



## Leták zimnej časti VI. 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 navyše vám 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<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 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](http://prask.ksp.sk). V časti **Zadania** a **vzoráky** nájdete okrem zadání 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 18. november 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>). Vrelo ti ju odporúčame riešiť tiež, naučíš sa veľa nových vecí a môžeš sa dostať aj na krajské kolo Olympiády.

#### 1. Poskakujúci kôň

15 bodov za riešenie

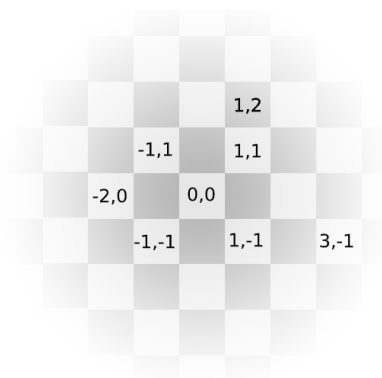
*Ak máte akékoľvek otázky ohľadom tejto úlohy, napíšte Romanovi na [roman.sobkuliak@trojsten.sk](mailto:roman.sobkuliak@trojsten.sk)*

Emovi sa končila stáž v Zurichu a potreboval pred odchodom upratať byt. Po chvíli upratovania našiel v kope neporiadku niečo zaujímavé – nekonečnú<sup>3</sup> šachovnicu a figúrku jazdca (koňa). Emo si síce uvedomoval, že by mal upratať, no napriek tomu si sadol k šachovnici a začal sa s jazdcom hrať. Všetkým len chvíľku. . .

Ale chvíľa sa natiahla a Emo si zrazu uvedomil, že ak v okamihu neodíde, zmešká lietadlo. Čo však s nevyriešenými problémami, ktoré mu pri hraní napadli?

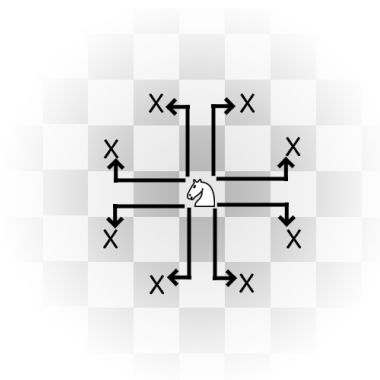
#### Úloha

Každému políčku Emovej nekonečnej šachovnice vieme priradiť dvojicu čísel, ktoré určujú jeho súradnice. Prvé číslo zodpovedá horizontálnej (stĺpec) a druhé vertikálnej (riadok) pozícii vzhľadom na začiatok (0,0). Zopár príkladov súradníc nájdete na obrázku nižšie.



Na začiatku stojí na políčku (0,0) jazdec, ktorý sa môže pohybovať podľa štandardných šachových pravidiel. Pohyb jazdca pripomína písmeno L – najskôr sa posunie o dve políčka v zvislom alebo vodorovnom smere a potom ešte o jedno políčko kolmo na pôvodný smer. Na nekonečnej šachovnici má teda pri každom skoku na výber osem políčok, na ktoré sa môže presunúť (viď obrázok).

<sup>3</sup>Políčko (0,0) je v strede Emovej obývačky a šachovnica je nekonečná do všetkých smerov. Áno, Emova obývačka je naozaj veľká :)



- (2 body) Políčko  $(0, 0)$  má bielu farbu a zvyšné políčka sú ofarbené šachovnicovo. Akú farbu bude mať políčko, na ktoré sa jazdec dostane po  $X$  skokoch? Prečo?
- (2 body) Jazdec zo svojho začiatočného políčka urobil presne dva skoky. Na koľkých rôznych políčkach môže teraz stáť? Aké súradnice majú tieto políčka?
- (2 body) Koľko najmenej skokov potrebuje jazdec na to, aby sa dostal zo svojho začiatočného políčka na ľubovoľné zo 4 políčk, ktoré sú začiatočným susedia stranou (políčka so súradnicami  $(0, 1)$ ,  $(1, 0)$ ,  $(0, -1)$  a  $(-1, 0)$ )? Prečo to nejde na menej skokov?
- (2 body) Jazdec zo svojho začiatočného políčka urobil presne tri skoky. Na koľkých rôznych políčkach môže teraz stáť? Popíšte postup, ktorým ste sa dopracovali k svojmu riešeniu. Na rozdiel od podúlohy b) nemusíte do popisu písať súradnice týchto políčk.
- (3 body) Koľko najmenej skokov potrebuje jazdec na cestu z políčka  $(0, 0)$  na políčko  $(9, 9)$ ? Prečo to nejde na menej? Popíšte postup, ktorým ste sa dopracovali k svojmu riešeniu.
- (4 body) Popíšte všeobecný postup, ako by ste zisťovali, na koľko najmenej skokov sa vie jazdec dostať z políčka  $(0, 0)$  na políčko  $(X, Y)$ . Skúste popísať, prečo je váš postup správny.

