

Úlohy 2. kola zimnej časti

Termín odoslania riešení tejto série je pondelok **22. január 2018.**

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/2016/sl-2016-1-zad-B.pdf>). Vrelo ti ju odporúčame riešiť tiež, naučíš sa veľa nových vecí a môžeš sa dostať aj na krajské kolo Olympiády.

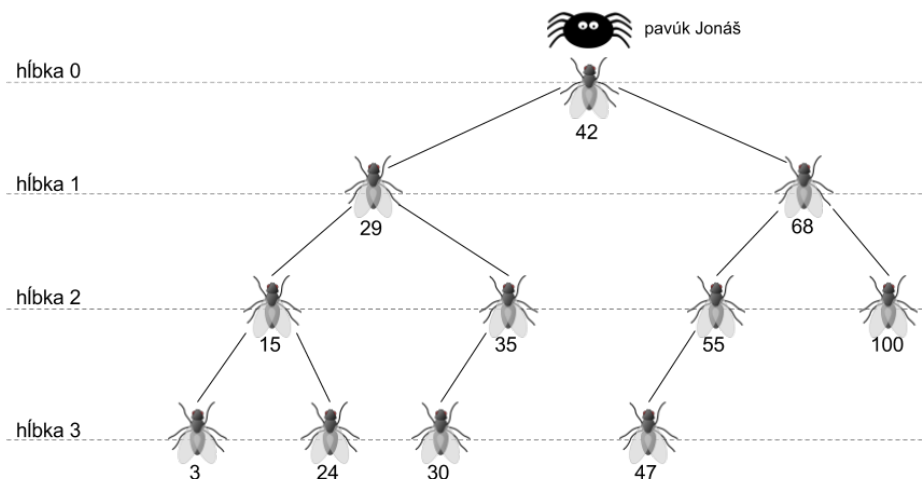
1. Pokračovanie príbehu s pavúkom Jonášom

15 bodov za riešenie

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

Táto úloha voľne nadväzuje na úlohu [Pavúčí sklad](#) z prechádzajúceho kola. Odporúčame si preto aspoň letmo prečítať jej zadanie aj vzorové riešenie. Na vyriešenie tejto úlohy to síce nie je nutné, myslíme si však, že vám to vie pomôcť.

Spomínate si ešte na pavúka Jonáša a jeho siete? Ukladal si do nich mušky, ktoré sa mu podarilo chytiť. Dodržiaval pri tom veľmi prísne pravidlá. Z každej mušky viedli dodola najviac dve vlákna – ľavé a pravé. Navyiac, pre každú mušku v sieti platilo, že všetky mušky, ktoré boli pod ňou **naľavo boli ľahšie** ako ona, a všetky mušky, ktoré boli pod ňou **napravo boli ťažšie** ako ona. Jonášova pavučina teda mohla vyzeráť napríklad takto.



Obr. 1

Úloha

Keďže Jonáš sa chcel naučiť tkať ešte krajšie siete, vybral sa na skusi do sveta. Ako prvých navštívil svojich bratrancov v Španielsku, kde sa dozvedel, že v móde sú hlboké siete. A keďže nevedel, čo je hĺbka siete, jeho bratrance mu to vysvetlili.

Každá muška v sieti je v nejakej hĺbke. Najvyššia muška v sieti je v hĺbke 0, tie zavesené priamo pod ňou sú v hĺbke 1, tie pod nimi v hĺbke 2 atď.. Presnejšie, muška zavesená priamo pod muškou v hĺbke x je v hĺbke $x + 1$. Hĺbka siete je potom najväčšia hĺbka, v ktorej je ešte nejaká muška. Na obrázku vyššie teda vidíte sieť hĺbky 3.

