



Leták letnej časti VII. 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 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 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 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 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 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 1. kola letnej časti

Termín odoslania riešení tejto série je pondelok **24. mája 2021.**

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

1. Permoníci 2

100 bodov za riešenie

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

Táto úloha je pokračovaním [Permoníkov](#)³ z minulej série.

Permoníkom sa zapáčil online svet. Keďže skúmanie prvej online bane medzi nimi zožalo úspech, nevedeli sa dočkať, kým intraweboví orgovia predstavia ďalšiu baňu.

Permoník Jozef práve ale zistil že táto úloha sa momentálne nedá odovzdávať. Musí preto počkať na koniec odovzdávania série — úloha bude zase (tentokrát nesúťažne) zverejnená.

2. Rámovanie

0 bodov za riešenie

Ak máte akékoľvek otázky ohľadom tejto úlohy, napíšte Adamovi na adamgramblicka@hotmail.com

V múzeu praskov dostali novú zbierku obrazov. Podľa popisu zásielky majú obsahovať mozaiky rôznych veľkostí a tvarov. Kurátor múzea a jeho tím však potrebujú rýchlo zistiť ktoré časti plátna zaplňajú takéto mozaiky, aby ich mohli vystaviť. Výstavu potrebujú otvoriť o jeden deň a nemajú dostatok času na manuálnu kontrolu, preto o pomoc požiadali vás. Keďže stroj na rezanie plátna potrebuje špeciálne značenie, je nutné každú mozaiku orámoviť.

Úloha

Vašou úlohou je napísať program, ktorý na vstupe dostane obraz veľkosti $r \times s$ znakov. Každý obraz pozostáva zo znakov anglickej abecedy a bodiek (.). Bodka znamená, že na danom mieste sa nenachádza časť mozaiky. V obraze je potrebné mozaiku nájsť a čo najtesnejšie ju orámoviť obdĺžnikom tvoreným zo znakov mriežky (#). Mozaika nemusí byť súvislá, môže obsahovať “diery”, vždy však stačí použiť iba jeden rám.

Formát vstupu

Na prvom riadku sa nachádzajú dve čísla r a s (platí $1 \leq r, s \leq 800$), výška a šírka obrazu. Na nasledujúcich r riadkoch sa nachádza reťazec dĺžky s pozostávajúci zo znakov anglickej abecedy a bodiek (.). Môžete počítať s tým, že obraz nie je prázdny, teda obsahuje aspoň jeden znak odlišný od bodky.

Formát výstupu

V prípade, že mozaiku ide zarámoviť, vypíšte obraz zo vstupu obsahujúci orámovanie zložené zo znakov # (r riadkov po s znakoch). Ak by rám presahoval rozmery obrazu, vypíšte na jediný riadok výstupu **Neda sa**.

Príklad

vstup

```
4 4
....
.xy.
.zq.
....
```

výstup

```
####
#xy#
#zq#
####
```

³<https://prask.ksp.sk/ulohy/zadania/1994/>

vstup

```
4 4
....
..a.
..a.
.b..
```

výstup

```
Neda sa
```

(Mozaiku nejde orámoviť, pretože jeden z jej znakov (b) sa dotýka steny obrazu. Rám by tak presahoval obraz.)

vstup

```
7 6
.....
.....
..aa..
.....
..bb..
.....
.....
```

výstup

```
.....
.####.
.#aa#.
.#..#.
.#bb#.
.####.
.....
```

vstup

```
18 30
.....
.....
.....qqj.....
.....qqq.....
.....j..kqk.j--...jpkqp...
.....qqkkq-qkqq-p-kkqkp...
..kqkk--qk--k-kk-p-kp.....
...j-qk--ppp-p-kkjjjj...
.....-qk-----kqkqp...
...kqqqq--prask--k-pj.....
.....pqk-----kp.....
.....kqkkqqppk-p-k.....
...jqk-j..pqk...pkk--kp...
.....pqk.....j-kkq-...
.....pj.....j-k-...
.....
.....
```

výstup

```
.....
.#####.
.#.....qqj.....#.
.#.....qqq.....#.
.#.....j..kqk.j--...jpkqp.#.
.#.....qqkkq-qkqq-p-kkqkp.#.
.#kqkk--qk--k-kk-p-kp...#.
.#.j-qk--ppp-p-kkjjjj.#.
.#....-qk-----kqkqp.#.
.#.kqqqq--prask--k-pj...#.
.#....pqk-----kp.....#.
.#....kqkkqqppk-p-k.....#.
.#.jqk-j..pqk...pkk--kp.#.
.#.jqk-j..pqk...pkk--kp.#.
.#.....pqk.....j-kkq-#.
.#.....pj.....j-k-#.
.#####.
.....
```