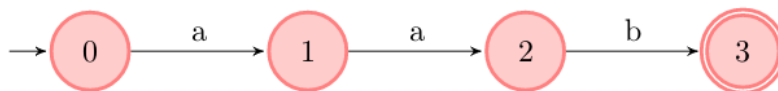
## 2. Receptomaty

15 bodov za riešenie

Ak máte akékoľvek otázky ohľadom tejto úlohy, napíšte Adamovi Královi na [adam.kral@trojsten.sk](mailto:adam.kral@trojsten.sk)

Čarodejnice (a čarodejnici) z Praskbury, ako všetky ostatné čarodejnice (a čarodejnici) po celom svete, varia kúzla. Veľmi im to ale nejde, pretože recepty sú zložité a nedávajú veľmi zmysel. Jednotlivé kroky sú však jednoduché, buď je to nejaký postup (ako zohriať, schladiť) alebo pridať ingredienciu. Je však veľmi dôležité, v akom poradí sa ingrediencie do kotla hodia a poradie krokov. Napríklad recepty “aab” a “aba” síce vyzerajú podobne, ale jeden vytvorí elixír neviditeľnosti a druhý výbuch o sile 3 megatony.

Čarodejnice sa už dostali do bodu, kedy sa boja úplne variť. Nevedia o svojich receptoch povedať, čo z nich vyjde. Pri revízii skladu posledný týždeň našli dole v sklade obrovskú knihu “Receptomaty”. Dokáže o recepte povedať, či navarí správne kúzlo alebo nie. Receptomat môže vyzeráť napríklad takto:



Tento konkrétny receptomat overuje správnosť elixíru na neviditeľnosť, ktorý bol spomenutý vyššie.

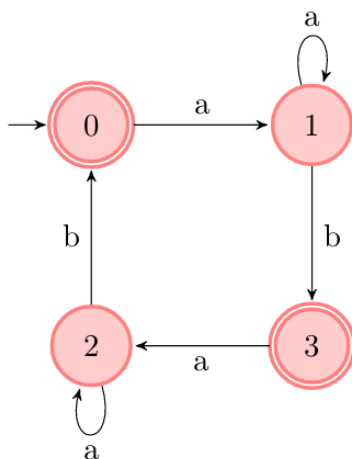
Na konci knihy zo skladu našli čarodejnici kapitolu “Teória receptomatov”. Tú vám oskenovali a zdieľali na [tejto adrese](#). Aby vám pomohli ešte viac, dokázali podľa knihy zostrojiť aj simulátor receptomatu. Ten je na [tejto adrese](#).

### Úloha

Niektoré strany z knihy receptomatov chýbajú. Pomôžte nájsť:

- (1 bod) Receptomat pre recept “abca”.
- (2 body) Receptomat pre recepty, ktoré obsahujú iba znaky “a”.

- c) (3 body) Receptomat, ktorý akceptuje recepty v tvare čísla deliteľného číslom 3.
- d) (3 body) Receptomat, ktorý akceptuje recepty s podreťazcom “abac”.
- e) (3 body) Receptomat, ktorý akceptuje len abecedne usporiadané recepty (vzostupne).
- f) (3 body) Aké recepty akceptuje tento receptomat? Dokážeš znížiť počet jeho stavov (množina správnych receptov musí ostať nezmenená)?



Podúlohy a, b, d, e, f sú z kapitoly “Písmenné receptomaty”. V týchto receptoch sa používajú len znaky ‘a’, ‘b’, ‘c’ a ‘d’. Podúloha c je z kapitoly “Číselné receptomaty”. V týchto sa používajú znaky ‘0’ až ‘9’.

## 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 netriviálna úloha.

### 3. A tak sa Denis stratil

15 bodov za riešenie

*Ak máte akékoľvek otázky ohľadom tejto úlohy, napíšte Prefixovi na [michal.sladecek@trojsten.sk](mailto:michal.sladecek@trojsten.sk)*

Denis mal napísať zadanie k príkladu 3. Ako ho išiel písať, zrazu si všimol na svojom počítači malú ikonku - bola to čierna ikonka ktorú nepoznal. “Hmm to vyzerá zaujímavo, čo sa asi stane ak na túto ikonku kliknem?” povedal si Denis. A tak na ňu klikol.

