



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 navyiac v ňom získate body, ktoré sa vám rátajú 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 sústreďenie**. 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 umiestení 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 zadaní 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 2. kola zimnej časti

Termín odoslania riešení tejto série je pondelok 4. marca 2019.

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. Podivné tabuľky

15 bodov za riešenie

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

Andrej s Dávidom sú šikovní a už dávno prelistovali všetky strany [podivného zošitu](#) a odhalili pravidlá, podľa ktorých sa správajú. Na poslednej strane však Dávid našiel vložený malý lístok s nápisom *Práskovňa*. Andrej si spomenul, že je to názov záložne neďaleko jeho obľúbenej reštaurácie. Zvedaví kamaráti sa tam ihneď vybrali a cestou rozmýšľali, čo len za tento lístok dostanú.

Po krátkom rozhovore a odovzdaní lístku vedúcemu *Práskovne*, dostali do rúk staré *podivné tabuľky* a kus kriedy, ktorou sa na ne kedysi písalo.

Hneď im napadlo, že podivné tabuľky budú podobné podivnému zošitu. Andrej si zobral jednu takúto tabuľku do rúk a celú ju poriadne prezrel. V pravom hornom rohu si všimol hrubo vyrytý text. Tento text bol rôzny na každej tabuľke. Celý natešený zobral Dávidovi z rúk kriedu a napísal na tabuľku číslo. A naozaj to bolo ako zošit, ktorý našiel v kúpeľni. Napísané číslo zmizlo a na jeho mieste sa objavil iný text. Chlapci už vedeli, že potrebujú prinútiť tabuľku vypísať text vyrytý v jej rohu, a tak neváhali a začali odhaľovať pravidlá jednotlivých *podivných tabuľiek*.

Úloha

V tejto úlohe dostanete prístup k *podivným tabuľkám*. Vždy, keď do nich vpíšete nejaké číslo, tabuľka vypíše odpoveď (číselnú, slovnú ...) dodržiavajúc jednoduché pravidlo.

Vašou úlohou bude toto pravidlo odhaliť a nájsť také vstupné číslo, pre ktoré tabuľka 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/5/1/2>

Hodnotenie

K dispozícii je 10 tabuľiek, za vyriešenie každej získate 1.5 bodu. Náročnosť postupne stúpa, môžete ich však riešiť v ľubovoľnom poradí.

2. Reťazce v Lemingove

15 bodov za riešenie

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

Zadanie možno vyzerá odstrašujúco dlhé. Je však jednoduchšie ako by sa mohlo zdať. Odporúčame ho riešiť úlohu po úlohe a veľa si kresliť. Už za nakreslenie niekoľkých obrázkov viete získať pomerne veľa bodov. Vaše kresby nám môžete odfoťiť a vložiť do riešenia.

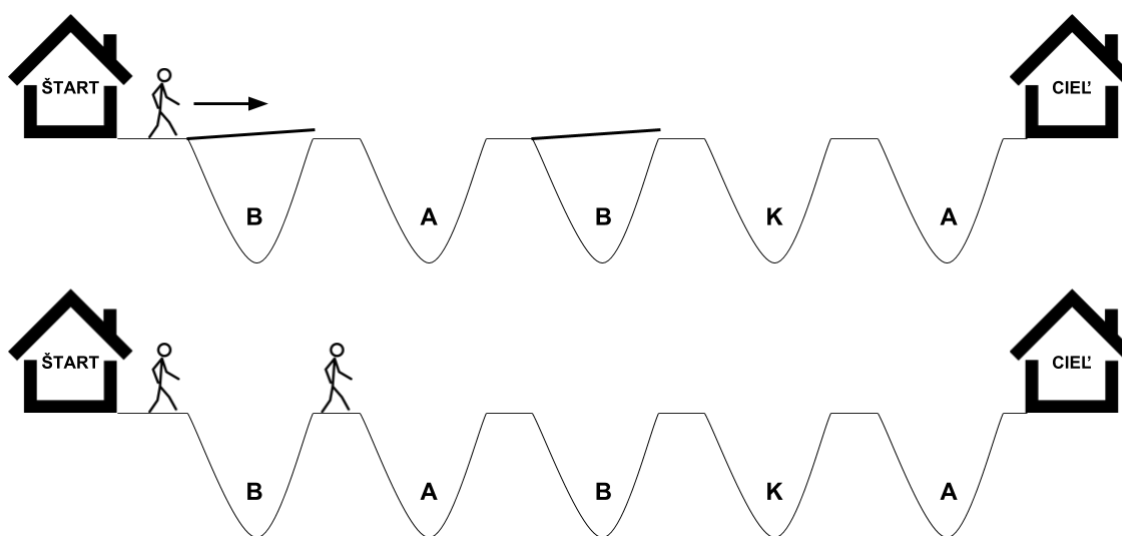
Poznáte lemingov zo známej počítačovej hry? Sú to malé postavičky, ktoré postupne vychádzajú zo svojho domčeka, idú neustále dopredu a snažia sa doraziť do cieľa. Niektorí z nich počas tejto cesty možno umrú, ale