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>

3. Aktuálny problém

0 bodov za riešenie

Ak máte akékoľvek otázky ohľadom tejto úlohy, napíšte Miškovi na michal.farnbauer@gmail.com

Táto úloha je o dátovej štruktúre - *Halda*. Ak nevieš čo Halda je, môžeš si o nej viac prečítať v [študijnom texte o Halde](#)⁴.

V Krtkove⁵ majú teraz nový problém. Je viac ľudí, ktorí sa chcú dať zaočkovať ako vakcín. Rozhodli sa preto, že spravia čakáreň, do ktorej sa môžu ľudia zapísať a počkať, kým pre nich bude k dispozícii vakcína. Ostáva už len jeden problém a to koho zavolať na očkovanie ako prvého. Keďže všetci v Krtkove sú zaneprázdnení, potrebujú niekoho kto im naprogramuje čakáreň.

Úloha

Vašou úlohou bude simulovať čakáreň. Čakáreň pozostáva z dvoch typov akcií. Buď sa niekto s celočíselnou prioritou⁶ p prihlási do čakárne alebo máme novú vakcínu, teda človek s najväčšou prioritou bude zaočkovaný a odíde z čakárne. Po každej akcii nás zaujíma, aká je najväčšia priorita medzi ľuďmi v čakárni. Na začiatku je čakáreň prázdna.

Formát vstupu

Na prvom riadku sa nachádza číslo n - počet akcií ktoré treba spracovať. Nasleduje n riadkov, i -ty z nich popisuje i -tu akciu a má tvar $t\ x$. Ak t je 0, tak prišla nová vakcína a x ignorujeme. Ak t je 1, tak sa do čakárne prihlásil nový človek s prioritou x . V prvej akcii sme ponechali na vstupe x kvôli ľahšiemu načítavaniu vstupu (pre obe akcie treba načítať dve čísla).

Je zaručené, že ak prišla nová vakcína v čakárni je aspoň jeden človek. Vo všetkých sadách platí, že $1 \leq n \leq 2 \cdot 10^5$ a $1 \leq x \leq 10^9$.

Formát výstupu

Vypíšete n riadkov, v i -tom jedno celé číslo - najväčšia priorita spomedzi ľudí v čakárni po i akciách. Ak je po i -tej akcii čakáreň prázdna, vypíšete namiesto toho *prazdna*.

Príklad

vstup	výstup
10	1
1 1	1
1 1	1
0 0	6
1 6	1
0 0	2
1 2	1
0 0	prazdna
0 0	4
1 4	prazdna
0 0	

4. Stovky programátorov

0 bodov za riešenie

Ak máte akékoľvek otázky ohľadom tejto úlohy, napíšte Miškovi na michal.farnbauer@gmail.com

Táto úloha je o dátovej štruktúre - *Halda*. Ak nevieš čo Halda je, môžeš si o nej viac prečítať v [študijnom texte o Halde](#)⁷.

Po tom ako v Krtkove požiadali o pomoc s naprogramovaním čakárne na očkovanie, začali sa programátori a programátorky zo širokého okolia predbiehať, kto túto čakáreň naprogramuje. Keďže čakáreň môže byť iba jedna, musia sa rozhodnúť, kto z nich ju má naprogramovať. To však nie je jednoduché, keďže takýto človek musí mať dostatok skúseností, no zároveň nemôže mať príliš veľa skúseností, lebo potom by cena systému prekročila rozpočet Krtkova.

Úloha

Postupne budú prichádzať ponuky od programátorov a programátoriek s rôznym počtom skúseností. Vašou úlohou bude po každej ponuke zistiť, koľko skúseností má vhodný kandidát. Vhodný kandidát je ten, ktorý

⁴<https://prask.ksp.sk/strudlohy/halda/>

⁵niečo ako Kocúrkovo

⁶Áno, každý v Krtkove má nejakú prioritu na očkovanie

⁷<https://prask.ksp.sk/strudlohy/halda/>

má *stredne* veľa skúseností. To znamená, že polovica kandidátov je horšia a polovica lepšia. Presnejšie sa tejto strednej hodnote hovorí *medián*. Formálne, medián M prvkov je taký prvok, ktorý je po utriedení vzostupne na $\lceil M/2 \rceil$ -tom⁸ mieste.

