



Leták letnej časti II. 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 základuškolákov, ktorých zaujíma matematika, informatika alebo by sa chceli naučiť programovať. Je to súťaž zameraná hlavne na 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 v ňom získate body, ktoré sa vám rátaajú 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 chcem 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 v informatike alebo Korešpondenčného Seminára z Programovania**. No a v neposlednom rade, pri **pohovoroch** do veľkých firiem ako Google, Facebook alebo Eset častokrát zaváži znalosť algoritmickeho programovania, ktoré si môžete pomocou nášho seminára trénovať.

Je tu však ešte jedna výhoda určená pre najlepších riešiteľov. Dvakrát ročne sa bude organizovať **týždenné sústredenie**. Naň pozývame niekoľko² 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 knihy, 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.

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

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

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.

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

Prvá 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 letnej časti

Termín odoslania riešení tejto série je pondelok 30. mája 2016.

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.

1. Princova dilema

15 bodov za riešenie

Ak máte akékoľvek otázky ohľadom tejto úlohy, napíšte Mišovi "Prefixovi" Sládečkovi na michalsladecek98@gmail.com

Kde bolo tam bolo, v jednom kráľovstve dal kráľ vyrobiť z rôznych kovov 16 guľičiek rôznych hmotností. Dlhو premýšľal, čo s nimi bude robiť, až kým mu v hlave neskrsol geniálny nápad – mal veľa synov a chcel nechať kráľovstvo v čo najšikovnejších rukách, tak prečo ich neotestovať? Každému zo svojich synov zadal ťažkú úlohu – na čo najmenej porovnaní dvoch guľičiek mali zistiť, ktorá guľička je tá druhá najľahšia. Kráľ si zároveň dáva pozor na to, aby synovia nemohli tipovať.

Úloha

Táto úloha je špeciálna, pri jej odovzdávaní budete používať webový formulár, ktorý nájdete tu: <https://prask.ksp.sk/specialne/prask/2/4/1/>. Do tohto formulára napíšete 2 čísla guľôčok a on vám povie, ktorá je ľahšia. Vašou úlohou je zistiť, ktorá guľôčka je druhá najľahšia. Guľôčky sú očíslované od 1 po 16. Z odpovedí na vaše otázky musí byť jednoznačne jasné, ktorá guľôčka je druhá najľahšia, za tipovanie body nedostanete. Čím menej otázok použijete, tým viac bodov dostanete. Za nájdenie správneho riešenia vo formulári môžete získať najviac 10 bodov.

Okrem toho odovzdajte aj pdf, v ktorom popíšete stratégiu, akou by ste na čo najmenej otázok vedeli nájsť druhú najľahšiu z n guľôčok pre všeobecné hodnoty n , nie iba $n = 16$. Nezabudnite napísať, koľko otázok pri vašej stratégii použijete v závislosti od počtu guľôčok. Za tento popis môžete získať až 5 bodov.

2. Prebľšený cirkus sa vracia

15 bodov za riešenie

Ak máte akékoľvek otázky ohľadom tejto úlohy, napíšte Edovi "Baklažánovi" Batmendijnovi na baklazan@ksp.sk

Pamätáte si ešte na blší cirkus?³ V tomto cirkuse majú obzvlášť veľké blchy, ktoré sa dajú trénovať. Minulý týždeň sa blšiemu cirkusu podarilo vyšľachtiť nový druh blch, ktoré sú ešte inteligentnejšie než tie staré a dajú sa s nimi robiť ešte zložitejšie predstavenia.

Pred blším vystúpením sa na zem nakreslí niekoľko (nepretínajúcich sa) kruhov a nejaké šípky medzi nimi. Každá šípka vedie od nejakého kruhu do nejakého (možno aj toho istého) kruhu. Ku každej šípke sa napíše buď jeden znak (písmeno alebo číslica), alebo sa nechá neoznačená, pričom musí platiť nasledovná vlastnosť: zo žiadneho kruhu nevedú von dve šípky označené rovnakým znakom, ani dve neoznačené šípky. Okrem toho sa jeden kruh označí ako štartovací a niektoré kruhy sa označia ako víťazné. Celé toto rozostavenie kruhov a šípok budeme nazývať *schéma*.

