



## Leták zimnej 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 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 rátaajú do PRASKu.

Na riešenie série je vyhradených niekoľko týždňov. Až do dňa odovzdania môžete doma riešiť zadané príklady. Môžete riešiť lubovoľné príklady z danej série, nemusíte vyriešiť všetko, nemusíte vyriešiť ani celú úlohu<sup>1</sup>. 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 niekoľko<sup>2</sup> 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.

<sup>1</sup>Aj keď budeme radi, ak sa vám to podarí.

<sup>2</sup>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](http://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](http://www.freepdfconvert.com).

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



# Úlohy 1. kola zimnej časti

Termín odoslania riešení tejto série je pondelok **25. januára 2021.**

## 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>). Vrelo ti ju odporúčame riešiť tiež, naučíš sa veľa nových vecí a môžeš sa dostať aj na krajské kolo Olympiády.

### 1. Permoníci

100 bodov za riešenie

*Ak máte akékoľvek otázky ohľadom tejto úlohy, napíšte Romanovi na [roman.sobkuliak@trojsten.sk](mailto:roman.sobkuliak@trojsten.sk)*

Permoníci sú malé tvory, ktoré najčastejšie možno stretnúť v doloch, kde baníkom pomáhajú nájsť cenné ložiská. Cez jednu zo Štiavnických baní bol nedávno natiiahnutý optický kábel, a tak miestny permoníci začali objavovať vo voľnom čase internet.

Zistili, že medzi permoníkmi sa stali populárne online bane. Tie sú skoro ako obyčajné bane, ale namiesto kopania stačí klikáť. Jednu z takýchto baní nájdete na adrese [prask.ksp.sk/specialne/prask/7/1/1](http://prask.ksp.sk/specialne/prask/7/1/1)<sup>3</sup>. Vašou úlohou je pomôcť permoníkom a dostať sa až na koniec tejto bane. Na danom linku nájdete všetky podstatné informácie k riešeniu tejto úlohy.

### 2. Rekreačný cyklista Kubko

100 bodov za riešenie

**Novinka:** táto úloha je rozdelená na niekoľko na seba nadväzujúcich **sád**, za ktoré viete dokopy získať 100 bodov. Zatiaľ máte k dispozícii iba zadania prvej sady. Na získanie zadaní ďalšej sady je potrebné **odovzdať riešenia** tej predchádzajúcej.

Hocikedy v priebehu kola môžete odovzdať vaše aktuálne riešenie. My vám ho v priebehu pár dní opravíme a pošleme späť aj s komentármi. Ak sa vám podúlohy podarilo vyriešiť správne, dostanete ďalšie zadania. Ak vaše riešenie nebolo správne, **nič sa nedeje**. V komentári vám skúsime poradiť kde nastala chyba a vy ju **môžete opraviť a poslať znova až kým nebudete úspešní**.

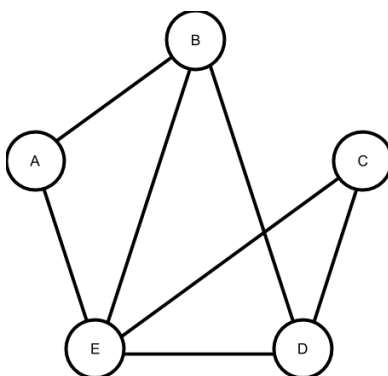
Veríme, že takto sa vám podarí vyriešiť viac podúloh a teda sa aj viac naučíte. Nabojte sa nám teda poslať aj rozpracované riešenie, alebo také, ktorým si nie ste úplne istí. Nehrozí vám žiadna penalizácia a môžete dostať dobrú radu. Odporúčame však **úlohu riešiť priebežne**.