Vžuuu Denisa počítač vtiahol do seba a Denis sa ocitol stratený niekde hlboko v príkazovom riadku Linuxu. Našťastie si sa ty, účastník/čka Prasku nachádzal nablízko a môžeš ho odtiaľ vyslobodiť.

Denis je v počítači roztrúsený v podobe kódov ktoré sa nachádzajú v jednotlivých úlohách. Každý kód vyzerá ako nejaké znaky, napr. `d003f58b96e6b0a543bf75`.

“Ale ja sa neviem orientovať v príkazovom riadku, ako ho tam mám nájsť?” hovoríš si. Žiadny problém! Na tomto počítači sa nachádza množstvo hintov, ktoré vám ukážu ako nájsť a zrekonštruovať Denisa. Veľa vecí si ale budeš asi musieť vygoogliť, no čo by si pre Denisa nespravil/a?

#### Úloha

V tejto úlohe budete musieť vykonávať rôzne podúlohy v príkazovom riadku Linuxu. Tento systém nebude bežať u vás - dostanete sa doň pomocou návodu, ktorý sme pre vás spísali na [tejto adrese](#). Takže sa nebojte, nemusíte si nainštalovať Linux.

Aby sme vám trochu pomohli, začneme drobným introm - v Linuxovom terminály sa vždy nachádzate v nejakom priečinku. Na začiatku budete v domovskom priečinku, v ktorom sú aj príklady. V priečinkoch sa môžu nachádzať iné priečinky alebo súbory. Obsah priečinku vypíšete príkazom `ls`. Ak chcete vidieť obsah nejakého súboru, viete to spraviť príkazom `cat subor`. Priečinkov viete meniť pomocou príkazu `cd - cd priečinok` vás premiestni do priečinku `priečinok`, pokiaľ taký existuje. Každý priečink obsahuje aj špeciálny priečink `..` - tento priečink vždy ukazuje na priečink nad ním. Príklady:

Nachádzate sa v priečinku `/home/user/priecinok` a chcete prečítať súbor `y` nachádzajúci sa v `/home/user/priecinok/priecinok2/x`. Môžete použiť príkaz `cd priecinok2`, čím sa dostanete do druhého priečinku a následne `cat x`, čím prečítate `x`.

Nachádzate sa v priečinku `/home/user/priecinok` a chcete prečítať súbor `y` nachádzajúci sa v `/home/user/y`. Môžete použiť príkaz `cd ..`, čo vás presunie na `/home/user/` (o priečinok vyššie), a následne `cat y`.

Tieto znalosti vám vystačia na prvé dve podúlohy. V ďalších podúlohach je google váš veľký kamarát. Aby sme vám ale ukázali približne, od čoho sa odraziť, tu sú príklady vecí, ktoré môžete googliť:

- príkazový riadok linuxu tutorial
- ako vyhľadať súbor v linuxe
- práva v linuxe
- linux príkazy procesy

Pokiaľ viete tak môžete skúsiť hľadať po anglicky, väčšinou dostanete viac výsledkov ktoré budú často aj presnejšie.

To, že máte vyriešený príklad viete, lebo sa dostanete ku kódu - písmena a čísla, ktoré vyzerajú asi takto: `d003f58b96e6b0a543bf75`. Tieto kódy budete na konci odovzdávať v popise - požadujeme od vás aj krátky popis toho, ako ste príklad riešili. Nepripravili sme pre vás žiadne chytky, takže raz keď už od príkladu máte niečo čo vyzerá ako kód, skoro určite je správny a dostanete bod.

## Programátorské úlohy

Tieto úlohy sú zamerané na praktickú tvorbu programov v niektorom vyššom programovacom jazyku ako je napríklad Python, C++ alebo Pascal. Na stránke odovzdávaš **iba zdrojový kód** svojho programu riešiaceho zadanú úlohu, ktorý bude okamžite automaticky otestovaný a do pár sekúnd sa dozvieš, koľko bodov tvoj program získal. Tieto body ti už nikto nemôže zobrať, ale ak si nezískal plný počet bodov, môžeš opakovane odovzdávať opravený program, až kým nebudeš spokojný s výsledkom.

Ak už vieš programovať, ale ešte si nepracoval s našim testovacím systémom, odporúčam ti zájsť na Programátorskú Liaheň (<http://liahen.ksp.sk>), kde si o tom môžeš prečítať úvodný text a vyriešiť si niekoľko jednoduchých úloh.