Samotné predstavenie vyzerá nasledovne: na začiatku sa jedna blcha postaví do štartovacieho kruhu a krotiteľ vyzve niektorého z divákov, aby mu niečo (v závislosti od druhu predstavenia to môže byť postupnosť písmen alebo číslic) napísal na papier. Toto niečo budeme nazývať *vstupný text*. Následne krotiteľ vstupný text nahlas číta znak po znaku. Vždy, keď krotiteľ prečíta nejaký znak, blcha urobí nasledovné:

- Najprv sa pozrie, či z kruhu, v ktorom práve stojí, vedie nejaká šípka označená znakom, ktorý krotiteľ práve prečítal. Ak áno, blcha skočí do kruhu, do ktorého vedie táto šípka.

³O jeho vzniku sa môžete dočítať v zadaniach druhej série zimnej časti: <https://prask.ksp.sk/ulohy/zadania/1107/>.

- V prípade, že z kruhu kde blcha stála, nevedla žiadna šípka označená práve prečítaným znakom, blcha sa pozrie, či z neho vedie aspoň neoznačená šípka. Ak áno, blcha skočí pozdĺž tejto šípky.
- V prípade, že z kruhu kde blcha stála nevedla ani neoznačená šípka, blcha sa na to vykašle, vyskočí von z kruhu a predstavenie sa predčasne skončí.

Na konci predstavenia sa krotiteľ pozrie, v ktorom kruhu blcha skončila. Ak skončila v niektorom z víťazných kruhov, zablahoželá divákovi a dá mu lízatko.

Všimnite si, že to, v ktorom kruhu blcha skončí závisí iba na schéme a na vstupnom texte. Ak si teda vezmeme jednu konkrétnu schému, je pre ňu jednoznačne určené, pri ktorých vstupných textoch skončí blcha v niektorom z víťazných kruhov. Tieto vstupné texty budeme nazývať *víťazné* pre danú schému. V tejto úlohe budeme pre zadané množiny víťazných vstupných textov kresliť schémy.

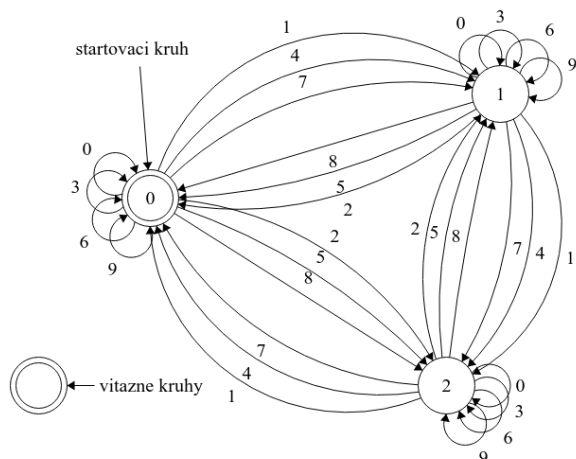
Príklad

Skúsme nakresliť schému, pre ktoré sú víťazné všetky čísla deliteľné tromi (ktoré môžu prípadne začínať zbytočnými nulami) a už nič iné. Vieme, že číslo je deliteľné tromi práve vtedy, keď je jeho ciferný súčet deliteľný tromi. Ak by si teda blcha vedela pamätať, aký je súčet prečítaných cifier a na konci podľa toho skočiť do správneho kruhu, mali by sme vyhrané. Dokonca by nám stačilo, aby si stále pamätala, aký zvyšok dáva súčet doteraz prečítaných cifier po delení tromi. Ak by bol tento zvyšok na konci nula, blcha by skočila do nejakého víťazného kruhu, inak by skočila do nejakého iného kruhu.

Blcha si, žiaľbohu, počas predstavenia nič nepamätá. Informáciu o zvyšku ciferného súčtu po delení tromi však môžeme mať počas predstavenia zakódovanú v pozícii blchy (teda v tom, v ktorom kruhu sa blcha práve nachádza). Môžeme teda nakresliť schému, ktorá bude mať tri kruhy (označme ich 0, 1 a 2). Pritom bude platiť, že:

- V kruhu 0 bude blcha vtedy, keď je súčet doteraz prečítaných cifier deliteľný tromi.
- V kruhu 1 bude blcha vtedy, keď súčet doteraz prečítaných cifier dáva zvyšok 1 po delení tromi.
- V kruhu 2 bude blcha vtedy, keď súčet doteraz prečítaných cifier dáva zvyšok 2 po delení tromi.