*Ak máte akékoľvek otázky ohľadom tejto úlohy, napíšte Miškovi na [michal.farnbauer@gmail.com](mailto:michal.farnbauer@gmail.com)*

Kubko sa rád bicykluje. V jeho okolí je však len zopár cyklistických chodníkov znázornených na obrázku. Napriek tomu vždy keď má nejaký voľný čas, sadne na bicykel a trénuje. Včera napríklad začal v mieste D, pokračoval cez C do E, potom sa vrátil do D a nakoniec skončil v B. Takúto cestu by sme skrátene zapísali ako D-C-E-D-B a jej **dĺžka** je 4.

Zajtra by sa chcel ísť znovu bicyklovať, no potrebuje si nájsť novú trasu, lebo ísť dvakrát po tej istej je príliš nudné. Zaujíma ho preto, koľko rôznych ciest určitej dĺžky v okolí existuje.

<sup>3</sup><https://prask.ksp.sk/specialne/prask/7/1/1>



### Sada úloh 1 – Krátke cesty

Kubko je síce zdatný cyklista, no počítanie nie je jeho silnou stránkou. Preto spočítať vhodné cesty pre Kubka ostalo na vás. Ako zvyčajne, budete riešiť niekoľko podúloh zoradených podľa nami odhadovanej zložitosti. Okrem výsledku vždy napíšte, aspoň pár vetami, ako ste sa k danému riešeniu dostali. Odporúčame vám podúlohy riešiť postupne, keďže riešenie jednej môže pomôcť pri riešení ďalšej.

- (8 bodov) Zistite, koľko je na obrázku rôznych ciest dĺžky 3 takých, že začínajú v A a končia v D. Dve cesty sú rôzne ak neprechádzajú tými istými miestami presne v rovnakom poradí. Teda napríklad cesty B-E-D-C a B-D-E-C sú dve **rôzne** cesty dĺžky 3.
- (10 bodov) Pre každý začiatok a koniec zistite, koľko existuje ciest medzi nimi s dĺžkou práve 3. Napríklad počet ciest medzi A a D už poznáte z predošlej podúlohy, ostáva vám to určiť pre zvyšné začiatky a konce. Nezabudnite, že cesty môžu začínať a končiť v tom istom mieste.
- (8 bodov) Zistite, koľko je takých ciest, ktoré začínajú aj končia v mieste A, majú dĺžku 6 a v strede, teda vo vzdialenosti 3, bude Kubko v mieste B. Jedna takáto cesta je napríklad cesta A-E-D-B-A-E-A.
- (8 bodov) Zistite koľko je takých ciest, ktoré začínajú v A, končia v C a majú dĺžku 6.

## 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. Ambiciózne múzeá

100 bodov za riešenie

**Novinka:** táto úloha je rozdelená na niekoľko na seba nadväzujúcich **sád**, za ktoré viete dokopy získať 100 bodov. Zatiaľ máte k dispozícii iba zadania prvej sady. Na získanie zadaní ďalšej sady je potrebné **odovzdať riešenia** tej predchádzajúcej.

Hocikedy v priebehu kola môžete odovzdať vaše aktuálne riešenie. My vám ho v priebehu pár dní opravíme a pošleme späť aj s komentármi. Ak sa vám podúlohy podarilo vyriešiť správne, dostanete ďalšie zadania. Ak vaše riešenie nebolo správne, **nič sa nedeje**. V komentári vám skúsime poradiť kde nastala chyba a vy ju **môžete opraviť a poslať znova až kým nebudete úspešní**.