to až tak nevádi, lebo ich je skutočne veľa. Byť lemingom však vôbec nie je také ľahké, hlavne keď máte pred domom obrovskú diery, ktorej sa vydáte v ústrety. Ak do nej spadnete, umriete.

V Lemingove je medzi štartovacím a cieľovým domčekom v rade niekoľko obrovitánskych (pre naše účely nekonečne hlbokých) dier, každá označená niektorým písmenom anglickej abecedy. Každú sekundu sa zo štartu vynorí nový leming, ktorý sa celý čas hýbe doprava a snaží sa dostať k cieľu.

Keď ste prišli na návštevu Lemingova s desom ste sa pozerali, ako sa každú sekundu vrhne ďalší leming do prvej diery. Nijakým spôsobom sa totiž cez ňu nevie dostať. V tomto okamihu hrôzy ste však v sebe objavili skryté magické schopnosti. Vysvitlo, že keď dostatočne nahlas povieť každú sekundu **jediné** písmeno, nad dierami označenými týmito písmenami sa na sekundu objaví most. Prechod cez most trvá lemingovi presne sekundu. To znamená, že leming stojaci na ľavom okraji jamy, nad ktorou sa objavil most, má akurát dosť času, aby sa dostal na jej druhú stranu.

Pozrite sa na obrázok nižšie. Na začiatku sekundy sa pri štarte objavil nový leming a vy ste zakričali písmeno B. Nad jamami s písmenom B sa objavil most. Cez prvú diery vedie most, leming po ňom môže prejsť a dostať sa na druhú stranu. Následne všetky mosty zmiznú a zo štartu sa vynorí ďalší leming. V tomto momente ste však v nezávideniahodnej situácii, nech totiž povieť akékoľvek písmeno, nezachránite oboch lemingov. Môžete povedať písmeno B a zachrániť druhého leminga, alebo písmeno A a pomôcť prvému. Alebo budete zlí a povieť napríklad Z a oboch ich pošlete do jamy.



Úloha

- a) (2 body) Nech v jamách sú postupne napísané písmená **BABKA** (ako na obrázku vyššie). Zo začiatkovej pozície, v ktorej je živý iba leming, ktorý práve vyšiel zo štartu, ste postupne povedali nasledovné písmená: **BBABKAABABABKA** – každú sekundu jedno v tomto poradí. Nakreslite, kde sa nachádzali lemingovia po tom, čo ste prečítali prvé štyri písmená, teda **BBAB** a po tom, čo ste prečítali prvých trinásť písmen, teda **BBABKAABABABK**.

Popri tom, ako ste vraveli text **BBABKAABABABKA** ste si pozorne všímali, kde sa nachádzajú lemingovia. Zaujímalo vás totiž koľkokrát sa v tomto texte vyskytlo slovo **BABKA**, ktoré je tvorené písmenami z jám. Ako vám vie poloha lemingov prezradiť, že v hovorenom texte sa nachádza slovo z jám? Prečo je tomu tak?