- a) (2 body) Jonáš by sa chcel pred Španielmi predviesť a čo najrýchlejšie spraviť sieť hĺbky 7. Na začiatku má prázdnu sieť, do ktorej bude postupne pridávať mušky, pričom bude celý čas dodržiavať všetky podmienky uvedené vyššie. Nájdite čo **najkratšiu** postupnosť mušiek, ktoré keď bude v tomto poradí pridávať do siete, dostane sieť hĺbky 7.

Pavučinu z prvého obrázku by Jonáš vytvoril, keby mušky pridával v nasledovnom poradí: 42, 29, 15, 68, 3, 35, 55, 47, 24, 30, 100¹. **Pri riešení môžete použiť** napríklad nasledovný nástroj btv.melezinek.cz/binary-search-tree.html. Keď vedľa tlačidla **Insert** (po slovensky pridaj) napíšete nejaké číslo a tlačidlo stlačíte, do stromu sa zadané číslo pridá dodržiujúc všetky Jonášove podmienky.

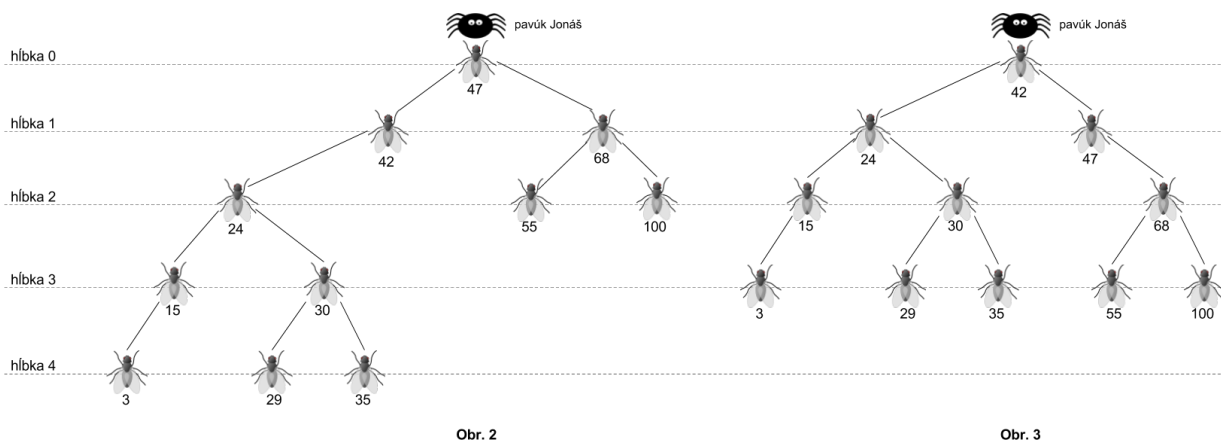
Po Španielsku sa Jonáš vybral do Nemecka. Nemecké pavúky sú oveľa praktickejšie a snažia sa tkať svoje siete čo najplytkejšie. Komu by sa totiž chcelo behať celý deň po sieti, keď môže mať všetky mušky poruke, iba pár vláknien od vrchu.

- b) (3 body) Jonáš by do svojej pavučiny rád vložil čo najviac mušiek tak, aby vzniknutá sieť mala hĺbku najviac 3. Opäť začína s prázdnu sieťou. Nájdite čo **najdlhšiu** postupnosť mušiek, ktoré keď bude v tomto poradí pridávať, nevytvorí sieť s hĺbkou 4.

Rovnako ako v predchádzajúcej úlohe pri riešení použite btv.melezinek.cz/binary-search-tree.html.

Jonáš sa z potuliek svetom vrátil múdrejší. Uvedomil si, že záleží na tom, ako jeho sieť vyzerá. A keďže je od prírody lenivý a za jedlom sa mu chodiť nechce, zapáčil sa mu viac nemecký spôsob stavania pavučín – čím menšia hĺbka, tým lepšie.

Keď sa však pozrel na svoju pavučinu (Obr. 2) zdala sa mu prihlboká. Hlavne vľavo bolo oveľa viac mušiek ako vpravo. Preto mu napadlo presunúť mušku 42 na vrch pavučiny. Výsledok jeho snaženia vidíte na Obr. 3. Teraz ho však zaujíma, či by takýto posun vedel robiť aj všeobecne.



