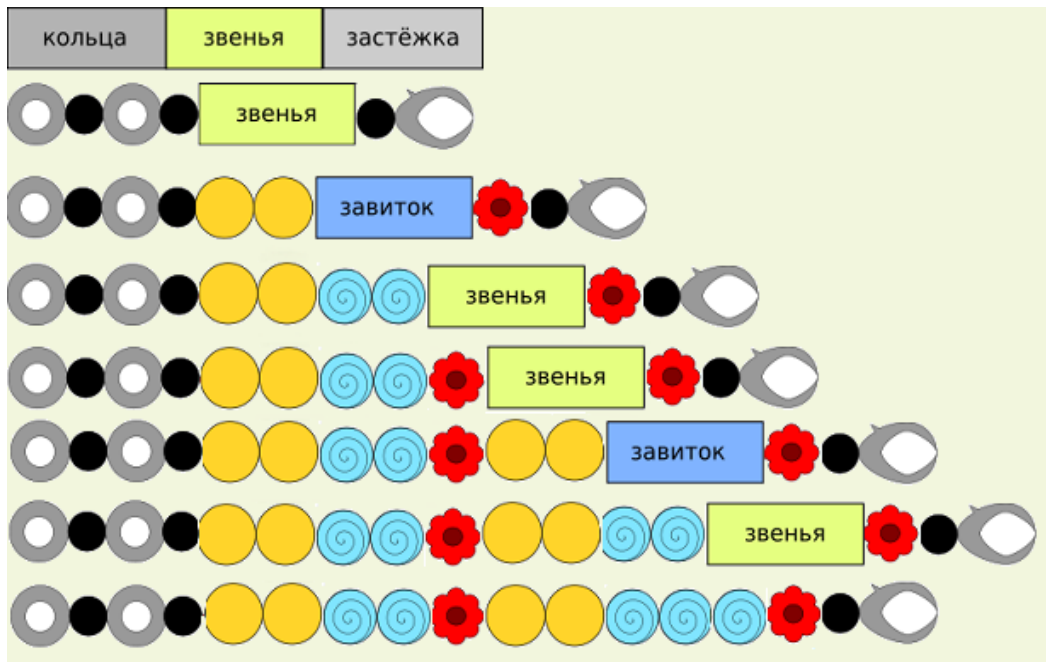
Вместо того, чтобы доказывать, что С имеет ошибку, было бы легче показать, что другие три браслета действительно могут быть сделаны с использованием этих правил. Для браслета D:



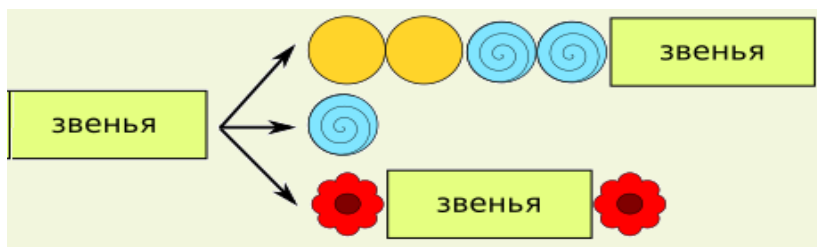
Не так сложно сделать похожие картинки для А





и для В.



Чтобы доказать, что браслет С не может быть создан с использованием установленных правил, мы рассмотрим правила: «звенья», «завиток» и «цветы». Если их объединить, получим:



Другими словами, если мы проигнорируем цветочную вставку, то каждый браслет состоит из набора , повторенного несколько раз (один раз или 0 раз), а затем используется одиночный .

Видно, что браслеты А, В и D следуют этому шаблону, а браслет С – нет: в середине браслета три бусинки с «завитком» подряд, что по правилам недопустимо.

Это информатика, потому что

Набор правил, приведенных в этом задании, может использоваться компьютером, чтобы проверить, мог ли Саша изготовить браслет. Подобные правила используются компьютерами для проверки того, допустил ли программист какие-либо опечатки в своих

программах, или для проверки того, что печатается в веб-форме. Например, следующие правила

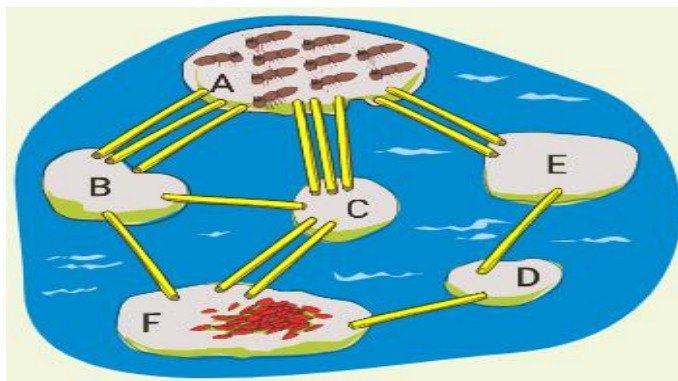
- целое число → натуральное число *или* +натуральное число *или* -натуральное число
- натуральное число → 0 *или* 1 *или* 2 *или* 3 *или* 4 *или* 5 *или* 6 *или* 7 *или* 8 *или* 9

могут быть использованы, чтобы сказать компьютеру, какой номер разрешено использовать в поле ввода.

8. Муравьи на болоте

Автор: Валентина Дагиене (Литва)

На камне А ползает 10 муравьёв, которые пытаются добраться до еды, расположенной на камне F. По одному пруту одновременно может перемещаться только один муравей. Чтобы добраться с одного камня на другой, муравью понадобится 1 минута.



Вопрос

Каково максимальное количество муравьев, которое сможет в течение первых 3 минут добраться до еды на камне F?

Ответ

7

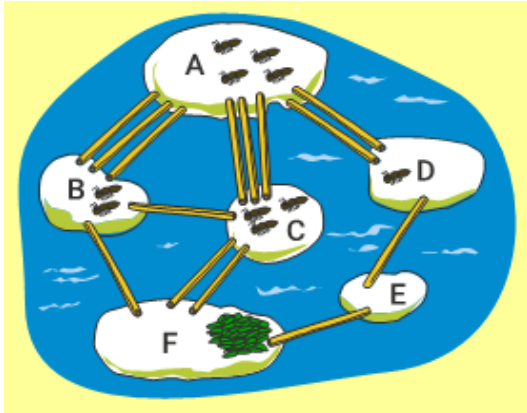
Объяснение решения

В данном задании, не используя существующий алгоритм, мы можем утверждать, что:

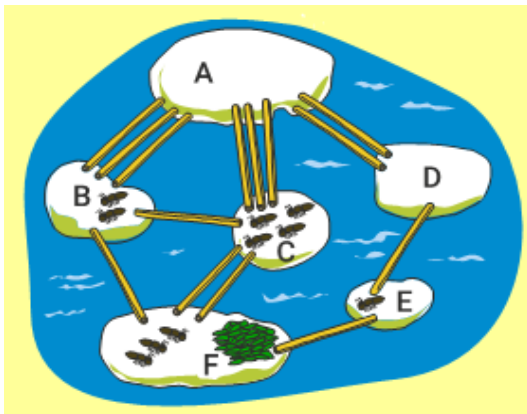
- не имеет смысла отправлять более одного муравья через камни D-E;
- не имеет смысла отправлять более двух муравьёв через камни A-B;
- тростинка между B-C не участвует в движении и может быть проигнорирована;
- ограничивающим фактором являются соломинки B-F и C-F.

Возможная ситуация после каждой минуты представлена на картинках ниже.

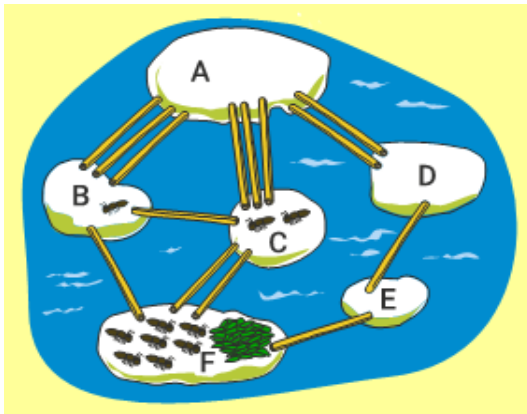
После первой минуты:



После второй минуты:



После третьей минуты:



Это информатика, потому что

Цель задачи состоит в том, чтобы оптимизировать поток муравьев через сеть, состоящую из прутиков и камней и, тем самым, доставить как можно больше муравьев к еде за 3

минуты. Муравьи, которые не знают о структуре сети, в которой они путешествуют, не смогут найти лучшее решение. Но наблюдатель, который видит структуру, может найти оптимальную стратегию.

В этой задаче мы предполагаем, что муравьи знают о структуре сети, в результате чего они выбирают движение по определенному пути.

Графы — это абстрактные структуры данных, которые используются для моделирования сетей, и существует множество алгоритмов для оптимизации потока при определенных обстоятельствах.




9. Вечеринка

Автор: Флорентина Воборил (Австрия)

Полина и Тимур получили приглашение на вечеринку. Вечеринка начинается в 15:00.

Ребята решили взять с собой свежеспечённую выпечку: печенье, пирог и пирожные.

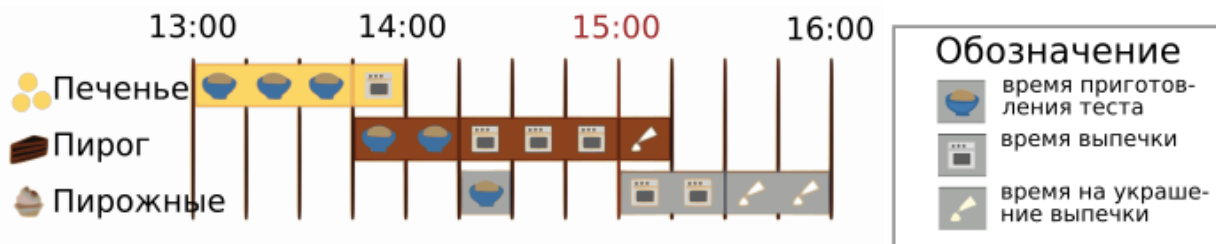
В книге рецептов указаны следующие временные затраты:

 Печенье Тесто: 45 мин Выпекание: 15 мин Украшение: 0 мин	 Пирог Тесто: 30 мин Выпекание: 45 мин Украшение: 15 мин	 Пирожные Тесто: 15 мин Выпекание: 30 мин Украшение: 30 мин
--	---	--

Полина занимается приготовлением теста и ставит его в печку, а после выпекания Тимур занимается украшением выпечки.

В печке есть место только для одной выпечки. Ни Полина, ни Тимур не могут выполнять несколько работ одновременно.

Ребята начали готовить в 13:00 и хотят всё приготовить к 15:00. Ребята составили следующий график работ:



Увидев, что для приготовления потребуется слишком много времени, ребята задумались о том, как успеть все сделать к 15:00. Для оптимизации работы им пришла идея изменить порядок приготовления выпечки.

Вопрос

Каково самое раннее время, когда эти 3 выпечки будут готовы?

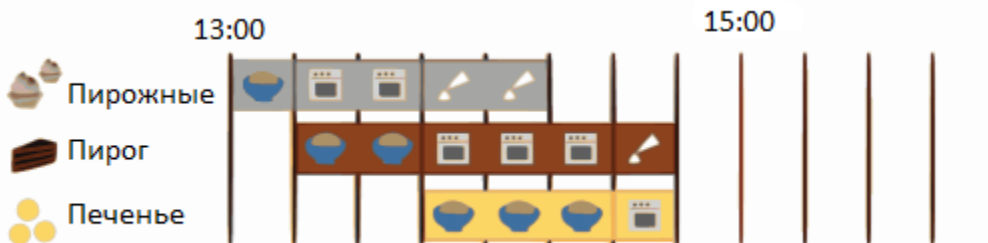
- A. 14:15
- B. 14:30
- C. 14:45
- D. 15:00

Ответ

C. 14:45

Объяснение решения

Если сначала готовятся пирожные, затем пирог и, наконец, печенье, ребята могут закончить в 14:45.



Полина и Тимур успеют вовремя выпечь всё необходимое и приведённый выше график самый оптимальный. Для приготовления теста требуется 1,5 часа (до 14:30), независимо от порядка, в котором оно делается. Если пирожные или пирог являются последним десертом,

то для выпечки и украшения нужно не менее одного часа после 14:30, поэтому именно печенье будет выпекаться последним и будет готово в 14:45.

Это информатика, потому что

Вместо того, чтобы готовить один пирог за другим, производство выпечки делится на несколько подзадач: тесто, выпекание, украшение. Это позволяет Полине готовить тесто для пирога, пока пирожные выпекаются в духовке. Данный процесс называется конвейерной обработкой и часто используется в микропроцессорах, что позволяет более эффективно использовать ресурсы процессора и выполнять некоторые команды параллельно.

10. Цветной китайский символ

Авторы: Янг Ксинг, Ланпинг Денг (Китай)

Внимание маленького бобра привлёк один китайский символ. Бобр хочет изучить строение символа. Бобр изготовил следующую цвето-узорчатую схему:

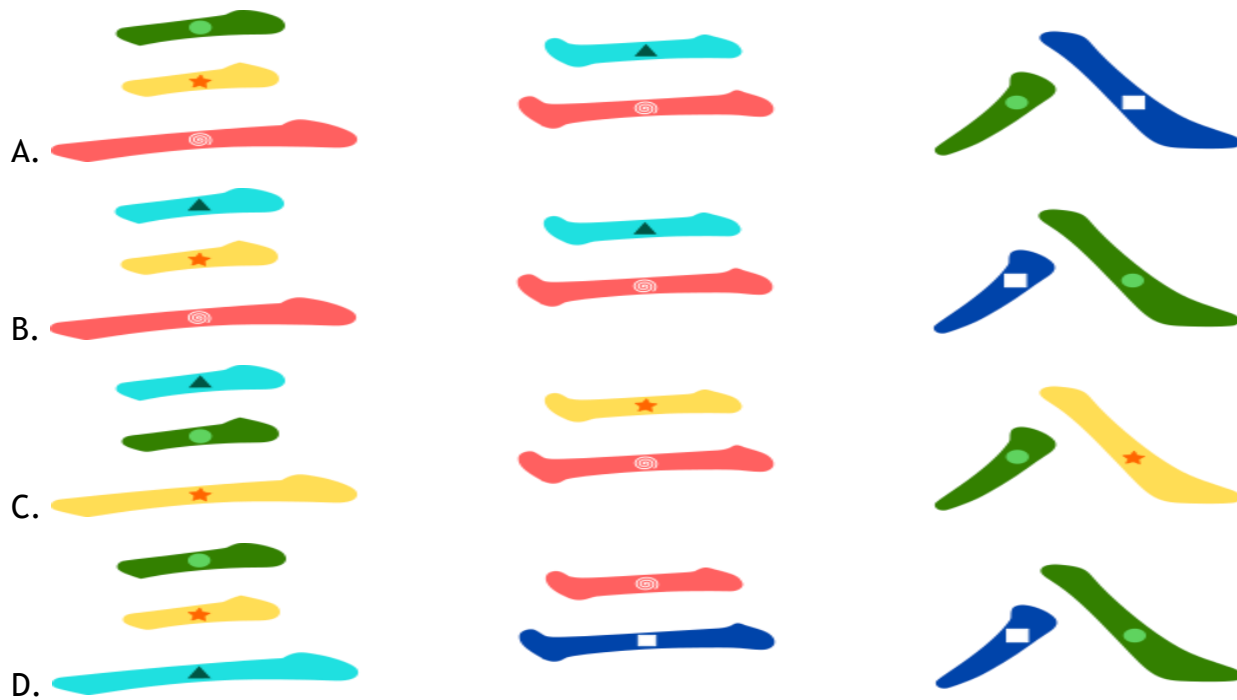


С помощью этой схемы нижеприведённые китайские символы можно представить следующим образом:

символ 川, расположение слева-по центру-справа	символ 儿, расположение слева-справа	символ 吕, расположение сверху-снизу

Вопрос




Какой вариант правильно отображает все следующие символы: Ξ , Ξ , \wedge ?







Ответ



Объяснение решения

Символ Ξ имеет структуру "сверху-по центру-снизу". Таким образом, верхняя часть имеет голубой символ , средняя жёлтый  и нижняя розовый .

Символ Ξ имеет структуру "сверху-снизу". Таким образом, верхняя часть имеет голубой символ  и нижняя розовый .

Символ \wedge имеет структуру "слева-справа". Таким образом, левая часть имеет синий символ  и правая зелёный .

Рассмотрим другие варианты:

A. Символ \equiv представлен правильно, а символы Ξ и \wedge нет.

C. Все символы представлены неправильно.

D. Символ \wedge представлен правильно, а символы \equiv и Ξ нет.

Это информатика, потому что

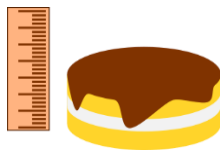
Данные могут принимать различные формы, например, быть представлены в виде изображения, текста или цифр. В этой задаче мы ищем последовательность изображений, которые помогают в решении проблемы. Идентифицируя узор, мы можем делать прогнозы, создавать правила и решать более общие проблемы.

У китайских иероглифов (кандзи) двумерная структура, которая состоит из частей, таких как «строительные блоки». Эти блоки, как правило, имеют форму слева направо, сверху вниз, образуя двумерный символ. Для большей информации смотрите: <http://www.littlechinesechannel.com/> или http://www.study-languages-online.com/ru/jp/kanji_01.html.

11. Торты и соседи

Автор: Крис Коолсает (Бельгия)

Пятничным утром каждая из трёх соседок – Анна, Белла и Валя, заказали у одного и того же пекаря торт для субботнего праздника. Все они заказали торт одного вида, высотой 3 см.



Однако после этого каждая из них позвонила пекарю и изменила свой заказ. Каждый раз пекарь записывал высоту торта согласно новому заказу, а данные по старому заказу выкидывал. Торты выпекаются ранним субботним утром.

Анна, первый звонок: Пожалуйста, сделай мой торт на 1 см толще по сравнению с моим предыдущим заказом.

Анна, второй звонок: Всё-таки сделай мой торт таким же толстым, как и торт Беллы.

Белла, первый звонок: Пожалуйста, сделай мой торт на 1 см тоньше по сравнению с моим предыдущим заказом.

Белла, второй звонок: Пожалуйста, сделай мой торт на 1 см тоньше по сравнению с моим предыдущим заказом.

Валя, первый звонок: Пожалуйста, сделай мой торт на 1 см толще по сравнению с заказом Анны.

Валя, второй звонок: Пожалуйста, сделай мой торт на 1 см толще по сравнению с моим предыдущим заказом.

Важно: Неизвестно, во сколько были сделаны звонки пекарю, за исключением того, что второй звонок каждой соседки был сделан после её первого звонка.

Вопрос

Какое из приведённых ниже утверждений будет верным в субботу в любом случае?

- A. Торты Анны и Беллы будут одинаковой толщины.
- B. Торт Беллы будет по меньшей мере на 1 см тоньше, чем торт Вали.
- C. Торт Вали будет точно на 2 см толще, чем торт Анны.
- D. Все три торта будут по меньшей мере 4 см толщиной.

Ответ

B. Торт Беллы будет по меньшей мере на 1 см тоньше, чем торт Вали.

Объяснение решения

Вариант A неверный: если Белла сделает все свои звонки после звонков Анны, то у Анны будет торт 3 см, а у Беллы – 1 см. По той же причине ответ D не является верным.

Вариант C неверный: если сначала Валя сделает все свои звонки, а затем Белла, а затем Анна, торт Вали будет 5 см в высоту, торт Беллы 1 см и Анны 1 см.

Если посмотреть на заказ Беллы, то высота её пирога была записана пекарем сначала 3 см, потом 2 см и в конце 1 см. Возможные варианты для пирога Анны, это – 3 см, 4 см, 1 см.

Это означает, что в субботу торт Вали будет иметь высоту 4, 5 или 6 см. Торт Беллы всегда будет 1 см в высоту. Таким образом, ответ B правильный.

Это информатика, потому что

Компьютер может делать много разных вещей одновременно, даже в одном приложении.

Например, в текстовом редакторе проверка орфографии работает во время набора текста.

Когда разные задачи выполняются одновременно, не всегда легко предсказать, каким будет конечный результат. Это связано с тем, что трудно угадать, когда именно каждая часть будет выполнена.

Написание программ, которые выполняют одновременные задачи, называют конкурентным программированием. Программисты должны знать о проблемах, связанных с этим типом программирования, и писать свои программы таким образом, чтобы они были уверены, что определенные действия всегда будут выполняться в одном и том же порядке. Например, программисты, которые создают программное обеспечение для сетевого принтера, должны убедиться, что два документа, которые отправляются на принтер одновременно, будут печататься один за другим. Программное обеспечение должно гарантировать, что страницы не будут перепутаны или, что ещё хуже, часть строк из первого документа напечатается следом за строками из второго документа.




В настоящее время компьютеры специально созданы для одновременного выполнения множества задач. Конкурентное программирование и связанные с ним методы позволяют избежать неправильного выполнения команд. В последнее время конкурентное программирование всё чаще входит в учебную программу по информатике.

12. Ранголи

Автор: Сонали Гогате (Индия)

Ранголи – это форма искусства, в котором орнамент выкладывается на полу при помощи цветных материалов.

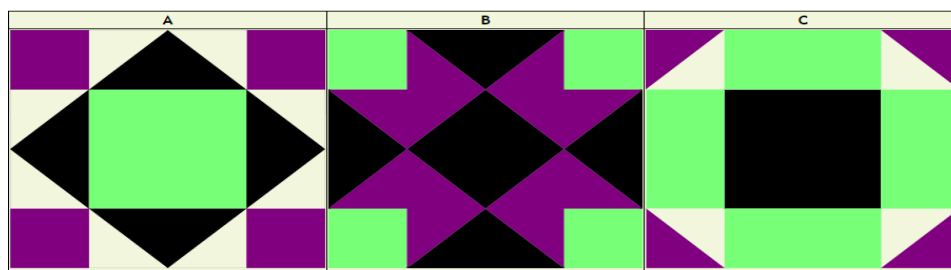
У Инны три вида плитки: 8 фиолетовых треугольников, 4 зеленых квадрата и 6 черных треугольников. Каждый вид плитки только одного размера.

Плитка			
Кол-во	8	4	6

Инна хочет создать ранголи только из этих плиток. Она не обязательно должна использовать все плитки или покрыть весь пол плитками.

Вопрос

Какой ранголи Инна сможет создать?



- A. Орнамент А
- B. Орнамент В
- C. Орнамент С
- D. Все три орнамента

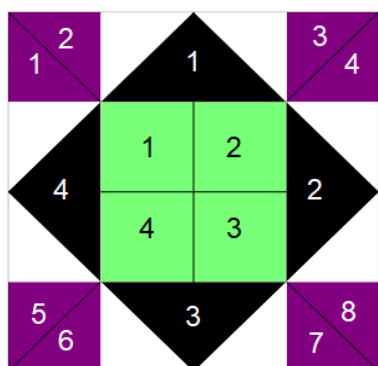
Ответ

- A. Орнамент А

Объяснение решения

Мы знаем какие плитки есть у Инны. Нам нужно выяснить, как можно их разместить для формирования заданных орнаментов ранголи и подсчитать их количество. Изображение ниже иллюстрирует один из способов, которым орнамент А может быть разрезан на плитки.

Цифры показывают итоговое количество каждого вида плиток.



В следующей таблице перечислено количество плиток необходимое для каждого орнамента. В последнем столбце указано, достаточно ли у Инны плиток для создания данного орнамента.

Орнамент	Количество плиток	Может ли Инна сделать такой орнамент?
Орнамент А	8 фиолетовых треугольников, 4 зеленых квадрата, 4 черных треугольника	Да, у неё достаточное количество плиток.
Орнамент В	12 фиолетовых треугольников, 4 зеленых квадрата, 6 черных треугольников	Нет, у неё есть только 8 фиолетовых треугольников.
Орнамент С	4 фиолетовых треугольника, 8 зеленых квадрата, 4 черных треугольника	Нет, у неё есть только 4 зелёных квадрата.

Таким образом, единственный возможный орнамент – это А.

Это информатика, потому что

Чтобы решить эту задачу, нам нужно найти плитки в каждом орнаменте и отследить их количество. Эта задача включает в себя декомпозицию и сопоставление с образцом.

В вычислениях сопоставление с образцом очень важно. Когда мы ищем слово в документе, файл на диске компьютера или что-то в Интернете, сравнение с образцом выполняется в фоновом режиме. Например, данный образец может быть первой частью слова, а поисковая система или файловый менеджер может искать все слова, имена которых начинаются с этого набора букв.

13. Копирование текста

Авторы: Лидия Феклистова, Ахто Труу (Эстония)

При копировании текста из одного документа в другой можно воспользоваться несколькими способами:



– текст вставляется в новый документ с сохранением стиля, что был в первоначальном документе;



– текст вставляется в новый документ; стиль скопированного текста объединяется с тем, что уже используется в новом документе;



– текст вставляется в новый документ простым текстом, не учитывая стиль первоначального документа.

Бобр пишет реферат и в качестве источника информации использует статью из Википедии:

Hunt ehk **hallhunt** ehk **susi** (*Canis lupus*) on põhjapoolkeral elutsev kiskjaliste seltsi koerlaste sugukonda kuuluv loomaliik. Arvatakse, et hunte elab maailmas kokku umbes 200 000 isendit.

Hunti kutsutakse **metsa sanitariks**, sest ta kütib haigeid või vigaseid loomi ning piirab näriliste ja sõraliste arvukust.

Hunt on üks koera esivanemaid. Koeratõugudest peetakse hundiga välimuselt sarnasemaks saksa lambakoeri ja teisi samalaadseid tõuge.









Huntidel on oluline osa eri rahvaste folklooris. Sealhulgas Eestis. 2018. aastal kuulutati hunt Eesti rahvusloomaks.



Бобр хочет вставить в свой реферат второй и третий абзацы статьи таким образом, чтобы во втором абзаце сохранился выделенный толстым шрифтом текст, а приведённые в третьем абзаце гиперссылки исчезли.

Вопрос

Как он должен вставить в свой реферат эти два абзаца?

- A. Второй абзац с помощью , а третий с помощью 
- B. Второй абзац с помощью , а третий с помощью 
- C. Второй абзац с помощью , а третий с помощью 
- D. Второй абзац с помощью A, а третий с помощью 

Ответ

- B. Второй абзац с помощью , а третий с помощью 

Объяснение решения

Покажем, что при использовании других вариантов желаемый стиль теряется.

Вариант А – во втором абзаце сохранится выделенный толстым шрифтом текст, но в третьем гиперссылки не исчезнут. Тоже действует и в случае варианта С.

В варианте D второй абзац будет вставлен в виде простого текста, то есть будут потеряны все стили; толстый шрифт не будет применён к тексту. В третьем абзаце стиль

скопированного текста будет объединён с тем, что уже используется в документе, но гиперссылки не исчезнут.

Это информатика, потому что

Обработка информации является частью науки о вычислительных машинах. Для упрощения обработки можно копировать повторяющиеся части. Затем при вставлении информации в новое место в нужном стиле не надо проделывать дополнительную работу по обработке данных.

14. Наблюдение

Автор: Михаэль Вейгенд (Германия)

Дигитальная камера делает фотографию с городской площадки через каждые 10 секунд.



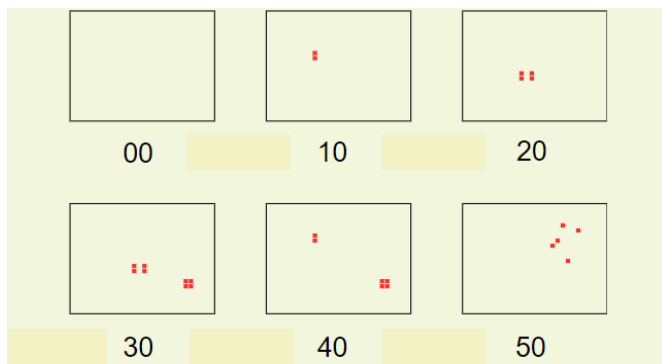
Компьютерная программа сравнивает каждую новую фотографию с предыдущей фотографией и образует "картину различий". На этой картине в каждом месте, где новая фотография отличается от предыдущей, изображен маленький красный квадратик.

Например, внизу рядом с двумя фотографиями располагается их "картина различий".



Если "картина различий" полностью белая, то различий между соответствующими двумя фотографиями нет.

Далее показана последовательность из пяти "картин различий", которые покрывают промежуток времени в 50 секунд. В этот промежуток времени произошло пять событий.



- A. Томас встречается с Тийной.
- B. Кто-то открывает дверь ратуши.
- C. Томас и Тийна идут за ручку.
- D. Поднимается ветер.
- E. Кто-то закрывает дверь ратуши.

Вопрос

В каком порядке происходили эти события?

Напиши пятизначный ответ (только буквы А-Е).

Ответ

Возможны два правильных варианта ответа: BACED и EACBD.

Объяснение решения

Между секундами 0 и 10: кто-то открывает дверь ратуши. 10 секунда – на "картине различий" на месте двери имеются красные квадраты, указывающие на изменение, которое может быть вызвано открытием двери (или закрытием двери).

Между секундами 10 и 20: на двадцатой секунде красные квадраты посередине – Томас встречается с Тийной. Они приблизились друг к другу, один слева, другой справа, поэтому их не было видно на 10й секунде.

Между секундами 20 и 30: Томас и Тийна идут за ручку. Томас и Тийна настолько близко друг к другу, что изображение с камеры отображает их как одно пятно.

Между секундами 30 и 40: кто-то закрывает (или открывает) дверь ратуши, а Томас и Тийна продолжают идти, уходя прочь.

Между секундами 40 и 50: поднимается ветер. На последнем изображении красные квадраты расположены в области кроны дерева. Это указывает на изменения (например, движущиеся листья), которые могут быть вызваны ветром.

Обратите внимание: события В и Е могут быть поменяны местами.

Это информатика, потому что

Автоматическая обработка и анализ изображений являются важными функциями в системах безопасности общественных мест, таких как аэропорты и вокзалы. Они могут быть использованы для обнаружения злоумышленников в отдаленных районах или для идентификации подозреваемого в ходе расследования. Тем не менее, постоянное просматривание общественных мест, может привести к потере конфиденциальности.

Пример иллюстрирует, что можно получать информацию с изображений веб-камеры, используя компьютерную программу. Задача анализа изображения может быть такой же простой, как считывание QR-кодов или штрих-кодов с продуктов супермаркета, или такой сложной, как оценка пола и возрастной группы человека по изображению его лица.

15. Браузер

Авторы: Лидия Феклистова, Ахто Труу (Эстония)

Дима находится в гостях у друга и решил воспользоваться его компьютером, чтобы почитать новости в Интернете. На рабочем столе имеются иконки различных программ. Дима сомневается, какая из них является иконкой браузера. Помогите ему!

Вопрос

Что из нижеприведённого является иконками браузеров?


Выбери все правильные ответы!





Ответ





Объяснение решения


A.  – Файловый менеджер (компьютерная программа для работы с файлами). Позволяет выполнять создание файлов, просмотр, редактирование, переименование, перемещение, копирование, назначать права доступа к файлу и многое другое.


B.  – Веб-браузер Google Chrome, самый популярный браузер в мире. Выпущен в 2008 году.


C.  – Adobe Acrobat Reader – бесплатное приложение международного стандарта для просматривания, подписания, комментирования PDF документов.

D.  – Веб-браузер Safari, входящий в состав macOS и iOS. Был выпущен в 2003 году, после того как подошёл к концу договор о разработке Internet Explorer для Mac устройств.

E.  – Веб-браузер Microsoft Edge, выпущенный в 2015 году изначально для Windows 10 и XboxOne, позднее для Android, iOS и macOS.

F.  – Skype, программное обеспечение для общения через Интернет посредством текста, голосового или видеозвонка. Впервые выпущен в 2003 году. Над созданием Skype работали программисты из Эстонии: Ахти Хейнла (эст. *Ahti Heinla*), Прийт Казесалу (эст. *Priit Kasesalu*), Яан Таллинн (эст. *Jaan Tallinn*) и Тойво Аннус (эст. *Toivo Annus*).

G.  – Файловый менеджер Finder в составе операционной системы macOS.

H.  – Веб-браузер Mozilla Firefox, изначально выпущенный в 2002 году под кодовым названием "Phoenix" ("Феникс"). Позже был переименован в "Firebird" ("Жар-птица") и потом получил название Firefox ("Огненная лиса").

В [Интернете](#) можно найти статистику популярности веб-браузеров.

Это информатика, потому что

Веб играет большую роль в науке о вычислительных машинах. Пользоваться вебом мы можем с помощью веб-браузеров, созданных программистами для более удобного использования веб-ресурсов.

3.3 Juuniorid (9.–10. klass)

1. Юбилей WWW

Задание 1 из раздела Бенджамины.

2. Фотографирование в школе

Задание 3 из раздела Бенджамины.

3. Покупка компьютера

Авторы: Лидия Феклистова, Ахто Труу (Эстония)

Денис хочет купить себе новый компьютер. Он обратил внимание на специальное предложение для компьютера со следующей конфигурацией:

Intel Core i3-9100, 8GB DDR4, Dell 21.5" FHD, 256GB PCIe SSD, Intel UHD 630, RJ45, SD/MMC/MS, 2xUSB 2.0, 2xUSB 3.1, VGA, HDMI, Ubuntu 19.04

Вопрос

Какие компоненты входят в этот комплект? Выбери все правильные ответы!

