



Leták zimnej časti V. ročníka

Ahojte milí riešitelia.

Sme veľmi radi, že ste sa dozvedeli o PRASKu a asi by vás zaujímalo, čo to vlastne je, ako to celé funguje a prečo by ste to mali riešiť. Na všetko z toho sa vám teraz pokúsime odpovedať.

Čo to je a pre koho je to určené?

PRASK je korešpondenčný seminár určený pre všetkých základoškolákov, ktorých zaujíma matematika, informatika alebo by sa chceli naučiť programovať. Je to súťaž zameraná hlavne pre siedmakov a starších, môžete ju však riešiť aj keď ste v nižšom ročníku.

Seminár je organizovaný študentami informatiky na Fakulte matematiky, fyziky a informatiky na Univerzite Komenského.

Priebeh súťaže

Počas roka prebiehajú dve nezávislé časti – letná a zimná. Priebeh častí je už potom úplne rovnaký. Každá časť pozostáva z dvoch sérií piatich príkladov – dvoch teoretických, jedného praktického a dvoch programátorských. Ak aj neviete programovať nezúfajte. Namiesto programátorských úloh si môžete prejsť programátorským tutoriálom, ktorý vás to naučí a navyše vám získate body, ktoré sa vám rátať do PRASKu.

Na riešenie série je vyhradených niekoľko týždňov. Až do dňa odovzdania môžete doma riešiť zadané príklady. Môžete riešiť ľubovoľné príklady z danej série, nemusíte vyriešiť všetko, nemusíte vyriešiť ani celú úlohu¹. Najneskôr do dňa odovzdania (ktorý je napísaný na zadaniach aktuálnej série) je potrebné poslať vaše riešenia pomocou webového rozhrania.

Po konci série si pozrieme vaše odovzdané riešenia a opravíme ich. Pre každý príklad je v zadaní napísané, koľko bodov sa zaň dá dostať. Samozrejme, je možné získať čiastkové body, aj keby ste nevyriešili celú úlohu, alebo by vaše riešenie nebolo úplne správne. Dokonca, ak nás prekvapíte originálnym riešením, môžete získať bonusové body. Opravené riešenie vám potom pošleme späť aj s poznámkami ohľadom vašeho riešenia.

Prečo to chceme riešiť?

Riešenie korešpondenčného seminára prináša mnoho výhod. Riešením úloh a čítaním našich vzorových riešení **objavíte a naučíte sa** mnoho nových vecí, ktoré by ste sa v škole skoro určite nenaučili. Napríklad sa môžete naučiť **programovať**. To vám potom vie **pomôcť pri prijímačkách**, či už na stredné alebo vysoké školy. Takisto vám to pomôže pri **riešení Olympiády z informatiky alebo Korešpondenčného Seminára z Programovania**. No a v neposlednom rade, pri **pohovoroch** do veľkých firiem ako Google, Facebook alebo Eset častokrát zaváži znalosť algoritmického programovania, ktoré si môžete pomocou nášho semináru trénovať.

Je tu však ešte jedna výhoda určená pre najlepších riešiteľov. Dvakrát ročne sa bude organizovať **týždenné sústredenie**. Naň pozývame niekoľko² najlepších riešiteľov. Na sústreďení si užiješ kopec zábavy, športu, nových ľudí a možno sa aj niečo naučíš.

A samozrejme, víťazov čakajú pekné **vecné ceny** vo forme knižky, hry alebo menšej elektroniky.

Ako má vyzeráť správne riešenie

To závisí od typu úlohy, ktorú riešite. Pri teoretických úlohách musí správne riešenie okrem výsledku obsahovať aj popis postupu, akým ste sa k danému výsledku dopracovali. Dôraz sa pri opravovaní dáva hlavne na tento slovný popis, ktorý by mal byť napísaný čo najzrozumiteľnejšie, aby sme si pri opravovaní nemuseli lámať hlavu. Mal by obsahovať všetky podstatné kroky, ktoré vás viedli k riešeniu.

V prípade praktických úloh sa to líši. Občas od vás chceme slovný popis, občas sa stačí dostať k nejakému tajnému heslu alebo kliknúť na správnu linku. Presný spôsob nájdete v zadaní.

¹Aj keď budeme radi, ak sa vám to podarí.

²zhruba 15, ale aj nižšie umiestnení riešitelia sa môžu dostať ako náhradníci

No a pri programátorských úlohách a programátorskej liahni odovzdávate iba váš program, ktorý sa vám okamžite automaticky otestuje a do pár sekúnd sa dozviete, či ste úlohu vyriešili správne. A ak nie, môžete skúsiť odovzdať opravený program znova.

A nebojte sa, ak ste ešte nikdy nespisovali postupy svojich riešení. Keď vám riešenia opravíme, napíšeme vám k nim aj komentáre, ktoré vám môžu pomôcť v riešení ďalšej série. To je najlepší spôsob, ako sa zlepšovať.