- c) (3 body) Navrhňte postup, ktorým bude Jonáš vedieť presunúť mušku priamo naľavo alebo napravo od najvyššej mušky na vrch pavučiny. Nezabudnite zdôvodniť, prečo je vaše riešenie správne.

Presunutím ľavej mušky dostal Jonáš z obrázku 2 obrázok 3, presunutím pravej mušky mohol zase z obrázku 3 vzniknúť obrázok 2. Dajte si pozor, že obrázok neukazuje všeobecný a ani najťažší prípad, ktorý sa Jonášovi pri presúvaní môže stať.

Ani toto Jonášovi však nestačilo. Prišiel na to, že ak by na vrch pavučiny dal mušku, ktorej váha je v strede medzi všetkými muškami, napravo aj naľavo by bolo rovnako veľa mušiek. V jeho pavučine je to muška s váhou 35, lebo práve päť mušiek je ľahších a práve päť je ťažších. Ako však dostať mušku 35 na vrch siete?

- d) (3 body) Jonáš stojí pri niektorej muške v pavučine. Túto mušku by rád presunul na samý vrch pavučiny. Nechce si však pokaziť celú pavučinu a neustále musia v jeho pavučine platiť uvedené podmienky. Pomôžte mu nájsť postup, ktorý mu bude hovoriť, ako preväzovať vlákna tak, aby sa zvolená muška dostala na vrch siete. Nezabudnite, že váš postup musí fungovať pre ľubovoľnú sieť a mušku v nej, nie len pre sieť na obrázku.

Presúvanie mušiek bez konkrétneho plánu je únavné a častokrát zbytočné. Jonáš si preto nakreslil, ako má jeho pavučina vyzeráť. Ostáva už len presunúť mušky v jeho sieti. Ako to však spraviť?

¹Toto samozrejme nie je jediná postupnosť, ktorá by vytvorila túto pavučinu.

- e) (4 body) Predstavte si, že máte dva obrázky sietí, ktoré obsahujú tie isté mušky, akurát inak usporiadané. Prvý obrázok ukazuje aktuálny stav siete, druhý ukazuje ako má sieť vyzerieť. Popíšte postup, ktorý bude presúvať mušky tak, že zo siete z prvého obrázku vytvorí sieť z druhého obrázku. Samozrejme, váš postup musí fungovať na ľubovoľných dvoch sieťach s tými istými muškami.

Pri riešení môžete použiť riešenia podúlohy c) a d). Nemôžete však sieť stavať od začiatku. Dovoľené máte robiť iba také zmeny, po ktorých bude sieť stále celistvá a bude spĺňať podmienky zadania. To znamená, že môžete presunúť mušku naľavo na vrch siete (ako v podúlohe c)) alebo previazať zopár vlákien, ale nemôžete povedať, že na vrch dáte mušku 27 a zvyšné mušky si zatiaľ niekam odložíte.

Ako príklad si môžete predstaviť, že Jonášova pavučina vyzerá ako na obrázku 2 a jeho cieľom je ju upraviť tak, aby vyzerala ako na obrázku 1.

2. Príbeh o Dášenke a minci

15 bodov za riešenie

Ak máte akékoľvek otázky ohľadom tejto úlohy, napíšte Baške na baska@ksp.sk

“Dášenska, pamätáš si ako som ťa učil počítať?”, pýta sa ocko Bob.

“Áno. Nula, jeden, dva, tri, štyri až po veeeelmi veľa.”, odpovedá malá Dáša s úsmevom.

“Tak dnes ti ukážem hru, ktorú sa s tými číslami môžeš hrať.”

“Jéj, ďalšia hra. Aké sú pravidlá?”