Ak však **nevieš programovať, tak nezúfaj!** Pripravili sme pre teba **Programátorskú Liaheň**, ktorá ťa **naučí základy programovania** v jazyku C++. Navyše, za riešenie týchto tutoriálových úloh na Liahni môžeš získať body priamo do PRASKu a tým si vynahradiť neriešenie niektorej z programátorských úloh.

Presnejšie to funguje takto. Na Liahni sa nachádzajú dve sady úloh, prvá zameraná na premenné a druhá na podmienky v jazyku C++. V týchto sadách sa nachádzajú bodované aj nebodované úlohy, ktoré môžeš postupne riešiť a ktoré ti postupne vysvetlia danú problematiku. Dokopy sa v jednej sade dá získať až 15 bodov.

Týmto 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. Sladké náramky

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](http://liahen.ksp.sk)). Keď budeš riešiť sadu **variables\_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 Jitke na [jitka.mur@gmail.com](mailto:jitka.mur@gmail.com)*

V jedno pekné popoludnie sa Timka vracala celá šťastná domov zo sústredu Prasku. Nielenže bolo sústredenie skvelé, tešila sa aj z dvoch náramkov plných cukríkov všetkých chutí, ktoré vyhrala v tombole.

Aby spravila radosť Romanovi, ktorý na sústredenie nemohol prísť, rozhodla sa, že mu vyrobí nový špeciálny náramok. Tento náramok by mal byť čo najdlhší a zároveň by malo platiť, že cukríky, z ktorých bude zložený, sa budú dať preusporiadať tak, aby tvorili podpostupnosť (pozri časť Úloha) jedného z pôvodných náramkov. A to nie je všetko! Ak cukríky znovu preusporiadame, vytvoríme podpostupnosť druhého z pôvodných náramkov.

Timka, ešte nabudená zo sústredka, hneď vedela, z ktorých cukríkov má náramok pre Romana poskladať. Len čo prišla domov pustila sa do práce a aj zabudla, akú mala sama na cukríky chuť.

## Úloha

Vašou úlohou je napísať program, ktorý na vstupe dostane popis náramkov, ktoré Timka vyhrala a vypíše náramok, ktorý Timka vyrobila pre Romana.

Pre jednoduchosť si budeme cukríky reprezentovať malými písmenami anglickej abecedy – rovnaké písmená predstavujú rovnaké cukríky. Náramok si teda vieme predstaviť ako reťazec písmen, napr. **abac** je náramok so štyrmi cukríkmi, kde sú prvý a tretí cukrik rovnaké.

Romanov náramok musí byť najdlhší možný. Navyše cukríky v ňom musíme vedieť preusporiadať tak, aby vytvorili podpostupnosť prvého náramku, a potom preusporiadať tak, aby vytvorili podpostupnosť druhého náramku.

Podpostupnosť reťazca dostaneme tak, že z neho vymažeme ľubovoľné znaky a zostávajúce písmená bez zmeny poradia spojíme dokopy. Napríklad **nrm**, **ramo**, **naok** sú podpostupnosti slova **naramok**, ale **aaa** ani **kom** nie sú.

Samozrejme, môže existovať viac možností ako vytvoriť náramok pre Romana, nás preto bude zaujímať lexikograficky najmenší. Teda ak by sme zobrali všetky možné prípustné najdlhšie náramky a zoradili by sme ich podľa abecedy, tento by bol prvý v poradí.

## Formát vstupu

Na prvom riadku vstupu je reťazec reprezentujúci prvý Timkin náramok. Na druhom riadku je reťazec reprezentujúci druhý náramok.

## Formát výstupu

Na výstup vypíšete do jediného riadku lexikograficky najmenší reťazec, ktorý tvorí najdlhší možný náramok pre Romana. V prípade, že takýto náramok neexistuje, vypíšete **neda sa**.

## Hodnotenie

Číslo sady	1	2	3	4	5
maximálna dĺžka náramkov	100	1 000	2 000	100 000	500 000

## Príklady

vstup

```
prask
trojsten
```

výstup

```
rs
```

Náramok **rs** je podpostupnosťou náramku **prask** (cukríky na indexoch 1 a 3) a aj podpostupnosťou náramku **trojsten** (cukríky na indexoch 1 a 4).

vstup

```
krk
nos
```

výstup

```
neda sa
```

Timka nedokáže vyrobiť špeciálny náramok pre Romana.

vstup

```
hlava
aha
```

výstup

```
aah
```

Aj náramok **haa** by spĺňal požiadavky, čo sa týka dĺžky a podpostupností (v prvom náramku cukríky na indexoch 0, 2 a 4. V druhom náramku sa preusporiadanie **aha** zhoduje s cukríkmi na indexoch 0,1 a 2), no nebol by lexikograficky najmenší. Preto vypisujeme náramok **aah**.