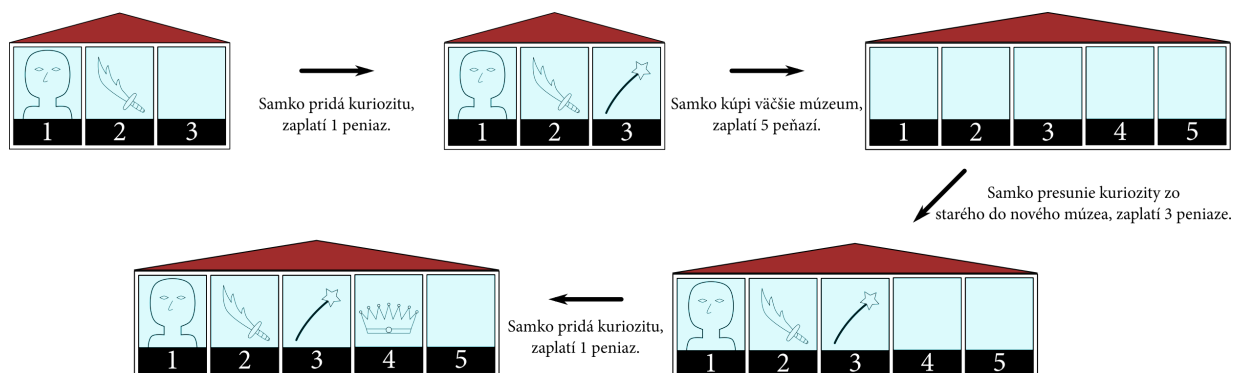
Veríme, že takto sa vám podarí vyriešiť viac podúloh a teda sa aj viac naučíte. Nabojte sa nám teda poslať aj rozpracované riešenie, alebo také, ktorým si nie ste úplne istí. Nehrozí vám žiadna penalizácia a môžete dostať dobrú radu. Odporúčame však **úlohu riešiť priebežne**.

*Ak máte akékoľvek otázky ohľadom tejto úlohy, napíšte Marianke na [mariannahronska@gmail.com](mailto:mariannahronska@gmail.com)*

Sem-tam je čas na zmenu, pomyslel si Samko, keď dostal nápad otvoriť si múzeum Kuriózných Starodávnych Predmetov. Takéto múzeum sa skladá z niekoľkých vitrín, v ktorých sú vystavené kuriozity. Vitríny sú očíslované od 1 po  $n$  (počet vitrín v múzeu) a v každej môže byť nanajvýš jedna kuriozita.

Do múzea chodia návštevníci a čím viac kuriozít v ňom je, tým sú spokojnejší. Samko preto usilovne pracuje na rozširovaní svojej zbierky. Vždy keď získa novú kuriozitu, uloží ju do voľnej vitríny. V nejakom momente sa však stane, že naplní všetky vitríny a viac sa mu ich do budovy už nezmestí. Bude preto musieť kúpiť priestory s väčším počtom vitrín a všetky staré kuriozity tam pracne preniesť.

Pri rozbiehaní takého vážneho biznisu je dôležité sledovať tok peňazí a práve s výdavkami sa Samko trápi. Beh múzea totiž obnáša nasledovné náklady. Vždy keď Samko získa novú kuriozitu a uloží ju do vitríny, zaplatí 1 peniaz (za jej kúpu). Navyše, musí platiť za nákup nových, väčších budov. Kúpenie budovy, do ktorej sa zmestí  $n$  vitrín ho stojí  $n$  peňazí. No a nakoniec, stojí ho aj prenášanie kuriozít do novej budovy, za každú prenesenú kuriozitu zaplatí 1 peniaz.



Všimnite si, že pri kúpe novej budovy do nej Samko **musí** preniesť všetky staré kuriozity a navyše, za starú budovu nedostane späť žiadne peniaze (to viete, inflácia, opotrebovanie a tak).

### Interaktívna pomôcka

Pripravili sme pre vás [webové prostredie](#)<sup>4</sup>, ktoré simuluje Samkovo múzeum – najmä jeho výdavky. Odporúčame ho využiť pri riešení podúloh z prvej sady.

### Sada úloh 1 – Stratégie kupovania budov

Samko chce zozbierať všetky kuriozity sveta, nevie však koľko ich je, môže ich byť 10 alebo 100 000 000. Zistí to až v momente, keď získa tú poslednú z nich (daný predmet je totiž kurióznym tým, že je poslednou kuriozitou sveta, ktorú Samko nevlastní). Dovtedy sa jeho múzeum postupne rozrastá a on chce, samozrejme, optimalizovať svoje výdavky. Vymyslel si preto rôzne stratégie, podľa ktorých nakupuje nové priestory. Pomôžte mu ich ohodnotiť a navrhnete vlastnú, ešte lepšiu.

