PRASK

Leták letnej 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 korešpodenčný 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 naviac 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 úlohu[1]. 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 chcem riešiť?

Riešenie korešpodenčného 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 Korešpodenčného 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 niekoľko[2] najlepších riešiteľov. Na sústredení 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 vyzerať 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.

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

[2] zhruba 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**.

PRASK
V. ročník, 2018/19
Katedra základov a vyučovania informatiky FMFI UK,
Mlynská Dolina, 842 48 Bratislava

Úlohy 2. kola letnej časti

Termín odoslania riešení tejto série je pondelok **6. mája 2019**.

Teoretické úlohy

V tejto časti ťa čaká niekoľko matematickejších úloh, ktoré úzko súvisia s informatikou. Ako riešenie týchto úloh treba poslať podrobne spísaný 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. Poťiaže s peniazmi

15 bodov za riešenie

Dano čerstvo zmaturoval. Od septembra sa chystá nastúpiť na matfyz, kde bude študovať informatiku a popri tom vedúcovať KSP aj PRASK. Počas prázdnin však chcel zažiť niečo nové. Zbalil si teda svoj modrý batôžtek a vybral sa stopom do Absurdistanu – krajiny neobmedzených možností.

V Absurdistane totiž máločo funguje tak akoby malo a ľudia sú zvyčajne jednoduchšieho charakteru. Šikovní junáci ako Dano teda vedia rýchlo spraviť dieru do sveta. Čo bolo v skutočnosti presne to, čo Dano spravil ako prvé, keď do Absurdistanu dorazil. Zobral do ruky svoj verný rýľ a vykopal dieru. Keď sa ho okolo idúci občan spýtal, prečo to spravil, Dano mu vysvetlil, že to preto, aby sa mohol do tej diery postaviť a vyzerať nižšie. Nižším ľuďom totiž zvyšní viac dôverujú. A okolo idúci občan mu uveril.

Netrvalo dlho a Dano v jame si získal dôveru všetkých občanov Absurdistanu. Túto dôveru následne premenil na politický kapitál a onedlho už sedel v ministerskej stoličke. Bohužiaľ, postaviť sa z nej nemohol, lebo potom by bol vyšší ako zvyšní ľudia a to by mu už nikto neveril. Dano sa stal ministrom financií v Absurdistane.

Úloha

Byť ministrom financií však nebolo také jednoduché. Dano veľmi rýchlo pochopil, aká problématická je menová politika Absurdistanu. Každá mena používa niekoľko platidiel, mincí alebo bankoviek rôznych nominálnych hodnôt, pre jednoduchosť ich však volajme dokopy **mince**. V rámci tohto zadania je teda v poriadku povedať, že na Slovensku používame 20 eurovú mincu. Sada mincí je navyše dobrá ak spĺňa nasledovné dve vlastnosti.

1. Majme neobmedzený počet mincí každého druhu. Pomocou týchto mincí vieme zaplatiť ľubovoľnú celočíselnú kladnú sumu peňazí.
2. Majme neobmedzený počet mincí každého druhu. Ak pri platení ľubovoľnej sumy postupujeme tak, že vždy vyberieme mincu s najväčšou hodnotou, ktorá nepresahuje sumu, ktorú máme ešte zaplatiť, na zaplatenie použijeme **najmenší** možný počet mincí.

Napríklad euro spĺňa obe tieto vlastnosti. Pomocou euromincí vieme platiť všetky celočíselné sumy peňazí. A dodržiac vyššie uvedený postup použijeme aj najmenší počet mincí. Zoberme si napríklad, že chceme zaplatiť sumu 34 euro. Najväčšia minca (opäť pripomínam, že pod pojmom minca rozumieme mince aj bankovky), ktorá sa do 34 zmestí je 20 eurová bankovka. Po jej použití nám ostáva zaplatiť 14 euro. Opäť najväčšia minca, ktorá je nanajvýš takto veľká je 10 eurovka, použijeme aj tú. Na zaplatenie zvyšných 4 eur potom použijeme najskôr 2 eurovú mincu, a potom opäť 2 eurovú mincu. Dokopy sme teda zaplatili $20 + 10 + 2 + 2 = 34$ peňazí a použili sme 4 mince. To je naozaj najmenší počet mincí, ktoré sme mohli použiť. Túto sumu sme mohli zaplatiť aj ako $20 + 5 + 5 + 2 + 1 + 1$, vtedy by sme však použili až 6 mincí.