Spôsob odovzdávania

Ako prvú vec, ktorú musíte urobiť pred tým, ako budete môcť odovzdávať svoje riešenia, je **zaregistrovanie** sa na našej webovej stránke prask.ksp.sk. V časti **Zadania** a **vzoráky** nájdete okrem zadání aj odkaz, na ktorom môžete odovzdať vaše riešenie.

Riešenie každej teoretickej úlohy má byť jeden súbor formátu **.pdf**. Ten nahráte na našu stránku a stlačíte zelené tlačítko **Submit**. Opravovať sa bude **posledné odovzdané** riešenie, takže si dajte pozor, aby ste si niečo neprepísali.

Myslím, že vytvoriť pdf súbor by pre vás nemal byť problém, ak by ste s tým predsa len problém mali, pokúste sa použiť nejaký online converter ako napríklad www.freepdfconvert.com.

V prípade programátorských úloh sa dá rovnakým spôsobom odovzdať zdrojový kód vášho programu, teda súbor s príponou **.cpp**, **.py** alebo **.pas**.



Úlohy 1. kola zimnej časti

Termín odoslania riešení tejto série je pondelok 10. decembra 2018.

Teoretické úlohy

V tejto časti ťa čaká niekoľko matematickejších úloh, ktoré úzko súvisia s informatikou. Ako riešenie týchto úloh treba poslať podrobne písaný postup toho, ako si riešil danú úlohu.

A ak by ťa to zaujímalo, podobné úlohy môžeš nájsť aj v Olympiáde v informatike, kategória B (<http://oi.sk/archiv/2018/sl-2018-1-zad-B.pdf>). Vrelo ti ju odporúčame riešiť tiež, naučíš sa veľa nových vecí a môžeš sa dostať aj na krajské kolo Olympiády.

1. Podivný zošit

15 bodov za riešenie

Ak máte akékoľvek otázky ohľadom tejto úlohy, napíšte Emovi na siegrift@ksp.sk

V spoločných kúpeľniach v Mlynskej doline sa dá nájsť ozaj všeličo. Jeden zamračený večer sa Andrej po dlhom dni vrátil na intrák a netúžil po ničom inom ako si ľahnúť a spať až do rána. Ešte sa však rozhodol umyť si zuby. Ani vo sne by mu nenapadlo, že na prahu kúpelne nájde pohodený zošit.

Andrej bol samozrejme zvedavý, zošit zdvihol a prelistoval. Zošit bol prekvapivo úplne prázdny, až na čísla strán. Tie však neboli vôbec obyčajné, nešli v poradi za sebou a niektoré čísla ani neboli čísla, ale slová.

Andrej aj zabudol, že si prišiel umyť zuby a hneď utekal ukázať *podivný zošit* Dávidovi. Tomu napadlo napísať na jednu z prázdnych strán číslo. Hneď ako to spravil, číslo zmizlo a na jeho mieste sa objavila iná hodnota. Pokus ešte pár krát zopakoval s rovnakým výsledkom. Prizerajúci sa Andrej si uvedomil, že číslo, ktoré sa objaví nie je náhodné, ale nejakým spôsobom súvisí s číslom, ktoré napísal Dávid.

Napadlo im, že by mohli skúsiť zošit donútiť vypísať číslo aktuálnej strany. Možno sa vtedy stane niečo zaujímavé. A veruže sa stalo. Hneď ako bolo číslo napísané zošitom rovnaké ako číslo strany, v zošite sa objavil ďalší riadok textu: “Získavaš 1.5 bodu do PRASKu”. Super nie?

Úloha

V tejto úlohe dostanete prístup k *podivnému zošitu*. Vždy, keď doň vpíšete nejaké číslo, zošit vypíše odpoveď (číselnú, slovnú ...) dodržiavajúc jednoduché pravidlo.

Vašou úlohou bude toto pravidlo odhaliť a nájsť také vstupné číslo, pre ktoré zošit vypíše požadovaný text.

Odovzdávanie

K tejto úlohe netreba odovzdávať žiadny popis ani program. Simulátor, spolu so všetkými potrebnými informáciami o odovzdávaní nájdete na stránke <https://prask.ksp.sk/specialne/prask/5/1/1>

Bodovanie

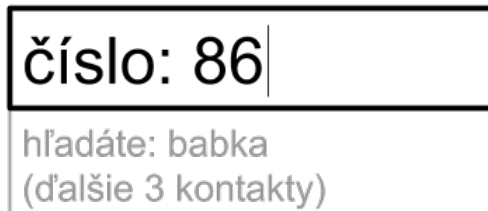
K dispozícii je 10 strán zošita, za vyriešenie každej získate 1.5 bodu. Náročnosť stúpa každou ďalšou stranou, môžete ich však riešiť v ľubovoľnom poradí.