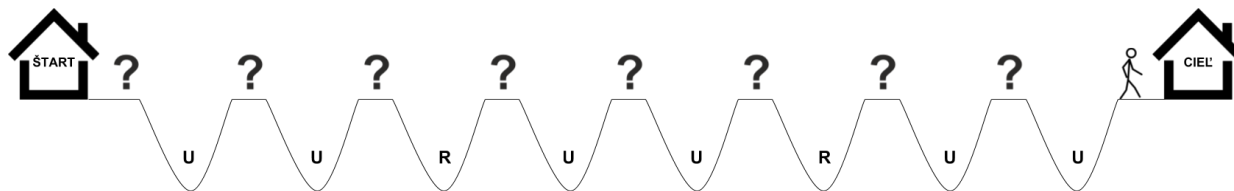
- b) (2 bod) Hľadanie slov v texte sa vám zapáčilo a povedali ste si, že by to mohlo byť aj pomerne užitočné. Pozor však, že hľadané písmená sa v texte musia nachádzať za sebou. V texte **BAZBKA** sa teda slovo **BABKA** nenachádza ani raz.

Vymyslíte text dlhý 15 písmen, v ktorom sa slovo **KATKA** nachádza čo najviackrát.

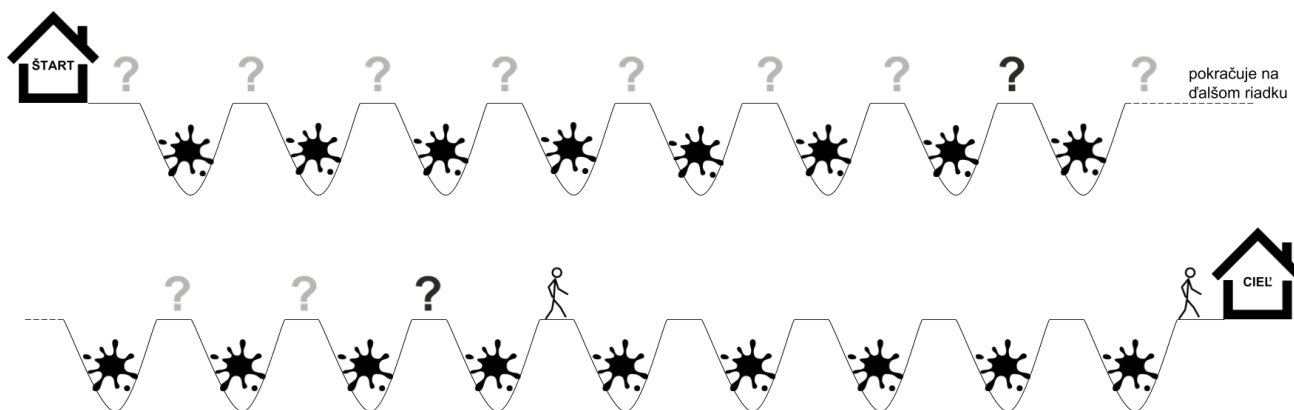
- c) (4 body) Keď ste sa vrátili z Lemingova, po rozume vám stále behal ich zvláštny spôsob života. Začali ste si teda kresliť na papier, akoby sa lemingovia pohybovali pri rôznych písmenách v jamách a textoch, ktoré by ste čítali. Prekreslovať zakaždým všetkých lemingov je ale strašne otravné. Radšej by ste sa sústredili iba na jedného – toho čo je najviac vpravo a teda najbližšie k cieľu. Posúvať jedného leminga je totiž oveľa jednoduchšie. Keď však tento leming prišiel do cieľa a zmizol v domčeku, zrazu ste nevedli, kde sú ostatní lemingovia. Možno to viete spätne vypočítať.

Predstavte si, že poznáte písmená v jamách a pozíciu najpravejšieho leminga, ktorý ešte nevošiel do cieľového domčeka. Vedeli by ste zistiť, kde sa nachádzajú ostatní lemingovia? Viete o každom mieste naľavo od neho povedať, či sa tam leming nachádza alebo nie? Je toto umiestnenie jednoznačné? Odôvodnite svoje tvrdenia v popise. Takisto do popisu napíšte postup, ktorým by ste zisťovali pozície zvyšných lemingov.