Podúloha a. (2 body)

Nie každá sada mincí však musí mať tieto dve vlastnosti. Navrhnite sadu mincí, ktorá poruší **obe** podmienky. To znamená, že sa ňou nebudú dať zaplatiť všetky celočíselné sumy peňazí a existuje aspoň jedna hodnota S, pri ktorej platení opakovaním použitím najväčšej možnej mince nepoužijeme najmenší počet mincí.

K obom pravidlám uveďte aj príklad, v ktorom daná podmienka nie je splnená. V druhom prípade to znamená, že ukážete, že pri platení zvolenej sumy S uvedeným postupom je nutné použiť viac mincí ako pri platení iným spôsobom (vydávanie nie je dovolené).

Navrhnúť sadu mincí znamená určenie ich nominálnych hodnôt. Môžete napríklad povedať, že vaše mince majú hodnotu 1, 2 a 4. Takáto sada však spĺňa obe podmienky.

Podúloha b. (4 bodov)

Ani platenie v Absurdistane však nie je jednoduché. Ich mince majú totiž nasledovné hodnoty: 1, 4, 7 a 19. Niet preto divu, že pri platení nastávajú zmätky. Ktovie, ktoré hodnoty sa vôbec dajú zaplatiť a koľko najmenej mincí pri tom použiť. O tomto probléme sa Dano dozvedel, keď k nemu ráno prišiel človek a spýtal sa ho: "Koľko najmenej mincí mám použiť na zaplatenie sumy 1?". Hneď potom však prišiel druhý a spýtal sa to isté, akurát pre sumu 2. A takto to postupne pokračovalo až do obeda.

Dano sa preto rozhodol vydať brožúrku, v ktorej bude vysvetlené, koľko najmenej mincí je potrebných na zaplatenie všetkých súm, postupne od 1 po 20. Pre každú z týchto hodnôt určite, koľko najmenej mincí je potrebných na jej zaplatenie.

Zoberte si sadu mincí, ktorú ste navrhli v podúlohe a. Pre každú z hodnôt 1 až 20 zistite, koľko najmenej mincí je potrebných na jej zaplatenie.

V oboch prípadoch uvažujte stav, v ktorom máte z každej mince k dispozícii neobmedzene veľa kusov a vydávanie nie je povolené.

Podúloha c. (5 bodov)

Dano sa rozhodol, že zmení nominálne hodnoty mincí v Absurdistane. Je mu však jasné, že ho potom čaká odpovedanie na všetky tie otázky znovu. A čo keď sa rozhodne zmeniť hodnoty viackrát? Zišiel by sa mu postup, ktorým by tieto hodnoty vedeli vypočítať všetci občania Absurdistanu.

Predstavte si, že máte zadanú hodnotu n rôznych mincí. Táto sada môže, ale nemusí spĺňať vyššie uvedené podmienky. Navyše máte všetkých mincí neobmedzene veľa. Popíšte postup, ktorým by ste vedeli vypočítať, koľko najmenej mincí je potrebných na zaplatenie všetkých súm od 1 po S. Tento postup by mal byť všeobecný a mal by fungovať aj pre väčšie hodnoty n a S (napríklad 500).

Podúloha d. (4 bodov)

Nastal problém. V Absurdistane fungujú s novou sadou n mincí rôznych hodnôt. Dve z týchto mincí sú špeciálne a majú hodnoty x a y. Čím sú špeciálne? Na minci x je Danova podobizeň, na minci y je podobizeň aktuálneho premiéra. Dana majú všetci radi, lebo je nižší ako oni a mincu x všetci radi používajú. Naopak, premiéra nemá nikto rád, lebo sa týči nad nimi, a mincu y ľudia priam zahadzujú do koša. V krajine je teda v obehu oveľa viac mincí x ako y a to nie je dobré.