“Každé ráno odo mňa dostaneš dve rovnaké nenulové čísla. Potom si vezmeš túto mincu, a budeš si ňou hádzať. Vždy keď hodíš *hlavu*, pripočítáš prvé číslo k druhému a na papier si zaznačíš **H**. Keď hodíš *znak*, tak pripočítáš druhé číslo k prvému a na papier si zaznačíš **Z**.”

“Môžeš mi to ukázať na príklade?”

“Samozrejme. Začnem napríklad s dvoma štvorkami. Povedzme že som hodil postupne znak, hlava, znak a znak. Aké budú výsledné čísla?”

Dášenska sa zamyslela, zobrala si papier a začala sčítavať. Nakoniec vyhlásila: “Tridsať dva a dvanásť!”.

Bob si pozrel jej papier a našiel tam nasledovnú tabuľku:

Hody	Z	H	Z	Z	
Prvé	4	8	8	20	32
Druhé	4	4	12	12	12

“Správne. Na konci dňa mi potom donesieš ukázať aj postupnosť znakov, aj výslednú dvojicu čísel, dobre?” To ho už ale Dášenska poriadne nepočúvala. Začala si mincu hádzať a pokračovala sčítavaním 32 a 12.

“Ešte jedna vec, kým dávaš pozor. Aby si sa neupísala do zbláznenia, tak ak hodíš viacero rovnakých znakov za sebou, tak stačí ak napíšeš číslo, kolkokrát to bolo, a písmenko **Z** alebo **H**. Teda náš príklad by sa dal zapísať ako **1Z,1H,2Z**. Dobre?”

“Áno oci.” a už jej nebolo.

Ďalší deň

Na druhé ráno dostala Dášenska od Boba dve rovnaké čísla a celý deň si hádzala mincou.

Avšak večer prišla za svojim ockom celá nešťastná, lebo stratila všetky svoje výpočty, a s nimi aj začiatkové dve čísla a postupnosť hodov. Jediné čo jej ostalo boli výsledné dve čísla.

Nanešťastie si ani Bob už nepamätá čísla, ktoré jej ráno zadal. Obrátil sa preto na vás:

“Ahojte PRASKÁči. Vedeli by ste z posledných dvoch čísel zistiť, ktoré čísla som zadal ráno Dášenke a ako vyzerala celá postupnosť hodov? Alebo to už nejde?”

Úloha

Tak si to zhrňme. Na začiatku máme čísla p a d , ktoré sú **rovnaké a kladné**. Postupne s nimi robíme operácie dvoch typov:

- **Z**: pripočítame d ku p , takže vznikne dvojica čísel $p + d$ a d
- **H**: pripočítame p ku d , takže vznikne dvojica čísel p a $d + p$

Potom, ako sme spravili niekoľko takýchto operácií, sme získali nejaké dve čísla P a D . V prvých troch podúlohách je vaša úlohou zistiť (ak je to možné) začiatkovú hodnotu čísel p a d . Okrem toho zistíte aj postupnosť hodov mince v skrátenom formáte.

Skrátený formát je postupnosť dvojíc: číslo a písmenko (**Z** alebo **H**). Číslo je vždy kladné väčšie ako nula, a hovorí koľkokrát po sebe Dášenska hodila konkrétnu stranu mince. V rámci tejto skrátenej postupnosti sa vždy strieda *hlava* a *znak*, takže nemôže nastať napríklad takýto prípad:

$$3Z, 5H, 7Z, 6Z, 8H, 1Z, 5H$$

pretože by sme to vedeli skrátiť a prepísať 7Z, 6Z na 13Z.