Aby ste lepšie pochopili, čo v tejto úlohe treba robiť, skúste si ju vyriešiť na konkrétnom príklade z obrázku nižšie. Za správne určenie pozícií lemingov v tomto obrázku môžete **získať až 2 body**.



- d) (4 body) Povedať niečo o lemingoch keď poznáte písmenká v jame je ľahké, čo však, keď nevíete ani to? Predstavte si, že medzi štartom a cieľom je 17 jam a nepoznáte písmená, ktoré im prislúchajú. Medzi jamami sa nachádzajú nejaký lemingovia. Vy ste si všimli jedného, ktorý práve dorazil na koniec slova a chystá sa vstúpiť do domčeka s cieľom. K nemu najbližšie stojí leming za 12-tou jamou, medzi nimi žiaden ďalší leming nie je. Rozhodnite a svoje tvrdenie odôvodnite, či za siedmou a jedenástou jamou stojí leming.



- e) (3 body) Počas toho ako ste sa hrali s rôznymi slovami sa vám najviac zapáčili také, v ktorých sa pravidelne opakovala nejaká postupnosť písmen, mali takzvanú periódu. Slovo je periodické práve vtedy, ak vzniklo opakovaním tej istej postupnosti písmen. Napríklad periodické slovo MAMA vzniklo opakovaním postupnosti MA. Slovo ABABABABAB mohlo vzniknúť aj opakovaním štyroch písmen ABAB aj opakovaním dvoch písmen AB. V takomto prípade nás však zaujíma najkratšia možná perióda.

Presnejšie, slovo má periódu p , ak platí, že ľubovoľné dve písmená, ktoré sú od seba vo vzdialenosti p , sú rovnaké. Slovo ABABABAB má preto periódu 2, slovo PAAPPAAPPAAP má periódu 4 a slovo LLLLLL má periódu 1. Dokonca, mierne prekvapivo, napríklad slovo KATKA má periódu 3 – naozaj platí, že písmená vo vzdialenosti tri sú v ňom rovnaké. Ako ďalšie príklady si môžeme ukázať slovo KATKAKAK s periódou 7 a slovo MACKA s periódou 5.

Zaujíma vás, či vám vedia lemingovia pomôcť s určovaním periódy. Do jam ste teda napísali písmená slova, ktorého periodickosť chcete zistiť. Následne ste postupne čítali písmená tohto slova a sledovali, ako sa hýbu lemingovia. Keď ste skončili, pri cieľovom domčeku určite stál leming a ďalší boli možno niekde po ceste. Viete na základe tohto zistiť, akú periódu má zadané slovo? Svoje tvrdenia odôvodnite. Snažte, aby bolo vaše riešenie všeobecné a fungovalo na ľubovoľnom slove.

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 2

15 bodov za riešenie

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

V minulom kole mali *Práskači* plné ruky práce a aj vďaka vašej pomoci stihli rozlúsknuť do konca roka všetky audio šifry. Prišiel však nový rok a s ním aj veľa nového kompromitujúceho materiálu. Ľudia majú o posielanie materiálu taký veľký záujem, že sa *Práskači* rozhodli zriadiť automatizovanú horúcu linku. Tá by im mala ušetriť dosť času na to, aby zvyšok zvládli sami. Predtým ale potrebujú pomôcť s 3 najnovšími šiframi a so zavedením horúcej linky. Pomôžete *Práskačom* s ich výzvami?

Úloha

Táto úloha je pokračovaním úlohy *Audio Práskači* z minulého kola 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.

Podúlohy D, E (5 bodov)

V podúlohách D a E budete pomáhať *Práskačom* s AHL (Automatizovanou Horúcou Linkou). Najskôr si ukážeme príklad hovoru s AHL:

AHL - "Dobrý deň, dovolali ste sa k Práskačom. Ak chcete pokračovať v slovenčine, zvolte 1. If you wish to switch to English press 2."