Dano preto vydal nové nariadenie. Podľa neho je nutné pri platení ľubovoľnej sumy použiť **aspoň toľko mincí y ako mincí x**. To znamená, že nemôžete pri platení použiť 2 mince x a iba jednu mincu y, bez ohľadu na to, akú sumu platíte a aké zvyšné mince ste použili.

Danova brožúrka a jeho postup však s takýmto niečom nepočítali. Ako sa dá upraviť postup z podúlohy c. tak, aby dodržiaval nové pravidlo? Popíšte teda postup, ktorý pre všetky sumy od 1 až S vypočíta, aký najmenší počet mincí je potrebný na ich zaplatenie s dodržaním nového pravidla.

2. Rôzne Merlinove problémy

15 bodov za riešenie

Ďaleko na severe, hlboko v Británii, v obrovskej knižnici ukrytej v útrobách podzemnej kamennej jaskyne už celé stáročia pracuje čarodejník Merlin. Merlin má na starosti veľké množstvo písomností z čarodejníckej literatúry, ktorá je pre ostatných čarodejníkov veľmi dôležitá.

Títo čarodejníci za ním najčastejšie chodia s troma typmi dotazov:

- Sú tieto dve kúzla vo *Veľkom zvitku* rovnaké?
- Nachádza sa v tejto knihe tento reťazec písmen?
- Koľko many potrebujem na použitie tohto kúzla?

Merlin potrebuje čo najrýchlejšie odpovedať na tieto otázky, aby sa mohol venovať aj iným prospešným činnostiam.

Úloha

Merlin nevie veľmi rýchlo pracovať s textom a písmenami. Keď chce napríklad porovnať, či sú dva reťazce

písmen rovnaké, postupne prechádza oba reťazce znak po znaku, až kým nenájde miesto, v ktorom sa líšia, alebo nepríde na koniec reťazcov. Ak napríklad porovnáva, či sú "abrakadabra" a "abrakazebra" rovnaké, porovnáva postupne "a"="a", "b"="b", "r"="r", "a"="a", a tak ďalej. Postupne overí, že prvých šesť znakov je rovnakých a potom na siedmom znaku má jeden reťazec písmeno "d" a druhý "z". Tieto reťazce teda nie sú také isté.

Overiť, že "athraighanboscaadhmaidseoinachloch" je to isté ako "athraighanboscaadhmaidseoinachloch" mu trvá 34 krokov. Pokiaľ by sa nejaké reťazce zhodovali na prvých dvetisíc znakoch a odlišovali sa na dvetisíc prvom, trvalo by mu zistiť, že sú rôzne, 2001 krokov. To je príliš dlho.

Podobne overiť, či sa nejaký reťazec nachádza v nejakej knihe je ťažké. Napríklad aby zistil, či sa $R =$ "abababababc" nachádza v $K =$ "abababababababababab", postupoval Merlin tak, že postupne overoval: Je prvých 10 znakov K rovnakých ako R? Sú znaky 2 až 11 v K rovnaké ako R? Sú znaky 3 až 12 v K rovnaké ako R? Sú znaky 4 až 13 v K rovnaké ako R? Sú znaky 5 až 14 v K rovnaké ako R? A tak ďalej až po: Sú znaky 11 až 20 v K rovnaké ako R?

Na to potreboval $10+1+10+1+10+1+10+1+10+1 = 55$-krát porovnať nejaké dva znaky (10 porovnaní v prípade, že dané písmená v K tvorili reťazec „ababababa" a až potom prišlo na nezhodu s písmenom „c", 1 porovnanie v prípade, že postupnosť v K začínala písmenom „b" a teda chybu objavil hneď pri prvom porovnaní). Overiť, či sa 1000 znakový reťazec nachádza v milión znakov dlhej knihe môže trvať skoro až miliardu krokov. A to je dosť veľa.

