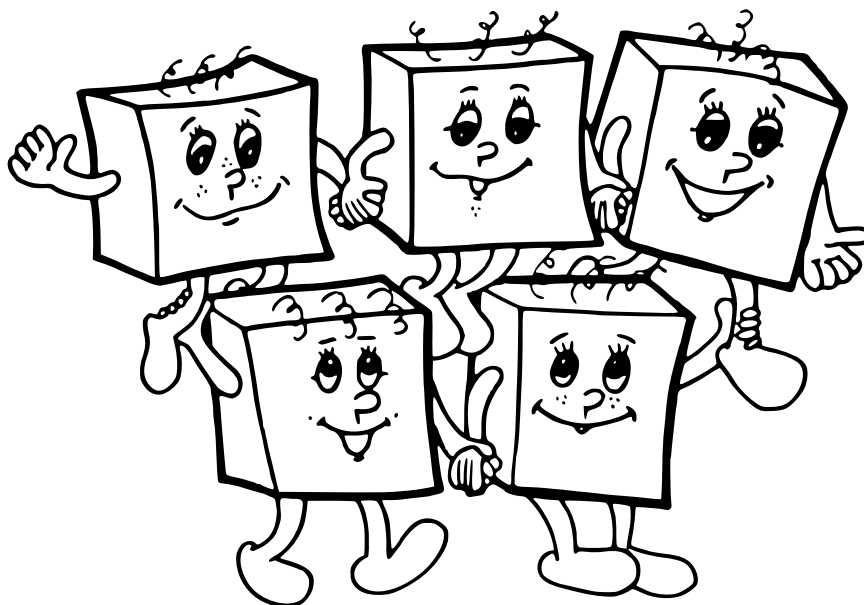


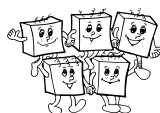
OLYMPIÁDA V INFORMATIKE NA STREDNÝCH ŠKOLÁCH



dvadsiaty štvrtý ročník
školský rok 2008/09

zadania krajského kola
kategória B

- **Olympiáda v informatike** je od školského roku 2006/07 samostatnou súťažou. Predchádzajúcich 21 ročníkov tejto súťaže prebiehalo pod názvom **Matematická olympiáda, kategória P** (programovanie).
- Oficiálnu **webstránku** súťaže nájdete na <http://oi.sk/>.



Priebeh krajského kola

Krajské kolo 24. ročníka Olympiády v informatike, kategórie B, sa koná 13. januára 2008 v dopoludňajších hodinách. Na riešenie úloh majú súťažiaci 4 hodiny čistého času. Rôzne úlohy riešia súťažiaci na samostatné listy papiera. Akékoľvek pomôcky okrem písacích potrieb (napr. knihy, výpisy programov, kalkulačky) sú zakázané.

Čo má obsahovať riešenie úlohy?

- Slovné popíšte algoritmus.
Slovný popis riešenia musí byť jasný a zrozumiteľný i bez nahliadnutia do samotného zápisu algoritmu (do programu).
- Zdôvodnite správnosť vášho algoritmu.
- Uveďte a zdôvodnite jeho časovú a pamäťovú zložitosť.
- Podrobne uveďte dôležité časti algoritmu, ideálne vo forme programu v Pascale alebo C/C++.
- V prípade, že používate vo svojom programovacom jazyku knižnice, ktoré obsahujú implementované dátové štruktúry a algoritmy (napr. STL pre C++), v popise algoritmu stručne vysvetlite, ako by ste napísali program s rovnakou časovou zložitou bez použitia knižnice.

Hodnotenie riešení

Za každú úlohu môžete získať od 0 do 10 bodov.

Pokiaľ nie je v zadaní povedané ináč, základným kritériom hodnotenia riešenia je **správnosť** a **efektívnosť** navrhnutého algoritmu. Hodnotí sa aj kvalita popisu riešenia a zdôvodnenie tvrdení o jeho správnosti a efektívnosti.

Efektívnosť algoritmu posudzujeme vypočítaním jeho časovej zložitosti – funkcie, ktorá hovorí, ako dlho vykonanie algoritmu trvá v závislosti od veľkosti vstupných parametrov.