## 5. Kamenná cesta

15 bodov za riešenie

Táto úloha je prebraná z domáceho kola **Olympiády v Informatike** kategórie B. Riešením tejto úlohy sa teda viete zapojiť aj do Olympiády (odovzdať to však musíte samostatne druhýkrát).



Ak sa Vám úlohy v PRASKu páčia, **vyskúšajte si aj riešenie Olympiády, termín jej domáceho kola je do 30. novembra 2019**. Riešením domáceho kola máte možnosť postúpiť do kola krajského a skúsiť si riešenie aj v súťažnejších podmienkach a kratšom čase. Na postup na krajské kolo obvykle stačí 10 bodov, takže ak vyriešite túto úlohu a ešte niečo, postup budete mať zaručený.

**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](http://liahen.ksp.sk)). Keď budeš riešiť sadu **conditions\_cpp**, bodmi, ktoré získaš si môžeš nahradiť riešenie tejto úlohy. Stačí ak na spodku tejto stránky odovzdáš pdf-ko s prezývkou, ktorú používaš na Liahni.

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

Jitka sa rozhodla ísť pozrieť Ema do Švajčiarska. Ten sa na jej príchod veľmi tešil a pripravil pre ňu prekvapenie – romantický pobyt v horskej chate. Keď sa však už tretiu hodinu drali hustým lesom na ceste k chate, uvedomil si Emo, že ceny vo Švajčiarsku sú odlišné od tých na Slovensku. A tak za cenu priemerného slovenského ubytovania objednal Emo pomerne podpriemernú chatku uprostred ničoho.

Našťastie, to ich už našiel chatár, ktorý poznal zopár šikovných skratiek a o chvíľu už mali chatku na dohľad. Poslednú prekážku tvorila riečka, z ktorej vytrčalo veľké množstvo kameňov, po ktorých sa dalo preskákať na druhú stranu. Blíži sa však obdobie dažďov a Emo je nervózny. Ako dlho potrvá, kým sa hladina zdvihne natolko, že sa už nebude dať dostať na druhú stranu?

## Úloha

V rieke sa nachádza cesta tvorená  $n$  kameňmi. Pre každý kameň máme zadané jedno číslo  $k_i$  – poradové číslo posledného dňa kedy je ešte tento kameň nezatopený a dá sa po ňom prejsť.

Pri prechode riekou môžeme samozrejme stúpať iba na nezaplavené kamene. Jedným krokom sa však vieme presunúť nielen na nasledujúci kameň, ale aj na kameň hneď za ním. Vďaka tomu je možné niektoré zaplavené kamene prekonať.

Vašou úlohou je zistiť, ktorý najneskorší deň sa ešte stále dá dostať z jednej strany rieky na druhú.

## Formát vstupu

Na prvom riadku vstupu je číslo  $n$  ( $2 \leq n \leq 1\,000\,000$ ) – počet kameňov v rieke, ktoré tvoria cestu medzi dvoma brehmi.

Na druhom riadku vstupu je  $n$  medzerami oddelených čísel  $k_1, k_2 \dots k_n$ . Hodnota  $k_i$  ( $1 \leq k_i \leq 1\,000\,000\,000$ ) označuje posledný deň v ktorom sa bude dať prejsť po  $i$ -tom kameni predtým ako bude zatopený.

## Formát výstupu

Na výstup vypíšete jediné číslo  $x$  – posledný deň kedy sa dá prejsť po nezatopených kameňoch z jednej strany rieky na druhú.

## Hodnotenie

Sú 3 sady vstupov po 5 bodov. Pre prvú sadu navyše platí, že  $n \leq 1\,000$ .

## Príklad

vstup	výstup
<pre>7 7 3 15 7 2 8 6</pre>	<pre>7</pre>

*V 7. deň je zatopený druhý, štvrtý a siedmy kameň, teda tie, ktoré majú ako deň zatopenia číslo menšie ako 7. Pomocou nezatopených kameňov sa dá však stále popreskakovať na druhú stranu rieky. Stačí skočiť z brehu na prvý kameň, odtiaľ preskočiť na tretí kameň, posunúť sa na štvrtý, z neho na šiesty a odtiaľ už rovno na druhý breh. Všimnite si, že pri každom skoku, vrátane tých z brehu a na breh, sme preskočili maximálne jeden kameň. V ôsmy deň už nemôžeme použiť prvý ani druhý kameň a teda z brehu nevieme skočiť na nezatopený kameň.*