Na druhú stranu, ako správny čarodejník, Merlin dokáže rýchlo pracovať s číslami. Sčítať, vynásobiť či porovnať dve čísla mu trvá jeden krok **bez ohľadu na to, aké sú čísla veľké**. Podobne jednoduché operácie s jedným alebo dvoma číslami, vie tiež robiť takto rýchlo, ale keď je čísel viac, nevie to spraviť naraz. Sčítať n čísel by mu stále trvalo $n-1$ krokov.

Často, keď Merlin pracuje s reťazcami, prevedie si ich na zoznam čísel. Písmena abecedy si očísluje od a=1 až po z=26 a potom reťazec písmen je zoznam k nim prislúchajúcich čísel. Napríklad "abrakadabra" = 1, 2, 18, 1 11, 1, 4, 1, 2, 18, 1.

Nasledovné pojmy *súčet* a *magické číslo* budeme používať v podúlohách, tak si ich teraz popíšme.

Súčet nejakého reťazca je súčet jeho písmen prevedených na čísla. Napríklad súčet "abrakadabra", je $S(\texttt{abrakadabra}) = 1+2+18+1+11+1+4+1+2+18+1 = 60$.

Magickú hodnotu nejakého reťazca vypočítame nasledovne: Opäť písmenám priradíme čísla a=1 až z=26, ale tentokrát tieto písmená pred sčítaním ešte niečim vynásobíme. Posledný znak vynásobíme jednotkou, predposledný číslom 47, predpredposledný číslom 47 krát 47. i-ty znak od konca (začínajúc od $i = 0$) i krát vynásobíme číslom 47.

Magická hodnota reťazca "zebra" je $M(\texttt{zebra}) = 47^4 \cdot 26 + 47^3 \cdot 5 + 47^2 \cdot 2 + 47 \cdot 18 + 1 \cdot 1 = 4879681 \cdot 26 + 103823 \cdot 5 + 2209 \cdot 2 + 47 \cdot 18 + 1 = 126871706 + 519115 + 4418 + 846 + 1 = 127396086$ (Zápis a^k voláme umocňovanie a je to zápis pre vynásobenie čísla a k-krát. Napríklad $47^4 = 47 \cdot 47 \cdot 47 \cdot 47 = 4879681$)

Podobne magická hodnota reťazca "abrakadabra" je $M(\texttt{abrakadabra}) = 55266621785409182$.

Vypočítať súčet či magické číslo nejakého reťazca dlhého ℓ zaberie Merlinovi rádovo ℓ krokov.

Podúloha a – úvod

V Merlinovom *Veľkom zvitku* sú postupne napísané všakovaké kúzla, v poradí v akom ich jeho predkovia objavili. Každé kúzlo je reťazec malých písmen anglickej abecedy. Mohlo sa stať, že nejaké kúzlo bolo objavené viackrát, a tak sa môže vo zvitku nachádzať rovnaké kúzlo opakovane. Ostatní čarodejníci pravidelne Merlina navštevujú a pýtajú sa ho otázky tvaru: "Je i-te kúzlo a j-te kúzlo vo *Veľkom zvitku* rovnaké?"

Merlin tiež vlastní *Malý zvitok*, ktorý si už dávnejšie vyrobil. Na i-tom riadku tohto zvitku je napísaný *súčet* i-teho kúzla a tiež jeho *magická hodnota*, ktoré spočítal postupom uvedeným vyššie.

Merlin teraz rozmýšľa, či by nedokázal použiť *Malý zvitok* na to,
aby dokázal by na otázky čarodejníkom rýchlo a okamžite, bez toho aby kúzla porovnával pracne, znak po znaku.

Podúloha a – (4 body)

Nájdite dve **rôzne kúzla**, ktoré majú **rovnaký súčet**, alebo stručne ukážte, prečo také kúzla neexistujú.

Nájdite dve **rôzne kúzla**, ktoré majú **rovnakú magickú hodnotu**, alebo stručne ukážte, prečo také neexistujú.

Stručne popíšte postup, ako Merlin môže odpovedať na otázky ostatných čarodejníkov, len pomocou *Malého zvitku*. Napríklad, čo s čím má Merlin porovnať a ako má vyhodnotiť výsledok, ak chce zistiť, či sú i-te a j-te kúzlo vo *Veľkom zvitku* rovnaké.