2. Reprézantácia telefónnych čísel

15 bodov za riešenie

Ak máte akékoľvek otázky ohľadom tejto úlohy, napíšte Žabovi na zaba@ksp.sk

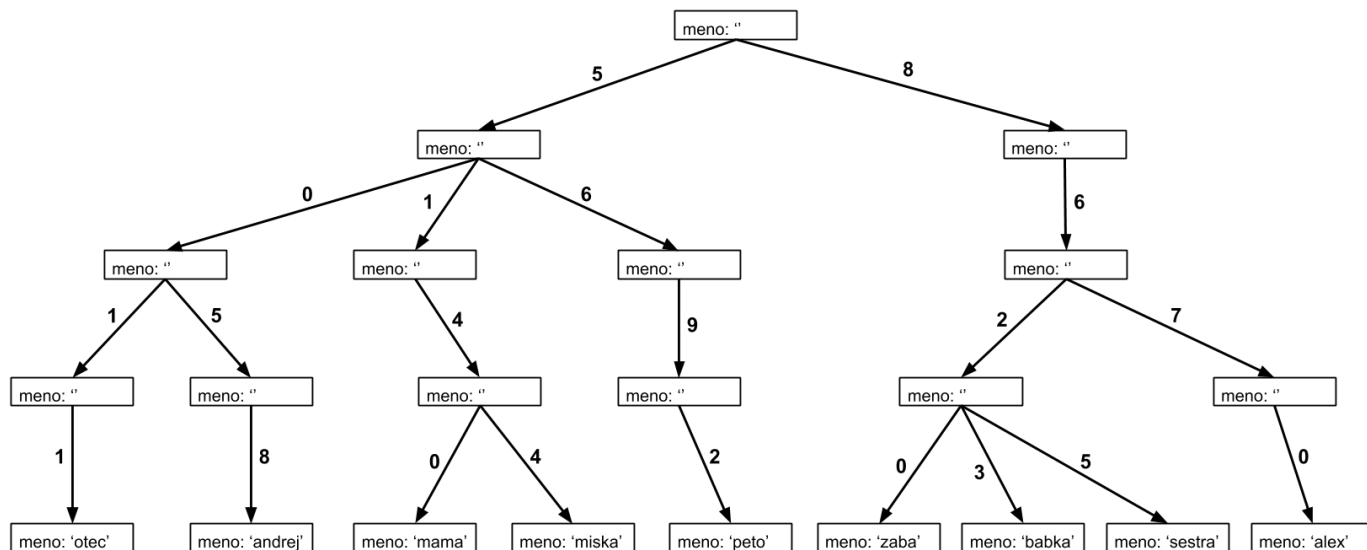
Adam sa minulú nedelu, ako vzorný vnuk, rozhodol zavolať svojej babke. Jej telefónne číslo si pamätá odmala, začal ho preto vyfukávať do mobilu. V polovici písania však Adamova obrazovka vyzerala zhruba takto a kontakt na babku bol automaticky doplnený.



Adam sa zháčil. Takúto funkcionalitu pokladal za samozrejmu, nikdy sa však nezamyslel, ako to vlastne ten mobil robí. Podľa čoho si vie tipnúť zadávané číslo a dokonca povedať koľko ďalších možností ešte existuje? Zrazu zabudol na to čo robil a začal sa vrtáť v pamäti telefónnu. Pomôžte Adamovi odpovedať na jeho otázky, nech stihne zavolať babke.

Úloha

- a) (1 body) Adam si do mobilu stiahol program, ktorý mu umožňuje prezeráť si vnútornú pamäť telefónnu. Hneď ho aj spustil s tým, že mu má zobrazíť pamäť patriacu kontaktom. Program preňho vygeneroval nasledovný obrázok.



Adam však vôbec nerozumie tomu, čo má táto reprezentácia znamenať. A jediný kontakt, ktorý si pamätá je na jeho babku, ktorá má číslo 8623 (nečudujte sa, štvormiestne čísla sú tam kde žije Adam úplne bežné).

Z uvedeného obrázku vydedukujte, aké všetky kontakty má Adam uložené v telefónne. Pre každý kontakt určite aj telefónne číslo, ktoré mu prislúcha.

- b) (2 body) Ako by vyzeral obrázok vygenerovaný stiahnutým programom, ak by mal Adam uložené nasledovné kontakty:

mama: 091576	ujo: 091497	brat: 091256	janka: 092730
sestra: 092576	stryko: 091483	lucka: 092578	sysel: 093037
babka: 091547	peter: 092530	jonas: 092098	radka: 091418