Z každého kruhu potom pôjde von 10 šípok označených ciframi 0 až 9. Šípky označené 0, 3, 6 a 9 budú vždy viesť do toho istého kruhu, lebo pripočítaním 0, 3, 6 alebo 9 sa zvyšok po delení tromi nezmení. Šípky označené 1, 4 a 7 budú vždy viesť do kruhu s o 1 väčším číslom (resp. z kruhu 2 do kruhu 0), lebo pripočítaním 1, 4 alebo 7 sa zvyšok po delení tromi zvýši o 1. Nakoniec, šípky označené 2, 5 a 8 budú viesť do kruhu s o 1 nižším číslom (resp. z kruhu 0 do kruhu 2). Za štartovací označíme kruh 0, lebo keď sme ešte nič neprečítali, súčet prečítaných cifier je 0. Jediným víťazným kruhom bude opäť kruh 0, lebo číslo je deliteľné tromi vtedy, keď jeho ciferný súčet dáva po delení tromi zvyšok 0.



Súťažná úloha

- a) (5 bodov) Nakreslite schému, pre ktorú budú víťazné všetky postupnosti znakov zložené z písmen „a“, „b“ a „c“ také, že žiadne dve za sebou idúce písmená **nie sú rovnaké**. Žiadne iné postupnosti už víťazné

byť nemajú. Teda napríklad postupnosť „abcbcbcbaba“ má byť víťazná, ale postupnosť „bacca“ nemá byť víťazná.

- b) (5 bodov) Nakreslite schému, pre ktorú budú víťazné postupnosti znakov, v ktorých sa niekde vyskytujú 4 áčka po sebe. Teda napríklad slovo „tadaaaa“ má byť víťazné, ale slovo „lalalalalala“ nemá byť víťazné.
- c) (5 bodov) Nakreslite schému, pre ktorú budú víťazné postupnosti znakov, v ktorých sa niekde (súvislo) vyskytuje slovo „tamtamtatam“⁴ (pozor, nie slovo „tamtamtamtam“). Slovo „alotamtamtambetam“ je víťazné, ale slovo „tamverutamtatamtam“ víťazné nie je.

V každej podúlohe okrem nakreslenia schémy aj zdôvodnite, prečo budú pre túto schému víťazné všetky postupnosti znakov špecifikované v danej podúlohe a prečo už žiadna iná postupnosť nebude víťazná. Nezapodíajte vo vašich schémach vyznačiť štartovací kruh a víťazné kruhy.

Praktická úloha

Pri práci s počítačom je potrebné vedieť pracovať aj s rôznymi nástrojmi, ktoré slúžia na úpravu obrázkov, prácu so zvukom či vyhľadáváním na internete. V tejto časti ťa preto zakaždým čaká nejaká netradičná úloha. V tomto kole to bude práca s programami na automatizáciu práce s klávesnicou a myšou.

3. Problémové pohostenie 3

15 bodov za riešenie

Ak máte akékoľvek otázky ohľadom tejto úlohy, napíšte Kubovi Havelkovi na kubo@ksp.sk

Je príjemný jarný deň. Vonku svieta slnko, na stromoch štebotajú vtáčiky, a tak ste sa rozhodli spraviť si výlet do hôr. Ako sa začal blížiť čas obeda, ku štebotu sa postupne pridalo škvŕkanie nenásytných žalúdkov. A potom padla tá otázka.

“Kto zbalil jedlo?” Z panikáriacich pohľadov členov výpravy bolo hneď každému jasné, že ho nezbalil nikto. Po krátkom dumaní a letmom pohľade na mapu ste sa rozhodli, že sa vyberiete k neďalekej horskej chate a skúsíte svoje šťastie tam.

Úloha a odovzdávanie

Úloha sa rieši interaktívne na tejto stránke: <https://people.ksp.sk/~prask/specialne/2/4/3/>.

Stránka vás prevedie ďalšími časťami vašeho dobrodružstva. Po splnení každej podúlohy vám stránka ukáže odkaz, kliknutím na neho dostanete body.