Podúlohy b a c. – úvod

Niektorí čarodejníci nevedia čítať. A tak chodia za Merlinom, donesú mu knihu dlhú k znakov a spýtajú sa, či sa v knihe nachádza nejaký konkrétny reťazec dlhý r znakov.

Napríklad "`braka`", "`dabra`" aj "`a`" sa nachádzajú v "`abrakadabra`", ale "`zebra`" ani "`aaa`" sa tam nenachádzajú.

Podúloha b (3 body)

Kráľ Pendragon mal rád čísla, pravidelnosť a rád písal knihy. Všetky jeho knihy majú presne milión znakov (malých písmen anglickej abecedy) a majú zaujímavú vlastnosť: Každý úsek 1000 znakov v nej má iný súčet.

Navrhnite postup, ako efektívne zistiť, či sa zadaný 1000 znakový reťazec nachádza v danej knihe napísanej kráľom Pendragonom.

Keď hovoríme "efektívne" máme tým namysli, že tento postup netrvá príliš dlho. Skúste nájsť postup, ktorý potrebuje niekoľko miliónov krokov, prípadne zopár desiatok miliónov krokov, nie však miliardu.
Bežné matematické operácie, ako previesť písmeno na číslo, sčítať dve čísla, porovnať dve čísla, vynásobiť dve čísla trvajú jeden krok. Avšak už sčítať n čísel, alebo pozrieť sa na n hodnôt trvá n krokov.

Podúloha c (3 body)

Navrhnite spôsob, ako efektívne zistiť, či sa zadaný reťazec R dĺžky r nachádza v zadanom reťazci K dĺžky k.

Poznámky: O reťazci K už nemáte žiadne záruky, ako v predošlej podúlohe. Možno sa vám bude lepšie spisovať riešenie, ak budete používať označenie K_i resp. $K[i]$ pre i-ty znak knihy a R_j resp. $R[j]$ pre j-ty znak hľadaného reťazca. Rada: zamyslite sa, či sa dajú použiť magické čísla popísané vyššie v zadaní.

Podúloha d. (5 bodov)

Každé kúzlo potrebuje na vyčarovanie toľko many, koľko jeho začiatkov (prvých niekoľko písmen) je zároveň aj jeho koncom (posledných niekoľko písmen). Napríklad kúzlo "`abrakadabra`" spotrebuje 3 many, lebo "`a`", "`abra`" a "`abrakadabra`" sú všetko začiatky, ktoré sú zároveň aj konce. Kúzlo "`aaaaa`" spotrebuje 5 many a "`ttottogttottogttottogtt`" tiež 5 many (začiatky a zároveň konce sú "`t`", "`tt`", "`ttottogtt`", "`ttottogttottogtt`" a celé kúzlo). Kúzlo "`abracaz`" potrebuje len jednu manu, lebo žiaden začiatok okrem celého kúzla nie je aj koncom.

Navrhnite postup, ktorý pre dané kúzlo čo najrýchlejšie vypočíta, koľko many potrebujeme na jeho vyčarovanie. Váš postup by mal použiť rádovo toľko krokov, aká je dĺžka kúzla.

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. Andrej vymyslí

15 bodov za riešenie

Ak máte akékoľvek otázky ohľadom tejto úlohy, napíšte Andrejovi na ajo@ksp.sk

Táto úloha nadväzuje na úlohu Akútny upgrade z prechádzajúceho kola. Odporúčame si preto aspoň letmo prečítať jej zadanie a aj vzorové riešenie. Na vyriešenie tejto úlohy to síce nie je nutné, myslíme si však, že vám to vie pomôcť.

Timkinmu učiteľovi informatiky sa veľmi páčilo jej vypracovanie domácej úlohy na PRAStarom Kalkulátore. Tak si povedal, že najbližšiu úlohu by mohla celá trieda vypracovať na tomto prastarom počítači. Kedže je ale Timkin kalkulátor jediný zachovaný svojho druhu, dal im najlepší online simulátor PRAStarého Kalkulátora na internete. Jitka sa práve vrátila nevyspatá zo sústredenia a to posledné čo sa jej chce robiť, je riešiť nejaké úlohy.