Táto úloha sa skladá z niekoľkých podúloh:

- (2 body) Dášenska doniesla Bobovi čísla $P = 15$ a $D = 35$. Viete zistiť postupnosť hodov a hodnotu čísel p a d , ktoré ráno od Boba dostala? Poprípade vysvetlite Bobovi, že sa to už zistiť nedá.
- (2 body) Ak by Dášenska doniesla Bobovi čísla $P = 111$ a $D = 9$, vedeli by ste zistiť postupnosť hodov a hodnotu počiatočných čísel p a d ?
- (5 body) Vedeli by ste riešiť takýto problém pre ľubovoľnú dvojicu čísel P a D ? Popíšte postup, ktorý z dvoch čísel P a D zistí počiatočné čísla p a d a tiež skonštruuje skrátenú postupnosť hodov.
Snažte sa, aby bol váš postup dostatočne rýchly. Vzorové riešenie by vedelo len s pomocou papiera a pera vypočítať výsledok aj pre čísla 1 000 000 a 14, poprípade pre ľubovoľnú dvojicu čísel veľkých až milión.
- (3 body) Vedeli by ste dokázať, že počiatočné číslo p delí obe konečné čísla P a D ?
- (3 body) Vedeli by ste dokázať, že ak nejaké číslo x delí obe výsledné čísla P a D , tak delí aj pôvodné číslo p ?

Čo vyplýva z vlastností d) a e) o čísle p ?

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. Pohodlnejšie Kreslo

15 bodov za riešenie

Ak máte akékoľvek otázky ohľadom tejto úlohy, napíšte Dávidovi na davidb@ksp.sk

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

Tomu už vďaka vám zistil, aký úžasný a krásny je jeho programovací jazyk Kreslo. Niečo mu v ňom však stále chýbalo. Nech sa snažil ako chcel, nedarilo sa mu v ňom nakresliť program, ktorý by zoradil čísla od najmenšieho po najväčšie. Uvedomil si, že na vyriešenie tohto problému potrebuje viac ako 26 registrov, začal si ich preto číslovať. Aby nemusel meniť už existujúce príkazy Kresla, rozhodol sa, že hodnota v registri D bude závisieť od hodnoty v registri A (adresa). To znamená, že pre každú hodnotu v registri A predstavuje D úplne iný register. Túto zmenu pochopíte najlepšie na praktickej ukážke, ktorú sme pre vás pripravili: <https://ksp.sk/~prask/specialne/4/2/3/index.html?ukazka=4>.

Úloha

Kreslo je veľmi jednoduchý jazyk. Pracuje iba s celými číslami, ktoré vie načítavať zo vstupu a vypisovať na výstup. Aby sa Tomimu ľahko kreslilo, celý program je nakreslený do mriežky a každá inštrukcia zaberá jeden štvorec. Podrobnosti o tom, ako tento jazyk funguje a ako v ňom písať programy nájdete na ksp.sk/~prask/specialne/4/2/3/manual.pdf.

Našťastie, my môžeme používať elektronický editor, ktorý dokáže simulovať beh programu. Nájdete ho spolu s ukázkovými programami na ksp.sk/~prask/specialne/4/2/3/.

Vašou úlohou bude vyriešiť nasledujúce podúlohy a odovzdať ZIP súbor, ktorý obsahuje zdrojové kódy vašich riešení (dajú sa stiahnuť z editora). ZIP by mal obsahovať súbory 1.kr až 5.kr s vašimi riešeniami jednotlivých podúloh.

Podúlohy

- a. (2 bod) Na vstupe je číslo n a potom dva krát n čísel. Na výstup vypíšte najskôr druhých n čísel a po nich prvých n čísel.

vstup

3 1 2 3 4 5 6

výstup

4 5 6 1 2 3

Prvé číslo vstupu je $n = 3$. Následne je na vstupe prvá trojica (1, 2, 3) a potom druhá trojica (4, 5, 6). Na výstup preto najskôr vypíšeme druhú trojicu a až potom prvú.

- b. (2 body) Na vstupe je číslo n a potom n čísel. Za nimi sú dve čísla a a b ($1 \leq a \leq b \leq n$). Na výstup vypíšte súčet čísel medzi a -tým až b -tým číslom (vrátane).

vstup

5 1 4 3 3 5 2 4

výstup

10

Na vstupe je 5 čísel, my máme vypísať súčet čísel od druhého po štvrté. To znamená $4 + 3 + 3 = 10$.