- A. Процессор
- B. Монитор
- C. Жесткий диск
- D. Сетевая плата
- E. Считыватель карты памяти
- F. Операционная система

Ответ

A. Процессор, B. Монитор, C. Жесткий диск, D. Сетевая плата, E. Считыватель карты памяти, F. Операционная система

Объяснение решения

Далее приведено описание каждого из компонентов, указанных в специальном предложении:

- **Core i3-9100** – процессор 2019 года с хорошей производительностью и относительно невысокой ценой;
- **8 GB DDR4** – *Double Data Rate 4* – DDR4 подразумевает четвёртое поколение оперативной памяти с повышенной скоростью передачи информации и небольшими энергетическими затратами. В массовом производстве находится с 2014 года;
- **Dell 21.5" FHD** – *Dell 21.5" Full High Definition* – монитор. Очень хорошая информация о том, что нужно учесть при покупке монитора, доступна на странице [Digitark](#);
- **256GB PCIe SSD** – *Peripheral Component Interconnect Express Solid-State Driver* – жёсткий диск, работающий намного быстрее, чем традиционный HDD;
- **Intel UHD 630** – *Intel Ultra-High Definition 630* – графическая карта (видеокарта);
- **RJ45** – *Registered Jack 45* – "зарегистрированный разъём", в системах передачи данных наиболее распространённые: RJ12, RJ11 – телефонный, RJ45 – компьютерный. Часто можно услышать другое название данного разъёма 8p8c, что означает 8 позиций для 8 контактов. Последовательность контактов имеет важное значение. Много информации о том как правильно собрать данный интернет разъём можно найти в [Интернете](#);
- **SD/MMC/MS** – *Secure Digital/Multimedia Card/Memory Stick* – адаптеры для подключения карт памяти;
- **2xUSB 2.0, 2xUSB 3.1** – *2xUniversal Serial Bus 2.0, 2xUniversal Serial Bus 3.1* – интерфейс для подключения периферийных устройств к компьютеру. USB 3.1 обеспечивает более быструю передачу данных. Порт USB 3.1 может быть использован для подключения устройств, использующих стандарты USB 2.0, скорость в таком случае будет соответствовать стандартам 2.0. Также с устройствами USB 3.1, подключёнными к порту USB 2.0, скорость будет соответствовать порту USB 2.0;
- **VGA** – *Video Graphics Array* – видеоадаптер, использующий аналоговый сигнал для передачи цветовой информации;

- **HDMI** – *High-Definition Multimedia Interface* – разъём, который в отличие от VGA, передаёт цифровые видеоданные высокого разрешения, а также поддерживает передачу многоканальных цифровых аудиосигналов;
- **Ubuntu 19.04** – операционная система.

Это информатика, потому что

Мы бы не могли использовать компьютер без технического обеспечения – компонентов, из которых он состоит. При выборе компьютера, необходимо внимательно посмотреть, из чего он состоит, могут ли его компоненты работать вместе и подумать, для чего он будет использован. При покупке компьютера надо обращать внимание на то, смогут ли компоненты выполнять необходимые требования.

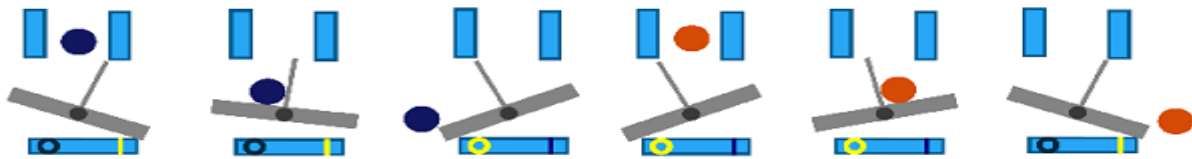
4. Счётчик

Автор: Вилфрид Бауманн (Австрия)

В машине есть четыре рычага, которые поворачиваются в одну или в другую сторону.

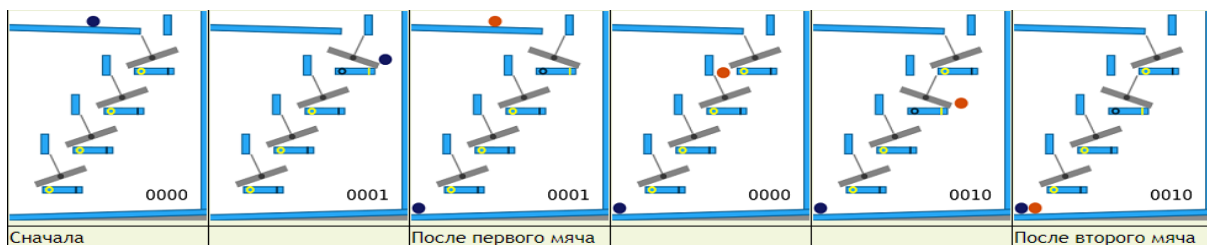
- Рычаг повернут налево = 0
- Рычаг повернут направо = 1

Если на рычаг падает мячик, то положение рычага меняется, и мячик катится дальше:



На следующей картинке изображено падение двух первых мячиков на рычаги:

Положение всех рычагов на первой картинке равно 0, поэтому счётчик показывает 0000.



Вопрос

Что покажет счётчик после того, как на рычаги упадет 5 мячиков? Напиши четырехзначный ответ (только числа 0 и 1).

Ответ

0101

Объяснение решения

Полная последовательность для пяти мячиков: 0001, 0010, 0011, 0100, 0101.

Это механический двоичный счётчик. Начальная позиция 0000.

Первый мяч поворачивает первый рычаг направо. Счётчик показывает 0001.

Второй мяч поворачивает первый рычаг налево и второй рычаг направо. Счётчик показывает 0010.

Третий мяч поворачивает первый рычаг направо и скатывается. Второй рычаг остаётся повернутым направо. Счётчик показывает 0011.

Четвёртый мяч поворачивает первый рычаг налево, второй рычаг налево и третий рычаг направо. Счётчик показывает 0100.

Пятый мяч поворачивает первый рычаг вправо и скатывается. Второй рычаг остаётся повернутым налево и третий рычаг остаётся повернутым направо. Счётчик показывает 0101.

На решение можно посмотреть и с другой стороны: мяч падает на каждый последующий рычаг до тех пор, пока не окажется на рычаге, находящемся в позиции 0. После этого рычаг поворачивается направо, и мяч скатывается окончательно.

Это информатика, потому что

Когда мы ограничены счётом только с двумя цифрами, 0 и 1, мы используем систему счисления с основанием 2. Эта система называется бинарной или двоичной. Двоичный код используется компьютерами, потому что каждый переключатель внутри него может быть только включен (1) или выключен (0). Современные компьютеры имеют много переключателей, которые контролируют всю информацию, которую использует компьютер.

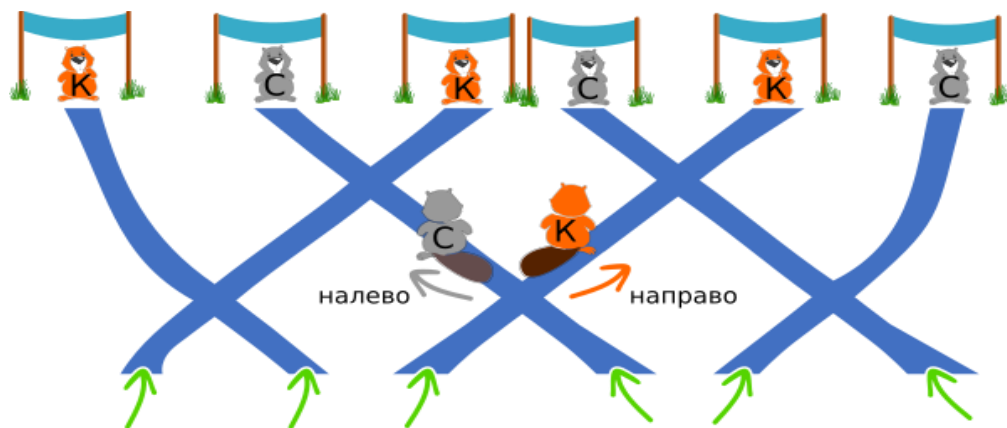
Простой в сборке механический двоичный счётчик, используемый в этом задании, имеет 4 цифры, поэтому он может считать до 15. Ниже приведена таблица всех этих чисел:

Двоичное число	Десятичное число
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	10
1011	11
1100	12
1101	13
1110	14
1111	15

5. Сеть туннелей для бобров

Автор: Нол Премасатиан (Таиланд)

Сеть туннелей состоит из ходов, по которым бобры могут передвигаться. В сети есть шесть входов и шесть выходов:



Есть два вида бобров: серые и коричневые. Если на перекрестке встречаются два бобра разного цвета, то коричневый идет направо, а серый - налево.

Шесть бобров заходят одновременно в сеть туннелей, с каждого входа по одному бобру. Бобры выходят из сети туннелей в следующей последовательности - КСККС (как показано на рисунке).

Вопрос

В какой последовательности зашли бобры в сеть туннелей? Напиши шестизначный ответ (только буквы С и К).

Ответ

ККСКСС или ККССС

Объяснение решения

Чтобы на крайнем левом выходе было К, то есть коричневый бобр, на левом перекрёстке должны встретиться коричневые бобры. Любая другая комбинация даст слева С.

Чтобы на крайнем правом выходе было С, то есть серый бобр, на правом перекрёстке должны встретиться серые бобры. Любая другая комбинация даст справа К.

По середине останется проверить две возможные комбинации: КС или СК и убедиться, что оба варианта верны.

Это информатика, потому что

Данное задание о том, как бобры движутся по сети в соответствии с заданными правилами. Это соответствует потоку данных через компьютерные сети и называется IP-маршрутизацией. IP-маршрутизация – это процесс перемещения пакетов данных между различными сетями. По умолчанию, две разные IP-сети не могут общаться друг с другом. Им нужно посредническое устройство, которое может перенаправлять пакеты между ними - маршрутизатор. Интерфейсы маршрутизаторов связаны с разными сетями. Эта связь хранится в таблице маршрутизации. Маршрутизаторы используют её для принятия решения о перенаправлении пакета.

Ещё один пример, который ассоциируется с данным заданием, основывается на принципе сортировочной сети. Последняя изменяет значения, если они находятся не в желаемом порядке. Такая система состоит из компаратора (электронного устройства для сравнения) и проволок. Каждая проволока несёт одно значение. Когда значения двух проволок встречаются в компараторе, то их заменяют только в том случае, если верхнее значение отличается от значения на нижней проволоке.

6. Склады

Автор: Илья Посов (Россия)

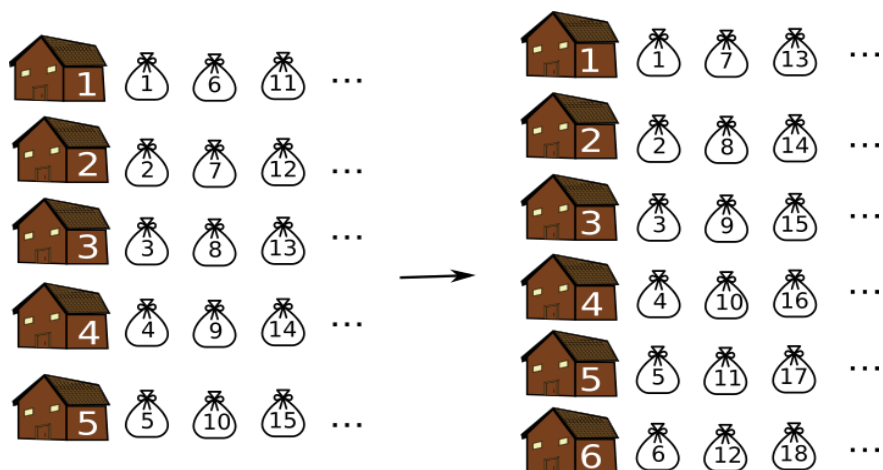
42 ежа хранят свои вещи на 5 складах. Первый еж хранит свои вещи на первом складе, второй еж – на втором складе, ..., шестой еж – на первом складе и так далее.

Однажды ежи построили шестой склад и дали ему номер 6. Ежи решили переставить вещи так, чтобы система распределение стала снова легкой: первый еж хранил бы вещи на первом складе, второй еж – на втором складе, ..., седьмой еж – на первом складе и так далее.

Вопрос

Сколько ежам не придется переносить свои вещи с одного склада на другой?

Например, вещи первого ежа всегда находятся на первом складе. А вот шестой еж должен перенести свои вещи с первого склада на шестой.



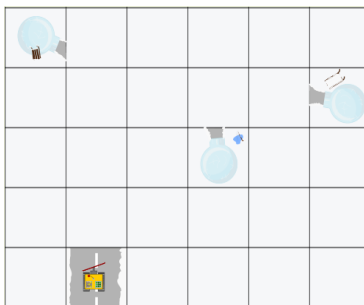
Это информатика, потому что

Это задача распределенного хранения ресурсов: для их извлечения можно использовать стратегию, называемую хешированием. В этой функции каждый ресурс связан с хеш-ключом, который может быть эффективно вычислен; ресурсы затем извлекаются путем вычисления хеш-ключа и поиска только среди тех, у кого такие же ключи хеширования. Например, если ресурсами являются имена людей, в стратегии хеширования может использоваться первая буква имени в качестве ключа хеширования: если нужно получить человека с именем "Марта", ключ хеша будет "М". В задаче хеш-ключ вычисляется с помощью функции MOD (остаток от деления). При увеличении количества хранилищ ключ поменялся бы. Хорошие хеш-функции имеют свойство не изменять все ключи, когда что-то в структуре хранилища данных поменялось. Последнее известно как свойство [согласованного хеширования](#).

7. Спасательная операция

Автор: Йиржи Ваничек (Чехия)

После снежной метели три человека хотят снова соединить свои иглу с общей трассой. Для чистки дорог выдан снегоочистительный робот. По завершению работы робот должен вернуться в исходный пункт назначения.



Передвижение из одного квадратного сектора в другой занимает один час при условии, что оба квадрата без снега. Передвижение из одного квадратного сектора в другой занимает два часа, если второй квадрат надо очистить от снега. Робот поворачивается в очищенном квадрате без потери времени.

Робот не должен очищать иглу от снега; достаточно, если робот очистит от снега квадрат перед дверью в иглу, чтобы люди смогли выйти из помещения.

Вопрос

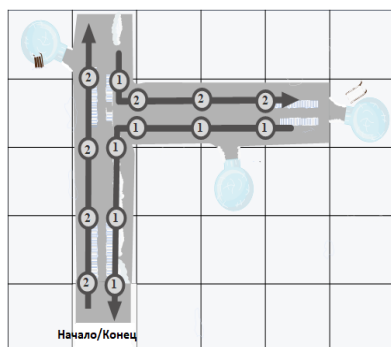
Как минимум сколько часов потребуется роботу для выполнения данной работы?

Ответ

21 час

Объяснение решения

Для повторного входа в уже очищенный квадрат потребуется меньше времени. Также очищенный путь должен быть как можно ближе к иглу. Имея в виду данные утверждения, мы можем построить путь, показанный на рисунке ниже.



Чтобы получить время работы, надо сложить все числа, приведённые на рисунке. В итоге получим 21 час. Как же проверить самый ли это быстрый вариант?

Поскольку очистка снега занимает больше времени, чем пересечение очищенного квадрата, мы хотели бы свести к минимуму количество очищаемых квадратов. На приведенном выше рисунке все очищенные пути используются повторно для сокращения времени. Другим способом сокращения времени могло бы быть использование других, более коротких путей, например, по возвращению назад, но, таковых нет.

Это информатика, потому что

Эта задача оптимизации является частным случаем поиска графа, который соединяет все заданные вершины с минимальными затратами и носит название [задачи Штейнера о минимальном дереве](#).

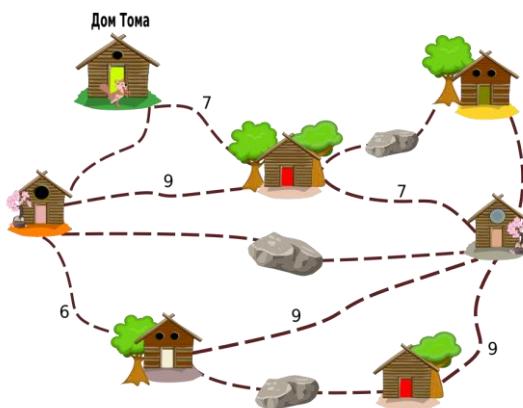
8. Визиты

Авторы: Лаура Унгуряну, Корина Винт (Румыния)

Маленький Том сидит дома и хочет навестить всех своих родственников.

Некоторые дороги, которые ведут к родственникам, платные (тарифы показаны на рисунке внизу). Если Том пройдет по платной дороге больше, чем один раз, то заново платить за эту дорогу не надо.

Некоторые дороги заблокированы камнями и по ним пройти невозможно.



Вопрос

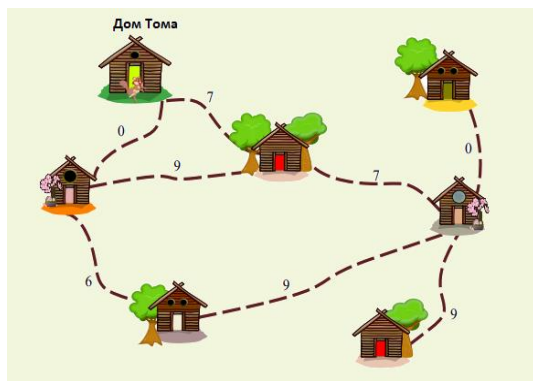
Какова наименьшая сумма денег, которую Том должен заплатить, чтобы навестить всех своих родственников?

Ответ

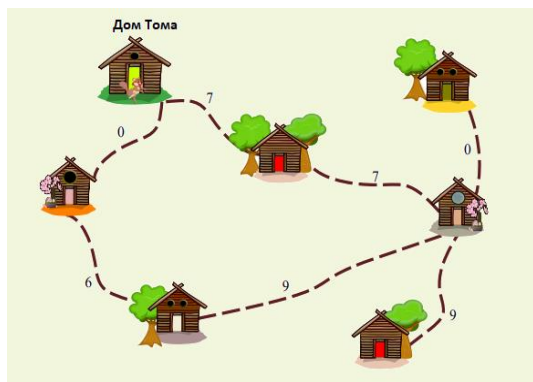
29

Объяснение решения

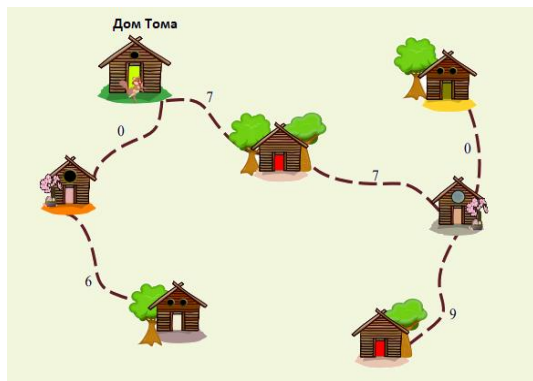
Дороги с камнями заблокированы, поэтому их мы не учитываем. Дороги, без указанного на них тарифа, бесплатные, обозначим 0-м. Учитывая последние утверждения, получим:



Как видно на картинке, от дома Тома до дома с двумя деревьями мы можем пройти по платной дороге с тарифом 7, либо через другой дом, что даст в сумме $0 + 9 = 9$. Выберем дорогу с более низким тарифом 7 и получим:

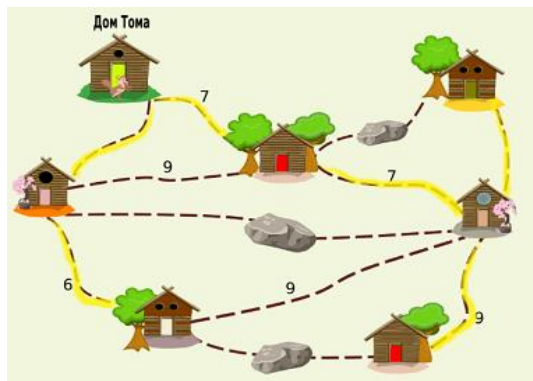


По карте видно, что мы бы могли пройти по кругу, но нам этого не надо, так как при повторном прохождении платной дороги за неё платить не надо. Без движения по кругу мы также сможем посетить каждый дом. Удалим дорогу с высокой ценой внутри круга:



На полученной карте вычислим сумму $6 + 7 + 7 + 9 = 29$.

Если взять изначальную карту, то это выглядело бы так:



Это информатика, потому что

Мы имеем дело с нахождением минимального остовного дерева. Минимальное остовное дерево – это дерево, подграф данного графа, где все вершины объединены между собой рёбрами с минимальным весом. Самый классический способ построения минимального остовного дерева – [алгоритм Краскала](#).

9. График полетов самолетов

Авторы: Випул Шах, Йогананда Йешпу (Индия)

Во избежание несчастных случаев для каждого приближающегося самолета на аэродроме выделяется коридор. Коридор – это часть воздушного пространства, куда не пускают другие самолеты.



На бобровом аэродроме запрещено выделять самолетам один и тот же коридор, если эти самолеты должны приземлиться в течение 15 минут.

Например, если один самолет приземляется в 6:10, второй – в 6:25 и третий – в 6:26, то первому и второму самолетам нельзя назначить один и тот же коридор. Однако третьему самолету можно назначить тот же коридор, что и первому; но третьему самолету нельзя назначить коридор, который использует второй самолет.

Сегодня тебя назначили диспетчером аэродрома. Твоя задача – выделить коридоры самолетам, время прибытия которых указано в таблице:

Самолет	Время
9W2400	7:00
9W1321	7:21
AI561	7:20
AI620	7:18
EK427	7:03
SG147	7:12

Вопрос

Каково наименьшее число коридоров, которое понадобится для безопасной посадки всех вышеприведенных самолетов?

Ответ

4

Объяснение решения

Запишем время прибытия самолётов в порядке возрастания:

Самолет	Время
9W2400	7:00
EK427	7:03
SG147	7:12
AI620	7:18
AI561	7:20
9W1321	7:21

Трём первым самолётам необходимо выделить три разных коридора, т.к. между их прибытием менее 15 минут. Таким образом, коридор 1 для самолёта 9W2400 в 7:00, коридор 2 для самолёта EK427 в 7:03 и коридор 3 для самолёта SG147 в 7:12. Для четвёртого самолёта AI620 назначим коридор 1, так как разница между прибытием первого и четвёртого самолёта более 15 минут. Для пятого самолёта AI561 коридор 2, разница в прибытии между вторым и пятым самолётом также более 15 минут. Шестой самолёт 9W1321 направим в коридор 4, так как разница с самолётами прибывающими в 7:12, 7:18, 7:20 менее 15 минут.

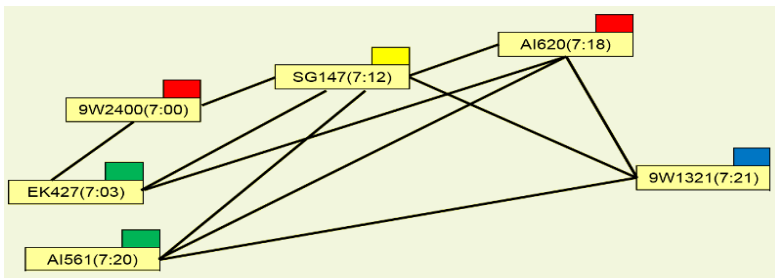
В результате получим:

Коридор	Самолет (время)
1	9W2400 (7:00), AI620 (7:18)
2	EK427 (7:03), AI561 (7:20)
3	SG147 (7:12)
4	9W1321 (7:21)

Это информатика, потому что

В данном задании есть ограничение – самолетам запрещено выделять один и тот же коридор, если эти самолеты должны приземлиться в течение 15 минут. Заданий такого типа много, например, размещая карточки с именами гостей на свадебном столе, надо учитывать, чтобы люди, которые не нравятся друг другу, не сидели за одним столом.

Один из возможных способов решения – [раскрашивание графа](#). Граф составляется из вершин (обозначающих полёт) и рёбер (обозначающих ограничение). Ребро между двумя вершинами рисуется в том случае, если действует ограничение, то есть два самолёта приземляются с разницей менее 15 минут. Затем находится минимальное количество цветов, необходимое для окрашивания вершин таким образом, чтобы две, соединённые ребром вершины, не имели один и тот же цвет. В итоге получим:



Такое решение для данной задачи не самое эффективное.

Когда в задании много элементов, хочется решить его побыстрее. В науке о вычислительных машинах есть возможности для решения подобных задач. Одним из примеров является [жадный алгоритм](#).

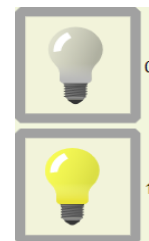
10. Сигнальные лампочки

Автор: Милан Райкович (Сербия)

Для передачи сообщения на большие расстояния бобры используют сигнальные лампочки.

Сообщения передаются при помощи следующей таблицы и правил:

A	01000001	J	01001010	S	01010011
B	01000010	K	01001011	T	01010100
C	01000011	L	01001100	U	01010101
D	01000100	M	01001101	V	01010110
E	01000101	N	01001110	W	01010111
F	01000110	O	01001111	X	01011000
G	01000111	P	01010000	Y	01011001
H	01001000	Q	01010001	Z	01011010
I	01001001	R	01010010		



Вопрос

Бобры отправили при помощи таблицы и правил следующее сообщение:

1	🔴	🟢	🔴	🔴	🟢	🔴	🔴	🔴
2	🔴	🟢	🔴	🔴	🟢	🟢	🟢	🟢
3	🔴	🟢	🔴	🟢	🔴	🟢	🔴	🟢
4	🔴	🟢	🔴	🟢	🔴	🔴	🟢	🟢
5	🔴	🟢	🔴	🔴	🔴	🟢	🔴	🟢

Какое сообщение отправили бобры?

Выбери один:

- A. HOUSE
- B. HAPPY
- C. HORSE
- D. HONEY

Ответ

- A. HOUSE

Объяснение решения

Заменим сигнальные лампочки цифрами. Полученному ряду найдём соответствующую букву из таблицы:

H	0	1	0	0	1	0	0	0
O	0	1	0	0	1	1	1	1
U	0	1	0	1	0	1	0	1
S	0	1	0	1	0	0	1	1
E	0	1	0	0	0	1	0	1

Это информатика, потому что

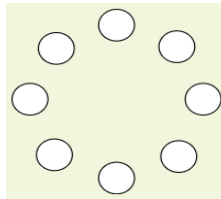
Создателем [двоичной системы](#) является Готфрид Лейбниц. Система использует только цифры 0 и 1. Это основание ко всем бинарным кодам, которые используются для написания данных, например, правил для компьютерных процессоров или для создания дигитального текста, с которым мы сталкиваемся каждый день.

Примеры из жизни, где используется бинарный код, очень легко найти. Лампы и переключатели являются хорошим примером. Коды и сообщения можно создавать и для того, чтобы общаться с другими. Например, посредством [азбуки Морзе](#).

11. Друзья за столом

Автор: Трой Васига (Канада)

Восемь друзей сидят за круглым столом лицом к центру стола.



Известно, что друзья рассажены следующим образом:

- Анна сидит напротив Даниэля.
- Генри сидит между Гретой и Элей.
- Франк не сидит рядом ни с Анной, ни с Даниэлем.
- Между Гретой и Кларой сидит один человек.
- Эля сидит рядом с Даниэлем с левой стороны.

Вопрос

Как рассажены друзья, если смотреть на друзей по часовой стрелке?

Выбери один:

- A. Анна, Бруно, Грета, Даниэль, Клара, Эля, Франк, Генри
- B. Анна, Грета, Генри, Эля, Даниэль, Бруно, Франк, Клара
- C. Анна, Клара, Франк, Бруно, Даниэль, Эля, Генри, Грета
- D. Анна, Генри, Эля, Грета, Даниэль, Франк, Бруно, Клара

Ответ

C. Анна, Клара, Франк, Бруно, Даниэль, Эля, Генри, Грета

Объяснение решения

Так как Анна сидит напротив Даниэля, то между ними должно быть три человека, поэтому вариант А не подходит – между Анной и Даниэлем 2 и 4 человека соответственно.

В тоже время, так как Эля сидит рядом с Даниэлем с левой стороны, то двигаясь по часовой стрелке, Эля должна быть сразу следом за Даниэлем. По этой причине варианты В и D не подходят, так как в варианте В Эля сидит перед Даниэлем, то есть справа от него, и в варианте D они находятся не рядом друг с другом.

Так как Генри сидит между Гретой и Элей, то варианты А и D не подходят, так как в этих вариантах рядом с Генри сидит Анна.

В варианте D Франк сидит рядом с Даниэлем, что по известным нам высказываниям является неверным.

Так как между Гретой и Кларой может сидеть только один человек, то не подходит вариант D, в котором между ними находятся три человека.

Поэтому остаётся только вариант С, который согласуется со всеми известными нам утверждениями.

Это информатика, потому что

В данном задании ключевой операцией является отрицание, которое мы используем каждый раз, когда говорим "не" или "нет". Утверждение "Франк не сидит рядом ни с Анной, ни с Даниэлем" можно записать таким образом: "Франк не сидит рядом с (Анной или Даниэлем)", что равносильно утверждению "(Франк не сидит рядом с Анной) и (Франк не сидит рядом с Даниэлем)".

В общей форме можно записать "не (А или В)" и равносильно этому "(не А) и (не В)".

Важно, что при внесении отрицания в скобки вместо изначального "или", появляется "и".

Преобразование логических выражений и их комбинирование – очень важное умение как в науке о вычислительных машинах, так и в других областях жизни.

12. Робот-художник

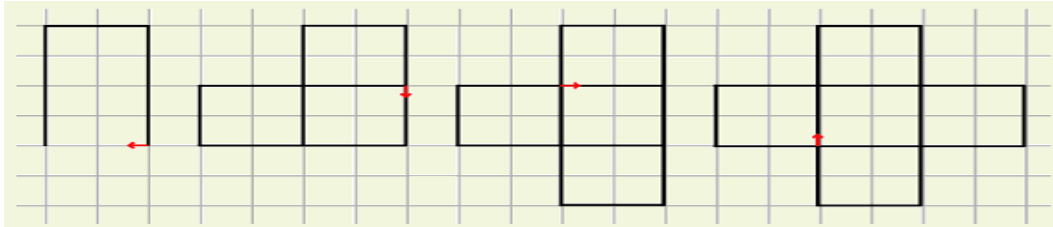
Автор: Даниэла Безакова (Словакия)

Робот двигается по сетке и чертит линию, которая образует картину. Каждую картину образует один набор из трех чисел.

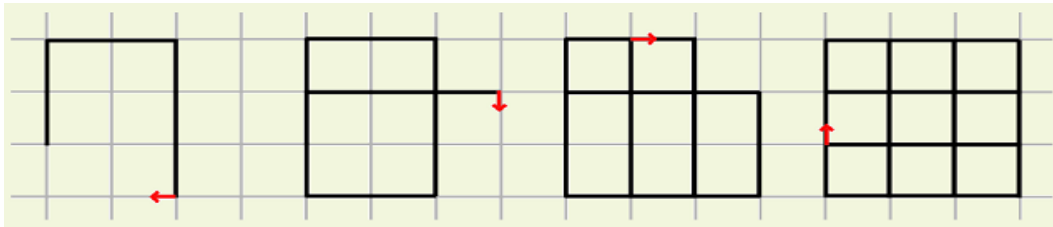
Объяснение решения

Рассмотрим какой рисунок получается.

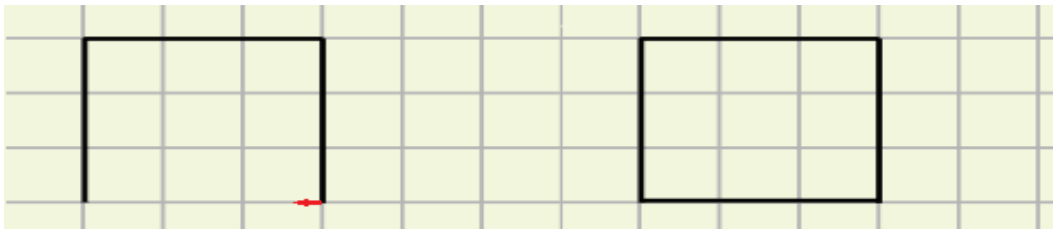
Повторив четыре раза набор из чисел 4, 2, 4, получим вариант А:



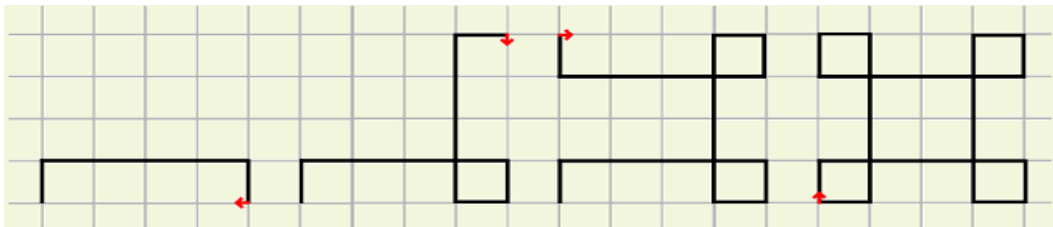
Повторив четыре раза набор из чисел 2, 2, 3, получим вариант В:



Повторив два раза набор из чисел 3, 3, 3, получим вариант С:



Повторив четыре раза набор из чисел 1, 4, 1, получим вариант D:



Это информатика, потому что

Выполнение программы – это процесс, где компьютер исполняет различные действия в соответствии с правилами, данными в программе. Каждое правило – это распоряжение сделать какое-то конкретное действие, которое подведёт решение ближе к ответу.

Прочтение и понимание распоряжений важно, чтобы понять, что происходит в процессе выполнения программы. Это очень важное умение программиста, особенно когда программа ведёт себя неправильно и надо найти, в чём заключается ошибка.

13. Копирование текста

Задание 13 из раздела Бенджамины.