- a) (12 bodov) Samkova prvá stratégia na rozbiehanie biznisu je, že vždy keď sa mu naplní priestor v múzeu, kúpi si novú budovu, do ktorej sa zmestí **o 10 vitrín s kuriozitami viac**. Na začiatku nevlastní žiadnu budovu, začne teda nákupom budovy veľkosti 10.

Predstavte si, že na svete je 100 kuriozít. Koľko peňazí Samko zaplatí, kým ich všetky získa? Popíšte, aj ako ste danú sumu vypočítali.

Pre aké počty kuriozít sveta sa táto stratégia nehodí, a pre aké je naopak dobrá?

- b) (12 bodov) Ďalšia Samkova stratégia je podobná, akurát po naplnení kúpi priestor **o 1 000 000 vitrín väčší**. Na začiatku nevlastní žiadnu budovu.

Predstavte si, že na svete je 10 000 000 kuriozít. Koľko peňazí Samko zaplatí, kým ich všetky získa? Popíšte, aj ako ste danú sumu vypočítali.

Pre aké počty kuriozít sveta sa táto stratégia nehodí, a pre aké je naopak dobrá?

- c) (25 bodov) Samko by chcel, aby ste mu navrhli vlastnú stratégiu, ktorá bude vhodná bez ohľadu na to, koľko kuriozít sa na svete nachádza.

Popíšte, ako by vaša stratégia fungovala, a koľko peňazí by Samko zaplatil, ak by na svete bolo 100 kuriozít, a ak by ich bolo 10 000 000.

Vysvetlite, prečo je vaša stratégia lepšia ako dve Samkove popísané v podúlohách a) a b).

## 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

<sup>4</sup><https://prask-interactive-playground.siegrift.vercel.app/>

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 nezáfaj!** 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. Substringy

100 bodov za riešenie

**Ak nevieš programovať, nezáfaj!** 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](http://liahen.ksp.sk)<sup>5</sup>). 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 Adamovi na [adamgramblicka@hotmail.com](mailto:adamgramblicka@hotmail.com)*

Bratia Artur a Boris medzi sebou radi súťažia a tak tomu bolo aj teraz. Keď Artur čítal knihu, všimol si zaujímavé slovo a prehlásil, že určite obsahuje viac častí<sup>6</sup> takých, že začínajú spoluhláskou. Boris mu hneď neveril a potreboval si túto domnienku overiť, lebo podľa neho bolo viac častí začínajúcich samohláskou. Keďže bratov zaujíma aj o koľko častí kto v tejto súťaži vyhral, potrebujú nájsť presný počet častí slova začínajúcich samohláskou a aj častí začínajúcich spoluhláskou. Keďže nechcú hľadať všetky takéto časti slova len za pomoci papiera, potrebujú vašu pomoc, aby im to šikovne zrátal program.

### Úloha

Vašou úlohou je napísať program, ktorý na vstupe načíta slovo dĺžky práve  $n$  a na jeho základe vypočíta výsledné body súrodencov. Slovo môže obsahovať len malé a veľké písmená 26 písmennej anglickej abecedy.

### Formát vstupu

Na prvom riadku sa nachádza jedno číslo  $n$  (platí  $1 \leq n \leq 10^6$ ), dĺžka slova. Na nasledujúcom riadku je reťazec dĺžky  $n$  z písmen anglickej abecedy. Môže obsahovať malé aj veľké písmená.

### Formát výstupu

Na jediný riadok výstupu vypíšete počet substringov (súvislých podúsekov slova) začínajúcich samohláskou a počet substringov začínajúcich spoluhláskou oddelené medzerou.