Úloha

Pomôžte Jitke s jej domácou úlohou a dožičte jej tak zaslúžený spánok.

Podúloha a. (3b)

V rúre sa nachádzajú dve čísla $a (1 \leq a \leq 50000)$ a $b (0 \leq b \leq 1000)$. Vypíšte na výstup jediné číslo x, pričom $x = a^b$. Inými slovami, vypíšte na výstup a umocnené na b.

vstup	výstup
5 3	125

$5^3 = 5 \cdot 5 \cdot 5 = 125$.

Podúloha b. (3b)

V rúre sa opať nachádzajú dve čísla a, b ($1 \leq a, b \leq 10\,000$). Vypíšte na výstup jediné číslo x, ich najväčší spoločný deliteľ. Pokiaľ vaše riešenie používa príliš veľa krokov, pozrite si priložený link ešte raz :-)

vstup	výstup
15 20	5

5 delí 15 aj 20 bezo zvyšku. Zároveň toto pre žiadne väčšie číslo neplatí.

Podúloha c. (3b)

c) (3 bod) V rúre sa nachádza číslo $x (0 \leq x \leq 50000)$ a postupnosť $n (1 \leq n \leq 100)$ kladných čísiel. Na výstup vypíšte jediné číslo p, pričom p je počet čísel v postupnosti, ktoré sú väčšie ako x.

vstup	výstup
4 12 5 2 4 8 9 1	4

Vidíme, že prvé číslo je x teda $x = 4$. V postupnosti sú potom väčšími číslami čísla 12, 5, 8, 9. Výsledok je teda 4

Podúloha d. (3b)

V rúre sa nachádza postupnosť $n (1 \leq n \leq 100)$ kladných čísel. Je garantované, že niekde v tejto postupnosti sa nachádza jednotka. Určite aspoň raz, možno viackrát. Vašou úlohou je vypísať, na koľkom mieste sa v tejto postupnosti nachádza prvá jednotka.

vstup	výstup
5 4 3 9 1 2 55 8 1 3	5

Jednotka sa v tejto postupnosti nachádza na dvoch miestach, na piatom a deviatom. Na výstup teda vypíšeme prvšiu pozíciu a to 5.

Podúloha e. (3 b)

V rúre sa nachádza postupnosť $n (1 \leq n \leq 100)$ kladných čísel. Na výstup vypíšte túto postupnosť zoradenú od najmenšieho po najväčší.

vstup	výstup
1 12 3 4 2	1 2 3 4 12

Odovzdávanie

Odovzdávanie funguje rovnako ako v predchádzajúcej úlohe. Počas minulého kola malo niekoľko z vás problém s odovzdávaním. V prípade problémov s testovaním, napr. keď testovač testuje váš program dlho, napíšte Andrejovi.

Programátorské úlohy

Tieto úlohy sú zamerané na praktickú tvorbu programov v niektorom vyššom programovacom jazyku ako je napríklad Python, C++ alebo Pascal. Na stránke odovdzávaš **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 naším 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. Sekanie bambusu

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 Danovi na danoravec@gmail.com

Globálne otepľovanie sa prejavuje stále viac a viac. Ľadovce sa topia. Hladiny oceánov sa dvíhajú. Živočíchy vymierajú. V ťažkej situácií je ale potrebné neprepadnúť panike a šíriť medzi ľuďmi radosť v podobe hudby. Na to však potrebujeme najdokonalejší hudobný nástroj. Tým je jednoznačne Panova flauta. Rozhodli ste sa, že nám ju vyrobíte z bambusu. My vás dovedieme ku bambusovému záhonu. V ňom sú bambusy rôznych výšok rastúce tesne vedľa seba v jednom dlhom rade. Sú medzi nimi také malé rozostupy, že nie je možné odseknúť bambus, ktorý je medzi dvoma inými bambusmi. V každom momente teda viete odseknúť iba bambus, ktorý je úplne vľavo, alebo úplne vpravo. Okamžite po odseknutí celý odseknutý bambus prilepíme na pravú stranu flauty.