14. Обработка текста в электронной таблице

Автор: Лидия Феклистова (Эстония)

Электронные таблицы можно использовать и для обработки текста:

Функция	Значение
LEFT(A;N)	из текста А возвращает N-ное количество знаков с левой стороны; например, результатом LEFT("KOBRAS";2) будет "КО"
RIGHT(A;N)	из текста А возвращает N-ное количество знаков с правой стороны; например, результатом RIGHT("KOBRAS";2) будет "АС"
MID(A;K;N)	из текста А возвращает N-ное количество знаков, начиная с позиции К; например, результатом MID("KOBRAS";3;2) будет "БР"
CONCATENATE(...)	соединяет приведенные в виде аргументов тексты в одно целое; например, результатом CONCATENATE("КО";"БР";"АС") будет "КОБРАС"

Вопрос

	A	B	C
1	OECD		
2	CODE		

Какую формулу необходимо записать в табличную ячейку A2, чтобы получить текст "CODE", если в ячейке A1 был записан текст "OECD"?

- A. =CONCATENATE(MID(A1;2;1);RIGHT(A1;1);LEFT(A1;1);MID(A1;3;1))
- B. =CONCATENATE(MID(A1;3;1);RIGHT(A1;1);LEFT(A1;1);MID(A1;2;1))
- C. =CONCATENATE(MID(A1;2;1);LEFT(A1;1);RIGHT(A1;1);MID(A1;3;1))
- D. =CONCATENATE(MID(A1;3;1);LEFT(A1;1);RIGHT(A1;1);MID(A1;2;1))

Ответ

- D. =CONCATENATE(MID(A1;3;1);LEFT(A1;1);RIGHT(A1;1);MID(A1;2;1))

Объяснение решения

Чтобы из OECD получить CODE, надо переместить третью букву в слове OECD в начало, первую букву на вторую позицию, четвёртую на третью позицию и вторую в самый конец. Для получения первой буквы C в слове OECD используем MID(A1;3;1) – функция MID возвращает одну букву начиная с третьей позиции.

Для получения второй буквы O используем LEFT(A1;1) – функция LEFT возвращает одну букву с начала слова OECD.

Для получения третьей буквы D используем RIGHT(A1;1) – функция RIGHT возвращает одну букву с конца слова OECD.

Для получения четвёртой буквы E используем MID(A1;2;1) – функция MID возвращает одну букву начиная со второй позиции.

Функция CONCATENATE() соединяет приведённые в виде аргументов тексты в одно целое. Таким образом, помещая ранее полученные буквы в виде аргументов функции, получим =CONCATENATE(MID(A1;3;1);LEFT(A1;1);RIGHT(A1;1);MID(A1;2;1)).

Это информатика, потому что

Обработка табличных данных играет свою роль в науке о вычислительных машинах. Благодаря обработке табличных данных, можно быстро работать с объёмными данными.

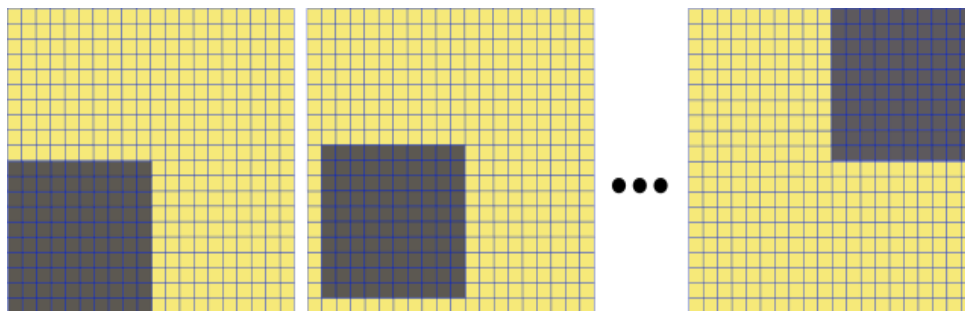
В данном задании слово составлялось посредством соединения букв. Используемые функции можно применять и в статистике. Вот некоторые примеры применения данных функций:

- [RIGHT\(A;N\)](#) – конвертирование текста в цифру (если дан текст ##678, применив функцию RIGHT(##678,3)+0, получим 678), удаление знака / в конце ссылки (=LEFT(url,LEN(B5)-(RIGHT(url)="/")))) и многое другое;
- [LEFT\(A;N\)](#) – подсчёт количества чисел, начинающихся на заданное значение (например, найти в документе количество чисел начинающихся на 25 можно, используя формулу =SUMPRODUCT(--(LEFT(range,chars)="25"))) и другое;
- [MID\(A;K;N\)](#) – чтобы получить количество часов, отработанных в промежутке между [двумя датами](#);
- [CONCATENATE\(...\)](#) – чтобы добавить "http://" в начало большого количества ссылок, соединить имя и фамилию для анализа и другое.

15. Сохранение видео

Автор: Сергей Поздняков (Россия)

Видео – это последовательность изображений, т.н. кадров, в которой каждый кадр чуть отличается от предыдущего. Самый простой способ для сохранения видео – сохранить каждый пиксель каждого кадра. Более эффективный способ сохранения видео – сохранить полностью только первый кадр, а дальше сохранять только те пиксели, которые изменяются при переходе от одного кадра к другому.



На вышеприведенном рисунке темный квадрат размером 10x10 движется из нижнего левого угла в верхний правый угол. Квадрат движется по светлому фону размером 20x20, перемещаясь за один кадр на один пиксель по горизонтали и вертикали. Таким образом, видео состоит из 11 кадров. Если сохранить данное видео простым способом, потребуется $(20 \times 20) \times 11 = 4400$ пикселей.

Вопрос

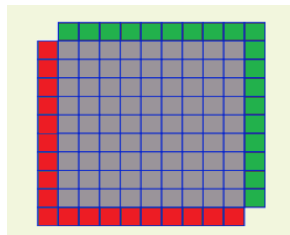
Сколько пикселей понадобится, если видео будет сохранено эффективным способом?

Ответ

780 пикселей

Объяснение решения

На первом кадре необходимо сохранить все пиксели, то есть $20 \times 20 = 400$ пикселей. На каждом последующем кадре изменяется только 38 пикселей (19 красных убирается, 19 зелёных добавляется):



За первым кадром следует ещё 10 кадров, где передвигается тёмный квадрат. Таким образом, получаем: $400 + 38 \times 10 = 780$ пикселей.

Это информатика, потому что

Сжатие данных – важная тема в информатике, особенно для видео и аудио данных. Некоторые известные форматы изображений, такие как JPEG, теряют данные при сжатии, поэтому некоторые цвета или границы искажаются.

Данное задание связано со сжатием без потерь, где информация о изображении не теряется. Складывая кадры один за другим, мы можем думать о каждом из них как о срезе трехмерного массива. Изменение в кадре происходит только в тех пикселях, которые отличаются от предыдущего кадра.

Следует иметь в виду, что в представленном алгоритме мы должны помнить не только цвет измененного пикселя, но и его координаты.

Больше про сжатие видео можно почитать на странице [Википедии](#).

3.4 Seeniorid (11.–12. klass)

1. Колосс

Автор: Свен Аллер (Эстония)

В 2019 году свой юбилей отпраздновал компьютер Колосс Марк 2, с помощью которого во время Второй мировой войны Великобритания расшифровывала засекреченные сообщения противников.

Вопрос

В каком году изобрели Колосс Марк 2?

- A. 1939
- B. 1940
- C. 1944
- D. 1949

Ответ

C. 1944

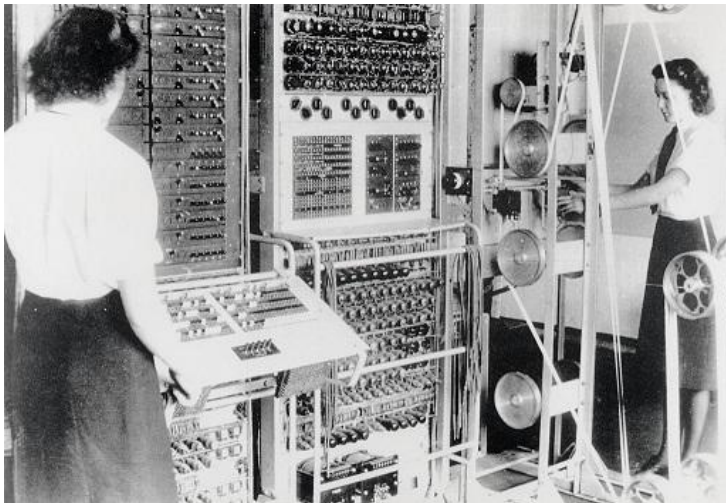
Объяснение решения

Колосс Марк 1 был изготовлен в 1943 году и его стали использовать в 1944. Колосс Марк 2 был изготовлен 1 июня 1944 года.

Это информатика, потому что

[Колосс](#) представляет собой компьютер, используемый британцами для прочтения немецких сообщений.

Колосс Марк 2 считается первым в мире программируемым, электронным и дигитальным компьютером.



Колосс Марк 2 со страницы [истории электронных компьютеров](#).

2.Фотографирование в школе

Задание 3 из раздела Бенджамини.

3. Покупка компьютера

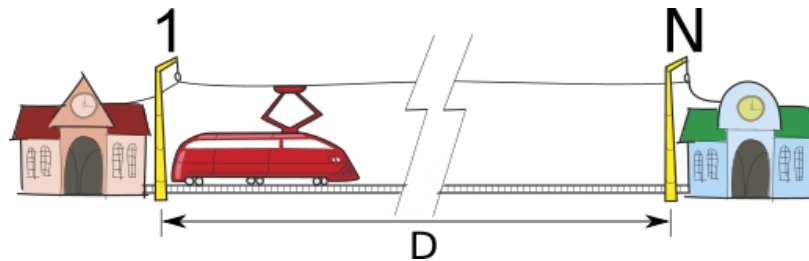
Задание 3 из раздела Юниоры.

4. Электрические столбы на железной дороге

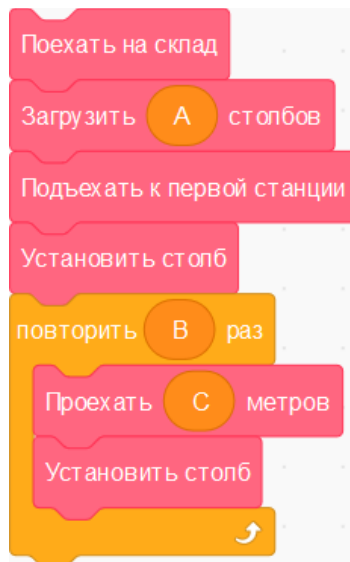
Автор: Йиржи Ваничек (Чехия)

Между двумя железнодорожными станциями решили провести электролинию.

Электрические столбы должны быть установлены через равные промежутки, первый из них должен располагаться на первой станции, а последний на второй станции.



Робот устанавливает электрические столбы в соответствии с программой:



Расстояние между станциями D метров, робот должен установить N столбов.

Вопрос

Какие значения подходят для мест, обозначенных в программе буквами А, В и С?

А	В	С
1	1	1
2	2	2
N	N	N
D	D	D
N+1	N+1	N+1
D+1	D+1	D+1
N-1	N-1	N-1
D-1	D-1	D-1
N+D	N+D	N+D
N-D	N-D	N-D
D-N	D-N	D-N
N/2	N/2	N/2
D/2	D/2	D/2
N/D	N/D	N/D
D/N	D/N	D/N
N/(D+1)	N/(D+1)	N/(D+1)
D/(N+1)	D/(N+1)	D/(N+1)
N/(D-1)	N/(D-1)	N/(D-1)
D/(N-1)	D/(N-1)	D/(N-1)

Ответ

A:N, B:N-1, C:D/(N-1)

Объяснение решения

Робот должен установить N столбов, поэтому значение А будет N – "Загрузить N столбов".

После установки первого столба останется N-1 столбов, поэтому значением для В будет – "Повторить N-1 раз".

Чтобы найти равные промежутки между столбами, надо расстояние D поделить на N-1 (количество промежутков). Таким образом, робот должен проехать D/N-1 метров, затем установить столб.

Это информатика, потому что

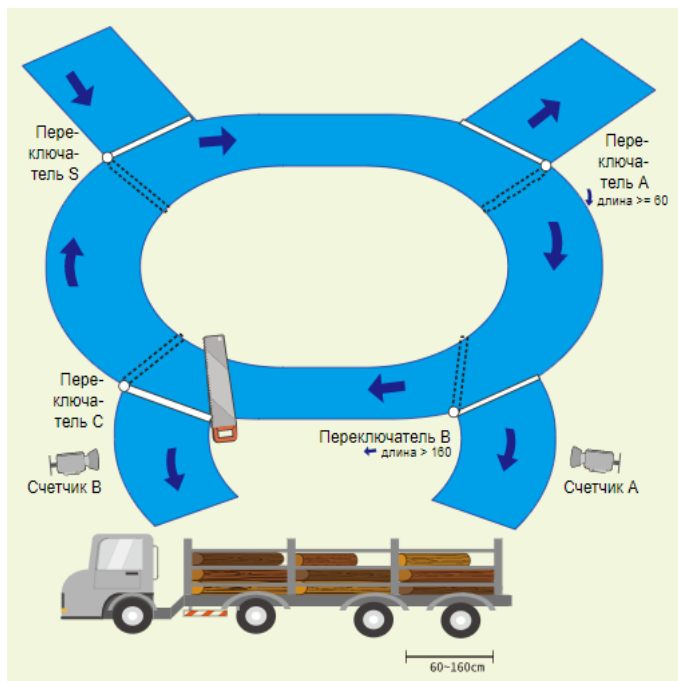
В функции, данной к заданию, используются переменные N и D. Переменная – это величина, имеющая значение. Переменные N и D получают свои значения в основной части программы, то есть задаются в самом начале.

Также в задании используется [цикл](#), многократно повторяющийся набор инструкций. Цикл вызывает один и тот же отрезок кода какое-то количество раз. В данном задании важно знать количество повторов. Робот должен установить N столбов, а количество промежутков между ними равно N-1. С последним очень легко ошибиться и забыть отнять единицу, так как количество промежутков на одно меньше количества столбов. Это очень распространённая ошибка в программировании, известная под названием [ошибка на единицу](#).

5. Лесопилка

Автор: Себастьян Комбефис (Бельгия)

Для строительства бревенчатого дома требуются бревна правильной длины. Бревна привозят разной длины, однако конвейер грузит в грузовик бревна длиной от 60 см до 160 см.



Бревна поступают на конвейер слева сверху. Конвейер состоит из следующих частей:

- переключатель S автоматически направляет бревна из двух входов на один выход;
- переключатель A пропускает бревна длиной по меньшей мере 60 см, более короткие бревна убираются с конвейера;
- переключатель B пропускает бревна длиннее 160 см, оставшиеся отправляет в грузовик;
- переключатель C включает пилу, которая распиливает бревно на две части. Первая часть бревна получается длиной 160 см и отправляется в грузовик, а вторая часть остается на конвейере для последующей обработки;
- счетчики A и B считают, сколько бревен было погружено в грузовик.

Вопрос

На конвейер поступило три бревна длиной 60 см, 140 см и 360 см.

Каковы будут показатели счетчиков, когда обработка этих бревен будет закончена?

- A. Счетчик A: 1 бревно, счетчик B: 3 бревна
- B. Счетчик A: 3 бревна, счетчик B: 1 бревно
- C. Счетчик A: 2 бревна, счетчик B: 2 бревна
- D. Счетчик A: 0 бревен, счетчик B: 4 бревна

Ответ

C. Счетчик A: 2 бревна, счетчик B: 2 бревна

Объяснение решения

Бревно длиной 60 см пройдет через переключатель A и переключатель B направит его в грузовик. Показатель на счётчике A увеличится на 1.

Бревно длиной 140 см пройдет через переключатель A и переключатель B направит его в грузовик. Показатель на счётчике A увеличится ещё на 1.

Бревно длиной 360 см пройдет через переключатель A, затем через переключатель B, после чего часть бревна длиной 160 см будет отпилена и отправлена в грузовик. Показатель на счётчике B увеличится на 1. Оставшаяся часть бревна длиной 200 см отправится на новый круг, пройдет через переключатель A, затем через переключатель B и будет распилена на две части: часть в 160 см отправится в грузовик. Показатель на счётчике B увеличится ещё

на 1. Оставшаяся часть длиной 40 см отправится на новый круг и, дойдя до переключателя А, будет убрана с конвейера.

В итоге получим 2 бревна на счётчике А и 2 бревна на счётчике В.

Это информатика, потому что

Движение брёвен по конвейеру можно представить как последовательность целых чисел. Данное задание является наглядным примером [реактивного программирования](#), где поток чисел (представленных в виде длин брёвен) обрабатывается с помощью операторов (функций, выполняющих определённое действие). Примером последних могут быть: объединить, фильтровать, преобразовать, просканировать.

Переключатель S объединяет брёвна из двух входов. Переключатели А и В фильтруют, разделяя брёвна в соответствии с их длиной. Переключатель С включает пилу, преобразовывая таким образом одно бревно в два. Счётчики осуществляют операцию сканирования, подсчитывая брёвна, погруженные в грузовик.

6. Склады

Задание 6 из раздела Юниоры.

7. Запасной путь

Авторы: Кристиан Дацко, Сузанн Дацко, Юрай Громкович (Швейцария)

Перед выходом вагоны, стоящие на запасном пути, необходимо упорядочить по номерам таким образом, чтобы самый крайний с левой стороны вагон был бы под номером 1.

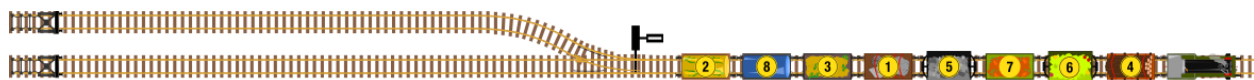
На запасном пути для маневров имеется два тупика, куда паровоз может на время загнать вагоны. Во время буксировки каждого вагона можно выбрать, в каком тупике он будет стоять. Когда все вагоны распределены по тупикам, их начинают цеплять к поезду таким образом, что в начале поезда будут вагоны, которые были в одном тупике, а в конце – вагоны, которые были в другом тупике. Всю эту процедуру будем считать одной операцией.

На нижеприведенном примере у нас имеется 4 вагона и для того, чтобы прицепить их к поезду в правильном порядке, нам необходимо выполнить две операции (это невозможно выполнить за одну операцию):



Вопрос

Если вагоны расположены в порядке 2 – 8 – 3 – 1 – 5 – 7 – 6 – 4, то каково наименьшее количество операций, чтобы, воспользовавшись вышеприведенной системой, их можно было бы прицепить в порядке 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8?

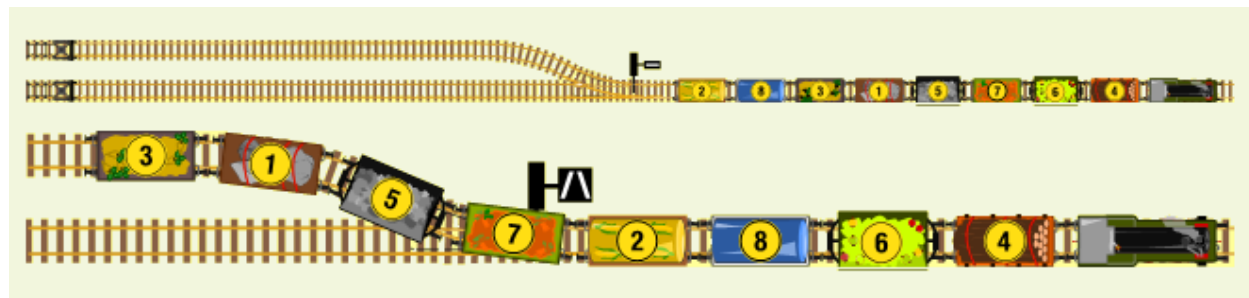


Ответ

3 операции

Объяснение решения

Конечно, есть много способов отсортировать вагоны, но один из лучших приведён ниже. Сначала направим вагоны под номерами 3, 1, 5, 7 в верхний тупик, а вагоны под номерами 2, 8, 6, 4 в нижний тупик. Цеплять вагоны к поезду будем сначала с нижнего тупика, а потом с верхнего. В результате получим:



Таким образом, в каждой паре из двух вагонов (1 и 2, 3 и 4, 5 и 6, 7 и 8), вагоны с наименьшим номером, будут располагаться отдельно от вагонов с большим номером.

Для второй операции имеет смысл отправить в верхний тупик вагоны под номерами 1, 5, 2, 6, а в нижний тупик под номерами 3, 7, 8, 4. Цеплять вагоны к поезду будем сначала с нижнего тупика, а потом с верхнего:



Сортировка пар из двух вагонов ранее не была случайной, потому что каждая пара из двух вагонов въезжала в один и тот же тупик. Кроме того, теперь вагоны с номерами от 1 до 4, а

также от 5 до 8, сортируются относительно друг друга, однако обе группы по-прежнему смешанные.

Для финальной, третьей процедуры, направим вагоны с номерами от 1 до 4 в верхний тупик, а вагоны с номерами от 5 до 8 в нижний тупик. Цеплять вагоны к поезду будем сначала с нижнего тупика, а потом с верхнего:



Сортировка двух групп не изменилась, поскольку все вагоны из группы от 1 до 4 свернули в один тупик, а все вагоны из группы от 5 до 8 свернули в другой тупик. Обе группы состоят из рассортированных вагонов, и в одной группе номера меньше, чем в другой группе.

Сортировать вагоны быстрее невозможно. Полное доказательство слишком длинное, но общая идея такова: одна операция над вагонами может изменить только порядок одного подмножества по сравнению с другим подмножеством, но не внутри каждого подмножества. Таким образом, первая процедура позволит переместить максимально два вагона, находящихся в неправильном порядке. Каждая последующая процедура удваивает количество вагонов, расположенных в неправильном порядке. Для этого задания было выбрано восемь вагонов, чтобы продемонстрировать невозможность упорядочить вагоны за две процедуры.

Это информатика, потому что

Железнодорожники во всем мире ежедневно сталкиваются с данной проблемой, потому что перестановка вагонов – это утомительная и трудоемкая работа: каждый раз, когда вагоны сцепляются и разъединяются, человеку приходится делать это вручную. Это требует времени и блокирует главную линию, особенно если некоторые вагоны необходимо оставить и закрепить на главной линии, пока локомотив доставляет вагон в тупик. Из-за этого более крупные железные дороги создали огромные сортировочные станции с множеством подъездных путей. В Швейцарии сортировочные станции существуют в Муттенце около Базеля, в Буксе, между Шпрайтенбахом и Дитиконом около Цюриха, в Денгесе около Лозанны и в Кьяссо. Эта задача предлагает только два тупика, куда поезд может на время загнать вагоны, что является проблемой для более крупных поездов.

Однако это стандартная ситуация для небольших железных дорог, особенно если это узкоколейная железная дорога, не имеющая прямого соединения с более крупной сетью железных дорог.

Чтобы эффективно сортировать эти вагоны, информатика нам помогает. В этом случае многократное решение частей проблемы делает её очень простой: метод, называемый в информатике "[разделяй и властвуй](#)". При этом сортируются сначала два вагона, затем четыре и, наконец, восемь вагонов.

Тупик для вагонов работает как [стек](#) – абстрактный тип данных, который широко используется в информатике. Разрешённые операции для стека: добавить новый элемент сверху (иначе протолкнуть, англ. *push*), удалить самый верхний элемент (иначе вытолкнуть, англ. *pop*). Также можно посмотреть на верхний элемент и проверить, пуст ли стек.

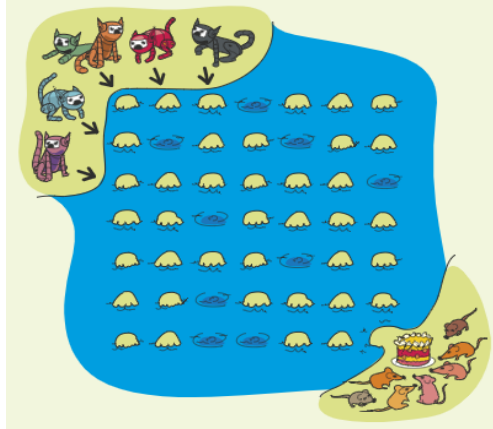
8. Мышиный праздник

Автор: Валентина Дагиене (Литва)

Шесть котов решило попасть на мышиный праздник, перепрыгивая с одного острова на другой. Кот может прыгнуть только на соседний остров (вперед, назад, влево, вправо), но не по диагонали. К тому же кот не может использовать острова, находящиеся под водой, и не может их перепрыгивать.

Коты перепрыгивают очень быстро и поэтому время прыжка можем не учитывать. Однако сам процесс перепрыгивания их утомляет настолько, что, попав на остров, они должны отдыхать одну минуту. По этой причине попавший на остров кот отдыхает здесь минуту и только после этого перепрыгивает на следующий остров, если остров пуст. Если следующий остров занят, то кот должен подождать еще минуту. Кот может перепрыгнуть на остров только в том случае, если предыдущий кот его покинул или покидает его в тот же момент.

Сначала каждый кот должен прыгнуть на любой обозначенный стрелкой остров.



Вопрос

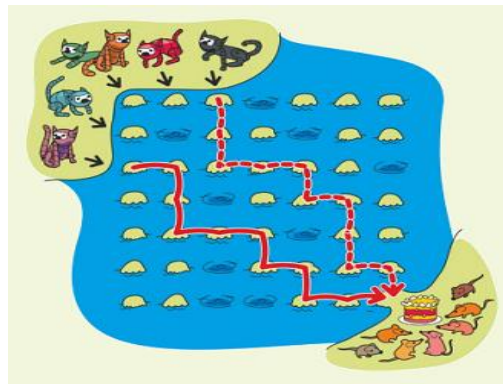
Сколько минут потребуются, чтобы все коты добрались от старта до цели самым быстрым способом?

Ответ

12

Объяснение решения

Для начала найдём самый короткий путь от каждой стрелки. Получим возможные варианты: 10, 11 и 12 минут. Важно отметить, что существует ровно два разных кратчайших пути (для двух параллельных потоков), как показано на рисунке.



Все коты начинают движение по двое, один идёт по одному из кратчайших путей, а другой по-другому, один за другим. Первые два кота доберутся до мышей за 10 минут, следующие два за 11 минут, так как в самом начале ожидают 1 минуту. Последние два кота доберутся до мышей за 12 минут.

12 минут – это наименьшее время, так как:

- если кот стартует на острове, который не является началом отмеченных на рисунке дорог, он должен пройти начало отмеченных путей позже. Поэтому сокращаем возможные заходы до двух стрелок;
- не найдётся пути короче, чем 10 минут;
- разом могут стартовать только два кота, поэтому разделив котов поровну, получим по 3 кота возле каждого старта. Прыгать все разом они не могут, поэтому двум последним котам потребуется на 2 минуты больше.

Это информатика, потому что

Учёные-информатики – специалисты по моделированию и оценке ситуаций, например, графических ситуаций. Одна из их основных задач – выяснить, какое поведение проявит математически смоделированная система, и как долго может выполняться процесс. Проблема оптимизации – это проблема поиска наилучшего решения из всех возможных решений.

В информатике важно интерпретировать реальные ситуации в форме, понятной компьютерам. Для этой задачи местоположение островов указывается координатами. Задача – найти кратчайший путь, двигаясь только по горизонтали и вертикали в системе координат, избегая подводного острова.

Мы также можем представить данное задание в виде графа, где острова – это вершины, а ребра – дороги между двумя островами, и надо найти кратчайший путь из одной точки в другую.

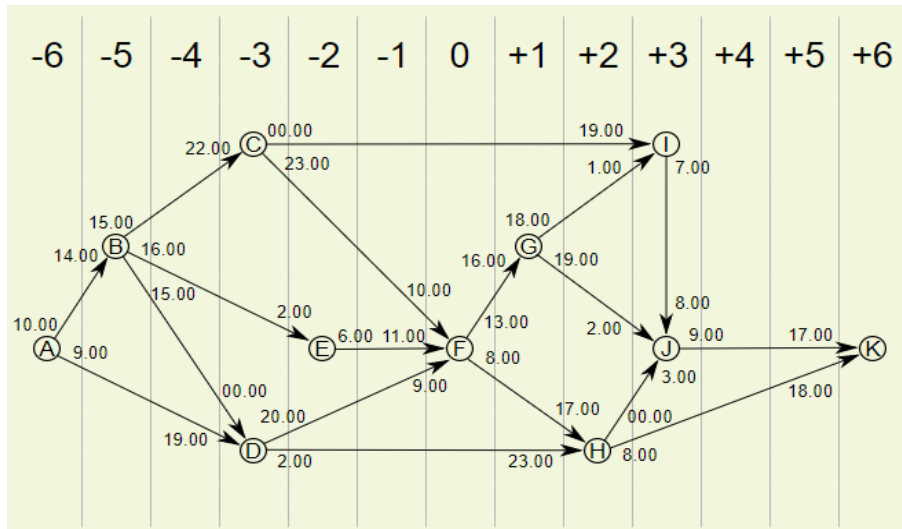
9. Воздушное сообщение

Автор: Крис Роффей (Великобритания)

Таня планирует совершить длинное путешествие. Чтобы определиться с самым быстрым путем, она нарисовала карту:

- 11 городов располагаются в тринадцати часовых поясах (часовые пояса обозначены цифрами на краю карты);
- между двумя городами в течение 24 часов совершается только один прямой рейс;
- самый короткий перелет длится 1 час, самый длинный – 16 часов;

- Таня успеет на следующий самолет только в том случае, если между посадкой предыдущего самолета и взлетом следующего самолета у нее есть один час времени;
- на карте время взлета и посадки отмечено по местному времени соответствующего аэропорта.



Например, самолет из города D в город F взлетает в 20.00 (по местному времени D) и приземляется в 9.00 (по местному времени F). Так как время в F впереди на 3 часа по сравнению со временем в D, то перелет занимает 10 часов.

Вопрос

Каково наименьшее время (в часах), которое понадобится Тане, чтобы добраться из города A в город K (учитывая как время перелета, так и время, затраченное на пересадки)?

Ответ

43

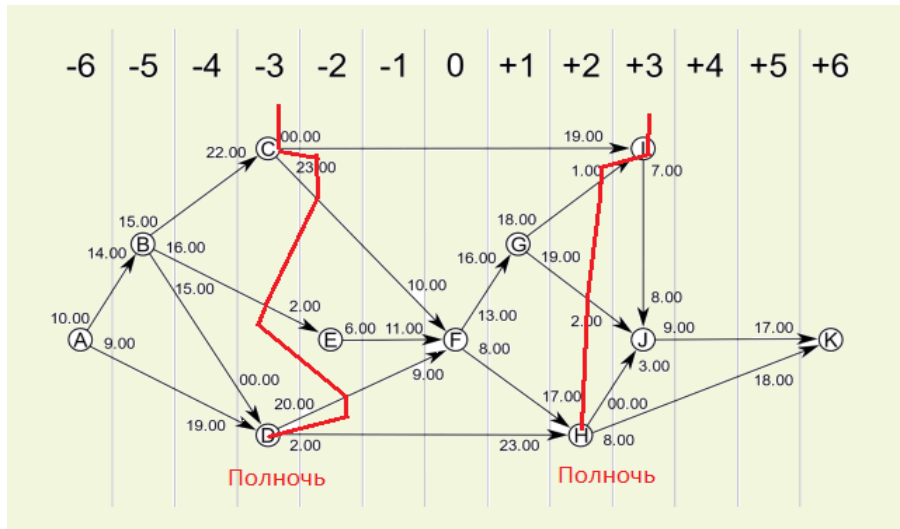
Объяснение решения

Чтобы найти самый оптимальный путь, надо изначально начать путешествие как можно позже по времени и закончить как можно раньше. То есть из A надо вылететь в 10.00 и время прибытия в K будет в 17.00. Также надо учесть, что Таня летит несколько дней и пересекает несколько часовых поясов.

Наша цель – найти решение, где не пришлось бы подсчитывать время каждого перелёта.

Для начала обозначим на рисунке полночь для каждого рейса, например, рейс из B в E занимает 7 часов, а время прибытия в 2.00. Делим линию из B в E на семь равных частей и

отнимаем две из них (там и будет 0.00 для B -> E). На рисунке видно, что красный рубеж, в любом направлении, пересекается два раза. Это означает, что, если бы рейс начинался и заканчивался по часам в одно и тоже время, продолжительность полета составила бы 48 часов.



Учитывая разницу в часовых поясах, надо от времени прибытия отнять 12. Получится, что 17.00 в К – это 5.00 в А, что по часам составляет 5 часов до времени взлета. Итак, если Таня отправляется из пункта А в 10 утра, она прибывает в пункт К через 48 минус 5 часов, что даёт в итоге 43 часа полёта и это самый быстрый маршрут.

Можно посчитать и немного по-другому: из пункта А до первой красной линии 14 часов, от красной линии до красной 24 часа, и от второй красной линии до пункта К 17 часов. Получим, $14 + 24 + 17 = 55$ часов и, учитывая часовые пояса между А и К, отнимем 12. Итого, $55 - 12 = 43$.

Это информатика, потому что

Поначалу кажется, что мы имеем обычное задание с графом и достаточно использовать имеющийся алгоритм для нахождения кратчайшего времени. Однако, учитывая наличие уникальных компонентов, стоит разработать конкретный алгоритм для этого конкретного сценария.

Уникальные аспекты этого сценария по сравнению с обычными проблемами кратчайшего маршрута:

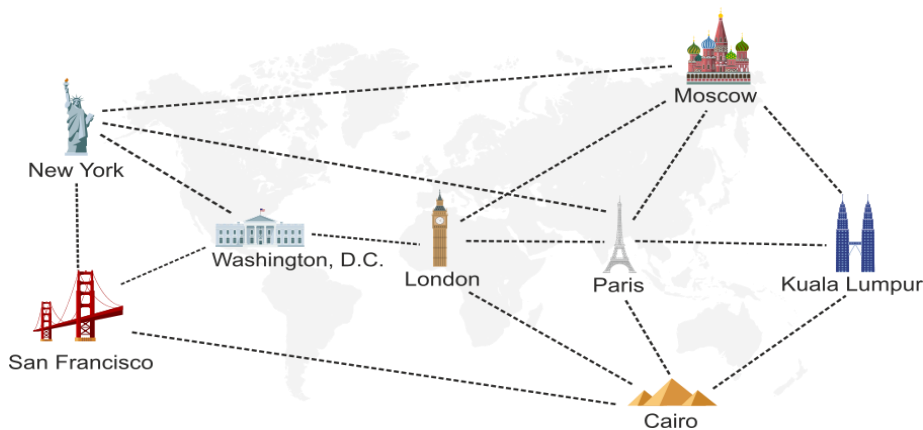
- разное время ожидания в аэропорту;
- часовые пояса.