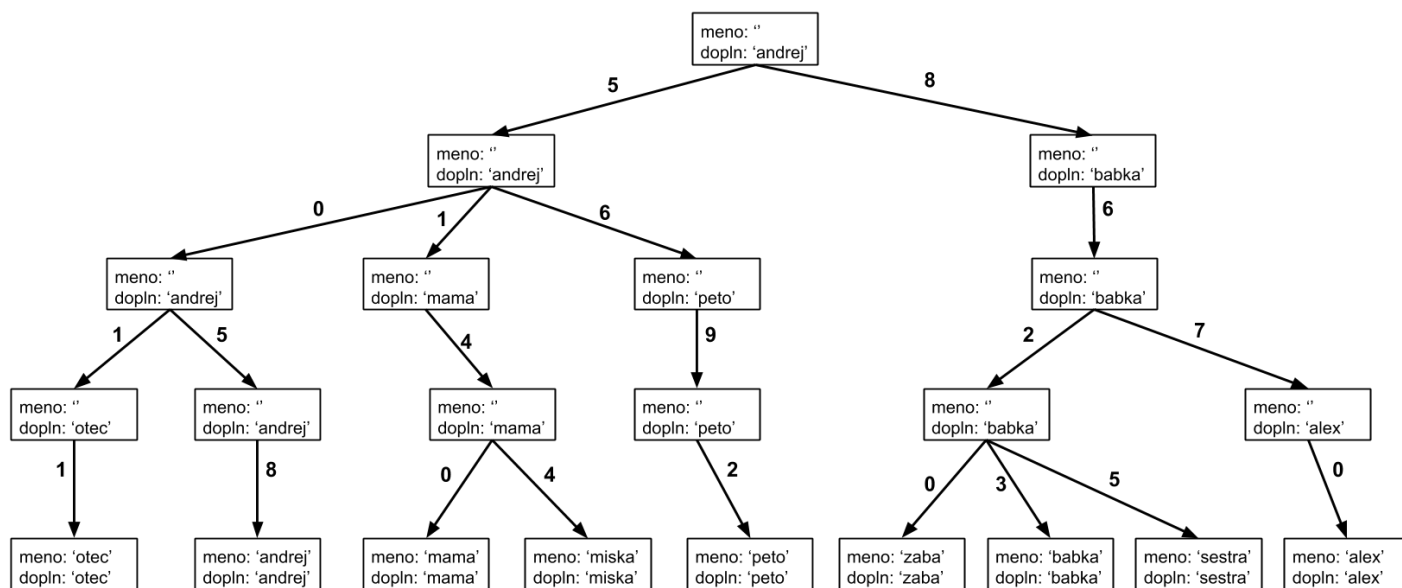
- c) (4 body) Adama by zaujímalo, ako vie jeho mobil pomocou uvedenej reprezentácie dopĺňať kontakt, ku ktorému by dané číslo mohlo patriť. Vašou úlohou je vymyslieť postup, ktorý po zadaní niekoľkých prvých číier čísla nájde ľubovoľný kontakt, ktorého číslo začína na zadané cifry. Popríklad oznámi, že žiadny taký kontakt sa v mobile nenachádza.

Napríklad, ak by sme sa pozreli na kontakty z podúlohy b), tak po napísaní 0914 by Adamov mobil mohol navrhnúť kontakt radka ale takisto kontakty ujo alebo stryko. A v prípade, že by Adam napísal iba 0, mohol by mu mobil ponúknuť ľubovoľný z jeho kontaktov. Ak by sa však pomýlil a napísal by 08 ako začiatok čísla, mobil by mu povedal, že žiaden vhodný kontakt neexistuje.

Vaše riešenie by malo fungovať pre každú možnú množinu kontaktov. Jednak sa teda môže meniť počet kontaktov a takisto dĺžka jedného čísla. Môžete však predpokladať, že všetky čísla majú rovnakú dĺžku a že tieto kontakty sú v pamäti reprezentované rovnako ako v podúlohách a) a b).

Taktiež sa snažte, aby bolo vaše riešenie čo najefektívnejšie, teda potrebovalo spraviť čo najmenej úkonov. Napríklad ak by sme mali 10 000 desaťmiestnych telefónnych čísel zapísaných vo vyššie uvedenej štruktúre, optimálne riešenie by sa stále dalo simulovať aj ručne.

- d) (5 body) Keď Adam do pamäte zapísal dlhšie čísla, hral sa aj s 1000 cifernými číslami, zdalo sa mu, že mu mobil začal sekať. Niekde v nastaveniach však našiel tlačidlo “Zoptimalizuj kontakty”, ktoré po stlačení citelne urýchlilo dopĺňanie kontaktov. Keď sa opäť pozrel do pamäte telefónu, s prekvapením zistil, že jej reprezentácia sa zmenila.



Upravte váš postup z podúlohy c) tak, aby využíval nové údaje z pamäte. Prečo je toto riešenie rýchlejšie ako to predošlé? Okrem toho vymyslíte postup, ktorým boli hodnoty `dopln` vypočítané. Máte teda popísať postup, ktorý sa vykoná keď je stlačené tlačidlo “Zoptimalizuj kontakty”, a ktorý doplní do pamäte nové hodnoty. Tento postup by mal fungovať na ľubovoľnej množine kontaktov.