Veľa šťastia, snáď aj dobrú chuť :)

Ako riešiť

Pri riešení tejto úlohy budete musieť zautomatizovať a zrýchliť svoju prácu s klávesnicou a myšou. Existujú na to rôzne nástroje, s ktorými vás naučíme pracovať.

Ak používate operačný systém Windows, tak jedným z najlepších nástrojov je AutoHotKey. Stručný návod na jeho použitie nájdete tu: <https://prask.ksp.sk/navody/autohotkey>. Na Linuxe si ukážeme, ako pracovať s programom xdotool. Návod: <https://prask.ksp.sk/navody/xdotool>.

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 naším testovacím systémom, odporúčame ti zájsť na Programátorskú Liaheň (<http://betaliahen.ksp.sk>), kde si o tom môžeš prečítať úvodný text a vyriešiť si niekoľko jednoduchých úloh.

Ak však **nevieš programovať, 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.

⁴Takto sa totiž začína krotiteľova obľúbená pesnička.

Presnejšie to funguje takto. Na Liahni sa nachádzajú dve sady úloh, prvá zameraná na cykly a druhá na polia 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. Toto môžeš urobiť **s každou sadou najviac raz**.

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

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

4. Programátori a zásuvky

15 bodov za riešenie

Táto úloha sa dá nahradiť riešením sady **loops_cpp** na Liahni (betaliahen.ksp.sk). Ak chceš, aby ti namiesto bodov za riešenie tejto úlohy boli započítané body získané riešením spomínanej sady, na stránke odovdžaj pdf-ko s prezývkou, ktorú používaš na Liahni.

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

V T2⁵ sa počas dňa vyskytuje veľa programátorov a tí chcú, samozrejme, používať svoje notebooky. Notebooky musia byť zapojené do elektriny, keďže baterka sa im minula pri používaní počas prednášok. V T2 je našťastie obrovská predlžovačka s n zásuvkami.

Problém však nastal, keď si m vedúcich kúpilo nové Macbooky, ktoré majú najmodernejšie, revolučné a absolútne nepraktické zástrčky.⁶ Sú o niečo širšie ako tie klasické a aj keď sa dajú strčiť do slovenskej zásuvky, na každej strane z nej trochu prečnievajú. Keď ich teda zastrčíte do predlžovačky, obe susedné zásuvky na predlžovačke sú blokované a nedá sa do nich vložiť žiadna iná zástrčka. To ale výrazne obmedzuje efektivitu spomínanej predlžovačky a KSP-ákov trápi, či si vôbec vedia všetci nabíjať svoje notebooky súčasne.

Úloha

Máme predlžovačku s n zásuvkami, m počítačov so širokými macovskými zástrčkami a k počítačov s normálnymi (úzkymi) zástrčkami. Zistite, či sa všetkých $m + k$ zástrčiek dá povkladať do predlžovačky. Normálna zástrčka zaberie práve jednu zásuvku a macovská zástrčka zaberie jednu zásuvku a zablokuje obe susedné zásuvky. Medzi dvoma macovskými zástrčkami však stačí mať jednu voľnú zásuvku.

Formát vstupu

Na prvom riadku dostanete n ($1 \leq n \leq 10^9$) – počet zásuviek v predlžovačke. Na druhom riadku budú čísla m a k ($0 \leq m, k \leq 10^9$) – počet macovských a počet normálnych zástrčiek.

Formát výstupu

Vypíšte **ano** ak sa notebooky zapojíť dajú a **nie**, ak to žiadnym spôsobom nejde.

Príklad

| vstup | výstup |
|--|----------------------------------|
| <input type="text" value="7"/> <input type="text" value="3 1"/> | <input type="text" value="ano"/> |

Rozložiť sa dajú napríklad takto: momonom (macovské – m , normálne – n , prázdna zásuvka – o). Všimnite si, že širokým koncovkám stačí medzi sebou iba jedna medzera.

| vstup | výstup |
|--|----------------------------------|
| <input type="text" value="5"/> <input type="text" value="1 2"/> | <input type="text" value="ano"/> |

Napríklad rozloženie nomon.

| vstup | výstup |
|--|----------------------------------|
| <input type="text" value="2"/> <input type="text" value="1 2"/> | <input type="text" value="nie"/> |

Žiadnym spôsobom sa nepomestia, nemáme totiž ani dostatok zásuviek.