Volajúci - "(stlačí 1)"

AHL - "Zvolili ste slovenčinu. Ak chcete napráskaať materiál z oblasti poľnohospodárstva, stlačte 1. Pre napráskanie IT, zvolte 2. Ak chcete zaslať medzinárodný materiál, stlačte 3. Inak zvolte 4."

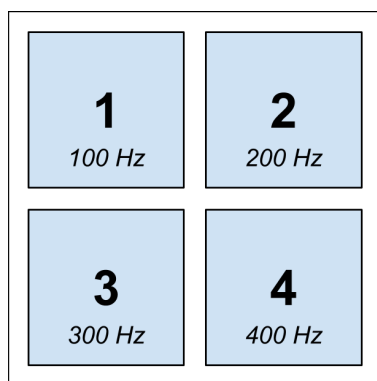
Volajúci - "(stlačí 3)"

AHL - "Zvolili ste medzinárodné napráskanie. Stlačte prosím 4 pre potvrdenie a spustíte nahrávku. Po ukončení stlačte 3."

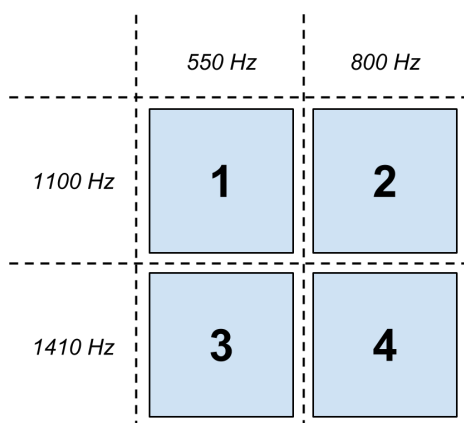
Volajúci - "(stlačí 4) (spustí nahrávku) (stlačí 3)"

AHL - "Ďakujeme za zaslania materiálu."

Vidíte ako jednoduché bude poslať *Práskačom* kompromitujúci materiál?! Pre zjednodušenie bude mať v tejto úlohe volajúci na výber iba zo 4 cifier na svojom telefóne. Zostáva vyriešiť, ako zakódovať stláčanie tlačidiel, aby sme ich vedeli poslať po telefónnej linke. Napríklad by sme mohli každej cifre priradiť jeden tón, ktorý by bol po jej stlačení vyslaný:



Systém AHL by potom vedel detegovať jednotlivé tóny. Problémom ale je, že jeden tón môže ľahko vzniknúť zašumením na linke. Mohlo by tak dochádzať k detekcii stlačenia klávesy aj v prípade, že ju volajúci vôbec nestlačil. Využijeme preto trochu iný systém. Každému riadku a každému stĺpcu priradíme jeden tón:



Stlačenie cifry potom reprezentujeme dvojicou **súbežne hrajúcich** tónov – tónom riadku a tónom stĺpca.³ Napríklad **zvuk cifry 3** budú tvoriť dva tóny s frekvenciami 550 Hz a 1410 Hz.

Podúloha D (2 body)

AHL je zatiaľ vo fáze testovania, preto *Práskachi* potrebujú hovory, na ktorých overia funkčnosť systému. Vašou úlohou je zostrojiť záznam **volajúceho** z ukážky vyššie. Vymyslený kompromitujúci materiál, ktorý môžete použiť v zázname nájdete na [Google Disku](#) v priečinku podúlohy D. Váš záznam by mal vyzeráť nasledovne:

(tóny stlačenia 1) (tóny stlačenia 3) (tóny stlačenia 4) (kompromitujúci materiál) (tóny stlačenia 3)

Výslednú nahrávku nám pošlite vo WAV formáte a nezabudnite **popísať váš postup**.

Podúloha E (3 body)

AHL zatiaľ nefunguje na 100% a niektoré hovory sa nepodarí dekodovať. Jeden taký hovor nájdete na [Google Disku](#) v priečinku podúlohy E. Vašou úlohou je dekodovať ho a poslať nám klávesy, ktoré volajúci stlačil. Vaše riešenie preto môže vyzeráť napríklad takto:

3, 1, 2, 2, záznam, 4

Postupoval(a) som nasledovne: Najskôr som...