- e) (3 body) S počítaním možností je však stále problém. Síce to Adamov mobil robí správne, trvá mu to príliš dlho. A žiadne tlačidlo “Zoptimalizuj počet” neexistuje. Rozhodol sa preto, že si ho do mobilu doprogramuje, nevie však, ako má toto tlačidlo pracovať. Vymyslíte a podrobne popíšete čo sa má stať po stlačení tlačidla “Zoptimalizuj počet”. Následne popíšete, ako bude telefón pre zadaný začiatok čísla zisťovať počet čísel s rovnakým začiatkom.

Praktická úloha

Pri práci s počítačom je potrebné vedieť pracovať aj s rôznymi nástrojmi, ktoré slúžia na úpravu obrázkov, prácu so zvukom či vyhľadávaním na internete. V tejto časti ťa preto zakaždým čaká nejaká netradičná úloha.

3. Audio Práskači

15 bodov za riešenie

Ak máte akékoľvek otázky ohľadom tejto úlohy, napíšte Romanovi na roman.sobkuliak@trojsten.sk

13:37 - “Rudo, máme tu tri nové nahrávky, toto bude veľké. Máš čas?”

13:40 - “Už znovu? Stále robím na tých nahrávkach z minulého týždňa.”

14:00 - “Veru už znovu. Škoda, na tieto bude treba naozaj odborníka...”

Práskači pracujú na odhaľovaní tých najväčších podvodov. Špecializujú sa na lúštenie audio šifier. Tvoria malý tím zložený z 3 členov a nahrávky dostávajú zvyčajne od anonymov. Za uplynulý rok mali plné ruky práce a inak tomu nie je ani teraz. Potrebujú rozlúsknuť 3 zaujímavé šifry a namixovať nový veľký hit, ale tím je vyčatený. Pomôžete *Práskačom* s ich najnovším prípadom?

Úloha

Táto úloha je o spracovaní zvuku. Odporúčame vám preto stiahnuť si voľne dostupný program *Audacity*. Nájdete ho napríklad na tejto adrese: <https://www.audacityteam.org/download/>. Okrem toho sme pre vás pripravili videotutoriál k tomuto programu: <https://youtu.be/CL6LaCsFSuI>. Nájdete v ňom všetko potrebné k vyriešeniu úloh, ale nie je *úplným* návodom k *Audacity*. Nebojte sa preto vyhľadávať na internete a experimentovať.

Nahrávky potrebné k riešeniu tejto úlohy nájdete na našom [Google Disku](#). Písmeno v názve priečinku označuje ku ktorej podúlohe jeho súbory patria.

Pri riešení tejto úlohy, a všeobecne pri spracovaní zvuku, vám odporúčame použiť sluchadlá namiesto reproduktorov.

Podúlohy A-C (10 bodov)

V týchto podúlohách sa stretnete s 3 nahrávkami od *Práskačov*. Každá z nahrávok v sebe obsahuje šifru a vašou úlohou je túto šifru vyriešiť. Riešením je vždy práve jedno slovo.

Nahrávky na seba nenadväzujú, takže ak nebudete vedieť rozlúsknuť niektorú zo šifier, môžete pracovať na inej.

Okrem riešenia šifry nám vo vašom riešení nezabudnite napísať aj **postup vášho riešenia**. Spomeňte aké úpravy ste s nahrávkami spravili a ako vám tieto úpravy pomohli k vyriešeniu šifry.

Za podúlohy A a B môžete získať po 3 bodoch, za podúlohu C až 4 body.

Podúloha D (5 bodov)

Práskači si okrem lúštenia šifier radi zahrajú na hudobné nástroje. Nedávno založili kapelu a práve nahráli svoj prvý singel. Zatiaľ však stále nemali čas ho *zmixovať* a tak potrebujú vašu pomoc. *Mixovanie nahrávky* znamená zmiešanie jednotlivých stôp s rozdielnymi nástrojmi tak, aby pesnička ako celok znela dobre.

Priečink na [Google Disku](#) pre túto podúlohu obsahuje nástroje nahrané oddelene. Okrem toho tam nájdete *Príklad_vysledku*. Táto nahrávka obsahuje náš jednoduchý *mix* so zámerne zníženou kvalitou, ktorým sa môžete inšpirovať.

Vašou úlohou bude:

- Presunúť stopy *Elektricka_gitara_clean*, *Crunch_gitara* a *Slaky* na správne miesta. Pozorne si vypočujte *Príklad_vysledku* a zistite kam spomenuté stopy patria.
- Vyčistiť jednotlivé nahrávky – odstrániť šum a nechcené časti (napríklad hlasné dýchanie pred začiatkom spevu).
- Upraviť hlasitosti nahrávok.
- Pridať efekty k stopám. Napríklad *Reverb* a *Bass boost* k spevom.
- Byť kreatívny. Môžete v podstate s nahrávkami spraviť čokoľvek. Zmeniť štruktúru pesničky (napríklad zopakovať slohu), pridať *Panning* (rozložiť stopy medzi ľavé a pravé ucho), použiť ľubovoľné efekty. Ak ste hudbone zdatný, môžete napríklad do pesničky nahráť svoje sólo.