⁵Tajná miestnosť KSP a Prask vedúcich na Matfyz.

⁶Ak si ešte stále nie ste istí pri používaní slov zástrčka a zásuvka, tento článok: <http://slovensky.diskusneforum.sk/clanky/zastrcka-nie-je-zasuvka/> vám určite pomôže.

vstup

```
4
2 1
```

výstup

```
nie
```

5. Poradie trpaslíkov

15 bodov za riešenie

Táto úloha sa dá nahradiť riešením sady `arrays_cpp` na Liahni (betaliahen.ksp.sk). Ak chceš, aby ti namiesto bodov za riešenie tejto úlohy boli započítané body získané riešením spomínanej sady, na stránke odovdžaj pdf-ko s prezývkou, ktorú používaš na Liahni.

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

Všetci určite poznáte rozprávku o Snehulienke a siedmich trpaslíkoch. Snehulienka sa stratila v lese, našla domček v ktorom bývali trpaslíci, bol tam strašný neporiadok, a ona začala upratovať. . . Viete prečo nemali trpaslíci doma upratané? Nie, nebolo to preto, že by boli leniví. . . Trpaslíci toho majú len veľa na práci. Preto majú presne rozdelené, kto má čo spraviť, podľa poradia v ktorom sa v daný deň zobudili. Prvý chystá raňajky, druhý pripravuje stôl, tretí varí čaj, štvrtý upratuje. . . Aby sa v činnostiach vystriedali, a tiež, aby nemusel Spachtoš vždy vstávať ako prvý, každý deň vstávajú v inom poradí.

Aby trpaslíci vedeli zistiť, že sa už vystriedali všetky poradia, používajú nasledovný systém:

- Každý z trpaslíkov má svoju obľúbenú cifru od 0 do 9.
- Keď trpaslík ráno vstane, napíše svoju obľúbenú cifru na tabuľu, hneď za poslednú, ktorá tam bola. Takto vznikne každý deň na tabuľi jedno číslo.
- Trpaslíci si počítajú súčet čísel, ktoré sa objavujú na tabuľi – predtým, ako idú spať, pripočítajú číslo z tabule k súčtu.
- Keď je súčet dostatočne veľký, vystriedali sa všetky poradia a môžu sa začať striedať (a sčítavať) odznova.

Systém je dokonalý, no trpaslíci majú jeden problém. Nevedia, aký súčet má byť na tabuľi po tom, čo sa vystriedali všetky poradia. Niekedy si tak nevšimnú, že sa už vystriedali a nasledujúce ráno nevstane nikto. Práve v takýto deň prišla Snehulienka a tá sa ako silná žena a nádejná programátorka rozhodla vyriešiť tento problém raz a navždy.

Úloha

Pre jednu domácnosť trpaslíkov dostanete jedno číslo c_i , ktoré vzniklo na tabuľi v prvý deň. Zistite, aký bude súčet čísel z tabule, keď sa vystriedajú všetky poradia trpaslíkov. Stručne povedané, zistite súčet všetkých permutácií cifier čísla c_i .

Úlohu vyriešte pre n rôznych domácností trpaslíkov.

Formát vstupu

Na prvom riadku je číslo n ($1 \leq n \leq 10^5$). Nasleduje n riadkov, v každom z nich je jedno číslo c_i ($0 \leq c_i \leq 10^{12}$). Žiadne číslo iné ako 0 nezačína nulou.

Formát výstupu

Vypíšte n riadkov, na i -tom riadku bude súčet permutácií cifier i -teho čísla zo vstupu. Každý súčet sa zmestí do 64 bitovej premennej (typu `Int64` v Pascale, `long long` v C++).

Príklad

vstup

```
5
2
47
33
750
4247
```

výstup

```
2
121
66
2664
113322
```

Trpaslík s cifrou 2 býva sám, preto musí vždy vstať prvý. Pre trpaslíkov s ciframi 4 a 7 existujú dve poradia, ako môžu vstať (47 a 74). Trpaslíci v tretej domácnosti majú rovnaké obľúbené cifry, ale aj tak ich rozlišujeme a preto majú tiež dve možnosti.