### Hodnotenie

Je 5 sád vstupov. Platia v nich nasledujúce obmedzenia:

Číslo sady	1	2	3	4	5
$1 \leq n \leq$	500	1000	100 000	750 000	$10^6$

Výsledky v posledných sadách môžu byť veľmi veľké čísla. Chcete preto použiť 64 bitové premenné (napr. `long long` v C++, `Int64` v Pascale), lebo do obyčajnej 32 bitovej premennej sa nemusia zmestiť. Ak programujete

<sup>5</sup><https://liahen.ksp.sk/>

<sup>6</sup>Časť slova, nazývaná aj substring, predstavuje akýkoľvek interval dĺžky od 1 po  $n$  nachádzajúci sa v slove (stringu).

v pythone, nemusíte sa tým zaoberať, on to vyrieši za vás.

### Príklad

vstup	výstup
5 prask	3 12
*Substringy začínajúce samohláskou sú len 3 a to ask, as, a. Spoluhláskou začína 12 substringov a to p, r, s, k, prask, pras, pra, rask, ras, ra, pr a sk.	
vstup	výstup
15 aekeocjekolskrj	66 54

## 5. Korytnačie preteky

100 bodov za riešenie

*Ak máte akékoľvek otázky ohľadom tejto úlohy, napíšte Žabovi na [zaba@ksp.sk](mailto:zaba@ksp.sk)*

Medzi matfyzákmi sú Korytnačie preteky veľmi obľúbenou spoločenskou hrou a to napriek tomu, že táto hra je primárne určená pre päťročné deti. Pri prvom opise je totiž naozaj veľmi jednoduchá. Hrací plán pozostáva z niekoľkých za sebou idúcich políčok a cieľom hráčov je dostať korytnačku ich farby čo najrýchlejšie do cieľa. Korytnačky sa však môžu hýbať iba pomocou kariet, ktoré hráči na striedačku vykladajú. Každá karta určuje farbu korytnačky a počet políčok, o ktoré sa má korytnačka posunúť (tento počet môže byť aj záporný, vtedy ide korytnačka dozadu).

Toto znie pomerne priamočiaro. Akurát, hráči poznajú iba farbu svojej korytnačky, nevedia za ktorú korytnačku hrajú súper a na ruke môžu mať karty ľubovoľnej farby. Navyše, korytnačky majú vskutku špecifický spôsob pohybu. Ak je totiž korytnačka presunutá na políčko, na ktorom sú iné korytnačky, nepostaví sa vedľa nich, ale na ne. Všetky korytnačky stojace na tom istom políčku teda vždy tvoria jeden *stĺpik*. Novú korytnačku, prichádzajúcu na políčko, vždy položíme **na vrch stĺpika**. A keď sa niektorá z korytnačiek tvoriacich stĺpik pohne, **zoberie so sebou (presnejšie, na sebe) všetky na nej stojace korytnačky**.

Pri hre naozaj často vznikajú takéto stĺpiky korytnačiek, ktoré sa spoločne presúvajú. Hráči sa snažia takticky držať svoje korytnačky na vrchu týchto kôpok (aby im ich posúvali aj ostatní hráči), a potom vo vhodnej chvíli rýchlo utiecť do cieľa.

Jano a Žaba sú vášniví hráči. Vytvorili si dokonca viacero hracích plánov, v ktorých je čoraz viac políčok aj korytnačiek. Nedávno ich pri hraní prišiel navštíviť Sysel, drgol však do stola a rozsypal im všetky korytnačky. Jano a Žaba by chceli túto partiu dohrať. Našťastie si pamätajú všetky karty, ktoré postupne zahráli. Teraz by potrebovali rýchlo zistiť, aké bolo rozloženie korytnačiek v momente, keď prišiel Sysel. Pomôžete im?

### Úloha

Dostanete popis hry, ktorá sa odohrávala na  $p$  políčkach s  $n$  korytnačkami. Políčka sú očíslované od 1 po  $p$ , pričom políčko 1 je začiatok a políčko  $p$  je cieľ. Korytnačky sú očíslované od 1 po  $n$ . Na začiatku všetky korytnačky tvoria jeden stĺpik na políčku 1. Poradie korytnačiek v štartovacom stĺpiku dostanete zadané.

Následne sa odohralo  $q$  kariet. Každá karta určuje korytnačku, ktorá sa pohla, a počet políčok, o ktoré sa posunula dopredu. Toto číslo môže byť kladné aj záporné.