Výslednú nahrávku nám pošlite vo WAV formáte a opäť nezabudnite **popísať úpravy**, ktoré ste spravili.

Odvzdávanie

Odvzdávanie tejto úlohy je trochu netradičné, preto prosím pozorne čítajte nasledujúce riadky:

1. Popisy riešení nám pošlite vo formáte pdf ako popis k tejto úlohe.
2. Zmixovanú pesničku z *Podúlohy D* nám nahrajte cez [špeciálne odovzdávateľko \(DriveUploader\)](#). Pri odovzdávaní nezabudnite správne vyplniť vaše meno a email.

Programátorské úlohy

Tieto úlohy sú zamerané na praktickú tvorbu programov v niektorom vyššom programovacom jazyku ako je napríklad Python, C++ alebo Pascal. Na stránke odovzdávaš **iba zdrojový kód** svojho programu riešiaceho

zadanú úlohu, ktorý bude okamžite automaticky otestovaný a do pár sekúnd sa dozvieš, koľko bodov tvoj program získal. Tieto body ti už nikto nemôže zobrať, ale ak si nezískal plný počet bodov, môžeš opakovane odovzdávať opravený program, až kým nebudeš spokojný s výsledkom.

Ak už vieš programovať, ale ešte si nepracoval s našim testovacím systémom, odporúčam ti zájsť na Programátorskú Liaheň (<http://liahen.ksp.sk>), kde si o tom môžeš prečítať úvodný text a vyriešiť si niekoľko jednoduchých úloh.

Ak však **nevieš programovať, tak nezúfaj!** Pripravili sme pre teba **Programátorskú Liaheň**, ktorá ťa **naučí základy programovania** v jazyku C++. Navyše, za riešenie týchto tutoriálových úloh na Liahni môžeš získať body priamo do PRASKu a tým si vynahradiť neriešenie niektorej z programátorských úloh.

Presnejšie to funguje takto. Na Liahni sa nachádzajú dve sady úloh, prvá zameraná na premenné a druhá na podmienky v jazyku C++. V týchto sadách sa nachádzajú bodované aj nebodované úlohy, ktoré môžeš postupne riešiť a ktoré ti postupne vysvetlia danú problematiku. Dokopy sa v jednej sade dá získať až 15 bodov.

Týmito bodmi si potom môžeš nahradiť úlohy 4 a 5. Samozrejme, toto môžeš urobiť **s každou sadou najviac raz**.

No a v budúcej sérii budeš môcť za body riešiť ďalšie dve sady z Liahne.

Samozrejme, nič ti nebráni riešiť aj úlohy z Liahne aj klasické programátorské úlohy v PRASKu.

Programátorskú Liaheň nájdeš na tejto stránke: <http://liahen.ksp.sk>

4. Skriňa plná tričiek

15 bodov za riešenie

Táto úloha je prebraná z aktuálneho domáceho kola [Olympiády v Informatike](#) kategórie B. Riešením tejto úlohy sa teda viete zapojiť aj do Olympiády (odovzdať to však musíte samostatne druhýkrát).

Ak sa Vám úlohy v PRASKu páčia, vyskúšajte si aj riešenie Olympiády, termín jej domáceho kola je do **30. novembra 2018**. Riešením domáceho kola máte možnosť postúpiť do kola krajského a skúsiť si riešenie aj v súťažnejších podmienkach a kratšom čase. Na postup na krajské kolo obvykle stačí 10 bodov, takže ak vyriešite túto úlohu a ešte niečo, postup budete mať zaručený.

Ak nevieš programovať, nezúfaj! Môžeš sa to naučiť a ešte za to získať body, ktoré sa ti budú počítať namiesto tejto úlohy.

Stačí, že pôjdeš na stránku Programátorskej Liahni (liahen.ksp.sk). Keď budeš riešiť sadu **variables_cpp**, bodmi, ktoré získaš si môžeš nahradiť riešenie tejto úlohy. Stačí ak na spodku tejto stránky odovzdaš pdf-ko s prezývkou, ktorú používaš na Liahni.

Ak máte akékoľvek otázky ohľadom tejto úlohy, napíšte Žabovi na zaba@ksp.sk

Jeden z príznakov toho, že chodíte na veľa informatických súťaží, je, že sa vám začnú hromadiť tričká. A Žabovi sa mu ich za tie roky nahromadilo vskutku neúrekom. Teraz ich má uložené v skrini na jednej veľkej kope. Problémom ale je, že Žaba nemá rád zmenu. Keď si teda ráno vyberá, ktoré tričko si dá na seba, zoberie vždy to, čo je navrchu. A večer ho dá do svojej ultramodernej práčky a sušičky v jednom, ktorá mu ho expresne operie a on si ho opäť položí na vrch kopy.