С учетом количества путешествий по всему миру на поезде, самолете, через часовые пояса (а иногда и через международную линию перемены даты), это проблема, решение которой очень важно для специалистов в области информатики.

10. Экологическая авиакомпания

Автор: Себастьян Комбефис (Бельгия)

У авиакомпании "Бобровый воздух" есть множество авиарейсов, которые соединяют самые большие города мира:



Чтобы уменьшить вредное влияние на климат, "Бобровый воздух" хочет уменьшить количество авиарейсов. Фирма хочет сделать это так, чтобы у клиентов все-таки осталась возможность долететь до всех городов, которые фирма обслуживает.

Например, если закрыть авиAPERелет между Сан-Франциско и Вашингтоном, то клиенты смогут добраться до Вашингтона через Нью-Йорк.

Вопрос

Каково наибольшее количество авиарейсов, которое можно закрыть, но при этом сохранить возможность летать во все города, показанные на вышеприведенном рисунке?

Ответ

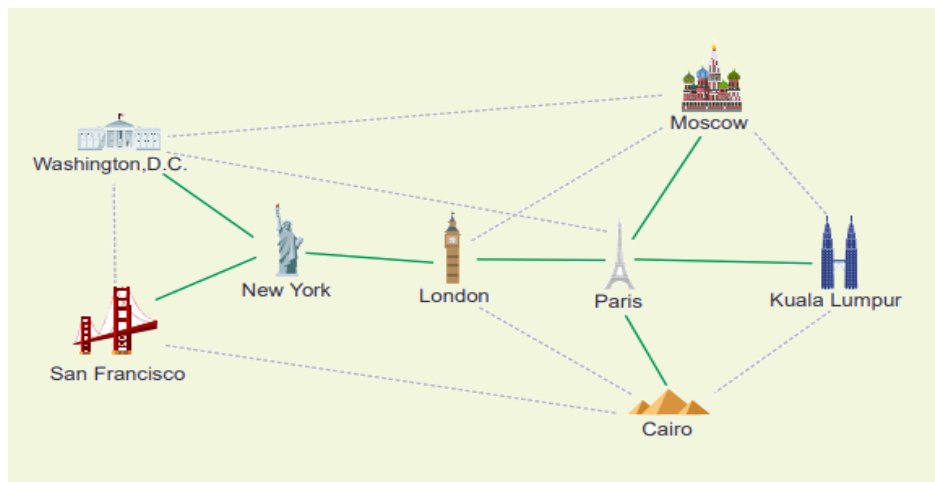
8

Объяснение решения

Изначально мы имеем 15 различных авиарейсов и 8 городов. 8 авиарейсов можно закрыть, таким образом, останется $15 - 8 = 7$ авиарейсов. Больше авиаперелетов закрывать нельзя, так как должна остаться возможность долететь до всех городов. Если бы у нас было 3 города, то хватило бы 2 авиалиний. Продолжив подобным образом, придём к выводу, что в случае 8 городов, нужно 7 авиалиний.

Если бы мы закрыли 9 авиалиний, то осталось бы $15 - 9 = 6$ авиалиний, что не покрыло бы соединение между всеми городами и один город остался бы без авиасвязи.

Один из возможных вариантов создания авиалиний между 8 городами выглядит следующим образом:



Это информатика, потому что

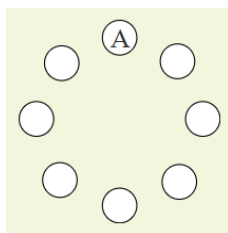
Авиарейсы можно изображать в виде графов, где города – это вершины, а соединения между ними – это рёбра. Задачей является нахождение [дерева](#) – связанного ациклического графа, где связанность означает наличие связи между любыми двумя вершинами, а ацикличность – отсутствие циклов.

В информатике графы используются для обозначения при проектировании сетей телекоммуникации, транспорта или водопровода.

11. Друзья за столом

Автор: Трой Васи́га (Канада)

Восемь друзей (Анна, Даниэль, Генри, Грета, Эля, Франк, Клара, Бруно) сидят за круглым столом лицом к центру стола. Анна сидит на стуле, обозначенном на рисунке буквой "А":



Известно, что друзья рассажены следующим образом:

- Анна сидит напротив Даниэля.
- Генри сидит между Гретой и Элей.
- Франк не сидит рядом ни с Анной, ни с Даниэлем.
- Между Гретой и Кларой сидит один человек.
- Эля сидит рядом с Даниэлем с левой стороны.

Вопрос

Рассади друзей за столом в правильной последовательности.

Ответ



Объяснение решения

Так как Анна сидит напротив Даниэля, то между ними справа и слева сидит по три человека.

Также уверенно можем обозначить Элю, которая сидит рядом с Даниэлем с левой стороны.

Ещё известно, что Генри сидит между Гретой и Элей. Последнее означает, что Генри сидит слева от Эли, так как справа от Эли уже сидит Даниэль и слева от Генри сидит Грета. Этим мы рассадим друзей по одну сторону между Анной и Даниэлем.

Между Гретой и Кларой сидит один человек. Так как по другую сторону друзья уже рассажены, то единственное место для Клары слева от Анны.

Так как Франк не сидит рядом ни с Анной, ни с Даниэлем, ему достаётся место слева от Клары. Остаётся последнее место между Франком и Даниэлем – место для Бруно.

Это информатика, потому что

В данном задании ключевой операцией является отрицание, которое мы используем каждый раз, когда говорим "не" или "нет". Утверждение "Франк не сидит рядом ни с Анной, ни с Даниэлем" можно записать таким образом: "Франк не сидит рядом с (Анной или Даниэлем)", что равносильно утверждению "(Франк не сидит рядом с Анной) и (Франк не сидит рядом с Даниэлем)".

В общей форме можно записать "не (А или В)" и равносильно этому "(не А) и (не В)". Важно, что при внесении отрицания в скобки вместо изначального "или", появляется "и". Данное правило известно под названием [закон де Моргана](#).

Преобразование логических выражений и их комбинирование – очень важное умение как в науке о вычислительных машинах, так и в других областях жизни.

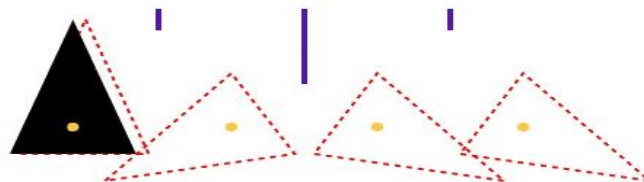
12. Вращайся-вращайся-вращайся

Автор: Илья Посов (Россия)

Треугольник движется по конвейеру и вращается. Центр треугольника находится ближе к двум углам треугольника, но подальше от третьего угла. Центр треугольника выбран так, что, когда треугольник поворачивается на 120 градусов, один из углов находится прямо над центром треугольника.

Над конвейером висят стержни. В зависимости от расположения треугольника и длины стержня, треугольник может начать вращаться на 120 градусов против часовой стрелки. В наличии имеются стержни двух длин:

- Длинный стержень всегда разворачивает треугольник.
- Короткий стержень разворачивает треугольник только в том случае, если соприкасается с углом, который дальше всех находится от центра треугольника.



Роберт пытается смоделировать такое расположение стержней, при котором треугольник всегда заканчивает движение по конвейеру в одном и том же положении независимо от того, в каком положении треугольник поступил на конвейер. Роберту не важно конечное положение.

Вопрос

Какая из следующих последовательностей стержней удовлетворяет условиям Роберта?

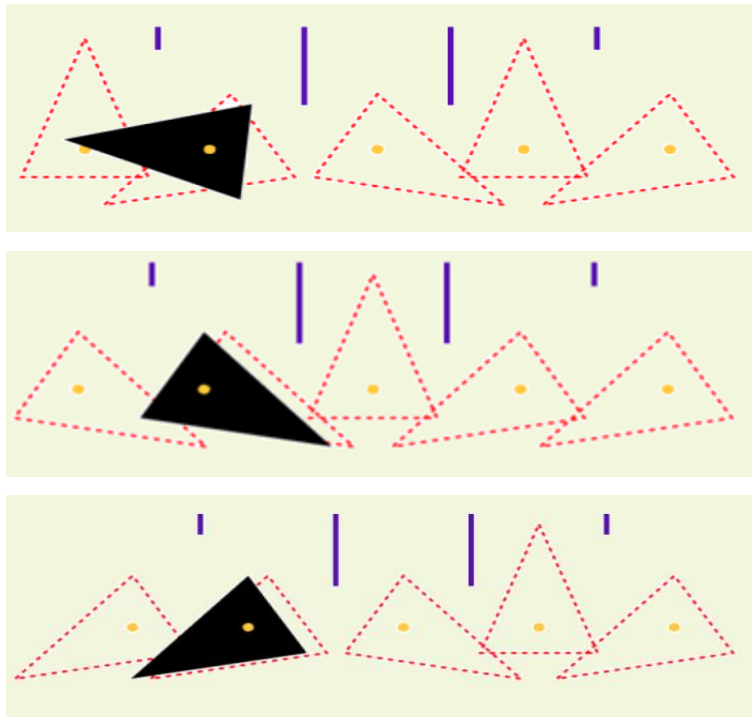
- A. Короткий, длинный, короткий, длинный
- B. Длинный, короткий, длинный, короткий
- C. Длинный, короткий, короткий, длинный
- D. Короткий, длинный, длинный, короткий

Ответ

- D. Короткий, длинный, длинный, короткий

Объяснение решения

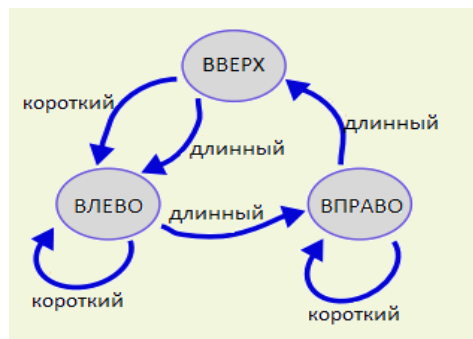
Убедимся, что с помощью варианта D получим желаемый результат:



В случае разных входов на конвейер треугольник всегда заканчивает движение в одном и том же положении.

Это информатика, потому что

В данном задании мы имеем дело с конечным автоматом, моделью дискретного устройства, изображённого ниже:



Здесь есть три состояния (направления самой дальней вершины от точки поворота): ВВЕРХ, ВЛЕВО или ВПРАВО. А также два перехода состояния в зависимости от длины стержня: короткий, длинный.

В данном задании важно найти общее конечное состояние, которое не зависит от начального состояния. Это довольно распространённая задача в информатике.

13. Копирование текста

Задание 13 из раздела Бенджамины.

14. Обработка текста в электронной таблице

Задание 14 из раздела Юниоры.

15. Сохранение видео

Задание 15 из раздела Юниоры.

3.5 Veebipõhise ülesannete kogumiku kasutamine

Alapeatükkides 3.2 – 3.4 esitatud ülesannete tekstid on kasutatud veebipõhises ülesannete kogumikus, mis töötab veebibrauseris. Kogumiku teksti lisamisel on kasutatud HTML märgistuskeelt. Programmeerimiskeel JavaScript on kasutatud näiteks nupude käitumise defineerimisel, vastuse näitamisel või peitmisel ning vastuste vahel liikumisel. Interaktiivse ülesannete kogumiku loomine toimus läbi GitHub-i privaatse hoidla – töofailid on kättesaadavad aadressilt https://github.com/vesta33/kobras_features.git.

Ülesannete kogumiku avaleht on välja toodud joonisel 18.

Задания II тура викторины по информатике KOBRA

На странице приведены задания второго тура викторины, который состоялся в феврале 2020 года. В конце каждого задания приведены правильный ответ и объяснение решения задания, а также связь данного задания с информатикой. Перейдите по ссылке к интересующему вас заданию, и попробуйте самостоятельно найти решение.

[Бенджамины \(6-8 класс\)](#) [Юниоры \(9-10 класс\)](#) [Сеньоры \(11-12 класс\)](#)

Copyright © 2020 [Bebras - International Challenge on Informatics and Computational Thinking](#)
Licensed under [Creative Commons Attribution-ShareAlike 4.0 International License](#)

[Закрыть](#)
[Раскрыть все](#)

Joonis 18. Ülesannete kogumiku avaleht

Vanuserühma valimisel näidatakse kogu rühma ülesannete pealkirju. Vajutades huvipakkuval pealkirjal avaneb kast koos ülesande püstituse ja selle lahendusega (vt. joonis 19).

Задания II тура викторины по информатике KOBRA

На странице приведены задания второго тура викторины, который состоялся в феврале 2020 года. В конце каждого задания приведены правильный ответ и объяснение решения задания, а также связь данного задания с информатикой. Перейдите по ссылке к интересующему вас заданию, и попробуйте самостоятельно найти решение.

[Бенджамины \(6-8 класс\)](#) [Юниоры \(9-10 класс\)](#) [Сеньоры \(11-12 класс\)](#)

[Юбилей WWW](#) [Ложные новости](#) [Фотографирование в школе](#) [Дополнительные устройства](#) [Переработка стекла](#)
[Перестановки](#) [Браслет дружбы](#) [Муравьи на болоте](#) [Вечеринка](#) [Цветной китайский символ](#)
[Торты и соседи](#) [Ранголи](#) [Копирование текста](#) [Наблюдение](#) [Браузер](#)

[Перейти к предыдущему заданию](#) **Фотографирование в школе** [Перейти к следующему заданию](#)

Автор: Axto Truu (Эстония)

Все чаще задают вопросы о сделанной в школе фото- и видеосъемке. Когда это можно делать без ограничений, а когда на съемку надо спрашивать разрешения?

Вопрос

Если фотография сделана в школе, то в каких случаях надо спрашивать разрешения на ее использование у запечатленных на фотографии людей? Выбери все правильные ответы!

A. Фотографирование себе на память
B. Публикование фотографии в социальных сетях
C. Публикование фотографии в печати

[Закрыть](#)
[Раскрыть все](#)

Joonis 19. Valitud ülesande vaade

Vastus on esialgu peidetud ja selle nägemiseks tuleb valida “Показать ответ”. Kui vastus on nähtav, siis tekst muutub – nüüd on kirjas “Скрыть ответ”. Viimane võimaldab vastust uuesti peita (vt. joonis 20).

<p>Вопрос</p> <p>Если фотография сделана в школе, то в ка Выбери все правильные ответы!</p> <p>A. Фотографирование себе на память B. Опубликование фотографии в социальны C. Опубликование фотографии в печати D. Фотографирование в качестве доказател</p> <p>Ответ</p> <p>Показать ответ</p> <p>Объяснение решения</p> <p>В законе сказано, что человек может сохр нельзя опубликовать без его разрешения.</p>	<p>Вопрос</p> <p>Если фотография сделана в школе, то в ка Выбери все правильные ответы!</p> <p>A. Фотографирование себе на память B. Опубликование фотографии в социальны C. Опубликование фотографии в печати D. Фотографирование в качестве доказатель</p> <p>Ответ</p> <p>Скрыть ответ</p> <p>B. Оубликование фотографии в социальных Объяснение решения</p>
--	---

Joonis 20. Peidetud ja kuvatud ülesande vastus

Teise ülesande valimisel eelmise ülesande tekst peidetakse. Ülesannete vahel on võimalik liikuda nii üleval toodud realkirjade abil kui ka nuppude abil: “Перейти к следующему заданию” või “Перейти к предыдущему заданию”. Mugavuse tõttu asuvad nad nii ülesande kasti alguses, kui ka lõpus.

Ülesannete kogumiku paremas alumises nurgas on olemas kaks lisanuppu:

- *Закреть* – paneb eelnevalt avatud küsimust või küsimusi kinni ja jätab alles ainult vanuserühma nimekirja;
- *Раскрыть все* – avab kõik küsimused kõikidest vanuserühmadest.

3.6 Kogumiku kohandamine järgmisteks viktoriinideks (arendajale)

Kogumiku töö põhineb kahel failil:

1. **examples.html** – veebileht ülesannetega
2. **answerBox.js** – fail, kus on defineeritud järgmised funktsioonid:
 - **function showTable(id1,id2,id3)** – määrab millise vanuserühma küsimuste lingid kuvada;
 - **function next(id1,id2)** ja **function prev(id1,id2)** – määravad, kas liikuda järgmise või eelmise küsimuse juurde;
 - **\$("#OpenAll").click(function())** – näitab kõik olemasolevad küsimused korraga, peidab ära küsimuste pealkirjad ning nupud “*Перейти к следующему заданию*” ja “*Перейти к предыдущему заданию*”;
 - **\$("#CloseBtn").click(function())** – sulgeb kõik avatud küsimused ja peidab ära küsimuste pealkirjad;
 - **function change(btnId, answ)** – määrab, kas kuvada või peita küsimuse vastus ning defineerib nupu pealkirja;
 - **function remove(id)** – määrab, kas kuvada küsimus või mitte.

3.7 Võimalik tuleviku arendus

Hetkel defineeritakse iga nupu kuvamist eraldi, mis tekitab väga palju koodi. Üheks optimeerimise võimaluseks on näiteks nupule “*Перейти к следующему заданию*” kirjutada kaks eraldi funktsiooni:

- 1) funktsioon, mis defineerib iga ülesande jaoks eraldi nupu *id* ja lisab neid HTML klassi ka *onclick* defineerimisel. Näiteks hetkel kasutatakse *type=“button“ id=“next1“ onclick=“next(‘#box1’, ‘#box2’)“* ja väärtused sisestatakse käsitsi;
- 2) funktsioon, mis võtab eelmisest funktsioonist defineeritud *id* väärtused ja peidab või kuvab neid lehel. Hetkel iga nupu kuvamine või peitmine kirjeldatud eraldi real kujul:

```
$(, #next1“).hide();
```

```
$(, #next1“).show();
```

4. Uued ülesanded

Selles peatükis on esitletud autori poolt koostatud uued ülesanded. Ülesanded on kirjutatud kolmes keeles, mis võimaldab nende kasutamist nii Eestis peetavas viktoriinis kui ka nende lisamist rahvusvahelisesse küsimuste panka. Käesolevas peatükis ülesanded on välja toodud eesti keeles. Inglise- ja venekeelsed tekstide versioonid on leitavad töö lisast 1.

4.1 Järgitud head tava

Uute ülesannete koostamisel autor on järginud head tava, mis rakendub ülesannete koostamisele rahvusvahelises informaatikaviktoriinis Kobras [17]:

- ülesandel peab olema seos informaatikaga;
- tekst peab olema hästi mõistetav ja võimalikult lühike;
- ülesande lahendamisele peaks kuluma umbes 3 minutit;
- ülesannet on võimalik lahendada ilma abimaterjalideta;
- iga ülesande kohta on märgitud soovituslik vanuserühm, raskusaste ja ülesannete tüüp;
- iga ülesande kohta on toodud ülesande püstitus ja õige vastus;
- iga ülesande kohta on toodud õige vastuse seletus ja seos informaatikaga.

4.2 Ülesanded eesti keeles

1. Tark programm

Autor: Anastassia Fomin (Eesti)

- Vanuserühm: [Benjaminid] / [keskmine], [Juuniorid] / [lihtne]
- Tüüp: Muu tarkvara

Väike kobras Danek vaatab õhtuti oma lemmik multikat. Multikas algab kogu aeg ühel ja samal ajal, kell 20:00. Isa kobras kirjutas programmi, mis lülitab televiisori sisse ainult teatud ajal.

Samuti kontrollib programm, kas majas on elekter olemas ja et Danek oleks sellel ajal kodus.

Programmi tingimus näeb välja järgmiselt:

KUI (tingimus1, tingimus2, tingimus3)

SIIS käsk1

MUIDU käsk2

LÕPP

On teada, et *käsk1* täidetakse ainult siis, kui kõik tingimused on õiged ja tingimusi kontrollitakse vasakult paremale. *Käsk2* täidetakse igal teisel juhul.

Küsimus

Mis järjekorras peavad olema täidetud programmi tingimused, et iga kord kontrollitud tingimuste arv oleks miimaalne?

- A. Kobras on kodus, Kodus on elekter, Kell on 20:00.
- B. Kell on 20:00, Kobras on kodus, Kodus on elekter.
- C. Kodus on elekter, Kell on 20:00, Kobras on kodus.
- D. Antud ülesandes järjestus pole oluline.

Ja mida täidab iga käsk?

- E. Käsk1 – lülitab TV välja, käsk2 – lülitab TV sisse.
- F. Käsk1 – lülitab TV sisse, käsk2 – lülitab TV välja.

Vastus

D. Antud ülesandes järjestus pole oluline; F. Käsk1 – lülitab TV sisse, käsk2 – lülitab TV välja.

Seletus

Pole oluline, mis järjekorras me hakkame tingimusi kontrollima, sest igaüks nendest võib osutada tõeseks või vääraks.

Kui kõik tingimused on tõesed, programm täidab käsu1 – lülitab TV sisse. Teistel juhtudel, programm lülitab TV välja ehk täidab käsu2.

See on informaatika, sest

Tingimuslause, mis sai kasutatud antud ülesandes, on programmeerimises kasutatav konstruktsioon, mis võimaldab käskude täitmist vastavalt etteantud loogilistele tingimustele. Kui tingimus on tõene, siis täidetakse üks käsk, kui mitte, siis teine käsk.

2. Vaatluspost

Autor: Anastassia Fomin (Eesti)

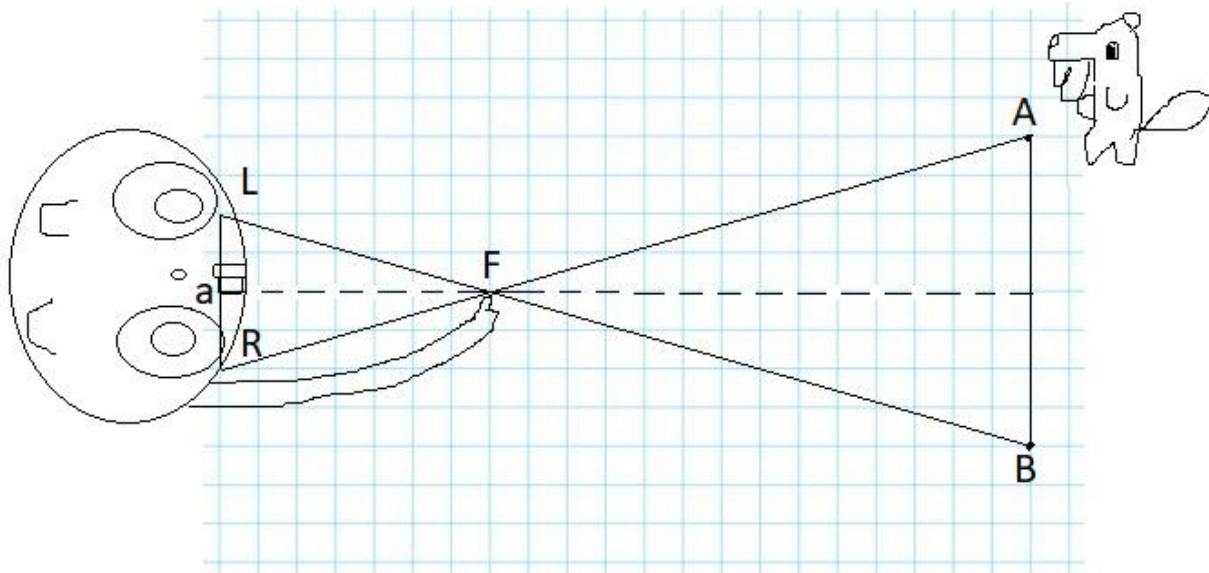
- Vanuserühm: [Benjaminid] / [keskmine], [Juuniorid] / [lihtne]
- Tüüp: Muu tarkvara

Kobras Robin on vaatluspostil. Ta peab teatama staabi kõigist nähtud objektidest ja kaugusest nendeni.

Pikemat aega toimuvat jälgides, ta märkas eemal ühte kobrast, kes liikus aeglase sammudega sirgjoonel punktist A punktini B ja tagasi.

Kobras Robin võttis liikuva kopra suhtes ristiasendi ja sirutas rusikas käpa, põial ülespoole ning viimase suunas. Kui põial oli sihitud kaugel oleva kopra kuju peale, pani Robin vasaku silma kinni. Kui võõras oli punktis A (kuhu oli Robini pilk pöidlast läbi suunatud), pani Robin kiiresti parema silma kinni ja avas vasaku silma. Sel hetkel liikus vaatenurk (vaateväli) punktist A punkti B. Kui põial nihkus, vaatenurga muutumise tagajärjel, hakkas Robin lugema kauguses kõndiva kopra samme ja sai teada, et punktist A punkti B on tehtud 20 sammu. On teada, et ühe sammu pikkus on $\frac{3}{4}$ meetrit, seega saame, et kaugus A-st B-ni on $20 * \frac{3}{4} = 15$ meetrit.

Samuti on teada, et kaugus vasakust silmast parema silmani on 6 cm (LR) ja kaugus silmade kõrgusest pöidlani on 60 cm (aF).



Robin laskis käpa alla ja otsustas arvutada, kui kaugel võõras temast asub.

Küsimus

Kui suur on Robini ja võõra kopra vaheline kaugus?

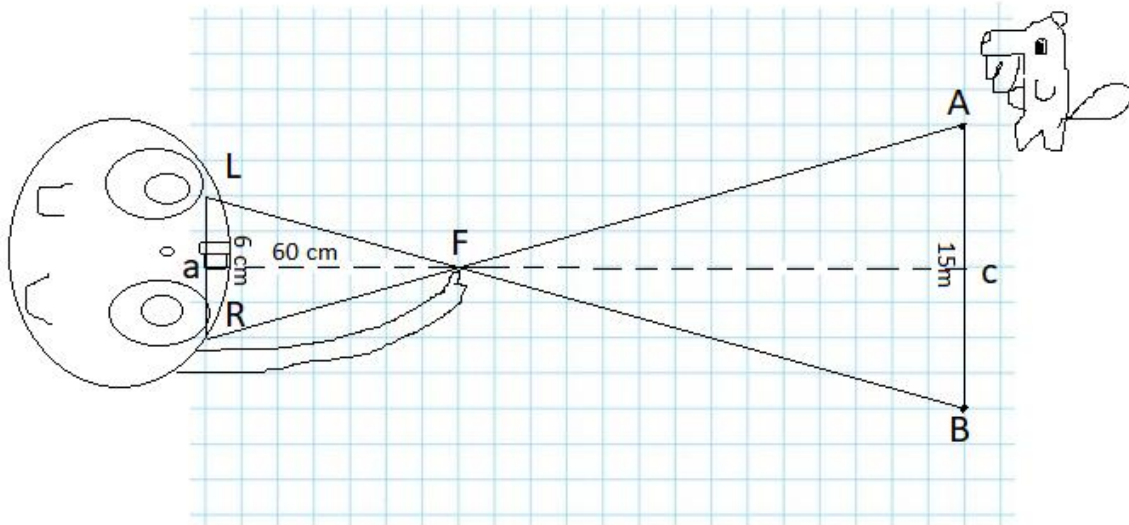
- A. 150 m
- B. 150,6 m
- C. 300 m
- D. 360 m

Vastus

B. 150,6 m

Seletus

Antud probleemi saab lahendada kolmnurkade sarnasuse põhjal. On teada, et vasaku ja parema silma vaheline kaugus on $LR = 6$ cm ja kaugus silmade kõrgusest pöidlani on $aF = 60$ cm. Nende suhet saab esitada järgnevalt $60/6$. Tähistagem lõigu AB keskel asuvat punkti tähega c , saame võrrandi $Fc / AB = aF / LR$, kus AB on ülesande eelduse põhjal 15 m. Saame võrratuse $Fc / 15 = 60/6$, kust $Fc = 15 * 10 = 150$ m. Viimasele numbrile tuleb lisada kaugus silmade kõrgusest väljasirutatud käpa pöidlani ja saame, et kaugus võõrani on $150 \text{ m} + 60 \text{ cm} = 150,6 \text{ m}$.



See on informaatika, sest

Virtuaalne reaalsus võimaldab meil reisida kohtadesse, kus me pole kunagi käinud. Prillid, mille tööpõhimõte tugineb stereoskoopia efektile, võimaldavad tunda virtuaalmaailma täiuslikkust. Ekraanil kuvatakse kaks pilti – üks parema silma jaoks, teine, veidi muudetuna, vasaku jaoks. Keskel on vahesein.

Silmad on vaatevälja subjekti suhtes erineva nurga all, mis seletab veidi erinevat rakurssi ja sügavust. Kui aga mõlemad silmad on avatud, siis seda me ei taju, kuna aju ühendab vastuvõetud andmed üheks mahupildiks.

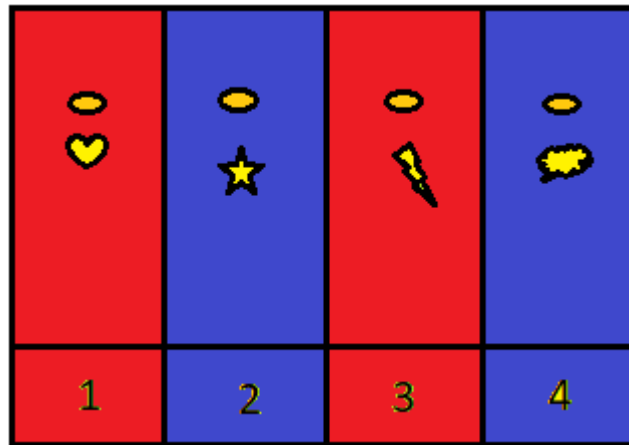
Stereoskoopilisi pilte saab teha asetades kaks kaamerat statiivile nii, et need oleksid samal kõrgusel ja läätsede keskpunktide vaheline kaugus oleks vähemalt 6 cm.

3. Kapid

Autor: Anastassia Fomin (Eesti)

- Vanuserühm: [Benjaminid] / [keskmine], [Juuniorid] / [lihtne]
- Tüüp: Muu tarkvara

Koprad Ann, Ben, Chris ja Diana lähevad lasteaeda. Õpetaja valmistab kapid ette niiviisi, et kahe poisi ja kahe tüdruku kapid ei oleks kõrvuti.



Küsimus

Mitu võimalust kappide jagamiseks eksisteerib, et mitte rikkuda ülendes antud tingimust?

Vastus

8

Seletus

On olemas 2 võimalust jagada kapid nii, et kahe poisi ja kahe tüdruku kapid ei oleks kõrvuti:

1. POISS, TÜDRUK, POISS, TÜDRUK
2. TÜDRUK, POISS, TÜDRUK, POISS

Selleks, et valida kes poistest saab vasakult poolt paikneva kapi, on olemas kaks võimalust – kapi saab Ben või Chris. Analoogselt kahe tüdruku puhul – kapi saab kas Ann või Diana.

Kokkuvõtteks saame $2*2*2=8$ võimalust.

Ülesannet saab lahendada ka vähem soodsama viisiga. Selleks on vaja kirja panna kõik võimalikud variandid ja loendada neid:

1. Ben, Ann, Chris, Diana
2. Ben, Diana, Chris, Ann
3. Chris, Ann, Ben, Diana
4. Chris, Diana, Ben, Ann
5. Ann, Ben, Diana, Chris
6. Ann, Chris, Diana, Ben
7. Diana, Ben, Ann, Chris
8. Diana, Chris, Ann, Ben

See on informaatika, sest

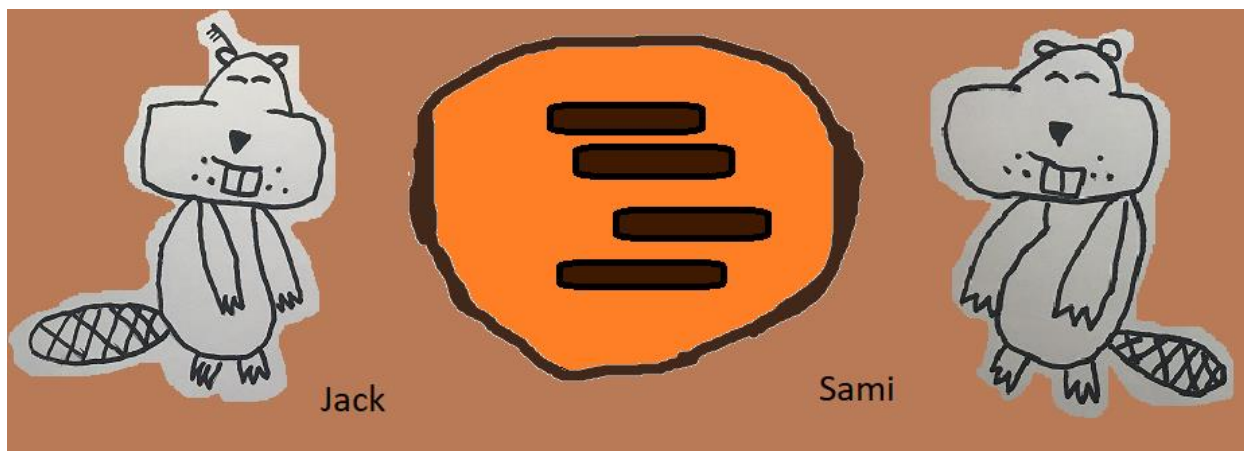
Antud ülesanne on otseselt seotud kombinatoorikaga – matemaatika haru, mis uurib diskreetseid objekte. Kombinatoorikat kasutatakse paljudel aladel, kaasa arvatud informaatikas. Näiteks paroolide sirvimisel, graafis lühema tee leidmiseks võimalikke variantide läbivaatamisel ning võrkude marsrutiseerimisel.