Formát vstupu

Na prvom riadku vstupu je jedno číslo $1 \leq n \leq 10^5$ - počet programátorov. Na nasledujúcich n riadkoch sa nachádza vždy jedno číslo, $0 \leq s_i < 10^9$, počet skúseností programátorov v poradí, v ktorom sa ponúkajú naprogramovať čakáreň.

Formát výstupu

Vypíšte n riadkov, v i -tom počet skúseností vhodného kandidáta po i ponukách.

Príklad

vstup

```
10
158
182
112
92
52
201
193
167
177
167
```

výstup

```
158
158
158
112
112
112
158
158
167
167
```

vstup

```
10
83
36
16
58
65
80
52
83
73
71
```

výstup

```
83
36
36
36
58
58
58
58
65
65
```

5. Krádež rezňov

0 bodov za riešenie

Ak máte otázky ohľadom tejto úlohy píšete Kalerábovi na kalerab@trojsten.sk

Malý Adam, ako každé leto, išiel na návštevu k jeho dedkovi. Adam k nemu chodí veľmi rád, pretože jeho dedko má na stole vždy veľkú misu s rezňami (akokoľvek čudne to znie). Samozrejme, Adam ju každé leto potajomky vykráda a tento rok nie je žiadnou výnimkou. Jeden večer, keď si už bol istý, že dedko už spí, potichu vyklzol z izby a preplazil sa ku miske pokladov. Na jeho veľké prekvapenie tam stál dedko, ktorý si všimol, že mu rezne miznú. Dedko dal Adamovi ponuku. Adam sa môže zahrať s dedkom hru o rezne. Ak Adam vyhrá môže si nechať všetky rezne, ale ak prehrá, dedko rezne schová na vysokú skriňu, kam Adam nedočiahne.

Úloha

Dedko položil všetky rezne na stôl. Hráči sa následne budú striedať v ťahoch. Každý hráč si z kopy zoberie nejaký počet rezňov, pričom vždy začína Adam. Počet rezňov, ktoré si môžu hráči brať je väčší rovný n a menší ako m . Obidve tieto čísla určí dedko na začiatku hry. Hráč, ktorý si zoberie posledný rezeň, vyhráva. Dedko je však dobrá duša a Adamovi sľúbi, že obe čísla určí tak, že ak Adam bude hrať správne, môže vždy vyhrať. Ak počas hry nastane situácia, že počet rezňov na kôpke je menší ako n , hráč na ťahu vyhráva (zoberie zvyšok).

⁸Hranaté zátvorky okolo $M/2$ znamenajú hornú celú časť, teda prvé celé číslo, ktoré je väčšie alebo rovné $M/2$. Napríklad horná celá časť 3.2 je 4 a horná celá časť 8 je 8.

Formát hry

Táto úloha je interaktívna. Na prvom riadku dostane hráč čísla k , n a m , kde k je počet rezňov na kôpke a n je najmenší počet rezňov, aký si hráči môžu brať. Musia si tiež brať len počet menší ako m . Ďalej sa hráči striedajú, počínajúc Adamom. Každý hráč vypíše jedno číslo $n \leq x < m$ vyjadrujúce, koľko rezňov si daný hráč berie z kopy. Hráč, ktorý si zoberie posledný rezeň, vyhráva. Na konci hry testovač vypíše `adam` ak vyhral Adam, inak vypíše `dedko`.

Aby testovanie fungovalo ako má, **je nutné**, aby sa po vypísaní tahu výstup presunul z pamäte na štandardný výstup pomocou príkazu `cout.flush()` v C++ alebo `sys.stdout.flush()` v Pythone. Pre iné jazyky hľadajte ekvivalent k príkazu `flush`.

Limity

V prvej sade máte garantované $n=1$.

Príklad

	17 1 6
2	
	5
4	
	1
5	

Na kôpke je 17 rezňov a hráči si môžu naraz brať 1, 2, 3, 4 alebo 5 rezňov. Adam si na začiatku vezme 2 rezne. Dedko nasleduje s 5timi. Adam si následne zoberie 4, a môžeme si všimnúť, že nech si dedko teraz zoberie hocikolko rezňov, Adam vždy vyhrá.