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. Váš záznam hovoru z *Podúlohy D* nahrajte cez [špeciálne odovzdávateľko \(DriveUploader\)](#). Pri odovzdávaní nezabudnite správne vyplniť vaše meno a email.

³Veľmi podobný systém sa v praxi skutočne používa a volá sa [DTMF](#).

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. Sladká odmena

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 Liahne (liahen.ksp.sk). Keď budeš riešiť sadu **loops.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 Denisovi na denis@ksp.sk

Kika sa rozhodla, že napečie palacinky pre celú rodinu. Spravila teda cesto, zapla sporák a zohriala na ňom panvicu. Na neuveriteľne dlhý stôl naukladala neuveriteľne veľa tanierov, pekne vedľa seba ako sa na skúsenu gazdinú patrí, a začala piecť palacinky. Keď bola prvá palacinka upečená, Kika zavolala k svojmu neuveriteľne dlhému stolu celú rodinu a položila palacinku na prvý tanier na stole. Vždy, keď bola ďalšia palacinka hotová, Kika ju položila na prvý tanier na stole.

Keďže Kika dáva palacinky len na prvý tanier, jej rodina musela vymyslieť, ako sa rozdelia. Predsa len, je fajn byť prvý pri stole, ale čo ostatní? Jej rodina si teda vymyslela zaujímavé stravovacie návyky. Ak má niekto na svojom tanieri tri palacinky, tak jednu posunie doprava a ostatné dve zje. Nemôže predsa posunúť ďalej viac palaciniiek ako by sám zjedol. Keď Kika dopečie poslednú palacinku, zaujíma ju, koľko palaciniiek zostalo každému z rodiny na tanieri.

Úloha

Vašou úlohou je napísať program, ktorý vypíše koľko palaciniiek zostalo každému z rodiny na tanieri po tom, ako Kika dopečie poslednú palacinku. Pri stole sa nachádza k členov rodiny a Kika upečie n palaciniiek. Rodina sa správa podľa týchto stravovacích návykov:

1. Kika vždy pokladá čerstvo upečenú palacinku na prvý tanier na stole.
2. Ak má nejaký člen rodiny na tanieri 3 palacinky, tak dve zje a jednu posunie nasledujúcemu členovi rodiny. Kika, ako správna gazdiná, vie odhadnúť, koľko cesta prichystať. **Preto posledný človek nikdy nebude musieť posúvať palacinku ďalej.**
3. Ak má akýkoľvek člen rodiny menej ako 3 palacinky, tak nerobí nič.

Formát vstupu

Na prvom riadku vstupu sú dve celé čísla n ($1 \leq n \leq 10^{15}$) a k ($1 \leq k \leq 40$). Číslo n reprezentuje počet palaciniiek, ktoré Kika napečie, a k reprezentuje počet osôb pri stole.

Pri riešení v jazyku C++, Pascal, Java, a podobných, si dávajte pozor na maximálne číslo, ktoré sú schopné celočíselné premenné uložiť. Počet palaciniiek môže v posledných troch sadách tieto limity prekročiť. Odporúčame použiť 64-bitové premenné. V jazyku C++ preto namiesto typu `int` použijete `long long int`.

Formát výstupu

Na výstup vypíšete riadok, na ktorom bude k čísiel, pričom k_i vyjadruje počet palaciniiek, ktoré zostali človeku i na tanieri po tom, ako Kika dopiekla poslednú palacinku. Tieto čísla vypisujete na výstup bez medzier a za posledným číslom vypíšete znak konca riadku.