Vašou úlohou je zistiť, ako vyzerá hrací plán po odohratí všetkých zadaných kariet.

### Formát vstupu

V prvom riadku vstupu sú tri čísla  $n$ ,  $p$  a  $q$  – počet korytnačiek, počet políčok a počet zahratých kariet.

V druhom riadku je  $n$  medzerou oddelených čísel udávajúcich začiatkové postavenie korytnačiek na políčku 1. Prvé číslo určuje číslo korytnačky na samom spodku stĺpika, druhé je číslo korytnačky priamo nad ňou a tak ďalej až po posledné číslo označujúce najvyššie položenú korytnačku. (Všetky tieto čísla sú od 1 po  $n$  a sú navzájom rôzne.)

Nasleduje  $q$  riadkov popisujúce karty v poradí, v akom boli zahraté. Každý z týchto riadkov sa skladá z troch čísel  $k_i$ ,  $p_i$  a  $x_i$ . Tento riadok udáva, že korytnačka  $k_i$  pri zahratí tejto karty stála na políčku  $p_i$  a posunula sa o  $x_i$  políčok dopredu. (Záporné  $x_i$  teda v skutočnosti označuje pohyb dozadu.)

Zadaný vstup je korektný. Môžete teda predpokladať, že korytnačky skončia po každom pohybe na niektorom z políčok 1 až  $p$  a korytnačka  $k_i$  pri zahratí  $i$ -tej karty naozaj stojí na políčku  $p_i$ .

## Formát výstupu

Na výstup vypíšete  $p$  riadkov. V  $i$ -tom z týchto riadkov vypíšete popis korytnáčiek na  $i$ -tom políčku. Prvé číslo riadku (označme ho  $x_i$ ) udáva počet korytnáčiek na tomto políčku. Zvyšok riadku má tvoriť ďalších  $x_i$  čísel: čísla korytnáčiek na tomto políčku v poradí od spodku stĺpika po jeho vrch.

## Hodnotenie

Váš program bude otestovaný na piatich vstupoch. Hodnoty  $n$ ,  $p$  a  $q$  v týchto vstupoch sú uvedené v tabuľke. Navyše, vo **treťom vstupe platí**, že každý presun korytnačky smeruje na prázdne políčko. V tomto vstupe sa teda nestane, že by sa korytnačka postavila na chrbát inej (stále však môžu cestovať v kôpke určenej z počiatočného rozostavenia).

Číslo vstupu	1	2	3	4	5
$n$	5	100	10 000	100 000	300 000
$p$	20	1000	10 000	1000	10 000
$q$	25	1000	5 000 000	2 000 000	5 000 000

## Príklad

vstup

```
5 7 4
2 4 1 3 5
1 1 3
3 4 -3
4 1 5
1 4 2
```

výstup

```
1 2
0
0
0
0
0
4 4 3 5 1
0
```

Máme päť korytnáčiek, sedem políčok a štyri zahrané karty. Na začiatku sú na políčku 1 korytnačky v poradí 2, 4, 1, 3, 5 zdola hore.

Potom sa postupne udeje nasledovné:

- Korytnačka 1 sa z políčka 1 pohne o 3 dopredu, na políčko 4. Na chrbte so sebou odnesie aj korytnačky 3 a 5.
- Korytnačka 3 sa z políčka 4 pohne o  $-3$  dopredu, teda naspäť na políčko 1. Na chrbte so sebou odnesie aj korytnačku 5. Na políčku 1 je teraz stĺpik 2, 4, 3, 5, zatiaľ čo na políčku 4 je samotná korytnačka 1.
- Korytnačka 4 sa z políčka 1 pohne o 5 dopredu. Na políčku 1 zostane len korytnačka 2, ostatné korytnačky odnesie korytnačka 4 so sebou na políčko 6.
- Korytnačka 1 sa zo svojho aktuálneho políčka 4 pohne o 2 dopredu a tam naskočí na vrch stĺpika.