Určite nechcete plytvať materiálom, preto bambus odseknete vždy celý a neostane z neho nad zem trčať ani kúsok. Zároveň však potrebujete bambusy sekať v takom poradí, aby bol každý ďalší odseknutý bambus dlhší, ako ten predchádzajúci. To je preto, že sa ponáhľame a nemáme dosť času na usporiadavanie toľkých paličiek ešte pred samotnou výrobou flauty. Bambusy musia byť vo flaute usporiadané podľa dĺžky, aby boli podobné tóny vedľa seba. No a nechceme predsa nejaký bambus odseknúť a zahodiť ho len preto, aby sme sa tým dostali k nejakému dlhšiemu. To je predsa zbytočný výrub lesov a proti tomu bojujeme.

Pozrime sa na príklad. Bambusy v záhone majú postupne zľava doprava výšky 1, 2, 5, 3, 2. Nám sa oplatí zobrať najskôr bambus naľavo s výškou 1. Máme teda flautu z jedného bambusu výšky 1. Ostane nám záhon s výškami 2, 5, 3, 2. Teraz zoberieme bambus úplne napravo s výškou 2. Flauta už je 1, 2 a záhon je ešte 2, 5, 3. Teraz zoberieme dvakrát bambus sprava a získame tak flautu 1, 2, 3, 5 a ostane nám záhon s jediným bambusom výšky 2. Postavili sme teda flautu šírky 4.

Čím viac tónov vieme zahrať, tým viac zachránime svet, to vie predsa každý. Chcete teda, aby flauta pozostávala z čo najviac bambusov. Akú najširšiu flautu nám viete postaviť?

Úloha

My vám popíšeme výšky bambusov tak, ako vedľa seba rastú v záhone. Vy chcete vyrobiť z bambusov flautu. To znamená, že sa v každom momente rozhodnete, či odseknete bambus vľavo, alebo bambus vpravo. Ten odseknete a pridáte ho ku flaute sprava. Odseknuté bambusy musia tvoriť rastúcu postupnosť.

Variant A

Zistite iba dĺžku najdlhšej rastúcej postupnosti, ktorú vieme vytvoriť vyššie popísaným spôsobom. Ináč povedané, vypíšte, koľko najviac bambusov vieme odseknúť tak, aby nám z toho vznikla flauta.

Variant B

Na získanie plného počtu bodov stačí vyriešiť Variant B

Nemusíte nám hovoriť dĺžku postupnosti. Vypíšte namiesto toho nejaký postup seknutí, ktorý nás dovedie k nejakej najdlhšej možnej flaute.

Formát vstupu

Na prvom riadku je číslo n - počet bambusov v záhone.

Na druhom riadku je n čísiel b_1 až b_n, výšky bambusov zľava doprava. Bude platiť, že $1 \leq b_i \leq 2n$.

Formát výstupu

Variant A

Na jediný riadok výstupu vypíšte číslo k - najväčší možný počet bambusov vo flaute.

Variant B

Na jedinom riadku výstupu vypíšte reťazec dĺžky k, kde k je najväčší možný počet bambusov vo flaute, ktorý sa dá dosiahnúť. Tento reťazec bude predstavovať postupnosť seknutí. Za každé odseknutie ľavého bambusu dajte znak `L`, za každé odseknutie bambusu sprava dajte znak `R`. Reťazec `LRR` teda bude znamenať, že sme najskôr odsekli ľavý potom pravý a potom znova pravý bambus.

Hodnotenie

Je 15 testovacích sád. Za každú viete získať 1 bod. Približne platí, že čím vyššie je číslo sady, tým väčšie je n.

Ak riešite *Variant A*, vaše riešenia sa otestujú na všetkých sadách okrem 3, 6, 12, 15. Za toto riešenie teda viete získať až 11 bodov. Plný počet bodov dostanete, ak vyriešite *Variant B*. Takéto riešenie sa otestuje na všetkých sadách.

Interne testovanie vašich riešení funguje jednoduchým pravidlom. Ak je výstupom vášho programu číslo, testovač sa k nemu správa ako k riešeniu *Variantu A*. V opačnom prípade ho označí za riešenie *Variantu B*.