4. Suvila

Autor: Anastassia Fomin (Eesti)

- Vanuserühm: [Benjaminid] / [raske], [Juuniorid] / [keskmine]
- Tüüp: Muu tarkvara

Kaks kobrast, Jack ja Sami, ehitavad palkidest suvilat. Jack viib palgid lattu ja Sami võtab nad sealt ära ning viib ehitusplatsile. Jack on väga tugev kobras ja toob korraga 2 palki, samuti liigub ta ühtlase kiirusega 0.5 meetrit sekundis. Sami seevastu liigub kiirusega 1.5 meetrit sekundis ja suudab kanda ainult ühte palki. Mõlemad koprad alustavad sama kauguselt – Jack metsarajuko-
hast, mis asub 1.5 m kaugusel laost ja Sami ehitusplatsilt, mis samuti asub 1.5 m kaugusel laost. Ühe sekundiga lattu saabudes ei leia Sami sealt palke. On teada, et kui palke pole, naaseb Sami ehitusplatsile ja teeb muid asju, et aega mitte raisata. Palgi laadimise ja mahalaadimise aega ei arvestata, ehk see võrdub 0 sekundiga.



Küsimus

Mitu palki saab Sami toimetada ehitusplatsile 10 sekundi jooksul?

Vastus

3

Seletus

10 sekundi jooksul jõuab Jack teha ainult kaks käiku ja tuua seega 4 palki. Seda on näha probleemi püstitusest – kaugus laost on 1.5 m ja Jacki kiirus on 0.5 m/s. Seega ta kuulutab palkidega lattu jõudmiseks 3 sekundit ja samuti kulub 3 sekundit, et minna uuele portsule järgi.

Samil on aega laos käia viis korda, kuna kahe sekundi jooksul jõuab ta minna lattu ja naasta tagasi ehitusplatsile. Esimese palgi toimetamine ehitusplatsile võtab aega neli sekundit, teine palk jõuab ehitusplatsile järgmise 2 sekundi möödumisel. Seitsmendal sekundil pole Samil jälle laost midagi võtta ja ta läheb tagasi ehitusplatsile, kuid kümnendal sekundil toob ta ehitusplatsile kolmanda palgi.

See on informaatika, sest

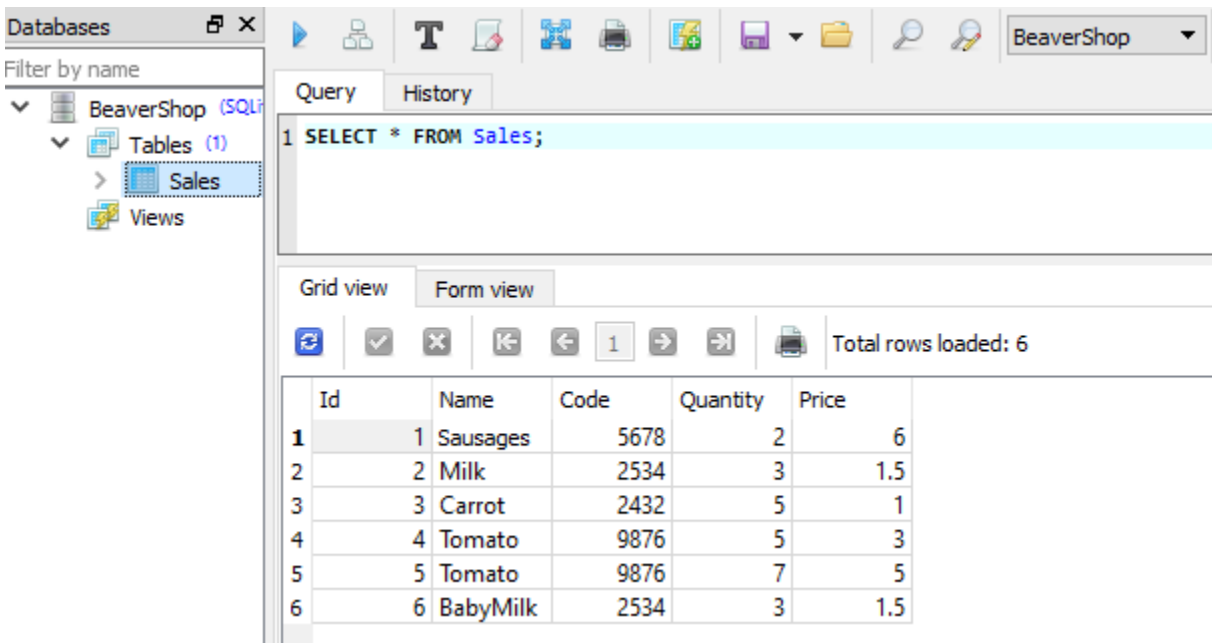
Antud ülesanne on seotud mõistega “protsessori taktsagedus”. Kobras Jack on taktsageduse generaatori rollis, seab takti ja mõjutab sellega Sami kiirust ja tööd. Viimast võib käsitleda arvuti protsessori rollis. Arvuti protsessor suudab teha miljoneid toiminguid sekundis ja tema kiirust mõõdetakse GHz-des.

5. Kauba arvelevõtmine

Autor: Anastassia Fomin (Eesti)

- Vanuserühm: [Seeniorid] / [raske]
- Tüüp: Muu tarkvara

Isegi väikeses kopra kaupluses kauba haldamiseks on vaja kasutada tabeleid, mis lubavad lihtsustada edasist andmetöötlust. Kopra kaupluse halduril on olemas Müügitabel (*Sales*):



The screenshot shows a database management interface with the following components:

- Databases:** BeaverShop (SQL)
- Tables:** Sales
- Query:** `1 SELECT * FROM Sales;`
- Grid view:** Total rows loaded: 6
- Table Data:**

Id	Name	Code	Quantity	Price
1	1 Sausages	5678	2	6
2	2 Milk	2534	3	1.5
3	3 Carrot	2432	5	1
4	4 Tomato	9876	5	3
5	5 Tomato	9876	7	5
6	6 BabyMilk	2534	3	1.5

Tarvis on:

- leida müüdnud kauba arvu (*Total*), mille kood (*Code*) on sama;
- esitada andmed nii, et nad oleksid kolmes tulpas nimedega *Code of item*, *Name*, *Total*;
- paigutada andmed kahanevas järjekorras.

Küsimus

Milline järgmistest päringutest on õigesti kirjutatud, et saada vastus?

- A. `SELECT Code AS 'Code of item', Name AS 'Name', SUM(Quantity) AS Total FROM SALES GROUP BY Id ORDER BY Total DESC;`
- B. `SELECT Code AS 'Code of item', Name AS 'Name', SUM(Quantity) AS Total FROM SALES GROUP BY Name ORDER BY Total DESC;`
- C. `SELECT Code AS 'Code of item', Name AS 'Name', SUM(Quantity) AS Total FROM SALES GROUP BY Code ORDER BY Total DESC;`
- D. `SELECT Code AS 'Code of item', Name AS 'Name', SUM(Quantity) AS Total FROM SALES;`
- E. `SELECT Code AS 'Code of item', SUM(Quantity) AS Total FROM SALES GROUP BY Code ORDER BY Total DESC;`

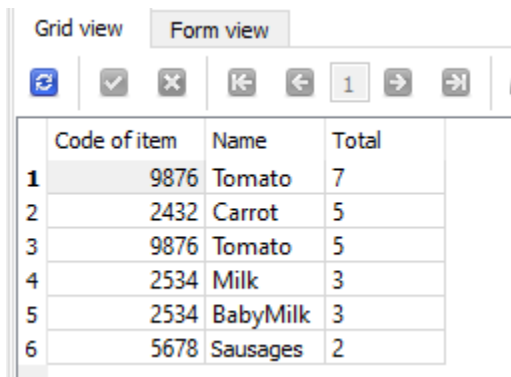
Vastus

C. *SELECT Code AS 'Code of item', Name AS 'Name', SUM(Quantity) AS Total FROM SALES GROUP BY Code ORDER BY Total DESC;*

Seletus

Vaatame iga päringut eraldi:

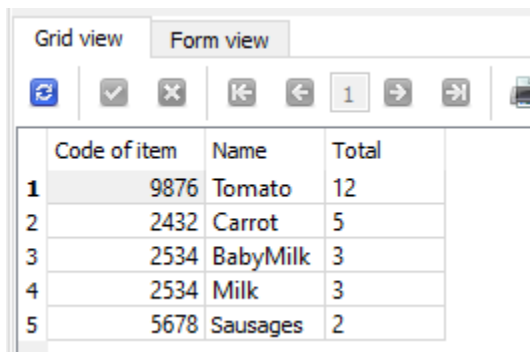
A. *SELECT Code AS 'Code of item', Name AS 'Name', SUM(Quantity) AS Total FROM SALES GROUP BY Id ORDER BY Total DESC;*



	Code of item	Name	Total
1	9876	Tomato	7
2	2432	Carrot	5
3	9876	Tomato	5
4	2534	Milk	3
5	2534	BabyMilk	3
6	5678	Sausages	2

Tulemus on esitatud kolmes tulpas nimedega *Code of item*, *Name*, *Total*. Andmed on välja toodud kahanevas järjekorras, kuid grupeeritud *Id* parameetri alusel, mitte *Code* alusel nagu on toodud ülesande püstituses.

B. *SELECT Code AS 'Code of item', Name AS 'Name', SUM(Quantity) AS Total FROM SALES GROUP BY Name ORDER BY Total DESC;*



	Code of item	Name	Total
1	9876	Tomato	12
2	2432	Carrot	5
3	2534	BabyMilk	3
4	2534	Milk	3
5	5678	Sausages	2

Tulemus on esitatud kolmes tulpas nimedega *Code of item*, *Name*, *Total*. Andmed on välja toodud kahanevas järjekorras, kuid grupeeritud *Name* parameetri alusel, mitte *Code* alusel. Kuna kauba nimi ei ole antud olukorras unikaalne, siis antud variant meile ei sobi.

C. *SELECT Code AS 'Code of item', Name AS 'Name', SUM(Quantity) AS Total FROM SALES GROUP BY Code ORDER BY Total DESC;*

	Code of item	Name	Total
1	9876	Tomato	12
2	2534	BabyMilk	6
3	2432	Carrot	5
4	5678	Sausages	2

Kõik tingimused on täidetud ja see on õige vastus.

D. *SELECT Code AS 'Code of item', Name AS 'Name', SUM(Quantity) AS Total FROM SALES;*

	Code of item	Name	Total
1	2534	BabyMilk	25

Antud päringus andmeid ei grupeerita üldse, vaid leitakse üldine müüdud kaupade summa.

E. *SELECT Code AS 'Code of item', SUM(Quantity) AS Total FROM SALES GROUP BY Code ORDER BY Total DESC;*

	Code of item	Total
1	9876	12
2	2534	6
3	2432	5
4	5678	2

Tulemus on toodud vaid kahes tulpas *Code of item, Total*.

See on informaatika, sest

Andmetega tabelid on aluseks iga andmebaasi jaoks. Selleks, et saada andmed tabelitest, peab teadma, kuidas päringut koostatakse ning mis käskude abil teostatakse. Väga suurte tabelitega töötamisel tuleb olla eriti ettevaatlik, vastasel juhul päringu sooritamine võib võtta mitu tundi ning anda vale tulemuse või andmebaasi töö saab häiritud.

6. Auhinnad (versioon 1)

Autor: Anastassia Fomin (Eesti)

- Vanuserühm: [Benjaminid] / [raske], [Juuniorid] / [keskmine], [Seeniorid] / [lihtne]
- Tüüp: Tabelitöötlus

Iga päev vastates tunnis küsimustele, saavad koprad õigete vastuste eest kleepse. Õppeaasta lõpus kõigi õpilaste vahel jagatakse auhinnad vastavalt kogutud kleebiste arvule. Eelmistel aastatel sisestati andmed arvutustabelisse ja töödeldi käsitsi. Sel aastal oli koolis palju kopraid ja auhinna kiireks määramiseks otsustati kasutada IF-funktsiooni, mis teeb kõik arvutused automaatselt. Sellel funktsioonil on järgmine üldine vorm:

=IF(loogiline_väljend;tagastada_kui_õige;tagastada_kui_vale).

Auhinna loosimiseks kirjutab žürii valemi üles. Näiteks esimese osaleja jaoks näeb valem välja järgmiselt:

=IF(B2<10;"komm";IF(B2<20;"šokolaad";IF(B2<30;"vihik";IF(B2<40;"pastapliiats";IF(B2<50;"raamat";IF(B2<60;"T-särk";IF(B2<70;"pall";IF(B2<80;"hüpits";IF(B2<90;"ehitusklotsid";"reis"))))))))

Siiski tehti viga ja auhindade asemel tabeli lahtritesse ilmus veateade "Err:508".

	A	B	C	D	E	F
1		Anna	Benjamin	Ciara	Diana	Elias
2	<u>Kleepsude arv</u>	45	65	35	50	90
3	<u>Auhind</u>	Err:508	Err:508	Err:508	Err:508	Err:508

Küsimus

Paranda viga ja tee kindlaks, millise auhinna said kõik viis kobrast.

Vastus

=IF(B2<10;"komm";IF(B2<20;"šokolaad";IF(B2<30;"vihik";IF(B2<40;"pastapliiats";IF(B2<50;"raamat";IF(B2<60;"T-särk";IF(B2<70;"pall";IF(B2<80;"hüpits";IF(B2<90;"ehitusklotsid";"reis"))))))))

	A	B	C	D	E	F
1		Anna	Benjamin	Ciara	Diana	Elias
2	<u>Kleepsude arv</u>	45	65	35	50	90
3	<u>Auhind</u>	raamat	pall	pastapliiats	T-särk	reis

Seletus

Veateade "Err: 508" tähendab, et valemis on sulguv sulg ilma sobiva avasuluta või valemi keskel puudub sulguv sulg (sulud valemi lõpus lisatakse automaatselt). Käesolevas ülesandes avasulud puuduvad kolmes kohas:

1. $IF(B2<30;"vihik";$
2. $IF(B2<50;"raamat";$
3. $IF(B2<90;"ehitusklotsid";$

Kui lisada sulud, saame toimiva valemi kätte:

```
=IF(B2<10;"komm";IF(B2<20;"šokolaad";IF(B2<30;"vihik";IF(B2<40;"pastapliiats";  
IF(B2<50;"raamat";IF(B2<60;"T-särk";IF(B2<70;"pall";IF(B2<80;"hüpits";IF(B2<90;  
"ehitusklotsid";"reis"))))))))
```

Õigest valemist lähtudes on näha, et Anna, kes kogus 45 kleepsu, saab raamatu. Benjamin, kes kogus 65 kleepsu, saab palli. Ciara saab oma 35 kleepsu eest pastapliiatsi. Diana, kes kogus 50 kleepsu, saab T-särgi ning Elias saab 90 kleepsu eest reisikupongi.

See on informaatika, sest

IF-funktsioon on arvutustabelites üks populaarsemaid funktsioone. Väärtuste loogilise võrdlemisega annab see oodatud tulemuse, mis võib olla nii arvuline kui ka sõnaline. Kuna tulemus võib olla kas tõene või väär, antakse funktsioonile kaks väärtust. On üsna levinud olukord, kus tahetakse väärtus võrrelda rohkem kui ühe tingimusega. Sel juhul mitme IF-funktsiooni kasutamine (pesastatud IF-funktsioonid) ühe valemi sees on hea lähenemine õige vastuse saamiseks.

7. Auhinnad (versioon 2)

Autor: Anastassia Fomin (Eesti)

- Vanuserühm: [Benjaminid] / [keskmine]
- Tüüp: Tabelitöötlus

Iga päev vastates tunnis küsimustele, saavad koprad õigete vastuste eest kleepse. Õppeaasta lõpus kõigi õpilaste vahel jagatakse auhinnad vastavalt kogutud kleebiste arvule. Eelmistel aastatel sisestati andmed arvutustabelisse ja töödeldi käsitsi. Sel aastal oli koolis palju kopraid ja auhinna kiireks määramiseks otsustati kasutada IF-funktsiooni, mis teeb kõik arvutused automaatselt. Sellel funktsioonil on järgmine üldine vorm:

```
=IF(loogiline_väljend;tagastada_kui_õige;tagastada_kui_vale).
```

Auhinna loosimiseks kirjutab žürii valemi üles. Näiteks esimese osaleja jaoks näeb valem välja järgmiselt:

=IF(B2<10;"komm";IF(B2<20;"šokolaad";IF(B2<30;"vihik";IF(B2<40;"pastapliiats";IF(B2<50;"raamat";IF(B2<60;"T-särk";IF(B2<70;"pall";IF(B2<80;"hüpits";IF(B2<90;"ehitusklotsid";"reis"))))))))

Siiski tehti viga ja auhindade asemel tabeli lahtritesse ilmus veateade "Err:508".

	A	B	C	D	E	F
1		Anna	Benjamin	Ciara	Diana	Elias
2	Kleepsude arv	45	65	35	50	90
3	Auhind	Err:508	Err:508	Err:508	Err:508	Err:508

Küsimus

Milles seisneb viga?

- A. Probleem sulgudega
- B. Ei tohi tagastada sõnalisi väärtusi
- C. Ühes ja samas valemis ei tohi kasutada nii palju tingimusi
- D. Alguses ei tohi olla "=" märki
- E. Jutumärgid tuleb eemaldada

Vastus

- A. Probleem sulgudega

Seletus

Veateade "Viga: 508" tähendab, et valemis on sulguv sulg ilma sobiva avasuluta või valemi keskel puudub sulguv sulg (sulud valemi lõpus lisatakse automaatselt). Käesolevas ülesandes avasulud puuduvad kolmes kohas:

1. IFB2<30;"vihik";
2. IFB2<50;"raamat";
3. IFB2<90;"ehitusklotsid";

Teised, pakutud vastusevariandid, ei ole õiged. IF-funktsioon võib tagastada mitte ainult numbrilisi väärtusi, vaid ka sõnalisi tulemusi, kuupäevi, lahtriviidet või isegi mõnda muud valemit. Tingimuste arv võib olla kuni 64, seega antud ülesandes kasutatud tingimuste arv on vastuvõetav. Funktsiooni alguses lisatakse "=" märk ja mitte numbrilised väärtused kirjutatakse jutumärkides.

See on informaatika, sest

IF-funktsioon on arvutustabelites üks populaarsemaid funktsioone. Väärtuste loogilise võrdlemisega annab see oodatud tulemuse, mis võib olla nii arvuline kui ka sõnaline. Kuna tulemus võib olla kas tõene või väär, antakse funktsioonile kaks väärtust. On üsna levinud olukord, kus tahetakse väärtus võrrelda rohkem kui ühe tingimusega. Sel juhul mitme IF-funktsiooni kasutamine (pesastatud IF-funktsioonid) ühe valemi sees on hea lähenemine õige vastuse saamiseks.

8. Sõnade otsing tekstis

Autor: Anastassia Fomin (Eesti)

- Vanuserühm: [Seeniorid] / [raske]
- Tüüp: Tekstitöötlus

Sõnade otsimiseks tekstist saab kobras kasutada regulaaravaldisi (ingl.k. *regular expressions*), mis koosnevad metamärkidest. Viimased määravad otsinguregleid.

Find & Replace

Search for ↖
regular_expression

Find
Find All

Replace with
Replace
Replace All

Match case
 Whole words only

Fewer Options ▲ Help Close

Current selection only
 Backwards
 Regular expressions ↖
 Similarity search
 Search for Styles
 Notes

Attributes...
Format...
No Format

Mõned metamärkide näited:

[:space:] – tühik, näiteks **b[:space:]e** leiab alltoodud tekstist **b e**.

? – erisümbol, mis määrab tähe või märgi olemasolu või puudumise selle ees. Näiteks **_?beb** leiab **_beb** ja **beb**.

Alltoodud tekstis, sõna "bebras" esineb viis korda ja iga kord ta on pandud kirja erinevalt:

There are two extant species of **b-e-b-r-a-s**; the North American **b e b r a s** and the Eurasian **bebras**. The American **bebras** was coined by Carl Linnaeus in 1758 who also classified the species name fiber. However, the two **bebas** were not conclusively shown to be separate species until the 1970s with chromosomal evidence.

Küsimus

Milline regulaaravaldis peab olema sisestatud otsinguaknasse, et leida tekstist kõik variandid sõnast "bebras"?

- A. `_?b-?e-?b-?r?-?a-?s_?`
- B. `_?b-?[:space:]?e[:space:]?-?b[:space:]?-?r?[:space:]?-?a[:space:]?-?s_?`
- C. `_?b[:space:]?e[:space:]?b[:space:]?r?[:space:]?a[:space:]?s_?`
- D. `_?b-?[:space:]?e[:space:]?-?b[:space:]?-?[:space:]?-?a[:space:]?-?s_?`

Vastus

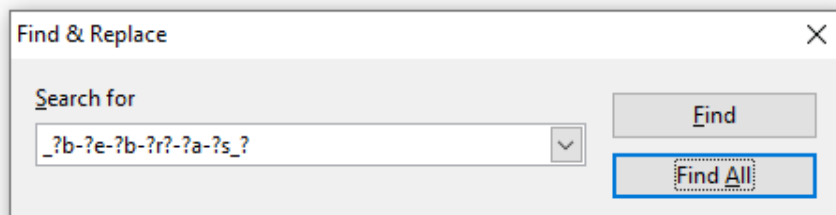
- B. `_?b-?[:space:]?e[:space:]?-?b[:space:]?-?r?[:space:]?-?a[:space:]?-?s_?`

Seletus

Vaatleme iga varianti eraldi.

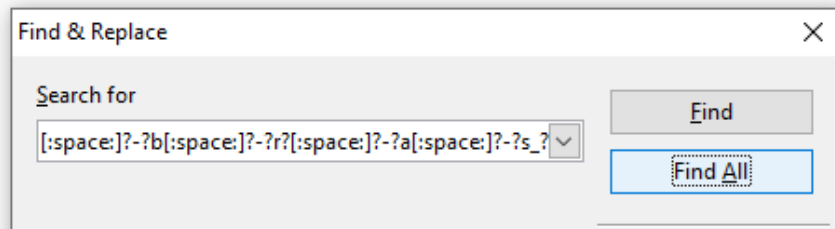
Variante A ei leia sõna **b e b r a s**, kus iga tähe järel on tühik.

There are two extant species of **b-e-b-r-a-s**; the North American **b e b r a s** and the Eurasian **bebras**. The American **bebras** was coined by Carl Linnaeus in 1758 who also classified the species name fiber. However, the two **bebas** were not conclusively shown to be separate species until the 1970s with chromosomal evidence.



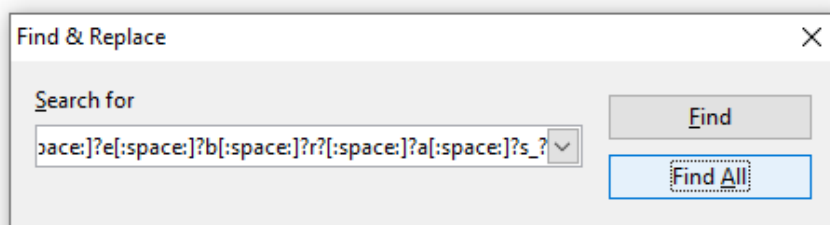
Variant B leiab kõik meile vajalikud sõnad.

There are two extant species of **b-e-b-r-a-s**; the North American **b e b r a s** and the Eurasian **bebras**. The American **bebras** was coined by Carl Linnaeus in 1758 who also classified the species name fiber. However, the two **bebas** were not conclusively shown to be separate species until the 1970s with chromosomal evidence.



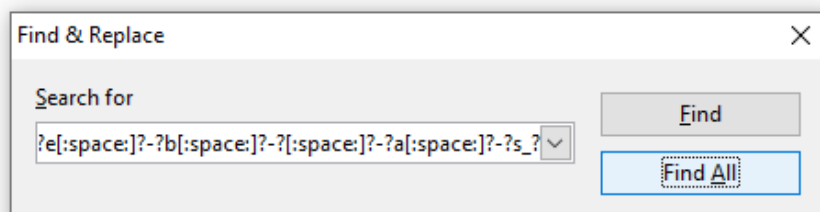
Valik C ei leia sõnavormi, kus tähtede vahel on sidekriips.

There are two extant species of **b-e-b-r-a-s**; the North American **b e b r a s** and the Eurasian **bebras**. The American **bebras** was coined by Carl Linnaeus in 1758 who also classified the species name fiber. However, the two **bebas** were not conclusively shown to be separate species until the 1970s with chromosomal evidence.



Ning variant D leiab vaid ühe sõna, kus täht "r" on puudu.

There are two extant species of **b-e-b-r-a-s**; the North American **b e b r a s** and the Eurasian **bebras**. The American **bebras** was coined by Carl Linnaeus in 1758 who also classified the species name fiber. However, the two **bebas** were not conclusively shown to be separate species until the 1970s with chromosomal evidence.



See on informaatika, sest

Regulaaravaldisi kasutatakse programmeerimises väga sageli, näiteks teatud märkide otsimisel, mitme tähemärgi otsimisel, sõna leidmisel, tervete sõnade valideerimisel, leidmisel ja asendamisel. Neid saab kasutada ka välja täitmise tingimuste määramisel, näiteks parooli sisestamise valideerimisel, kus uuele paroolile seatakse nõuded minimaalsele tähemärkide arvule, väikese / suure tähe, numbrilise või erimärgi olemasolule.

9. Kiri sõbrale

Autor: Anastassia Fomin (Eesti)

- Vanuserühm: [Benjaminid] / [keskmine]
- Tüüp: Eetika

Kobras Karl sai oma sõbra käest kirja.

Kallis sõber!

Ma ei saa aru, mida ma olen valesti teinud, aga meie klassi koprad vaatavad mind imelikult. Kõik algas sellest, et esmaspäeval toimunud koolipeol ma tegin Lionist pildi, kelle seljas oli tema kätega tehtud ülikond. Siis ma laadisin selle sotsiaalvõrku enda profiilipildiks. Miks mitte, oleme temaga nii sarnased?! Lion loomulikult märkas seda ja nüüd palub eemaldada. Kuidas ma saan aga eemaldada, sest tänu sellele pildile olin ma kutsutud saatesse "Moekas mina!"?!

Teisipäeval oli meil informaatika tund ja selleks, et aega mitte kulutada, panin oma lemmikmängu alla-laadimisele. Sel päeval oli Internet liiga aeglane, õppematerjalid ei tulnud lahti ja õpilased ei saanud kõik ülesanded valmis. Mäng sai aga ilusti alla laaditud ja sain vahetunnis natukene mängida.

Kolmapäeval palus õpetaja edasi öelda, et kooli peab tulema teiseks tunniks. Ma mõtlesin, et see on ikkagi liiga vara ja kirjutasin, et tulge kõik kolmandaks tunniks. Kõik nõustused minuga ja kui järgmisel päeval õpetaja pahandas, solvuti minu peale.

Neljapäeval suutsin e-päeviku lahti teha administraatori rollis. Nüüd terve klassi keskmine hinne on paremaks läinud.

Ja lõpuks reede! Lähen kinno oma uut kaamerat proovima. Salvestan uue filmi ja saadan sulle ka vaatamiseks.

Nii et oota!

Nägemiseni!

Sinu sõber Sandi

Küsimus

Mida Sandi tegi valesti? Valige kõik õiged vastused.

- A. Häiris teiste arvutikasutajate tööd, suurendades Interneti koormust.
- B. Omandas teisele kuuluvat intellektuaalset vara.
- C. Tegi Lionist pilti, küsides eelnevalt luba.
- D. Paigutas vale informatsiooni võrku.
- E. Sai lahti e-päeviku.
- F. Võttis kaamera kinno kaasa.

Vastus

- A. Häiris teiste arvutikasutajate tööd, suurendades Interneti koormust.
- B. Omandas teisele kuuluvat intellektuaalset vara.
- D. Paigutas vale informatsiooni võrku.

Seletus

Kui Sandi pani tunni ajal mängu allalaadimisele, häiris ta teisi õpilasi, sest võrk oli ülekoormatud ja vajalik andmevahetuse kiirus failide edastamiseks oli liiga madal.

Sandi oli kutsutud saatesse “Moekas mina!” ainult sellepärast, et omandas Lionile kuuluvat intellektuaalset vara ehk näitas tema ülikonda nagu enda oma. Järelikult variandid A ja B on õiged.

Küsidel luba, pildistamine ei ole keelatud, kuid avaldada pilte ilma luba küsimata või kellegi teisena esineda pole lubatud.

Sandi tegi õigesti, et pani õpetaja palvel info internetti üles, kuid see info oli vale ja seda ei tohtinud juhtuda.

Kui ta sai e-päeviku lahti, ei rikkunud ta veel reeglit, kuid sellest tarkvara nõrkusest pidi ta teavitama tarkvara arendajat ja muidugi mitte kasutama oma otstarbeks.

Kaamera kaasa võtmine kinno pole keelatud, kuid selle kasutamine filmi salvestamiseks ei ole lubatud.

See on informaatika, sest

Eksisteerib terve hulk eetika standarte. Mõni neist võib erineda vastavalt riigi eetilisele ja kultuuritraditsioonile.

Arvutieetikat hakati kasutama 1970-1980 aastatel ja juba siis tekkis palju uusi termineid – informatsioonieetika, kübereetika, blogijate eetika, häkkerieetika. Rohkem saab lugeda siin: <https://www.eetika.ee/et/arvutieetika>.

10. Arva ära sisu

Autor: Anastassia Fomin

- Vanuserühm: [Juuniorid] / [lihtne]
- Tüüp: Loogika, Muu tarkvara

Riiulil olid kastid, mis sisaldasid erinevat tüüpi puukoort. Ükskord, kastitega mängides, ajas väike kobras Matias kastide sildid segamini. Isa-kobras Peeter peab nüüd ära arvama, mis puukoor mis karbis asub. On teada, et Matias ajas segamini kõik sildid ja mitte ükski neist ei vasta enam kasti sisule.



Küsimus

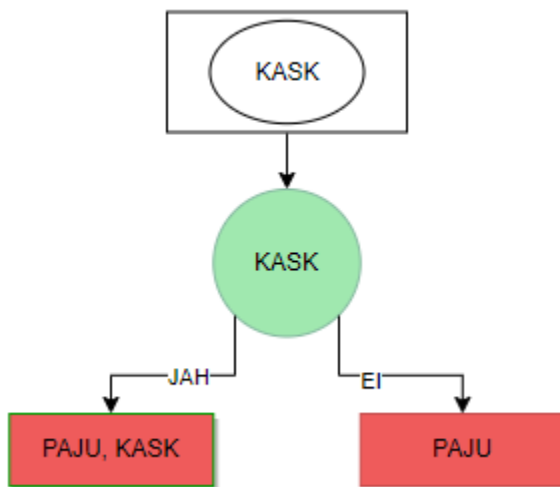
Mitu katset peab Peeter minimaalselt tegema, et ära arvata iga kasti sisu?

Vastus

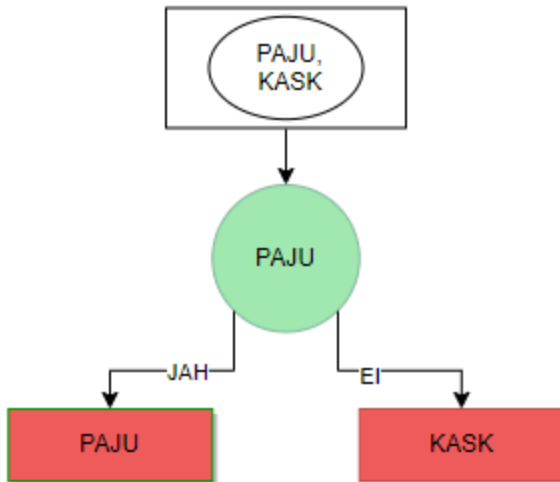
2

Seletus

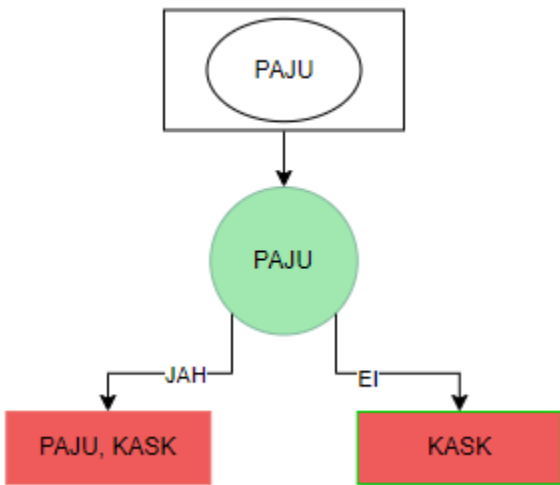
Võib alustada mis tahes kastiga. Oletame, et kobras Peeter tõmbas esimesest kastist kase puukoore välja. On teada, et silt “KASK” ei vasta tõele. Järelikult, lisaks kase puukoorele on kastis ka paju-koor ja sellele karbile oleks vaja kleepida silt “PAJU, KASK”.



Nüüd selgub, et teises kastis võib olla kas paju- või kasekoor. Oletame, et kobras Peeter tõmbas pajukoore välja ning kastile tuleks kleepida silt “PAJU”.



Edasi jääb vaid üks kast, mille sees on kasekoor.



Seega piisab kahest katses, et segatud sildid õigesse kohta kleepida.

See on informaatika, sest

Selle ülesande lahendamisel on kasutatud otsustuspuud. Antud puu lihtsustab andmete mõistmist ja tõlgendamist. Otsustuspuud kasutatakse tarkvaraarenduse protsessis juhtimisotsuste langetamisel, masinõppimises, andmeanalüüsis ning statistikas.

Kokkuvõte

Informaatikaviktoriin Kobras on populaarne viktoriin Eestis, mis on mõeldud üldhariduskooli õpilastele. Iga aasta osaleb viktoriinis rohkem kui 3000 õpilast.