Hodnotenie

Vaše riešenie bude testované na 5 rôznych sadách vstupov. Rôzne sady sa budú líšiť špeciálnymi obmedzeniami:

Číslo sady	1	2	3	4	5
maximálne n	100	10 000	10^{12}	10^{15}	10^{15}
maximálne k	10	15	30	35	40

Príklady

vstup	výstup
7 4	1200
vstup	výstup
12 5	01100

5. Kopa neumytého riadu

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 Liahne (liahen.ksp.sk). Keď budeš riešiť sadu `arrays_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 Andrejovi na ajo@ksp.sk

Dávidovi sa počas skúškového nič nechce. A už vôbec nie umývať taniere. Vždy keď niečo zje, spraví minimum námahy a tanier položí niekam vedľa seba. Občas, keď sa mu už ťažko chodí po izbe, tak nejaké taniere pozbiera a uloží ich na kopy neumytých tanierov. Ešte menej často si povie, že nejaký tanier na kope sa už časom mohol stať toxickým. Kopa neumytých tanierov medzitým poriadne narastie a on sa rozhodne, že problém trochu zmierni a umyje niekoľko najvrchnejších tanierov.

Aby vedel či už taniere treba umývať, tak si pre každý z nich pamätá jedno číslo, ktoré vyjadruje kedy naposledy bol daný tanier umytý. Dávidovi sa však nechce stále premýšľať, či má ísť už umyť taniere a aký tanier v kope je najdlhšie neumytý. Požiadal vás preto, aby ste mu na to napísali program a on už nemusel nad riadom tak intenzívne premýšľať.

Úloha

Vašou úlohou je napísať program, ktorý bude simulovať kopy neumytých tanierov. Program by mal preto vedieť spracovať 3 typy udalostí: 1. Pridanie taniera, ktorý bol naposledy umytý v čase x . 2. Umytie najvrchnejšieho taniera na kope. 3. Nájdienie najtoxickšieho, teda najdlhšie neumytého, taniera z kopy.

Formát vstupu

Na prvom riadku vstupu je jedno celé číslo n ($1 \leq n \leq 1\,000\,000$). Potom nasleduje n riadkov. Na každom riadku sa nachádza popis jedného príkazu. Ako sme spomínali, existujú 3 typy príkazov:

Ak ide o príkaz 1. typu, na riadku sa bude nachádzať reťazec 'PRIDAJ x ', kde x je celé číslo a platí $1 \leq x \leq 1\,000\,000\,000$.

Ak ide o príkaz 2. typu, na riadku sa bude nachádzať reťazec 'UMY'.

Ak ide o príkaz 3. typu, na riadku sa bude nachádzať reťazec 'NAJSPINAVSI'.

Formát výstupu

Pre každý príkaz 3. typu vypíšete na samostatný riadok výstupu jedno číslo. To hovorí, že najdlhšie neumytý tanier na kope bol umytý naposledy v čase x . Ak sa na kope nenachádza žiadny tanier, vypíšete -1 .

Hodnotenie

Vaše riešenie bude testované na 3 sady. Rôzne sady sa budú líšiť špeciálnymi obmedzeniami:

Číslo sady	1	2	3
obmedzenie na n	1 000	$n \leq 500\,000$	žiadne
obmedzenie na x	žiadne	$x \leq 1\,000$	žiadne

Príklady

vstup

```
7
PRIDAJ 30
NAJSPINAVSI
PRIDAJ 20
PRIDAJ 10
NAJSPINAVSI
UMY
NAJSPINAVSI
```

výstup

```
30
10
20
```

Na prvom riadku určíme počet príkazov, ktoré budú nasledovať – 7. Prvý príkaz pridá na kopy jeden tanier, ktorý bol umytý naposledy v čase 30. Keď sa spýtame na najšpinavší tanier v druhom príkaze, na kope sa nachádza jediný tanier s časom 30. Potom pridáme ešte 2 taniere, pričom jeden sa stane najšpinavším, ten je ale v zápätí vybraný a našpinavším sa stáva tanier s číslom 20.

vstup

```
6
NAJSPINAVSI
PRIDAJ 8
PRIDAJ 10
NAJSPINAVSI
UMY
UMY
```

výstup

```
-1
8
```

Na vstupe je 6 príkazov. Pri prvom príkaze na nájdenie najšpinavšieho taniera nie je žiadny na kope, takže vypíšeme -1. Potom pridáme dva taniere, najšpinavším sa stane tanier, ktorý bol naposledy umytý v čase 8 a potom oba taniere umyjeme.