- c. (3 body) Na vstupe je n čísel ($n \geq 2$). Vypíšte najmenšie a druhé najmenšie číslo.

vstup

4 8 1 5 -6

výstup

-6 1

- d. (3 body) Na vstupe je n čísel ($n \geq 1$). Vypíšte číslo, ktoré je na vstupe najviac krát. Môžete predpokladať, že na vstupe je také číslo vždy jednoznačné.

vstup

3 2 5 3 1 2 3

výstup

3

- e. (5 bodov) Na vstupe je n čísel ($n \geq 1$). Vypíšte všetky tieto čísla zoradené od najmenšieho po najväčšie.

vstup

3 -1 2 7 -4 5 -6

výstup

-6 -4 -1 2 3 5 7

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. Prefixove rezne

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 Liahni (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 Jarovi na jarko.paska@gmail.com

Prefix má rád rezne. Nedávno pribudla v Bratislave nová reštaurácia a všetci kamaráti, čo ju už navštívili Prefixovi vraveli, že tam robia tie najlepšie rezne. Keďže Prefix rád skúša nové chute rezňov, rozhodol sa, že tam zájde a objedná si všetky druhy rezňov, ktoré tam pripravujú. Po príchode však zistil, že objednávanie funguje v tejto reštaurácii veľmi zvláštnym spôsobom.

V reštaurácii je dlhý pult, na ktorom sú v jednom rade poukladané rezne rôznych druhov – černo-horský, kyjevský, horčicový... Pri objednávaní si kupujúci vyberie súvislý úsek rezňov na pulte a čašníci mu naservirujú všetky rezne z tohto úseku. Avšak, pri jednej návšteve je možné si vybrať iba jediný (aj keď ľubovoľný) úsek.

Prefix by si už pri prvej návšteve chcel kúpiť z každého druhu rezňa aspoň jeden (nech teda zistí, ktoré rezne sú najlepšie), nechce to však s kupovaním prehnáť a chcel by kúpiť čo najmenej rezňov.

Pomôžte Prefixovi vybrať najkratší úsek rezňov na pulte, ktorý obsahuje aspoň jeden z každého druhu rezňov. Samozrejme, môže sa stať, že na pulte sa nenachádzajú všetky druhy rezňov, ktoré reštaurácia ponúka (napríklad im práve došla horčica) a vtedy bude Prefix veľmi sklamaný.

Úloha

V reštaurácii ponúkajú m rôznych druhov rezňov a na pulte ich je v jednom rade vyložených n . Vašou úlohou je nájsť dĺžku **najkratšieho súvislého úseku**, ktorý obsahuje všetkých m druhov rezňov, poprípade prehlásiť, že taký úsek neexistuje.

Formát vstupu

Na prvom riadku vstupu dostaneme dve celé čísla n a m – počet rezňov na pulte a počet rezňov v ponuke. Platí $m \leq n$. Druhy rezňov si označujeme číslami od 1 po m .

Na druhom riadku je n čísel x_1, x_2, \dots, x_n z rozsahu 1 až m , číslo x_i reprezentuje druh i -teho rezňa na pulte.

Formát výstupu

Na jediný riadok výstupu vypíšete dĺžku najkratšieho úseku obsahujúceho aspoň jeden z každého druhu rezňov. Ak takýto úsek neexistuje, vypíšete “-1”.

Hodnotenie

Vaše riešenie bude otestované na 3 sadách vstupov. Za vyriešenie každej sady získate 5 bodov. Tieto sady sa líšia veľkosťou vstupných údajov.