Toimunud viktoriini tulemuste visualiseerimine ja analüüsimine on kulukas. Analüüsimise protsessi kiiremaks läbiviimiseks sai loodud tulemuste visualiseerimisvahend. Veebipõhine rakendus sisaldab interaktiivseid graafikuid ja tabelit. Käesolevas magistritöös rakenduse töö oli kontrollitud 2020. aasta II vooru andmete põhjal. Rakendus võimaldab saada ülevaate tulemustest, kus on välja toodud iga vanuserühma kohta üldine osalejate arv, mitu poissi ja tüdrukut osales viktoriinis ning nende keskmine tulemus. Lisaks on keskmine tulemus välja toodud iga ülesande kohta eraldi. Tulemuste visualiseerimisvahend lubab näha punktide tabelit, tulemusi ülesannete tüüpide kaupa, kus viktoriinis toodud küsimused on grupeeritud vastavalt teemale ja iga ülesannete tüübi juures saab näha protsentuaalset õigete, valede ning osaliselt õigete või puuduvate vastuste osakaalu. On lisatud ka kirjeldava statistika graafikud, kus on koondatud üldised andmed ning eraldi andmed poiste ja tüdrukute kohta. Iga küsimuse kohta ja ülesannete tüüpide kaupa on välja toodud maksimaalne ja minimaalne tulemus, keskmine ning standardhälve. Rakendus lubab teha edasist analüüsi, näiteks liigutada saadud andmed tabelisse ja kontrollida, kas poiste ja tüdrukute tulemused olid statistiliselt oluliselt erinevad või mitte ehk mis ülesandega said poisid või tüdrukud paremini hakkama.

Käesoleva töö üheks eesmärgiks oli tagada võimalus tutvuda ülesannetega sellisel viisil, et õpilane saaks lisaks vastusele näha ka täpsemat seletust, kuidas lahenduseni jõuda ja kui palju on ülesanne informaatikaga seotud. Selle eesmärgi saavutamiseks oli loodud interaktiivne veebipõhine ülesannete kogumik, kus on toodud 45 küsimust 2020. aasta II voor ülesannetega koos vastuste, seletuste ja informaatika seostega. Kogumik lubab kolme vanuserühma ja ülesannete vahel liikumist. Ülesannete vastust on võimalik peita või kuvada vastavalt soovile.

Mõeldes tulevastele informaatikaviktoriinidele, valmisid autori poolt 10 uut küsimust, mis on esitatud eesti, vene ja inglise keeles, et oleks võimalik need lisada rahvusvahelise viktoriini küsimuste pank.

Viidatud kirjandus

- [1] Bebras – International Challenge on Informatics and Computational Thinking. <https://www.bebas.org/> (20.12.2020)
- [2] Bebras history. <https://www.bebas.org/index.html%3Fq=history.html> (18.10.2020)
- [3] Bebras statistics. <https://www.bebas.org/index.html%3Fq=statistics.html> (18.10.2020)
- [4] Structure of a challenge. <https://www.bebas.org/index.html%3Fq=structure.html> (20.10.2020)
- [5] Kobras – informaatikavõistlus üldhariduskoolidele. <https://kobras.eio.ee/> (12.11.2020)
- [6] Bebras countries. <https://www.bebas.org/index.html%3Fq=countries.html> (05.12.2020)
- [7] Архів завдань. <http://bober.net.ua/page.php?name=archive&> (10.12.2020)
- [8] Über den Biber der Informatik – Statistiken. <https://www.ocg.at/node/269> (11.12.2020)
- [9] Aufgabensammlung. <https://bwinf.de/biber/archiv/aufgabensammlung/> (12.12.2020)
- [10] Archives. <https://www.bebas.be/archives> (12.12.2020)
- [11] Averages and Cutoffs. <https://www.cemc.uwaterloo.ca/contests/statistics.html> (13.12.2020)
- [12] Global statistics. <http://ibic.kangaroo.org.pk/global-statistics> (13.12.2020)
- [13] Výsledky národního kola 2020. <https://www.ibobr.cz/vysledky-a-statistiky> (14.12.2020)
- [14] Štatistické vyhodnotenie 2020/2021. <http://ibobor.sk/vyhodnotenie2020.php> (15.12.2020)
- [15] Pistikmoodul Chart.js. <https://github.com/chartjs/Chart.js> (06.06.2020)
- [16] Using XMLHttpRequest. https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest/Using_XMLHttpRequest (20.05.2020)
- [17] What is a Bebras task. <https://www.bebas.org/index.html%3Fq=goodtask.html> (18.12.2020)

Lisa 1. Uute ülesannete tekstide inglise ja venekeelsed versioonid

Ingliskeelsete versioonide vormistamisel on kasutatud viktoriini rahvusvahelise komitee poolt kinnitatud malli.

2021-EST-1: Smart program

Task Attributes

- Age Group / Difficulty: [Benjamins] / [medium], [Juniors] / [easy]
- Category: Other software
- Keywords: Conditional programming
- Answer type: Multiple-choice

Smart program

Beaver Danek watches his favorite cartoon every evening. The cartoon always starts at the same time at 8:00 P.M. Daddy beaver wrote a program that at the right time automatically turns the TV on. The program also controls that there is electricity in the house and the beaver Danek is at home. The program condition has the following form:

IF (condition1, condition2, condition3)

THEN command1

ELSE command2

END

It is known that *command1* is executed only if all conditions are true and the conditions are checked from left to right. *Command2* is executed in all other cases.

Question

In what order must conditions be checked to minimize the number of checks?

- Beaver is at home, There is electricity, Time is 8:00 P.M.
- Time is 8:00 P.M., Beaver is at home, There is electricity.
- There is electricity, Time is 8:00 P.M., Beaver is at home.
- In this case the sequence is not important.

And what is executed by each command?

- Command1 – turn the TV off, command2 – turn the TV on.
- Command1 – turn the TV on, command2 – turn the TV off.

Correct answer

D. In this case, the sequence is not important; F. Command1 – turn the TV on, command2 – turn the TV off.

Answer explanation

It doesn't matter in what order we check conditions since each of them may be true or false.

If all conditions are true, the program will execute command1 - turn the TV on. In any other cases the TV will be turned off, which means, command 2 will be executed.

It's Informatics

The conditional statement used in this task is a programming language feature that ensures the execution of commands in accordance with the specified logical expressions. If the expression is correct, one command is executed, if not, then another.

Comments

Lidia Feklistova (Estonia) lidia.feklistova@ut.ee

Authors and Contributors (incl. Graphics)

Anastassia Fomin (Estonia) vesta33@gmail.com

Websites

- [https://en.wikipedia.org/wiki/Conditional_\(computer_programming\)](https://en.wikipedia.org/wiki/Conditional_(computer_programming))
- <https://www.youtube.com/watch?v=XTf0n3CVx4g>

Умная программа

Автор: Анастасия Фомин (Эстония)

Бобрёнок Дима смотрит свой любимый мультик каждый вечер. Мультик всегда начинается в одно и тоже время в 20:00. Папа бобр написал программу, которая автоматически включает телевизор только в определённое время.

Программа также следит за тем, чтобы в доме было электричество и, конечно же, сам бобрёнок был бы дома.

Условие программы записывается в следующем виде:

ЕСЛИ (условие1, условие2, условие3)

ТОГДА команда1

ИНАЧЕ команда2

КОНЕЦ

Известно, что *команда1* выполняется только в том случае, если все условия окажутся верными, и условия проверяются слева направо. *Команда2* выполняется в любом другом случае.

Вопрос

В какой последовательности должны выполняться условия данной программы, чтобы каждый раз количество проверок было минимальным?

- A. Бобрёнок дома, В доме есть электричество, Время 20:00.
- B. Время 20:00, Бобрёнок дома, В доме есть электричество.
- C. В доме есть электричество, Время 20:00, Бобрёнок дома.
- D. В данном задании последовательность не важна.

И что выполняет каждая из команд?

- E. Команда1 – выключает телевизор, команда2 – включает телевизор.
- F. Команда1 – включает телевизор, команда2 – выключает телевизор.

Правильный ответ

D. В данном задании последовательность не важна; F. Команда1 – включает телевизор, команда2 – выключает телевизор.

Объяснение решения

Не важно в каком порядке мы будем проверять условия, так как каждое из них может оказаться верным или ложным.

Если все условия окажутся верными, то программа выполнит команду 1 – включит телевизор. В любом другом случае, телевизор будет выключен, то есть будет выполняться команда 2.

Это информатика, потому что

Условный оператор, используемый в данном задании, является конструкцией языка программирования и обеспечивает выполнение команд в соответствии с заданными логическими выражениями. Если выражение верное, выполняется одна команда, если нет, то другая.

2021-EST-02: Observation post

Task Attributes

- Age Group / Difficulty: [Benjamins] / [medium], [Juniors] / [easy]
- Category: Other software
- Keywords: virtual reality, stereoscopic displays
- Answer type: multiple-choice

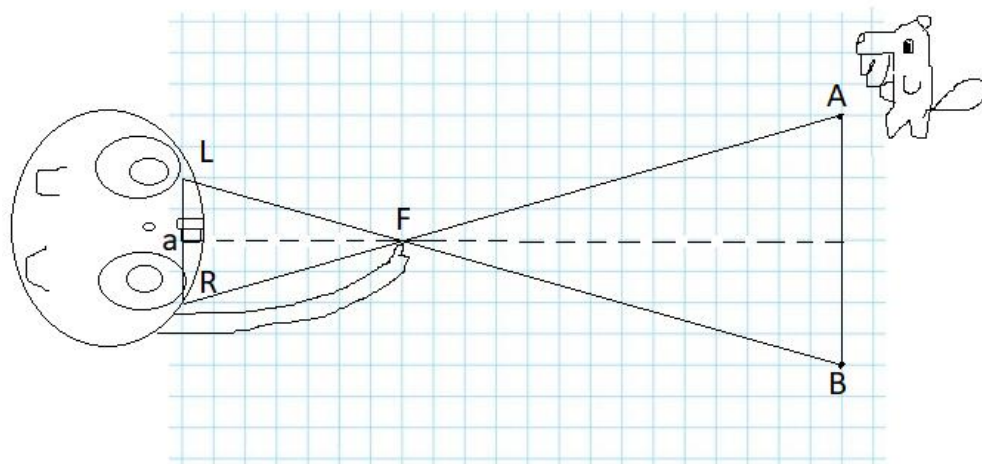
Observation post

Beaver Robin is in the observation post. He must report to the headquarters about all objects seen and the distance to them.

Looking closely at what has been happening for a long time, he noticed one beaver in the distance, who was moving with slow steps in a straight line from point A to point B and back.

Beaver Robin took a perpendicular position relative to the moving beaver and extended his paw in his fist, with his thumb raised up. When the thumb was hovering over the beaver in the distance, Robin closed his left eye. When the stranger reached point A (where the gaze was directed through an outstretched finger), the beaver Robin abruptly closed his right eye and opened his left eye. At this moment, the line of sight from point A moved to point B. When the finger shifted due to a change in the direction of the gaze, Robin started counting the steps of the beaver walking in the distance and counted that 20 steps were taken from point A to point B. It is known that the length of one step is $\frac{3}{4}$ meters, so we get that the distance from A to B is $20 * \frac{3}{4} = 15$ meters.

It is also known that the distance from the left eye to the right eye is 6 cm (LR), and the distance from the eye level to the thumb is 60 cm (aF).



Robin lowered his paw and decided to calculate how far away the stranger was from him.

Question

What is the distance to the stranger?

Answer options

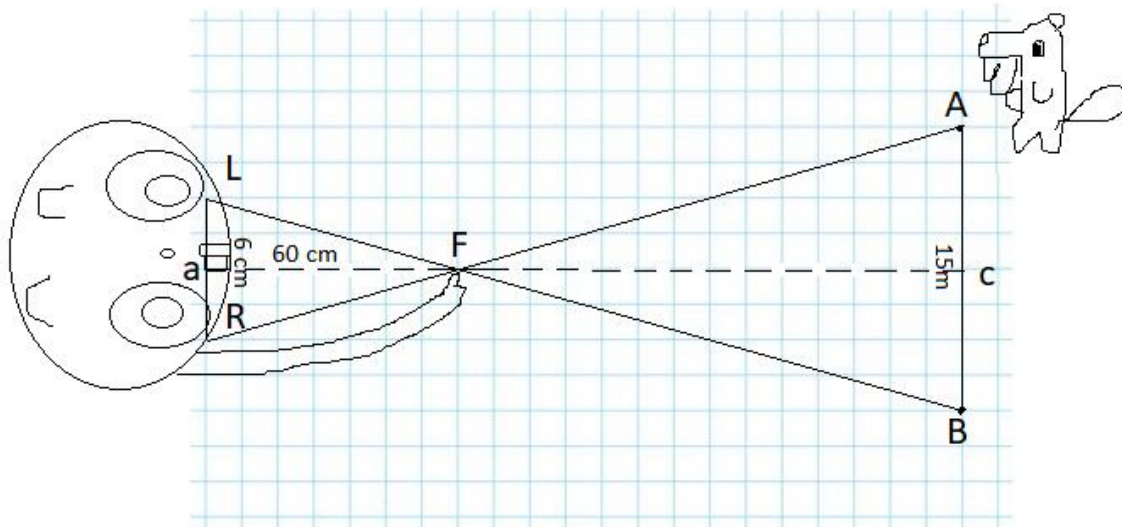
- A. 150 m
- B. 150,6 m
- C. 300 m
- D. 360 m

Correct answer

B: 150,6 m

Answer explanation

This problem is solved based on the similarity of triangles. We know that the distance between the left and right eyes is $LR = 6$ cm, and the distance from eye level to the thumb of the outstretched paw is $aF = 60$ cm. Their ratio can be represented as $60/6$. Let us denote a point in the middle of the segment AB by the letter c . This way, we get the equation $Fc / AB = aF / LR$, where AB is equal to 15 m. We get $Fc / 15 = 60/6$, whence $Fc = 15 * 10 = 150$ m. The distance from eye level to the thumb of the outstretched paw must be added to the last number, and thus we obtain that the distance to the stranger = 150 m + 60 cm = 150.6 m.



It's Informatics

Virtual reality gives the ability to be moved to places that we have never seen before, to feel something new. Glasses, the principle of which is based on the effect of stereoscopy, allow you to feel the fullness of the virtual world. The screen displays two images – one for the right eye, the other slightly modified for the left. There is a partition in the middle.

The eyes are at different angles relative to the subject in the field of view, which explains the slightly different angle and depth. However, when both eyes are open, we do not notice this since the brain combines the received data into one volumetric picture.

Stereoscopic images can be taken by placing two cameras on a tripod so that they are at the same height and the distance between the centers of the lenses is at least 6 cm.

Wording

*line of sight - visual axis or line of sight between an observer and a subject of interest

Comments

Lidia Feklistova (Estonia) lidia.feklistova@ut.ee

Files

Problem.png (original picture)

Answer.png (original solution)

Authors and Contributors (incl. Graphics)

Anastassia Fomin (Estonia) vesta33@gmail.com

Websites

- <http://osledah.ru/orientirovanie/kak-s-pomoshhyu-paltsa-mozhno-uznat-rasstoyanie-do-cheloveka-idushhego-vdaleke>
- <https://make-3d.ru/articles/kak-rabotayut-vr-ochki/>
- <https://fotogora.ru/что-такое-стереоскопия-и-что-о-ней-нужно-знать/>

Наблюдательный пункт

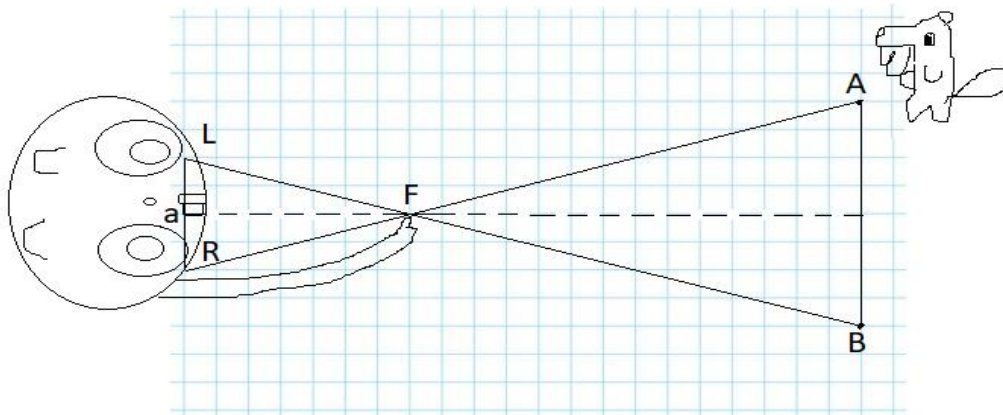
Автор: Анастасия Фомин (Эстония)

Бобр Робин находится в наблюдательном пункте. Он должен сообщать в штаб о всех увиденных объектах и расстоянии до них.

Долго присматриваясь к происходящему, он заметил вдали одного бобра, который медленными шагами передвигался по прямой от пункта А к пункту В и обратно.

Бобр Робин занял перпендикулярное положение относительно движущегося бобра и вытянул лапу в кулаке с приподнятым кверху большим пальцем. Когда большой палец был наведён на фигуру чужака вдали, бобр Робин закрыл левый глаз. Когда чужак достиг пункта А (куда был направлен взгляд через вытянутый палец), бобр Робин резко закрыл правый глаз и открыл левый. В этот момент луч зрения (взгляд) из точки А переместился в точку В. Когда палец сместился, вследствие перемены направления взгляда, Робин приступил к подсчёту шагов идущего вдали бобра и насчитал, что из пункта А в пункт В было сделано 20 шагов. Известно, что длина одного шага равна $\frac{3}{4}$ метра, таким образом получим, что расстояние от А до В равно $20 * \frac{3}{4} = 15$ метров.

Ещё известно, что расстояние от левого глаза до правого равно 6 см (LR), а расстояние от уровня глаз до пальца вытянутой лапы равно 60 см (aF).



Робин опустил лапу и решил рассчитать, на каком расстоянии от него находится чужак.

Вопрос

Какое расстояние до бобра вдали?

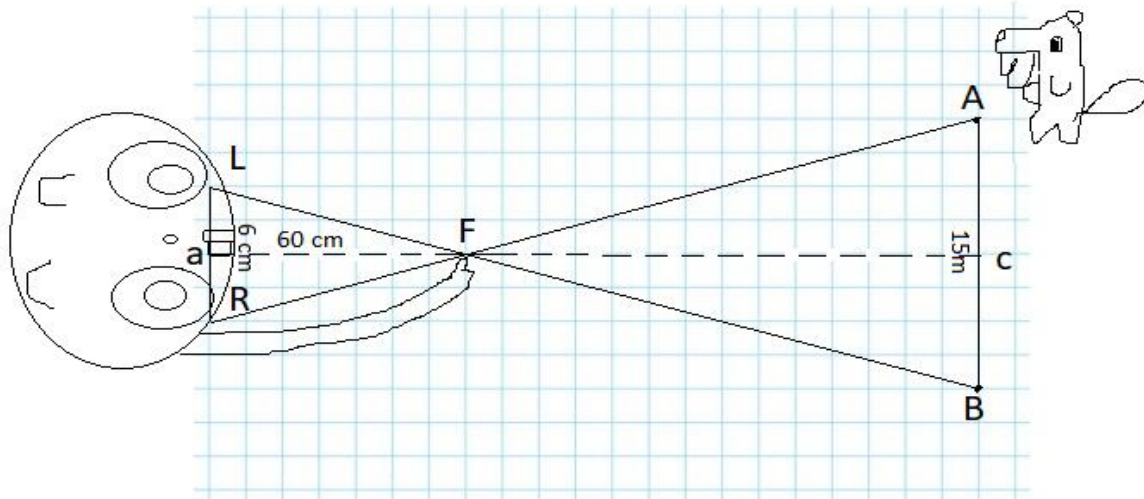
- A. 150 м
- B. 150,6 м
- C. 300 м
- D. 360 м

Правильный ответ

В: 150,6 м

Объяснение решения

Данная задача решается на основе подобия треугольников. Нам известно, что расстояние между левым и правым глазом равно $LR = 6$ см, а расстояние от уровня глаз до пальца вытянутой лапы $aF = 60$ см. Их соотношение можно представить как $60/6$. Обозначим точку посередине отрезка AB буквой c , получим уравнение $Fc/AB = aF/LR$, где AB по условию задачи равно 15 м. Получим $Fc/15 = 60/6$, откуда $Fc = 15 * 10 = 150$ м. К последнему числу надо прибавить расстояние от уровня глаз до пальца вытянутой лапы и таким образом получим, что расстояние до чужака = $150 \text{ м} + 60 \text{ см} = 150,6 \text{ м}$.



Это информатика, потому что

Виртуальная реальность позволяет нам переместиться в места, в которых мы никогда не были. Ощутить полноту виртуального мира позволяют очки, принцип работы которых основывается на эффекте стереоскопии. На экране отображается две картинки — одна для правого глаза, другая слегка видоизменённая для левого. По середине располагается перегородка.

Глаза находятся под разными углами относительно объекта в поле зрения, что объясняет немного отличающийся ракурс и глубину. Однако, когда открыто оба глаза, мы этого не замечаем, так как мозг совмещает полученные данные в одну объёмную картинку.

Стереоскопические изображения можно сделать, расположив две камеры на штативе так, чтобы они были на одной высоте и расстояние между центрами объективов было не менее 6 см.

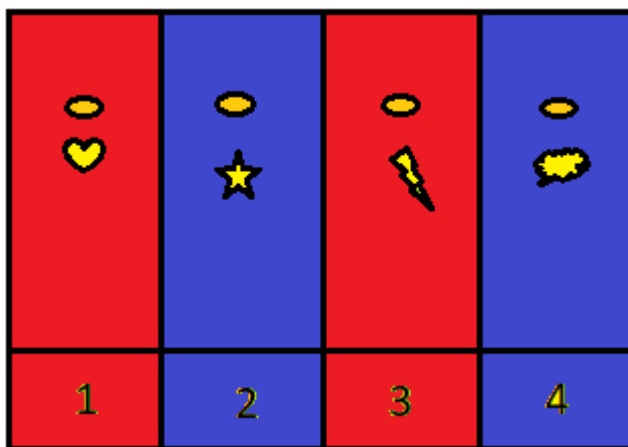
2021-EST-03: Lockers

Task Attributes

- Age Group / Difficulty: [Benjamins] / [medium], [Juniors] / [easy]
- Category: Other software
- Keywords: Combinatorics
- Answer type: open integer

Lockers

Bebras Ann, Ben, Chris, and Diana go to the kindergarten for the first time. The teacher is preparing lockers for them so that the lockers of two boys and two girls are not next to each other.



Question

How many ways are there to distribute lockers without breaking the rule?

Correct answer

8

Answer explanation

There are 2 ways to distribute lockers the way that lockers for two boys and two girls are not next to each other:

1. BOY, GIRL, BOY, GIRL
2. GIRL, BOY, GIRL, BOY

In order to choose which of the two boys will own the locker located on the left side, there are two possibilities – the locker will belong to Ben or Chris. Likewise, in the case of two girls – the locker will belong to Ann or Diana.

As a result, we get $2*2*2=8$ ways.

The task can be solved in another, less optimal way. To do this, you need to write down all the possible ways and calculate them:

1. Ben, Ann, Chris, Diana
2. Ben, Diana, Chris, Ann
3. Chris, Ann, Ben, Diana
4. Chris, Diana, Ben, Ann
5. Ann, Ben, Diana, Chris
6. Ann, Chris, Diana, Ben
7. Diana, Ben, Ann, Chris
8. Diana, Chris, Ann, Ben

It's Informatics

This task is directly related to combinatorics – a branch in mathematics that studies discrete objects. Combinatorics is used in many areas, including computer science, for example, for brute-force passwords, in a graph finding the shortest path by enumeration possible options, in network routing.

Comments

Lidia Feklistova (Estonia) lidia.feklistova@ut.ee

Files

Picture created by using the Paint tool.

Authors and Contributors (incl. Graphics)

Anastassia Fomin (Estonia) vesta33@gmail.com

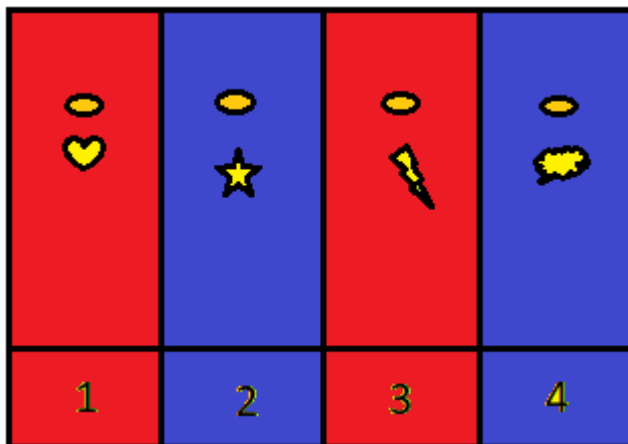
Websites

- https://acmp.ru/asp/do/index.asp?main=section&id_course=2&id_section=16
- http://mathprofi.ru/zadachi_po_kombinatorike_primery_reshenij.html
- <https://qna.habr.com/q/193497>
- <https://en.wikipedia.org/wiki/Combinatorics>

Шкафчики

Автор: Анастасия Фомин (Эстония)

Бобрята Аня, Боря, Ваня, Галя идут в детский сад. Воспитатель подготавливает им шкафчики таким образом, чтобы шкафчики двух мальчиков и двух девочек не оказались рядом.



Вопрос

Сколько всего способов раздать шкафчики с соблюдением условия существует?

Правильный ответ

8 способов

Объяснение решения

Существует 2 способа раздать шкафчики так, чтобы шкафчики для двух мальчиков и двух девочек не были рядом:

1. МАЛЬЧИК, ДЕВОЧКА, МАЛЬЧИК, ДЕВОЧКА
2. ДЕВОЧКА, МАЛЬЧИК, ДЕВОЧКА, МАЛЬЧИК

Для того, чтобы выбрать, кому из двух мальчиков будет принадлежать расположенный с левой стороны шкафчик, существует две возможности – шкафчик будет принадлежать Ване или Боре. Аналогично и в случае с двумя девочками – шкафчик будет принадлежать Ане или Гале.

В итоге получаем $2*2*2=8$ способов.

Задание можно решить и другим, менее оптимальным, способом. Для этого необходимо записать все возможные способы и подсчитать их:

1. Ваня, Аня, Боря, Галя
2. Ваня, Галя, Боря, Аня
3. Боря, Аня, Ваня, Галя
4. Боря, Галя, Ваня, Аня
5. Аня, Ваня, Галя, Боря
6. Аня, Боря, Галя, Ваня
7. Галя, Ваня, Аня, Боря
8. Галя, Боря, Аня, Ваня

Это информатика, потому что

Данная задача напрямую связана с комбинаторикой – разделом в математике, который изучает дискретные объекты. Комбинаторика используется во многих областях, в том числе и в информатике, например для перебора паролей, нахождения кратчайшего пути в графе с помощью перебора возможных вариантов, в маршрутизации сетей.

2021-EST-04: Summer house

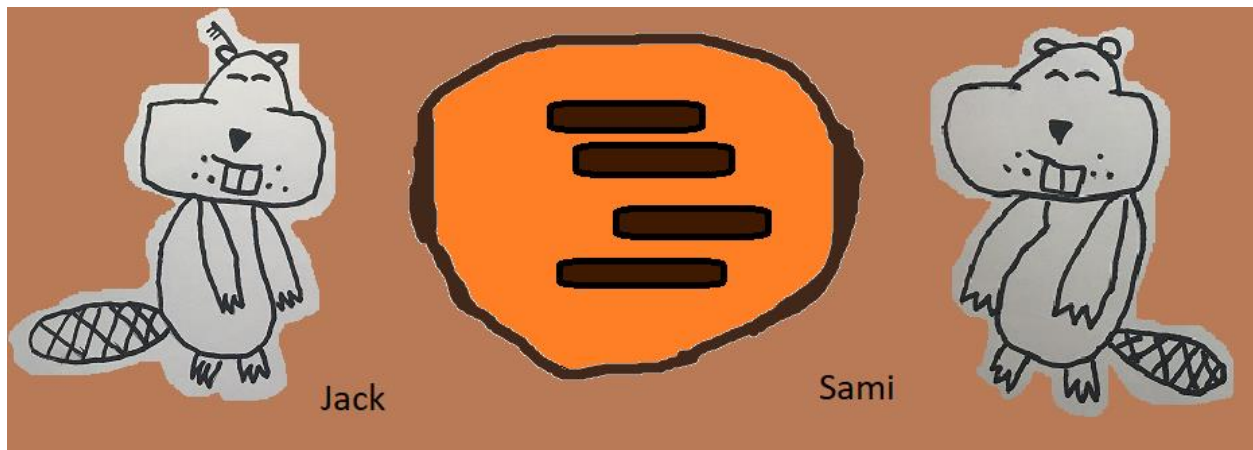
Task Attributes

- Age Group / Difficulty: [Benjamins] / [hard], [Juniors] / [medium]
- Category: Other software
- Keywords: Clock signal, clock rate
- Answer type: open integer

Summer house

Two beavers Jack and Sami are building a summer log house. At the same time, Jack carries the logs to the warehouse, and Sami takes them to the construction site. Jack is a very strong beaver and brings 2 logs at a time, and he also moves at a constant speed of 0.5 meters per second. Sami moves at a speed of 1.5 meters per second and can carry only one log. Both beavers start from the same distance - Jack from the felling site, located at a distance of 1.5 m from the warehouse, and Sami from the construction site, which is also 1.5 m away from the warehouse.

After the first second, when Sami arrives at the warehouse, there are no logs yet. It is known that if there is no log, Sami returns to the construction site and continues doing other things not to waste time. The time for loading and unloading a log is not counted, thus it is equal to 0 seconds.



Question

How many logs can Sami drag in 10 seconds?

Correct answer

3

Answer explanation

In 10 seconds, Jack will have enough time for two runs and thus will bring 4 logs. This can be seen from the condition of the problem - the distance to the warehouse is 1.5 m and the speed of Jack is 0.5 m/sec, so he spends 3 seconds to reach the warehouse with the logs and 3 seconds to return for a new portion.

Sami will have time to visit the warehouse 5 times since in two seconds he has time to go to the warehouse and return to the construction site. Delivery of the first log to the construction site will take him 4 seconds, the second log will be brought to the construction site in another 2 seconds. At the seventh second, Sami will again be left with empty hands, but at the 10th second, he will deliver the third log to the construction site.

It's Informatics

This task is related to the concept of “processor clock speed”. Beaver Jack acts as a clock frequency generator, sets the clock, and thus affects the speed and operation of Sami, who in this task works as a processor. A computer’s processor is capable of performing millions of operations per second and its speed is measured in GHz.

Comments

Lidia Feklistova (Estonia) lidia.feklistova@ut.ee

Files

Picture created by using the Paint tool.

Authors and Contributors (incl. Graphics)

Anastassia Fomin (Estonia) vesta33@gmail.com

Websites

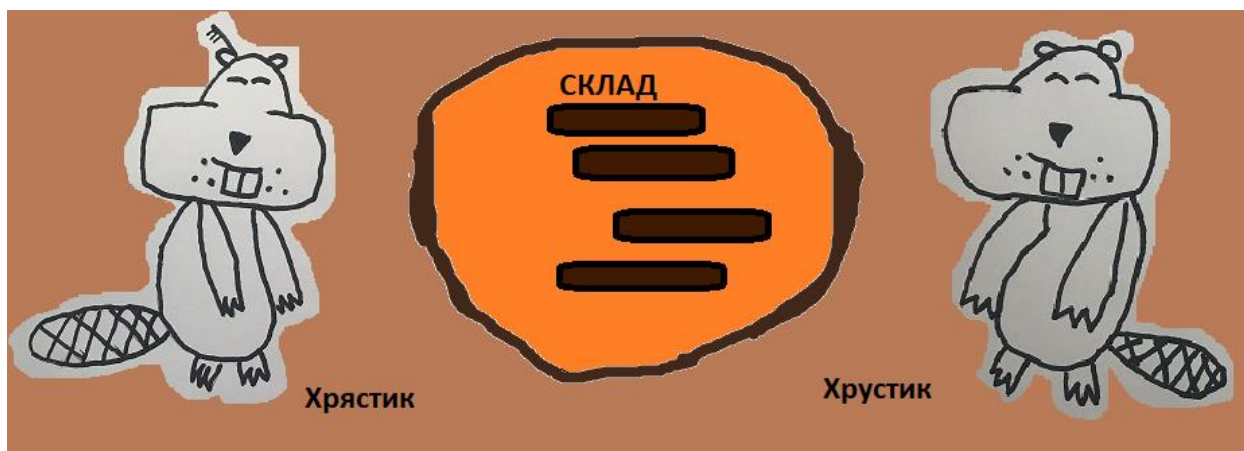
- https://en.wikipedia.org/wiki/Clock_signal
- <http://sgpek.ru/files/electronbook/AEVM/9.html>
- https://ru.qwe.wiki/wiki/Clock_rate

Летний домик

Автор: Анастасия Фомин (Эстония)

Два бобра Хрястик и Хрустик строят летний домик из брёвен. При этом Хрястик носит брёвна на склад, а Хрустик их забирает и относит на стройку. Хрястик очень сильный бобр и за один раз приносит 2 бревна, а также он передвигается с постоянной скоростью 0,5 м в секунду. Хрустик же передвигается со скоростью 1,5 м в секунду и может унести только одно бревно. Оба бобра стартуют с одинакового расстояния – Хрястик от лесоповала, расположенного на расстоянии 1,5 м от склада, а Хрустик от стройки, которая тоже находится на расстоянии 1,5 м от склада.

Придя на склад через одну секунду, Хрустик не застаёт там брёвен. Известно, что если бревна нет, Хрустик возвращается на стройку и занимается другими делами, чтобы не терять время. Время на погрузку и разгрузку бревна не учитывается, то есть равно 0 секунд.



Вопрос

Сколько брёвен успеет перетаскать Хрустик за 10 секунд?