Lucka sa už nemohla pozeráť na to, ako Žaba chodí stále v tom istom tričku. Ráno mu preto prikázala, aby si obliekol tričko, ktoré mal na sebe oblečené najmenší počet krát. A keďže takých bolo v kope viacero, zobral to, ktoré bolo najvyššie (t.j. najbližšie pri vrchu). Svoje ďalšie zvyky však Žaba nezmenil – keď večer prišiel domov, tričko opral a hodil na vrch kopy. A tak sa po pár dňoch aj toto nové tričko stalo okukaným.

Lucka mu preto vždy raz za čas povedala, aby si zobral nové tričko a Žaba jej želanie naplnil – teda zakaždým z kopy vytiahol tričko, ktoré mal na sebe dovtedy oblečené najmenejkrát (a v prípade rovnosti to z nich, ktoré bolo nablížšie k vrchu kopy). A keď mu nič nepovedala, tak použil svoj starý prístup a proste zobral tričko, ktoré bolo navrchu. Lucka má teraz problém sa k Žabovi zľadiť, preto by potrebovala pomôcť. Vedeli by ste jej povedať, ktoré tričko bude mať Žaba kedy oblečené?

Úloha

Žaba má n tričiek. Všetky tričká sú uložené na jednej veľkej kope, postupne jedno na druhom. Na začiatku sú očíslované od 1 po n v poradí zhora nadol – teda tričko 1 je na vrchu kopy, pod ním je tričko 2, a tak ďalej, až po tričko n úplne na spodku. Pre každé tričko poznáte hodnotu t_i , ktorá označuje, koľkokrát ho už Žaba mal oblečené.

Vašou úlohou bude pre nasledujúcich q dní zistiť, ktoré tričko bude mať Žaba oblečené. Pritom viete, či si v daný deň zoberie tričko z vrchu kopy, alebo nájde najvyššie tričko v kope, ktoré mal oblečené najmenší počet krát.

Pamätajte na to, že po každom dni si Žaba dané tričko vyperie a položí na vrch kopy, pričom počet nosení tohto trička sa zvýši o jedna.

Formát vstupu

Na prvom riadku vstupu sú dve čísla n a q – počet tričiek na kope a počet najbližších dní, ktoré Lucku zaujímajú.

Na druhom riadku vstupu je n medzerami oddelených čísel $t_1, t_2 \dots t_n$. Hodnota t_i označuje, koľkokrát mal Žaba už oblečené tričko i .

Nasleduje q riadkov. Na každom z nich je jeden znak – Z alebo L – určujúci rozhodnutie, ktoré v daný deň Žaba spraví. Ak je na vstupe Z, Žaba si zoberie najvrchnejšie tričko kopy. Ak je na vstupe L, vyberie si to tričko, ktoré mal na sebe najmenší počet krát. V prípade, že je takýchto tričiek viac, vyberie si to, ktoré je v kope najvyššie.

Formát výstupu

Pre každý z q dní vypíšte na samostatný riadok číslo trička, ktoré si v ten deň Žaba oblečie.

Hodnotenie

Vaše riešenie bude testované iba na 5 rôznych vstupoch. Za každý vstup, na ktorý odpovie váš program správne získate 3 body.

V prvých dvoch vstupoch platí, že $n = 1\,000$ a $q = 5\,000$.

V zvyšných troch je $n = 100\,000$ a $q = 500\,000$.

V treťom vstupe môžete navyše predpokladať, že počas všetkých dní platí, že rozdiel najväčšieho a najmenšieho počtu nosení niektorého trička nepresiahne 2.

Vo všetkých vstupoch platí, že začiatkové hodnoty nosení t_i nepresiahnu $1\,000\,000$.

Príklady

vstup

```
5 7
2 3 2 4 2
Z
L
Z
L
L
L
Z
L
```

výstup

```
1
3
3
5
5
5
5
1
```

Prvý deň si Žaba oblečie tričko z vrchu kopy, teda tričko s číslom 1. Na druhý deň si má obliecť tričko s najmenším počtom nosení. Také sú v kope dve, číslo 3 a 5, vyššie z nich je však tričko 3, preto si ho oblečie a dá na vrch kopy. Tretí deň opäť zoberie toto tričko z vrchu kopy, tričko číslo 3 teda bolo nosené už štyri dni. Na štvrtý deň je tričko s najmenším počtom nosení tričko číslo 5, Žaba si ho preto oblečie. Piaty deň si má opäť vybrať tričko, ktoré mal na sebe najmenší počet krát. V tomto prípade má na výber tričká 1, 2 a 5. Najvyššie z nich, a zrovna na vrchu kopy, je tričko číslo 5, preto si ho oblečie. Na šiesty deň svoju voľbu zopakuje a v posledný deň si vyberie vyššie z dvoch tričiek, ktoré mal na sebe iba trikrát – tričko číslo 1.