V nasledujúcej tabuľke označuje n počet rezňov na pulte. Vo vstupných sadoch platí:

Číslo sady	1	2	3
maximálne n	100	2 000	1 000 000

Príklady

vstup	výstup
8 4 1 4 3 4 3 2 2 1	5

Správnou možnosťou je vybrať úsek začínajúci na pozícii 4 a končiaci na pozícii 8. Tento úsek má dĺžku 5 a naozaj obsahuje všetky čísla od 1 po 4. Rozmyslite si, že kratší úsek s danou vlastnosťou naozaj neexistuje.

vstup	výstup
4 3 1 1 2 1	-1

Na pulte nie je vystavený rezeň druhu 3, preto neexistuje žiaden úsek spĺňajúci požadovanú vlastnosť.

5. Písmenková šifra

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 Liahni (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

Andrej a Kika medzi sebou už dlho tajne komunikujú. Ako to už ale so šifrovanými správami býva, šifrovanie a dešifrovanie nie je jednoduché. Boli dni, a nebolo ich málo, keď obaja presedeli celé hodiny nad svojimi šiframi. Keďže najviac času im zaberá dešifrovanie, požiadali vás, aby ste im s touto úlohou trochu pomohli.

Systém ich šifrovania je pomerne jednoduchý. Predstavte si, že zoberieme všetky možné slová zapísané veľkými písmenami anglickej abecedy a tieto slová zoradíme. Zoradovanie funguje nasledovne: kratšie slová dáme skôr ako dlhšie. V prípade, že sú dve slová rovnako dlhé, ako prvé dáme to, ktoré je skôr v abecede – budeme porovnávať písmená zo začiatku týchto slov až kým nenarazíme na prvú pozíciu, na ktorej sú rôzne. To slovo, ktoré má na tejto pozícii abecedne menšie písmeno zaradíme skôr ako to druhé.

Takýmto spôsobom dostaneme postupnosť všetkých možných slov. Prvé slovo je A, za ním nasleduje B, C ... Z, AA, AB, AC ... AZ, BA ... ZZ, AAA ... ANO ...

Predložka Z je tak na 26-tej pozícii, zatiaľ čo slovo AHA je 885. v poradí. Na zašifrovanie slova správy potom Andrej s Kikou použijú poradie slova v tejto postupnosti.

Úloha

Vašou úlohou je napísať program, ktorý na vstupe dostane n čísel. Každé z nich reprezentuje jedno zašifrované slovo správy, teda pozíciu slova v zoradenej postupnosti všetkých slov. Úlohou je rozšifrovať správu, čo znamená nájsť jednotlivé slová v postupnosti.

Formát vstupu

Na prvom riadku je číslo n – dĺžka správy. Na každom z n nasledujúcich riadkov je jedno **kladné** číslo. Každé z týchto čísel zodpovedá jednému zašifrovanému slovu správy.

Formát výstupu

Na n riadkoch výstupu výpíšte jednotlivé rozšifrované slová zo správy v takom poradí, v akom boli na vstupe.

Pri riešení v jazyku C++, Pascal, Java a podobných si dávajte pozor na maximálne číslo, ktoré sú schopné celočíselné premenné uložiť. Niektoré slová môžu byť zašifrované na veľké čísla prekračujúce tieto limity. Odporúčame preto použiť 64-bitové premenné, v jazyku C++ preto namiesto typu `int` použijete typ `long long int`.

Riešenie v Pythone sa takýto typ obmedzení netýka, pretože premenné v ňom majú neobmedzenú veľkosť.

Hodnotenie

Vaše riešenie bude otestované na troch sádach vstupov. Za vyriešenie každej sady získate 5 bodov. Tieto sady sa líšia veľkosťou vstupných údajov.

V nasledujúcej tabuľke označuje n dĺžku správy a x jeden jej prvok, teda jedno zašifrované slovo. Vo vstupných sádach platí:

Číslo sady	1	2	3
maximálne n	2 000	6 000	10 000
maximálne x	10 000	10^8	10^{15}

Príklady

vstup

```
4
5
28
46
100
```

výstup

```
E
AB
AT
CV
```

vstup

3
7629165
265
9062604

výstup

PRASK
JE
SUPER