Правильный ответ

3 бревна

Объяснение решения

За 10 секунд Хрястик успеет сделать всего 2 захода и принести таким образом 4 бревна. Это видно из условия задачи - расстояние до склада 1,5 м и скорость Хрястика 0,5 м/сек, таким образом он тратит 3 секунды, чтобы дойти с брёвнами до склада и 3 секунды, чтобы вернуться за новой порцией.

Хрустик успеет побывать на складе 5 раз, так как за две секунды он успевает сходить на склад и вернуться обратно на стройку. Доставка первого бревна на стройку займёт у него четыре секунды, второе бревно будет принесено на стройку еще через 2 секунды. На седьмой секунде Хрустик вновь останется ни с чем, но на 10 секунде доставит на стройку третье бревно.

Это информатика, потому что

Данное задание связано с понятием “тактовая частота процессора”. Бобр Хрястик выступает в роли генератора тактовой частоты, задаёт такт и таким образом влияет на скорость быстроедействия и работу Хрустика, который в данном задании предстает в роли процессора. Процессор компьютера способен выполнять миллионы операций в секунду и его скорость измеряется в ГГц.

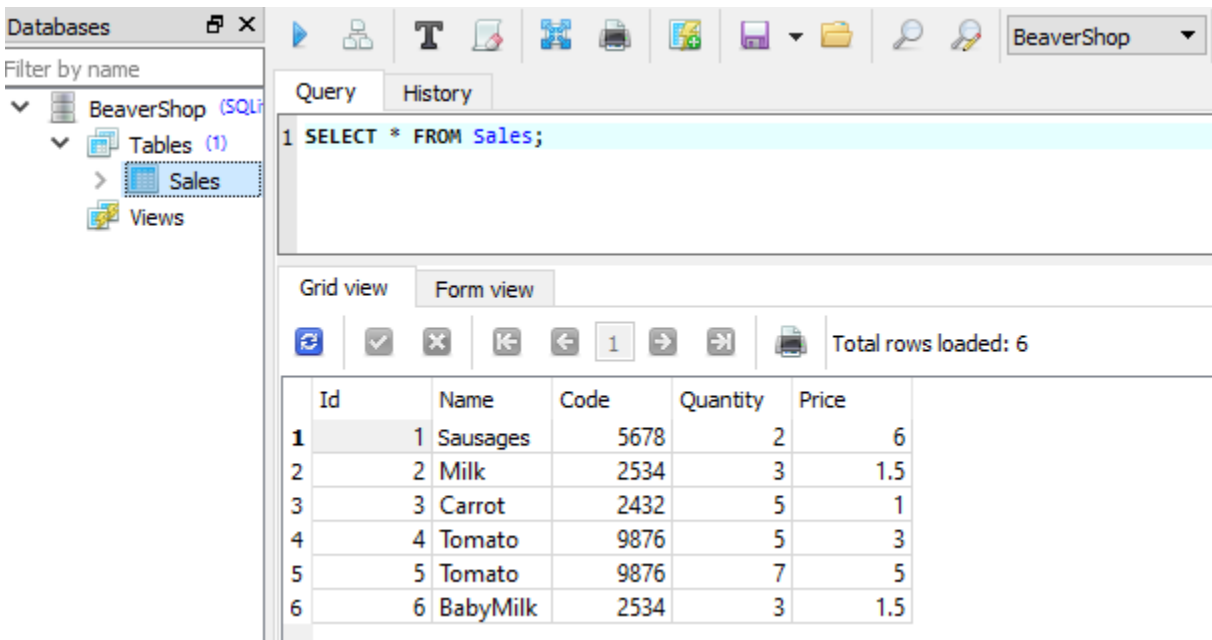
2021-EST-05: Item accounting

Task Attributes

- Age Group / Difficulty: [Senior] / [hard]
- Category: Other software
- Keywords: SQL
- Answer type: multiple-choice

Item accounting

Even in a small Bebras store, for an accounting of the goods, it is necessary to use tables to facilitate further data processing. Bebras manager has a Sales table:



The screenshot shows a database management interface for a database named 'BeaverShop'. The 'Sales' table is selected in the left-hand pane. The main window displays a query: `1 SELECT * FROM Sales;`. Below the query, the results are shown in a grid view. The grid has 6 rows and 5 columns: Id, Name, Code, Quantity, and Price. The data is as follows:

Id	Name	Code	Quantity	Price
1	1 Sausages	5678	2	6
2	2 Milk	2534	3	1.5
3	3 Carrot	2432	5	1
4	4 Tomato	9876	5	3
5	5 Tomato	9876	7	5
6	6 BabyMilk	2534	3	1.5

He needs to:

- find the number of sold products (Total) with the same code (Code);
- present data in the form of three columns named *Code of item*, *Name*, *Total*;
- arrange data in descending order.

Question

Which of the following queries is correct to meet all requirements?

Answer options

- A. *SELECT Code AS 'Code of item', Name AS 'Name', SUM(Quantity) AS Total FROM SALES GROUP BY Id ORDER BY Total DESC;*
- B. *SELECT Code AS 'Code of item', Name AS 'Name', SUM(Quantity) AS Total FROM SALES GROUP BY Name ORDER BY Total DESC;*
- C. *SELECT Code AS 'Code of item', Name AS 'Name', SUM(Quantity) AS Total FROM SALES GROUP BY Code ORDER BY Total DESC;*
- D. *SELECT Code AS 'Code of item', Name AS 'Name', SUM(Quantity) AS Total FROM SALES;*
- E. *SELECT Code AS 'Code of item', SUM(Quantity) AS Total FROM SALES GROUP BY Code ORDER BY Total DESC;*

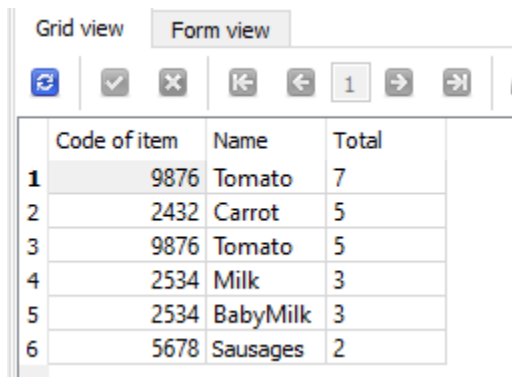
Correct answer

- C. *SELECT Code AS 'Code of item', Name AS 'Name', SUM(Quantity) AS Total FROM SALES GROUP BY Code ORDER BY Total DESC;*

Answer explanation

Let's see explanation for each query:

- A. *SELECT Code AS 'Code of item', Name AS 'Name', SUM(Quantity) AS Total FROM SALES GROUP BY Id ORDER BY Total DESC;*



The screenshot shows a database query result viewer with a table containing the following data:

	Code of item	Name	Total
1	9876	Tomato	7
2	2432	Carrot	5
3	9876	Tomato	5
4	2534	Milk	3
5	2534	BabyMilk	3
6	5678	Sausages	2

The result is presented in the form of three columns *Code of item*, *Name*, *Total*. The data is arranged in descending order, but is grouped by the *Id* parameter, not the *Code*, as stated in the condition.

B. *SELECT Code AS 'Code of item', Name AS 'Name', SUM(Quantity) AS Total FROM SALES GROUP BY Name ORDER BY Total DESC;*

	Code of item	Name	Total
1	9876	Tomato	12
2	2432	Carrot	5
3	2534	BabyMilk	3
4	2534	Milk	3
5	5678	Sausages	2

The result is presented in the form of three columns *Code of item*, *Name*, *Total*. The data is arranged in descending order, but is grouped by the *Name* parameter and not the *Code* parameter. Since the name of each product is not unique in this case, this option does not suit us.

C. *SELECT Code AS 'Code of item', Name AS 'Name', SUM(Quantity) AS Total FROM SALES GROUP BY Code ORDER BY Total DESC;*

	Code of item	Name	Total
1	9876	Tomato	12
2	2534	BabyMilk	6
3	2432	Carrot	5
4	5678	Sausages	2

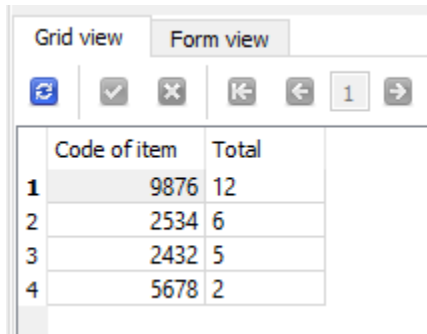
All requirements are met and this is the correct answer.

D. *SELECT Code AS 'Code of item', Name AS 'Name', SUM(Quantity) AS Total FROM SALES;*

	Code of item	Name	Total
1	2534	BabyMilk	25

In this query, the data is not grouped and returned the total of all sold products.

E. *SELECT Code AS 'Code of item', SUM(Quantity) AS Total FROM SALES GROUP BY Code ORDER BY Total DESC;*



	Code of item	Total
1	9876	12
2	2534	6
3	2432	5
4	5678	2

The result is presented only in the form of two columns *Code of item*, *Total*.

It's Informatics

Data tables are the foundation for a database. In order to retrieve data from tables, you need to know how the query is made, by which commands it is executed. When working with very large tables, you need to be especially careful, otherwise, the query may take several hours, the result may be incorrect, or even lead to a malfunction of the database.

Comments

Lidia Feklistova (Estonia) lidia.feklistova@ut.ee

Files

Screenshots from SQLiteStudio (3.2.1)

Authors and Contributors (incl. Graphics)

Anastassia Fomin (Estonia) vesta33@gmail.com

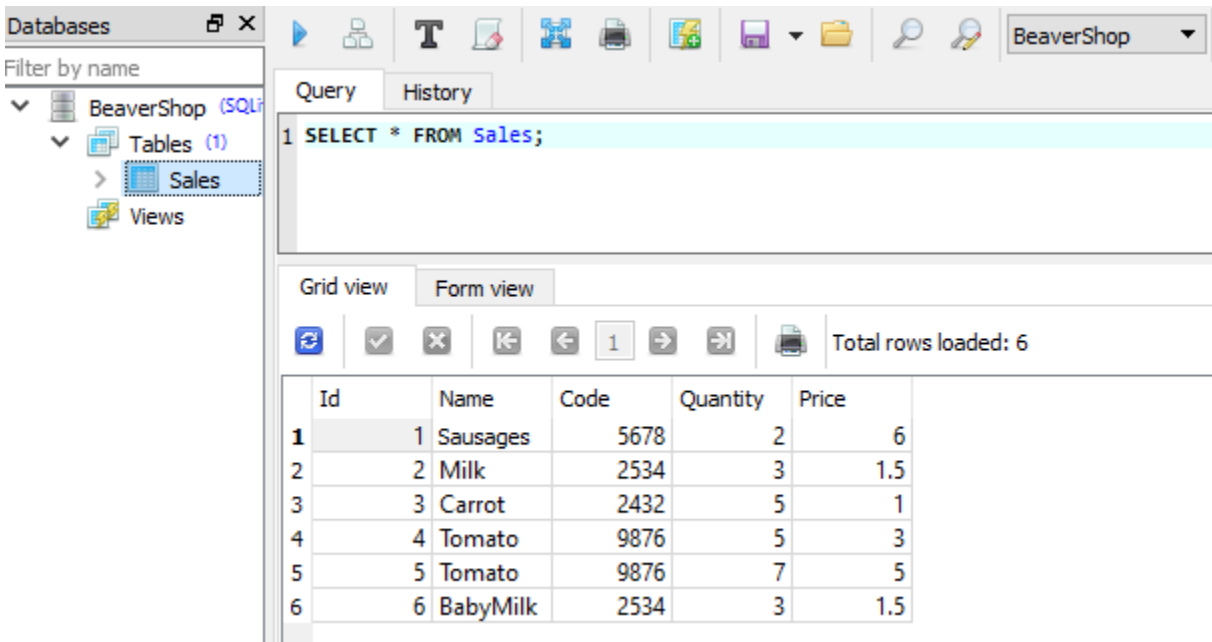
Websites

- <https://www.mysqltutorial.org/mysql-sum/>
- <https://www.mysqltutorial.org/mysql-order-by/>
- <https://www.mysqltutorial.org/mysql-group-by.aspx>
- [https://en.wikipedia.org/wiki/Table_\(database\)](https://en.wikipedia.org/wiki/Table_(database))

Учёт товара

Автор: Анастасия Фомин (Эстония)

Даже в маленьком магазине Бобринска для учёта товаров необходимо использовать таблицы, позволяющие облегчить дальнейшую обработку данных. У бобра менеджера есть таблица Продаж (*Sales*):



The screenshot shows a database management interface for a database named 'BeaverShop'. The 'Tables (1)' folder is expanded to show the 'Sales' table. The query editor contains the SQL statement: `1 SELECT * FROM Sales;`. The results are displayed in a grid view with 6 rows and 5 columns: Id, Name, Code, Quantity, and Price.

Id	Name	Code	Quantity	Price
1	Sausages	5678	2	6
2	Milk	2534	3	1.5
3	Carrot	2432	5	1
4	Tomato	9876	5	3
5	Tomato	9876	7	5
6	BabyMilk	2534	3	1.5

Ему надо:

- найти количество (*Total*) проданных товаров с одинаковым кодом (*Code*);
- представить данные в виде трёх столбцов, имеющих названия *Code of item*, *Name*, *Total*;
- расположить данные в порядке убывания.

Вопрос

Какой из следующих запросов соответствует всем требованиям?

- SELECT Code AS 'Code of item', Name AS 'Name', SUM(Quantity) AS Total FROM SALES GROUP BY Id ORDER BY Total DESC;*
- SELECT Code AS 'Code of item', Name AS 'Name', SUM(Quantity) AS Total FROM SALES GROUP BY Name ORDER BY Total DESC;*
- SELECT Code AS 'Code of item', Name AS 'Name', SUM(Quantity) AS Total FROM SALES GROUP BY Code ORDER BY Total DESC;*
- SELECT Code AS 'Code of item', Name AS 'Name', SUM(Quantity) AS Total FROM SALES;*
- SELECT Code AS 'Code of item', SUM(Quantity) AS Total FROM SALES GROUP BY Code ORDER BY Total DESC;*

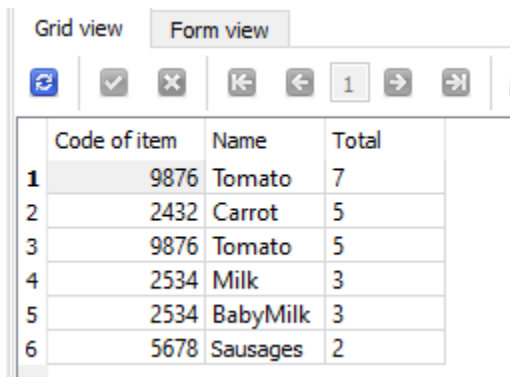
Правильный ответ

C. `SELECT Code AS 'Code of item', Name AS 'Name', SUM(Quantity) AS Total FROM SALES GROUP BY Code ORDER BY Total DESC;`

Объяснение решения

Рассмотрим по очереди каждый запрос:

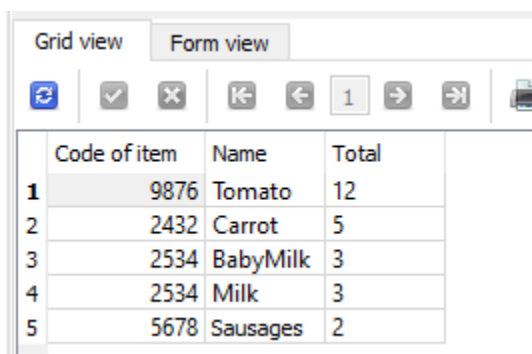
A. `SELECT Code AS 'Code of item', Name AS 'Name', SUM(Quantity) AS Total FROM SALES GROUP BY Id ORDER BY Total DESC;`



	Code of item	Name	Total
1	9876	Tomato	7
2	2432	Carrot	5
3	9876	Tomato	5
4	2534	Milk	3
5	2534	BabyMilk	3
6	5678	Sausages	2

Результат представлен в виде трёх столбцов *Code of item*, *Name*, *Total*. Данные расположены в порядке убывания, но группируются по параметру *Id*, а не *Code*, как сказано в условии.

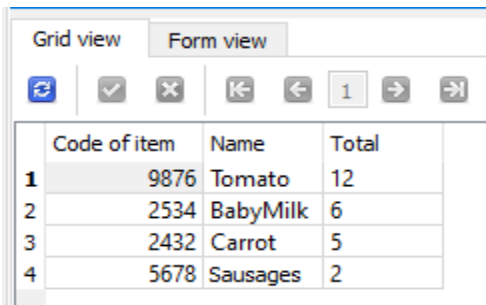
B. `SELECT Code AS 'Code of item', Name AS 'Name', SUM(Quantity) AS Total FROM SALES GROUP BY Name ORDER BY Total DESC;`



	Code of item	Name	Total
1	9876	Tomato	12
2	2432	Carrot	5
3	2534	BabyMilk	3
4	2534	Milk	3
5	5678	Sausages	2

Результат представлен в виде трёх столбцов *Code of item*, *Name*, *Total*. Данные расположены в порядке убывания, но группируются по параметру *Name*, а не *Code*. Так как имя не является в данном случае уникальным для каждого товара, то данный вариант нам не подходит.

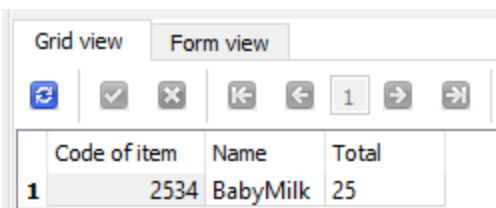
C. *SELECT Code AS 'Code of item', Name AS 'Name', SUM(Quantity) AS Total FROM SALES GROUP BY Code ORDER BY Total DESC;*



	Code of item	Name	Total
1	9876	Tomato	12
2	2534	BabyMilk	6
3	2432	Carrot	5
4	5678	Sausages	2

Все условия соблюдены, и это правильный ответ.

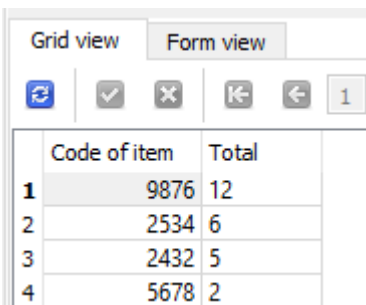
D. *SELECT Code AS 'Code of item', Name AS 'Name', SUM(Quantity) AS Total FROM SALES;*



	Code of item	Name	Total
1	2534	BabyMilk	25

В этом запросе данные не группируются, а просто находится общая сумма всех проданных товаров.

E. *SELECT Code AS 'Code of item', SUM(Quantity) AS Total FROM SALES GROUP BY Code ORDER BY Total DESC;*



	Code of item	Total
1	9876	12
2	2534	6
3	2432	5
4	5678	2

Результат представлен лишь в виде двух столбцов *Code of item, Total*.

Это информатика, потому что

Таблицы с данными являются основой для любой базы данных. Для того чтобы запрашивать данные из таблиц, необходимо знать как составляется запрос, посредством каких команд он выполняется. При работе с очень большими таблицами надо быть особенно внимательным, иначе выполнение запроса может занять несколько часов, результат может быть неверным, или вообще произойдет сбой в работе базы данных.

2021-EST-06: Prizes

Task Attributes

- Age Group / Difficulty: [Benjamins] / [hard], [Juniors] / [medium], [Seniors] / [easy]
- Category: Spreadsheet, work with data
- Keywords: Excel IF function
- Answer type: Open text

Prizes (version 1)

Every day, answering the lesson questions, beavers receive stickers for correct answers. At the end of the school year, prizes are awarded to all pupils according to the number of collected stickers. Previous years the data were entered into a spreadsheet and processed manually. This year there were a lot of beavers at school, and it was decided to use the IF function to determine the prize automatically. This function has the following general form:

=IF(logical_expression;return_if_true;return_if_false).

The jury members wrote down the formula for drawing the prize. For example, for the first participant, the formula looked like this:

=IF(B2<10;"sweets";IF(B2<20;"chocolate";IF(B2<30;"notebook";IF(B2<40;"pen";IF(B2<50;"book";IF(B2<60;"T-shirt";IF(B2<70;"ball";IF(B2<80;"skipping rope";IF(B2<90;"constructor";"travel voucher"))))))))

However, a mistake was made, and instead of the prizes, the message "Err: 508" appeared in the cells.

	A	B	C	D	E	F
1		Ann	Ben	Ciara	Diana	Elias
2	Number of stickers	45	65	35	50	90
3	Prize	Err:508	Err:508	Err:508	Err:508	Err:508

Question

Correct the mistake and determine what prize has received each of the five beavers.

Correct answer

`=IF(B2<10;"sweets";IF(B2<20;"chocolate";IF(B2<30;"notebook";IF(B2<40;"pen";IF(B2<50;"book";IF(B2<60;"T-shirt";IF(B2<70;"ball";IF(B2<80;"skipping rope";IF(B2<90;"constructor";"travel voucher"))))))))`

	A	B	C	D	E	F
1		Ann	Ben	Ciara	Diana	Elias
2	Number of stickers	45	65	35	50	90
3	Prize	book	ball	pen	T-shirt	travel voucher

Answer explanation

The error “Err: 508” means that the formula has a closing parenthesis without a matching opening parenthesis, or there is no closing parenthesis in the middle of the formula (closing parentheses at the end of the formula are added automatically).

In this task open parentheses are missing in three places:

1. `IFB2<30;"notebook";`
2. `IFB2<50;"book";`
3. `IFB2<90;"constructor";`

By adding them, we get the working formula:

`=IF(B2<10;"sweets";IF(B2<20;"chocolate";IF(B2<30;"notebook";IF(B2<40;"pen";IF(B2<50;"book";IF(B2<60;"T-shirt";IF(B2<70;"ball";IF(B2<80;"skipping rope";IF(B2<90;"constructor";"travel voucher"))))))))`

Based on the correct formula, it is clear that Ann, who has collected 45 stickers, will receive a book. Ben with 65 stickers will receive a ball. Ciara, who has collected 35 stickers, will receive a pen. Diana, who has collected 50 stickers, will receive a T-shirt, and Elias, for 90 stickers, will receive a travel voucher.

It's Informatics

The IF function is one of the most popular functions in spreadsheets. By comparing values logically, it produces the expected result, which can be either numeric or text. Since the result can be either true or false, the function uses two response options. It is quite common to have more than one condition to be checked. In this case, using several IF functions (nesting IF functions) in one formula is a good approach to get the correct answer.

Comments

Lidia Feklistova (Estonia) lidia.feklistova@ut.ee

Files

Screenshots from OpenOffice Calc

Authors and Contributors (incl. Graphics)

Anastassia Fomin (Estonia) vesta33@gmail.com

Websites

- https://wiki.openoffice.org/wiki/Calc_Error_Codes
- <https://exceljet.net/excel-functions/excel-if-function>
- <https://support.microsoft.com/en-us/office/if-function-69aed7c9-4e8a-4755-a9bc-aa8bbff73be2?ui=en-us&rs=en-us&ad=us>

Призы (версия 1)

Автор: Анастасия Фомин (Эстония)

Каждый день, отвечая на уроке, бобрята получают наклейки за правильные ответы. В конце учебного года среди всех учащихся разыгрываются призы в соответствии с количеством собранных наклеек. В прошлые года данные заносились в электронную таблицу и обрабатывались вручную. В этом году в школе училось очень много бобрят, и для быстрого определения приза было решено использовать функцию IF, которая делает расчёты автоматически. Эта функция имеет следующий общий вид:

$=IF(\text{логическое_выражение}; \text{значение_если_истина}; \text{значение_если_ложь})$.

Для розыгрыша призов члены жюри записали формулу. Например, для первого участника формула выглядит следующим образом:

$=IF(B2 < 10; "конфета"; IF(B2 < 20; "шоколад"; IF(B2 < 30; "тетрадь"; IF(B2 < 40; "ручка"; IF(B2 < 50; "книга"; IF(B2 < 60; "футболка"; IF(B2 < 70; "мяч"; IF(B2 < 80; "скакалка"; IF(B2 < 90; "конструктор"; "поездка"))))))))$

Однако была допущена ошибка, и вместо призов в ячейках появилось сообщение об ошибке "Err:508".

	A	B	C	D	E	F
1		Аня	Боря	Валя	Галя	Дима
2	Количество наклеек	45	65	35	50	90
3	Приз	Err:508	Err:508	Err:508	Err:508	Err:508

Вопрос

Исправив ошибку, определи, какой приз получил каждый из пяти бобрят.

Правильный ответ

=IF(B2<10;"конфета";IF(B2<20;"шоколад";IF(B2<30;"тетрадь";IF(B2<40;"ручка";IF(B2<50;"книга";IF(B2<60;"футболка";IF(B2<70;"мяч";IF(B2<80;"скакалка";IF(B2<90;"конструктор";"поездка"))))))))

	A	B	C	D	E	F
1		Аня	Боря	Валя	Галя	Дима
2	Количество наклеек	45	65	35	50	90
3	Приз	книга	мяч	ручка	футболка	поездка

Объяснение решения

Ошибка "Егг:508" означает, что в формуле имеется закрывающая скобка без соответствующей открывающей скобки или отсутствует закрывающая скобка в середине формулы (закрывающие скобки в конце формулы добавляются автоматически).

В данном задании открывающие скобки пропущены в трёх местах:

1. IFB2<30;"тетрадь";
2. IFB2<50;"книга";
3. IFB2<90;"конструктор";

Добавив их, получим работающую формулу:

=IF(B2<10;"конфета";IF(B2<20;"шоколад";IF(B2<30;"тетрадь";IF(B2<40;"ручка";IF(B2<50;"книга";IF(B2<60;"футболка";IF(B2<70;"мяч";IF(B2<80;"скакалка";IF(B2<90;"конструктор";"поездка"))))))))

Исходя из правильной формулы, видно, что Аня, набравшая 45 наклеек, получит книгу. Боря, набравший 65 наклеек, получит мяч. Валя, набравшая 35 наклеек, получит ручку. Галя, набравшая 50 наклеек, получит футболку, а Дима, за 90 наклеек, получит поездку.

Это информатика, потому что

Функция ЕСЛИ (IF) – одна из самых популярных функций в электронных таблицах. Путем логического сравнения значений, она выдаёт ожидаемый результат, который может быть как числовым, так и словесным. Так как результат может быть либо истинным, либо ложным, то функции задаётся два варианта ответа. Довольно часто приходится решать случаи, когда значение сравнивается больше, чем с одним условием. В этом случае использование нескольких функций ЕСЛИ (вложение ЕСЛИ функций) в рамках одной формулы является хорошим подходом для получения правильного ответа.

2021-EST-07: Prizes

Task Attributes

- Age Group / Difficulty: [Benjamins] / [medium]
- Category: Spreadsheet, work with data
- Keywords: Excel IF function
- Answer type: multiple-choice

Prizes (version 2)

Every day, answering the lesson questions, beavers receive stickers for correct answers. At the end of the school year, prizes are awarded to all pupils according to the number of collected stickers. Previous years the data were entered into a spreadsheet and processed manually. This year there were a lot of beavers at school, and it was decided to use the IF function to determine the prize automatically. This function has the following general form:

=IF(logical_expression;return_if_true;return_if_false).

The jury members wrote down the formula for drawing the prize. For example, for the first participant, the formula looked like this:

=IF(B2<10;"sweets";IF(B2<20;"chocolate";IF(B2<30;"notebook";IF(B2<40;"pen";IF(B2<50;"book";IF(B2<60;"T-shirt";IF(B2<70;"ball";IF(B2<80;"skipping rope";IF(B2<90;"constructor";"travel voucher"))))))))

However, a mistake was made, and instead of the prizes, the message “Err: 508” appeared in the cells.

	A	B	C	D	E	F
1		Ann	Ben	Ciara	Diana	Elias
2	Number of stickers	45	65	35	50	90
3	Prize	Err:508	Err:508	Err:508	Err:508	Err:508

Question

What is the error?

Answer options

- A. Parenthesis problem
- B. You cannot display text results
- C. So many conditions in one formula cannot be used
- D. There should be no "=" sign
- E. Quotes must be removed

Correct answer

- A. Parenthesis problem

Answer explanation

The error “Err: 508” means that the formula has a closing parenthesis without a matching opening parenthesis, or there is no closing parenthesis in the middle of the formula (closing parentheses at the end of the formula are added automatically).

In this task open parentheses are missing in three places:

1. *IFB2<30;"notebook";*
2. *IFB2<50;"book";*
3. *IFB2<90;"constructor";*

The other suggested answers are incorrect. The IF function can output not only numeric results, but also text results, dates, a cell reference, or even another formula. The number of conditions can be up to 64, so the number of conditions used in this task is acceptable. The "=" sign is placed at the beginning of the function, and non-numeric values are written in quotes.

It's Informatics

The IF function is one of the most popular functions in spreadsheets. By comparing values logically, it produces the expected result, which can be either numeric or text. Since the result can be either true or false, the function uses two response options. It is quite common to have more than one condition to be checked. In this case, using several IF functions (nesting IF functions) in one formula is a good approach to get the correct answer.

Comments

Lidia Feklistova (Estonia) lidia.feklistova@ut.ee

Files

Screenshots from OpenOffice Calc

Authors and Contributors (incl. Graphics)

Anastassia Fomin (Estonia) vesta33@gmail.com

Websites

- https://wiki.openoffice.org/wiki/Calc_Error_Codes
- <https://exceljet.net/excel-functions/excel-if-function>
- <https://support.microsoft.com/en-us/office/if-function-69aed7c9-4e8a-4755-a9bc-aa8bbff73be2?ui=en-us&rs=en-us&ad=us>

Призы (версия 2)

Автор: Анастасия Фомин (Эстония)

Каждый день, отвечая на уроке, бобрята получают наклейки за правильные ответы. В конце учебного года среди всех учащихся разыгрываются призы в соответствии с количеством собранных наклеек. В прошлые года данные заносились в электронную таблицу и обрабатывались вручную. В этом году в школе училось очень много бобрят, и для быстрого определения приза было решено использовать функцию IF, которая делает расчёты автоматически. Эта функция имеет следующий общий вид:

=IF(логическое_выражение;значение_если_истина;значение_если_ложь).

Члены жюри записали формулу, чтобы разыграть призы. Например, для первого участника формула выглядит так:

=IF(B2<10;"конфета";IF(B2<20;"шоколад";IF(B2<30;"тетрадь";IF(B2<40;"ручка";IF(B2<50;"книга";IF(B2<60;"футболка";IF(B2<70;"мяч";IF(B2<80;"скакалка";IF(B2<90;"конструктор";"поездка"))))))))

Однако жюри допустило ошибку, и вместо призов в ячейках появилось сообщение об ошибке "Err:508".

	A	B	C	D	E	F
1		<u>Аня</u>	<u>Боря</u>	<u>Валя</u>	<u>Галя</u>	<u>Дима</u>
2	<u>Количество наклеек</u>	45	65	35	50	90
3	<u>Приз</u>	Err:508	Err:508	Err:508	Err:508	Err:508

Вопрос

В чём заключается ошибка?

- A. Проблема со скобками
- B. Нельзя выводить словесные результаты
- C. В одной формуле не может быть использовано такое количество условий
- D. В начале не должно быть знака "="
- E. Кавычки надо убрать

Правильный ответ

- A. Проблема со скобками

Объяснение решения

Ошибка "Err:508" означает, что в формуле имеется закрывающая скобка без соответствующей парной открывающей скобки или отсутствует закрывающая скобка в середине формулы (закрывающие скобки в конце формулы добавляются автоматически).

В данном задании открывающие скобки пропущены в трёх местах:

1. $IFB2<30$;"тетрадь";
2. $IFB2<50$;"книга";
3. $IFB2<90$;"конструктор";

Другие, предложенные варианты ответов, неверные. Функция IF может выводить не только числовые, но и словесные результаты, даты, ссылку на ячейку или даже другую формулу. Количество условий может достигать до 64, таким образом используемое в нашем задании количество условий допустимо. В начале функции ставится знак "=", а не числовые значения записываются в кавычках.

Это информатика, потому что

Функция ЕСЛИ (IF) – одна из самых популярных функций в электронных таблицах. Путем логического сравнения значений она выдаёт ожидаемый результат, который может быть как числовым, так и словесным. Так как результат может быть либо истинным, либо ложным, то функции задаётся два варианта ответа. Довольно часто приходится решать случаи, когда значение сравнивается больше, чем с одним условием. В этом случае использование нескольких функций ЕСЛИ (вложение ЕСЛИ функций) в рамках одной формулы является хорошим подходом для получения правильного ответа.

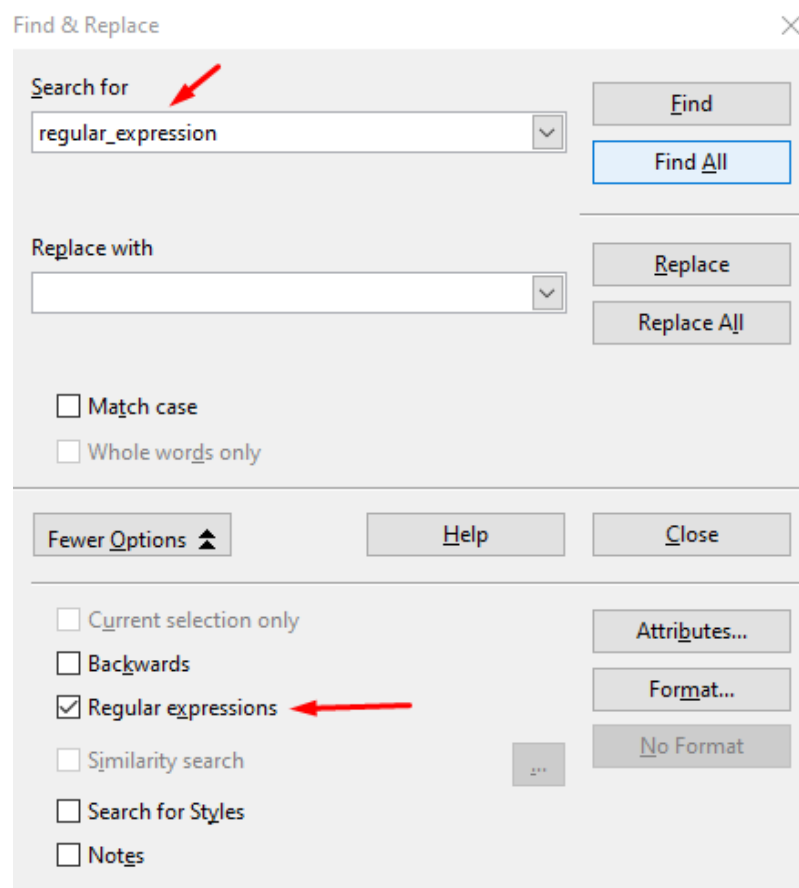
2021-EST-08: Searching for words in text

Task Attributes

- Age Group / Difficulty: [Seniors]/[hard]
- Category: word processing
- Keywords: Regular expressions
- Answer type: multiple-choice

Searching for words in text

To search for words in a text, a bebras can use regular expressions which specify search rules by using metacharacters.