Číslo sady	1, 2, 4, 5	3,6	7	8	9	10 až 15
maximálne n	10	20	100	10 000	50 000	200 000

Navyše, sadách 1, 2, 3, 8, 9, 10, 11 a 12 sú všetky výšky bambusov rôzne. V ostatných sadách môžu byť niektoré výšky rovnaké.

Príklady

Variant B

vstup

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

výstup

```
LRRR
```

Vznikne flauta, kde dĺžky bambusov budú zľava doprava postupne 1, 2, 3, 5. Určite nemôžeme použiť všetky bambusy zo vstupu, pretože sú tam dva rovnakej dĺžky, takže by vzniknutá postupnosť určite nebola ostro rastúca.

Variant A

vstup

```
4
1 2 4 3
```

výstup

```
4
```

Môžeme použiť všetky bambusy zo vstupu

5. Krajina špiónov

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 Romanovi na roman.sobkuliak@trojsten.sk

V Absurdistane už od nepamäti funguje inštitúcia OIA (*Ochrana Integrity Absurdistanu*). V poslednej dobe sa ale na ňu valí veľká vlna kritiky. Špión Paroháč pri poslednej misii v Blahostane špiónoval tak dobre, že sa mu podarilo preniknúť do radov miestnej inštitúcie OIB (*Ochrana Integrity Blahostanu*). Tí mu ponúkli lukratívne miesto, ktorému Paroháč neodolal a prezliekol kabát. Pre Absurdistan teraz Paroháč predstavuje hrozbu.

Zatiaľ o Paroháčovom prešľape vie iba Hlaváč, šéf OIA. Na zatknutí Paroháča už aktívne pracuje. Zatiaľ vytvoril sieť zo svojich špiónov, v ktorej **každý špión sleduje práve jedného iného špióna**. Navyše, pretože Hlaváč neverí nikomu, platí, že **každý špión je sledovaný práve jedným iným špiónom**.

Hlaváč plánuje vyslať všetkých špiónov do Blahostanu. Potrebuje z nich vybrať čo najviac takých, ktorým o Paroháčovom prešľape povie. Nemôže však vybrať všetkých – aby dokonale fungovala jeho stratégia vzájomnej kontroly, musí každý zo špiónov, ktorý o prešľape vie, byť sledovaný špiónom, ktorý o prešľape nevie. Hlaváčovi by teraz veľmi pomohlo, keby ste zistili, koľko takýchto špiónov môže vybrať.

Úloha

Na vstupe dostane sieť špiónov, o ktorej platí, že každý špión sleduje práve jedného iného špióna a práve jedným špiónom je aj on sám sledovaný. Vašou úlohou je nájsť maximálny počet špiónov, ktorým môže Hlaváč vyzradiť tajomstvo o Paroháčovi tak, aby platilo, že každý z týchto špiónov je sledovaný špiónom, ktorý o tajomstve nevie.

Formát vstupu

Na prvom riadku vstupu je jedno kladné číslo n ($2 \leq n \leq 10^6$), počet špiónov. Tých si očíslujeme od 1 po n. Na druhom riadku vstupu je n medzerou oddelených čísel, pričom i-te číslo je číslo špióna, ktorý je sledovaný i-tym špiónom.

Formát výstupu

Na výstup vypíšte jediné číslo – maximálny počet špiónov, ktorým vie Hlaváč vyzradiť svoje tajomstvo.

Hodnotenie

Číslo sady	1	2	3	4	5
maximálne n	10	20	1000	200 000	1 000 000

Príklady

vstup

```
5
5 3 2 1 4
```

výstup

```
2
```

Hlaváč môže svoje tajomstvo povedať napríklad špiónom 3 a 4. Špión 3 je sledovaný špiónom 2 a špión 4 špionom 5, pričom 2 ani 5 o tajomstve nevedia. Uvedomte si, že viac špiónov vybrať nemôžeme, pretože inak by nutne aspoň jeden z nich bol sledovaný iným z vybraných špiónov. Odpoveď je preto 2.