5. Koráľkový náhrdelník

15 bodov za riešenie

Ak nevieš programovať, nezúfaj! Môžeš sa to naučiť a ešte za to získať body, ktoré sa ti budú počítať namiesto tejto úlohy.

Stačí, že pôjdeš na stránku Programátorskej Liahni (liahen.ksp.sk). Keď budeš riešiť sadu `conditions_cpp`, bodmi, ktoré získaš si môžeš nahradiť riešenie tejto úlohy. Stačí ak na spodku tejto stránky odovzdáš pdf-ko s prezývkou, ktorú používaš na Liahni.

Ak máte akékoľvek otázky ohľadom tejto úlohy, napíšte Paulinke na psmolarova@gmail.com

Kde bolo tam bolo, žili si Etká a Jemo. A žili by si šťastne až kým by nepomreli, ak by nenastal ten obávaný deň. Výročie. A Jemo naň, ako inak, zabudol. Spomenul si naň až v poslednú možnú chvíľu a rýchlo trielil do obchodu, kde úspešne kúpil pestrofarebný koráľkový náhrdelník a hrdý na svoj počín zazvonil Etké pri dverách.

Lenže, beda prebada! Jemo vrámci tohto stresujúceho nákupu zabudal na jednu mimoriadne podstatnú vec. Etká nie je veľký fanúšik príliš farebných vecí. Konkrétne, vyžaduje aby všetky jej veci mali nanajvýš dve farby.

Jemo teraz stojí pred dverami, Etko sa blíži a on potrebuje svoju chybu rýchlo napraviť. Najjednoduchšie bude vybrať časť náhrdelníka, ktorá má najviac dve farby, zvyšok odstrihnúť a vybraný úsek zviazať dokopy a podarovať Etko krásny dvojfarebný náhrdelník.

Úloha

Pomôžte Jemovi opraviť náhrdelník skôr ako Etko otvorí dvere.

Náhrdelník sa skladá z farebných koráliek. Na vstupe dostane popis farieb jednotlivých koráliek tak, ako sú za sebou poukladané v náhrdelníku. Dajte si pozor, že po poslednej korálke nasleduje hneď korálka prvá.

Vašou úlohou je nájsť **najdlhší**³ súvislý úsek náhrdelníka, ktorý obsahuje najviac korálky dvoch farieb.

Formát vstupu

V prvom riadku je číslo $1 \leq n \leq 100\,000$ udávajúce počet koráliek na náhrdelníku.

V druhom riadku je n medzerami oddelených čísel c_1 až c_n ($1 \leq c_i \leq 10^9$) – farby koráliek v náhrdelníku.

Formát výstupu

Na prvý riadok výstupu vypíšte dĺžku najdlhšieho náhrdelníka ktorý vie Jemo vytvoriť.

Na druhý riadok vypíšte indexy prvej a poslednej korálky ktoré budú tvoriť nový náhrdelník. Dajte si pozor, že náhrdelník je kruhový.

V prípade, že je riešenie viac, vypíšte ľubovoľné z nich.

Hodnotenie

Vaše riešenie bude otestované na 5 sadách vstupov. Za vyriešenie každej sady získate 3 body. Tieto sady sa líšia veľkosťou vstupných údajov.

Číslo sady	1	2	3	4	5
maximálne n	100	1 000	1 000	\$100,000	100 000
Ďalšie obmedzenia	$1 \leq c_i \leq 100$	$1 \leq c_i \leq 3$	$1 \leq c_i \leq 1\,000$	$1 \leq c_i \leq 3$	$1 \leq c_i \leq 10^9$

Príklady

vstup

```
7
3 1 2 1 2 2 3
```

výstup

```
5
2 6
```

V tomto prípade je najlepší náhrdelník 1-2-1-2-2. Tento vstup by sa mohol nachádzať aj v sadách 2 a 4.

vstup

```
6
1 2 5 3 1 2
```

výstup

```
4
5 2
```

Všimnite si, že vo výstupe je číslo prvej korálky väčšie ako číslo poslednej. To je v poriadku, pretože náhrdelník je cyklický (posledná korálka je spojená s prvou). Chceme vybrať korálky 1-2-1-2.

vstup

```
5
3 9 9 3 9
```

výstup

```
5
4 3
```

V tomto prípade je aj pôvodný náhrdelník vyhovujúci. Správne odpovede by boli aj 1 5 alebo 5 4.

vstup

```
8
4 5 1 4 6 5 1 2
```

výstup

```
2
1 2
```

Tu sa žiaľ nedá spraviť nič lepšie ako použiť niektoré dve susedné korálky. Všimnite si, že **nemôžeme** vytvoriť náhrdelník 1-4-1, pretože úsek musí byť na pôvodnom náhrdelníku súvislý.

³Bolo by fajn keby si ho Etko zvládla pretiahnuť cez hlavu.