Some of the metacharacters are:

[:space:] – space, for example, used in case **b[:space:]e**, will find the following text **b e**.

? – a special character that determines the presence or absence of a letter or sign in front of it. For example **_?beb** matches **_beb** and **beb**.

In the text below, the word “bebras” occurs five times and is spelled differently each time:

There are two extant species of **b-e-b-r-a-s**; the North American **b e b r a s** and the Eurasian **bebras**. The American **bebras** was coined by Carl Linnaeus in 1758 who also classified the species name fiber. However, the two **bebas** were not conclusively shown to be separate species until the 1970s with chromosomal evidence.

Question

What regular expression should a bebras enter into the search box to find all versions of the word “bebras” in the text?

Answer options

- A. `_?b-?e-?b-?r-?a-?s_?`
- B. `_?b-?[:space:]??e[:space:]?-?b[:space:]?-?r[:space:]?-?a[:space:]?-?s_?`
- C. `_?b[:space:]??e[:space:]??b[:space:]??r[:space:]??a[:space:]??s_?`
- D. `_?b-?[:space:]??e[:space:]?-?b[:space:]?-?[:space:]?-?a[:space:]?-?s_?`

Correct answer

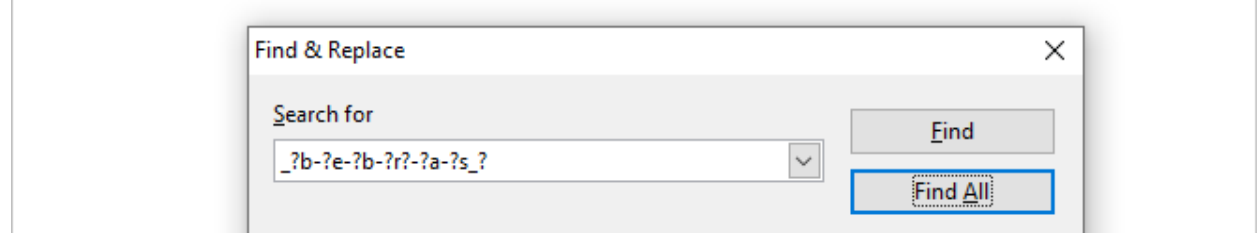
- B. `_?b-?[:space:]??e[:space:]?-?b[:space:]?-?r[:space:]?-?a[:space:]?-?s_?`

Answer explanation

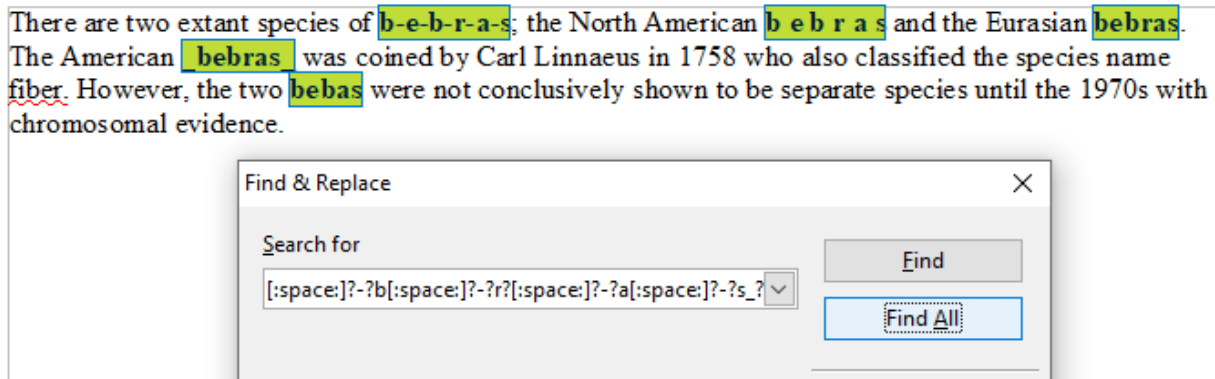
Let’s consider each option separately.

Option A will not find the word b e b r a s, where there is a space after each letter.

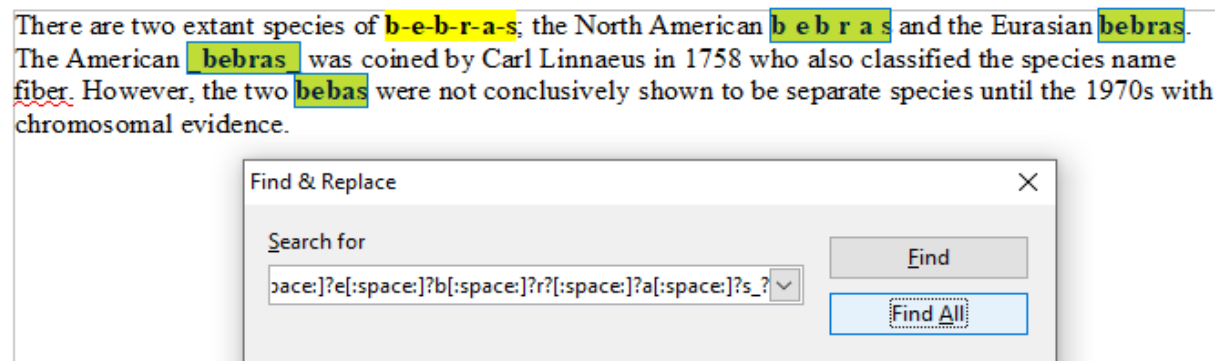
There are two extant species of **b-e-b-r-a-s**; the North American **b e b r a s** and the Eurasian **bebras**. The American **bebras** was coined by Carl Linnaeus in 1758 who also classified the species name fiber. However, the two **bebas** were not conclusively shown to be separate species until the 1970s with chromosomal evidence.



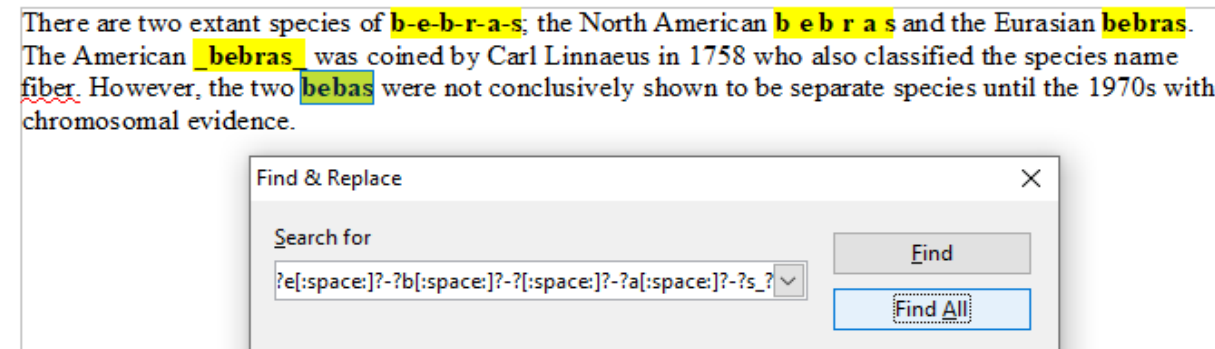
Option B will find all the words we need.



Option C will not find a word form with a hyphen between letters.



And option D will only find one word with the missing letter “r”.



It's Informatics

Regular expressions are very often used in programming, for example, to find matching characters, multiple characters, words, to validate entire strings, to find and replace words or characters. They can also be used to determine the conditions of filling in a field, for example, validation for entering a password, where the new password has a set of requirements like a minimum number of characters, the presence of a small/capital letter, number, a special character.

Comments

Lidia Feklistova (Estonia) lidia.feklistova@ut.ee

Files

Screenshots from OpenOffice Writer.

Authors and Contributors (incl. Graphics)

Anastassia Fomin (Estonia) vesta33@gmail.com

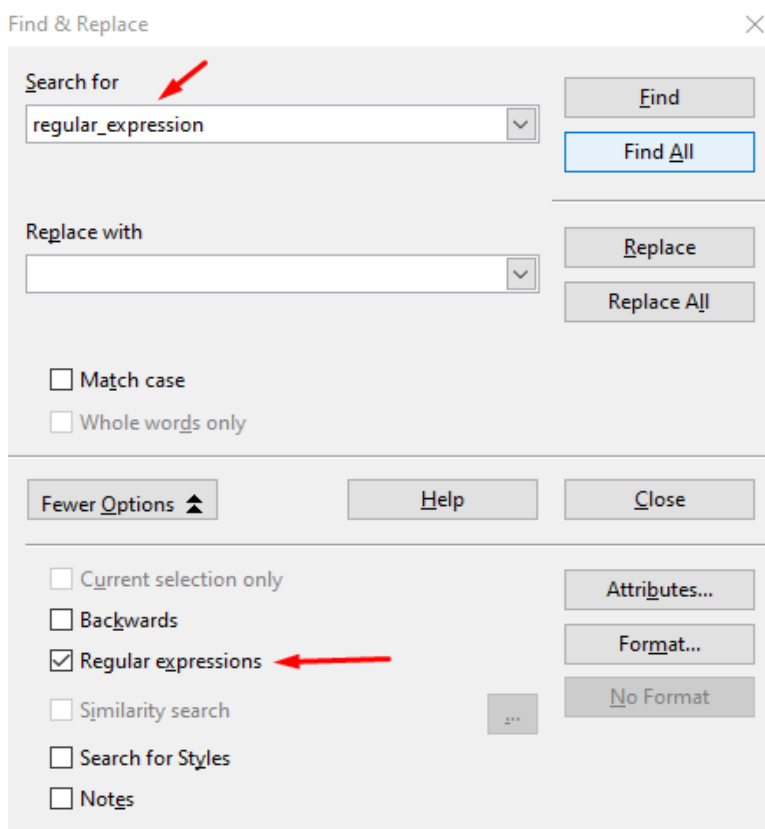
Websites

- https://wiki.openoffice.org/wiki/Documentation/How_Tos/Regular_Expressions_in_Writer#Repeating_match_.2B_.2A_.7Bm.2Cn.7D
- https://en.wikipedia.org/wiki/Regular_expression

Поиск слов в тексте

Автор: Анастасия Фомин (Эстония)

Для поиска слов в тексте бобр может использовать регулярные выражения (англ. *regular expressions*), состоящие из задающих правила поиска метасимволов.



Вот некоторые из метасимволов:

[:space:] – пробел, например **b[:space:]e** найдёт в ниже приведённом тексте **b e**.

? – специальный символ, определяющий наличие или отсутствие буквы или знака перед ним. Например **_?beb** найдёт **beb** и **beb**.

В приведенном ниже тексте слово “bebras” встречается пять раз и каждый раз оно записано по-разному:

There are two extant species of **b-e-b-r-a-s**; the North American **b e b r a s** and the Eurasian **bebras**. The American **bebras** was coined by Carl Linnaeus in 1758 who also classified the species name fiber. However, the two **bebas** were not conclusively shown to be separate species until the 1970s with chromosomal evidence.

Вопрос

Какое регулярное выражение следует ввести в окно поиска, чтобы найти в тексте все варианты слова “bebras”?

- A. `_?b-?e-?b-?r?-?a-?s_?`
- B. `_?b-?[:space:]?e[:space:]?-?b[:space:]?-?r?[:space:]?-?a[:space:]?-?s_?`
- C. `_?b[:space:]?e[:space:]?b[:space:]?r?[:space:]?a[:space:]?s_?`
- D. `_?b-?[:space:]?e[:space:]?-?b[:space:]?-?[:space:]?-?a[:space:]?-?s_?`

Правильный ответ

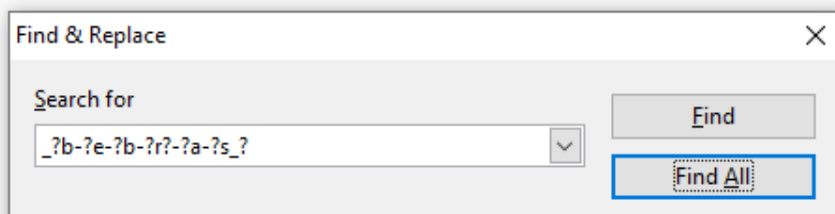
- B. `_?b-?[:space:]?e[:space:]?-?b[:space:]?-?r?[:space:]?-?a[:space:]?-?s_?`

Объяснение решения

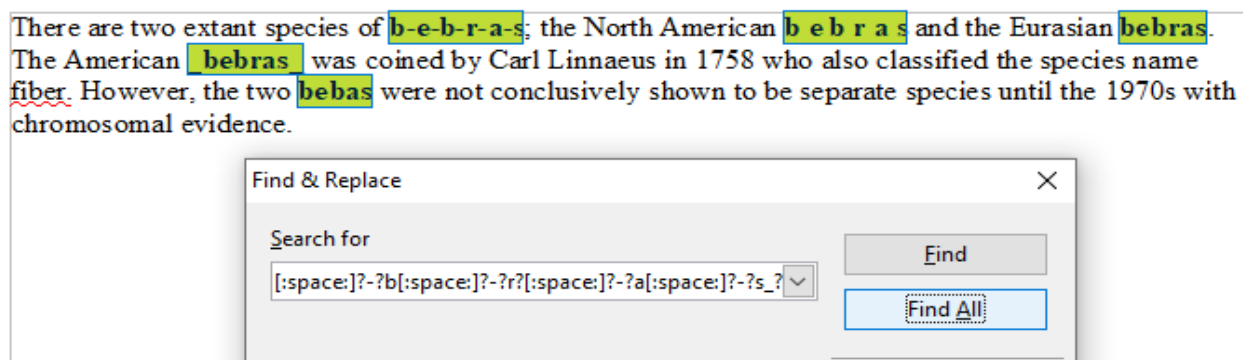
Рассмотрим каждый вариант по отдельности.

Вариант А не найдёт слово **b e b r a s**, где после каждой буквы встречается пробел.

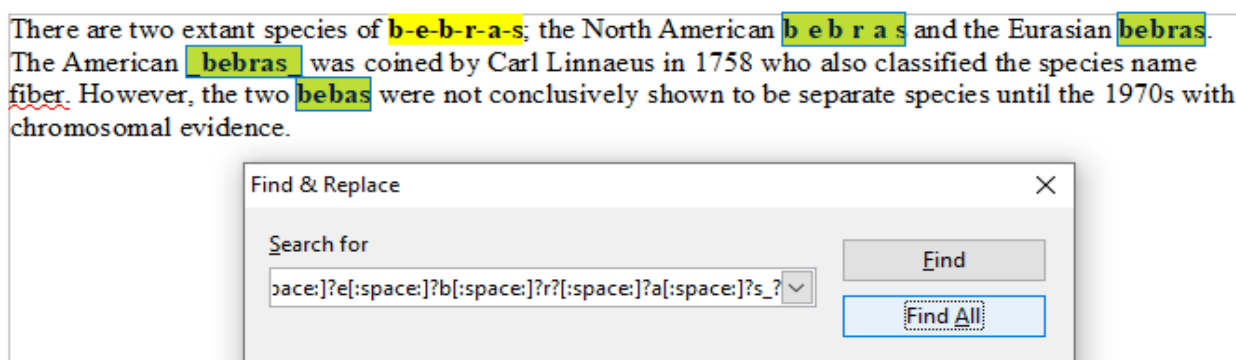
There are two extant species of **b-e-b-r-a-s**; the North American **b e b r a s** and the Eurasian **bebras**. The American **bebras** was coined by Carl Linnaeus in 1758 who also classified the species name fiber. However, the two **bebas** were not conclusively shown to be separate species until the 1970s with chromosomal evidence.



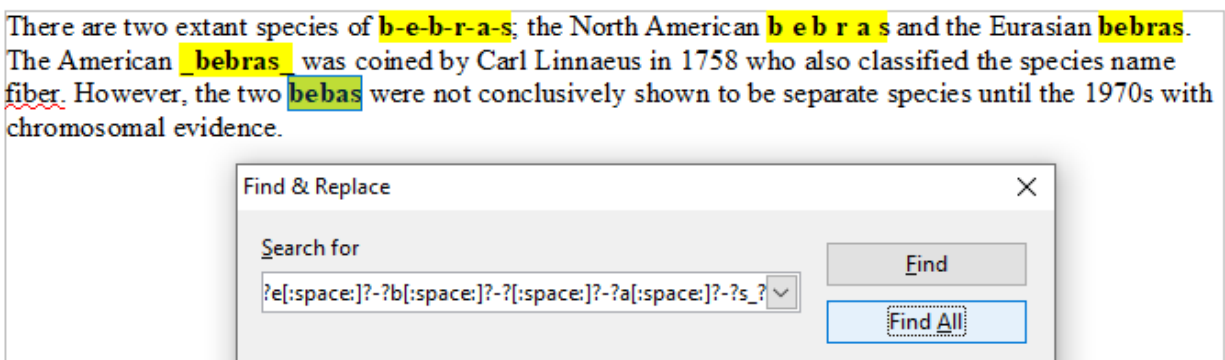
Вариант В найдёт все нужные нам слова.



Вариант С не найдёт форму слова со знаком дефиса между буквами.



И вариант D найдёт только одно слово с пропущенной буквой "r".



Это информатика, потому что

Регулярные выражения очень часто используются в программировании, например, для поиска совпадающих символов, множества символов, слов, валидации целых строк, для поиска и замены слов или символов. Их также можно использовать для определения условия заполнения поля, например, валидации по вводу пароля, где новому паролю задаются требования в виде минимального количества символов, наличия маленькой/большой буквы, наличие цифры или специального символа.

2021-EST-09: Letter to a friend

Task Attributes

- Age Group / Difficulty: [Benjamins] / [medium]
- Category: Ethics
- Keywords: Computer Ethics
- Answer type: multiple-choice

Letter to a friend

Beaver Karl received a letter from his friend.

Dier friend!

I don't understand what I did wrong, but the beavers of our class are looking strangely at me. It all started at a Monday school party, where I made a picture of Lion in a fashionable suit made by his hands and posted this picture on social networks like my own avatar. Why not, we are so similar to him?! Lion, of course, noticed it and now asks to remove it. But how can I clean it up if, thanks to this, I was invited to the show "Fashionable Me!"?!

On Tuesday there was a computer science lesson and, in order not to waste time, I put my favorite game to download. The Internet connection was terribly slow that day, the materials did not open and the pupils were unable to complete assignments. But the game was downloaded and during the pause, I managed to play a little.

On Wednesday a teacher asked me to tell everyone that they should come to school for the second lesson. I thought it was too early, and wrote that they should come for the third lesson. All the pupils first liked my post, but when the teacher scolded everyone, they took offense at me.

On Thursday, I was able to open the e-diary as an administrator, and now the average grade for our class has been improved.

Finally Friday! I'm going to try my new camera, record a new film and send it to you tomorrow.

See you later!

You friend Sandy

Question

What did Sandy wrong? Pick all correct answers.

- A. Interfering with other computer users by increasing the load of Internet traffic.
- B. Took over someone's intellectual property.
- C. Made a photo of Lion with his permission.
- D. Placed false information on the network.
- E. Opened the e-diary as an administrator.
- F. Took a camera to the cinema.

Correct answer

- A. Interfering with other computer users by increasing the load of Internet traffic.
- B. Tooked over someone else's intellectual property.
- D. Placed false information on the network.

Answer explanation

By downloading the game on the computer science lesson, Sandy interfered with other pupils, since the network was overloaded and the speed needed to transfer files was very low. By getting to the show "Fashionable Me!" Sandy appropriated Lion's intellectual property, by passing Lion's suit as his own. This way, options A and B are correct.

Taking pictures with permission is not prohibited, but publishing without permission or impersonating someone else is not permitted.

Sandy did the right thing by posting information on the Internet, that was requested by a teacher. But the wrong information was presented and this is false.

By opening the e-diary, Sandy didn't violate the law, however, he had to inform the creator of this program about such a vulnerability and, of course, not to use the last one for his own purposes.

It is not forbidden to take a camera to the cinema, but it is forbidden to use it to record new movies.

It's Informatics

There are many ethical standards. Some of them may differ from each other depending on the ethical and cultural traditions of different countries.

Computer ethics was born in the 70-80s last century and has already many different terms like information ethics, cyberethics, blogging ethics, hacker ethics.

Comments

Lidia Feklistova (Estonia) lidia.feklistova@ut.ee

Authors and Contributors (incl. Graphics)

Anastassia Fomin (Estonia) vesta33@gmail.com

Websites

- https://en.wikipedia.org/wiki/Computer_ethics
- <https://www.aki.ee/ru/polezno-znat>

Письмо другу

Автор: Анастасия Фомин (Эстония)

Бобр Кеша получил письмо от своего друга.

Дорогой друг!

Не понимаю, что я сделал не так, но бобры нашего класса странно на меня смотрят. Началось всё с того, что в понедельник на школьном празднике я сфотографировал нашего модника Леона в костюме, сделанном его руками и поставил это фото в социальной сети в виде своего аватара. А что, мы ведь с ним так похожи?! Леон, конечно же, это заметил и теперь просит убрать. Но как я уберу, если благодаря этому меня позвали на шоу “Я - модник!”?!

А во вторник у нас был урок информатики и, чтобы не терять время зря, я поставил скачиваться свою любимую игру. Интернет был в этот день ужасно медленным, материалы не открывались, и ученики не смогли сделать все задания. Зато игра скачалась, и на перемене я успел немного поиграть.

В среду учитель попросил меня передать всем, что в школу надо прийти ко второму уроку. Я подумал, что рановато, и написал к третьему. Все ученики сначала одобрили мой пост, а потом, когда учитель всех наругал, обиделись на меня.

В четверг у меня получилось открыть электронный дневник в роли администратора. Теперь у всего класса средний балл повысился.

И наконец пятница! Пойду попробую свою новую камеру, сниму один новый фильм и тебе завтра вышлю.

Так что жди :)

До скорого!

Твой друг Тик

Вопрос

Что Тик сделал не так? Выберите все правильные ответы.

- A. Создавал помехи для других пользователей компьютера, увеличивая нагрузку Интернет-трафика.
- B. Присвоил чужую интеллектуальную собственность.
- C. Сфотографировал Леона с его разрешения.
- D. Разместил недостоверную информацию в сети.
- E. Смог открыть электронный дневник.
- F. Взял камеру в кинотеатр.

Правильный ответ

- A. Создавал помехи для других пользователей компьютера.
- B. Присвоил чужую интеллектуальную собственность.
- D. Разместил недостоверную информацию в сети.

Объяснение решения

Скачивая игру на уроке, Тик создавал помехи для других учеников, так как сеть была загружена и скорость передачи нужных ученикам файлов была очень низкой. А попав на шоу “Я - модник!”, Тик присвоил себе интеллектуальную собственность Леона, то есть выдал его костюм за свой собственный. Таким образом варианты А и В правильные.

Фотографировать с разрешения человека не запрещается, однако публиковать без спроса, или ложно выдавать себя за кого-то другого не разрешено.

Тик сделал правильно, что разместил информацию по просьбе учителя в интернете. Однако неправильно, что информация была представлена ложной.

Открыв электронный дневник, Тик не нарушил закон, однако об обнаруженной уязвимости он должен был сообщить создателю данной электронной программы и, конечно же, не использовать данную уязвимость в своих целях.

Брать камеру в кинотеатр не запрещено, но пользоваться с целью отснять фильм запрещается.

Это информатика, потому что

Существует множество этических стандартов. Некоторые из них могут отличаться друг от друга в зависимости от этических и культурных традиций разных стран.

Компьютерная этика зародилась в 70-80 гг. XX века и уже имеет много разных терминов - информационная этика, киберэтика, блогерская этика, хакерская этика.

2021-EST-10: Guess what's inside

Task Attributes

- Age Group / Difficulty: [Juniors] / [easy]
- Category: Logic, Other software
- Keywords: Decision Tree
- Answer type: Open Integer

Guess what's inside

On a shelf, there were boxes containing the bark of various types of trees. Once, playing with the boxes, the little beaver Matthew re-pasted all the labels. Beaver-dad Peter has now to guess which bark is in which box. It is known that Matthew mixed up all the labels and none of them corresponds to the contents of the box.



Question

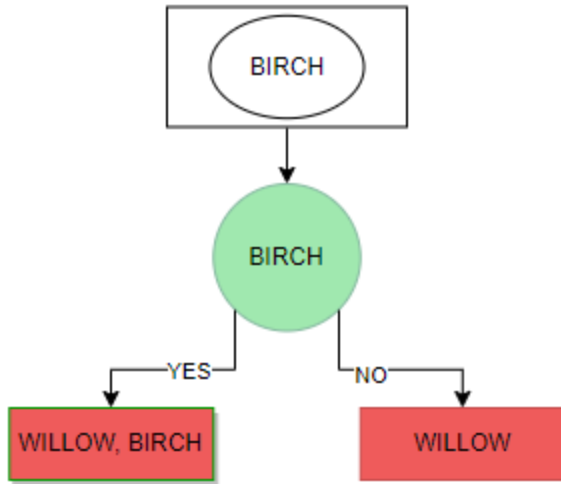
What is the minimum number of attempts Peter has to do, to get known the correct content of each box?

Correct answer

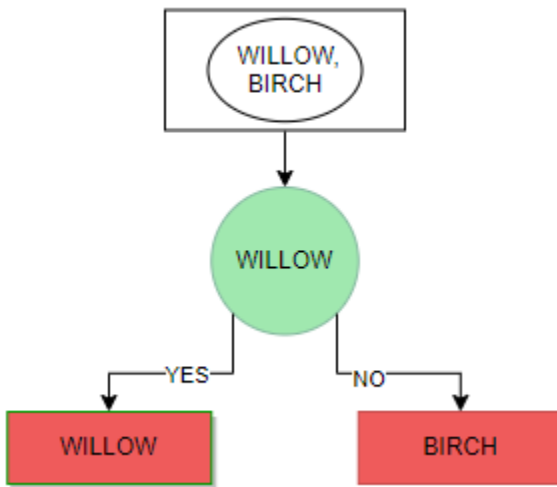
2

Answer explanation

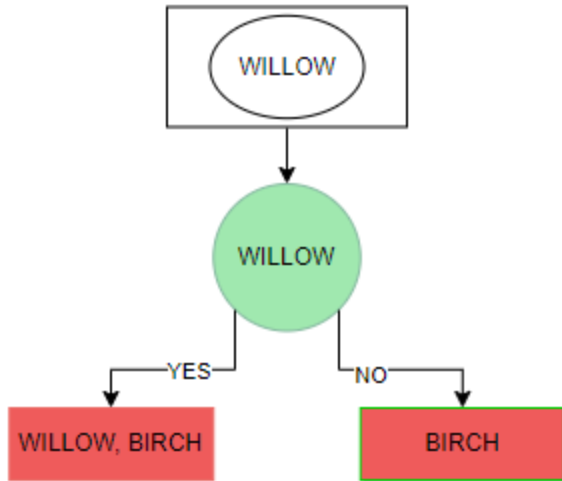
You can start with any box. Suppose that the beaver Peter pulled birch bark out of the first box. It is known that the “Birch” sticker is not true. So, in addition to birch bark, this box also contains willow bark. Therefore, it is necessary to stick the “Willow, birch” label on this box.



It turns out that the second box can contain either willow or birch bark. Suppose that the beaver Peter pulled out the willow bark, then a label with the name "WILLOW" should be stuck to the box.



Next, there is only one box left in which there will be birch bark.



That way, two attempts are enough to properly apply all the labels.

It's Informatics

To simplify the analysis of this task, a decision tree was used. This tree simplifies the understanding and interpretation of the data. The decision tree is widely used in the process of making managerial decisions in software development, machine learning, data analysis, statistics.

Comments

Lidia Feklistova (Estonia) lidia.feklistova@ut.ee

Files

Picture created using the Paint tool.

Decision Tree by using online tool: <https://online.visual-paradigm.com/drive/#diagram-list:proj=0&new=DecisionTree>

Authors and Contributors (incl. Graphics)

Anastassia Fomin (Estonia) vesta33@gmail.com

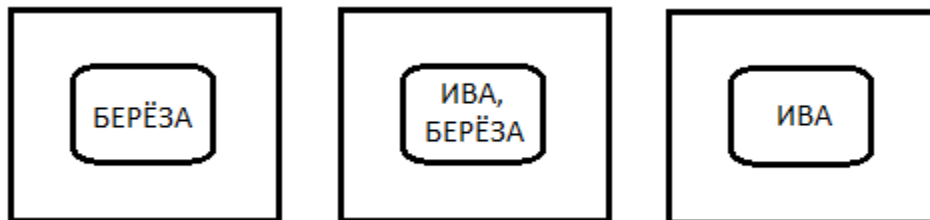
Websites

- <https://qa-academy.by/qaacademy/news/logicheskie-zadachi-na-sobesedovanii-testirovshhika/>
- https://en.wikipedia.org/wiki/Decision_tree

Угадай содержимое

Автор: Анастасия Фомин (Эстония)

На полке стояли коробки с корой различных видов деревьев. Однажды, играя с коробками, маленький бобрёнок Митя переклеил все этикетки. Бобру-папе Пете теперь предстоит отгадать, какая кора в какой коробке находится. Известно, что Митя перепутал все этикетки и ни одна из них не соответствует содержимому коробки.



Вопрос

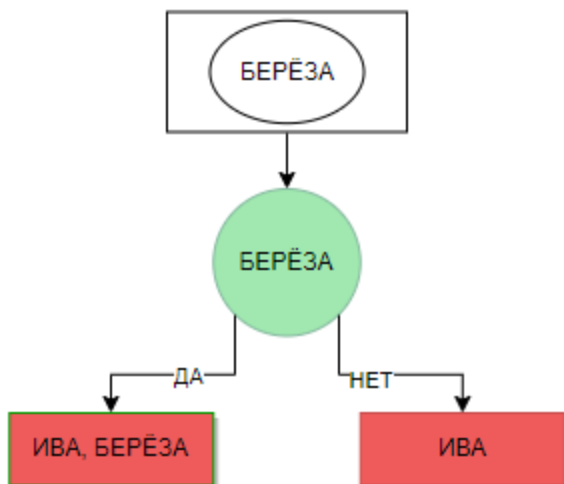
Каково минимальное количество попыток будет у Пети, чтобы узнать содержимое коробки?

Правильный ответ

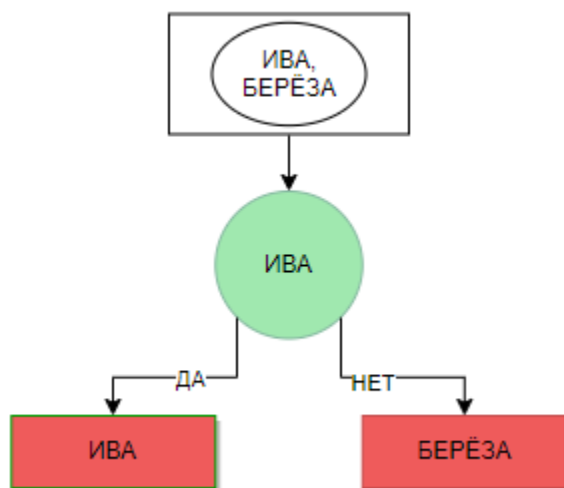
2 попытки

Объяснение решения

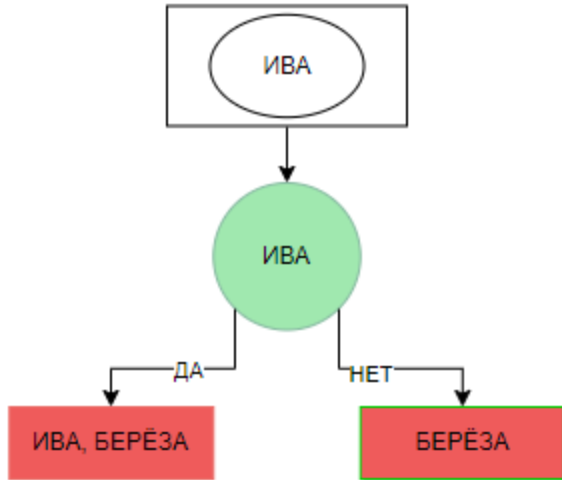
Можно начать с любой коробки. Предположим, что бобр Петя вытянул березовую кору из первой коробки. Известно, что наклейка “Берёза” не соответствует действительности. Значит, помимо коры берёзы в этой коробке есть ещё и кора ивы. Следовательно, на эту коробку необходимо наклеить этикетку “Ива, берёза”.



Получается, что во второй коробке может быть кора либо ивы, либо берёзы. Предположим, что бобр Петя вытянул кору ивы, тогда на коробку надо наклеить этикетку с названием “ИВА”.



Далее остаётся одна коробка, в которой будет кора берёзы.



Таким образом, достаточно двух попыток, чтобы правильно наклеить все этикетки.

Это информатика, потому что

Для упрощения анализа данного задания использовалось дерево решений. Данное дерево упрощает понимание и интерпретацию данных. Дерево решений находит широкое применение в процессе принятия управленческих решений при разработке программного обеспечения, в машинном обучении, анализе данных, статистике.