V zadaní úlohy môžu byť uvedené limity na veľkosť premenných. Tieto môžete použiť na odhad toho, ako dobré vaše riešenie je. Na počítači, ktorý vykoná miliardu inštrukcií za sekundu, zrieši vzorové riešenie ľubovoľný povolený vstup do niekoľko sekúnd.

Zadania kategórie B

B-II-1 Binárny súčet

Bitlandia je jediná známa krajina, v ktorej obyvatelia poznajú len dvojkovú sústavu. Binárne čísla sa tam používajú úplne všade – v obchodoch, školách, na hotelových izbách, ba dokonca aj čísla električiek sú písané v tejto sústave. Obyvatelia Bitlandie takisto radi riešia rôzne krížovky či hlavolamy. Najnovším hitom sa stal takzvaný Bitlandský súčtový hlavolam. Zadanie hlavolamu pozostáva z troch binárnych čísel (označme ich A , B , C), pričom niektoré cifry v číslach A , B chýbajú. Úlohou riešiteľa je doplniť chýbajúce cifry tak, aby platil súčet $A + B = C$. Nasledujúci obrázok ilustruje jedno zadanie hlavolamu.

$$\begin{array}{r} 0010**0 \\ 001**1* \\ \hline 0101001 \end{array}$$

Jedno z možných riešení tohoto hlavolamu je súčet $0010110 + 0010011 = 0101001$.

Súťažná úloha

Napište program, ktorý načíta reťazce A , B , C a rozhodne, či zadaný hlavolam má riešenie. Ak hlavolam je riešiteľný, váš program by mal vypísať aj jedno riešenie.



Formát vstupu a výstupu

Vstup obsahuje tri riadky, ktoré postupne obsahujú reťazce A , B a C . Chýbajúce cifry sa vyskytujú len v reťazcoch A a B a sú označené znakom '*'. Môžete predpokladať, že všetky tri reťazce na vstupe majú rovnakú dĺžku.

V prípade, že hlavolam zo vstupu má riešenie, váš program by mal vypísať jedno riešenie – dva riadky, určujúce sčítance hlavolamu v binárnom zápise. Ak hlavolam nemá riešenie, výstupom by mal byť jediný riadok s textom „Nema riesenie.“.

Príklady

vstup	výstup
<pre>010**0 01**1* 101001</pre>	<pre>010110 010011</pre>
vstup	výstup
<pre>01** 01*0 1111</pre>	<pre>Nema riesenie.</pre>

Hodnotenie

- Riešenia, ktoré dokážu efektívne spracovať 100 000-ciferné reťazce, môžu získať maximálne 10 bodov.
- Riešenia, ktoré dokážu efektívne spracovať 1 000-ciferné reťazce, môžu získať maximálne 8 bodov.
- Riešenia, ktoré dokážu efektívne spracovať 10-ciferné reťazce, môžu získať maximálne 5 bodov.

B-II-2 Výber dovolenky

Peter sa rozhodol, že si už teraz v zime naplánuje dovolenku. Chce ísť do jedného letoviska, kde sa každý večer koná jedna atrakcia. Petra konkrétne zaujímajú koncerty, futbalové zápasy a preteky člnov v prístave. Chcel by počas svojej dovolenky každú z týchto atrakcií zažiť aspoň raz. Dovolenka samozrejme musí byť jedno súvislé časové obdobie, a keďže Peter chce za ňu zaplatiť čo najmenej, čím bude kratšia, tým lepšie. Pomôžte Petrovi nájsť najlepšie obdobie na dovolenku.

Súťažná úloha

Daný je reťazec tvorený N malými písmenami anglickej abecedy. Tento reťazec popisuje atrakcie, ktoré sa budú v jednotlivé noci v letovisku konať. Písmeno a predstavuje koncert, písmeno b futbalový zápas, c sú preteky člnov a ostatné písmená predstavujú iné atrakcie. Napíšte program, ktorý nájde dĺžku najkratšieho úseku, ktorý aspoň raz obsahuje každú z troch udalostí, ktoré Petra zaujímajú.

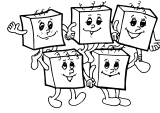
Formát vstupu a výstupu

V prvom riadku je jedno celé číslo N udávajúce počet dní, z ktorých si Peter môže vyberať. Druhý riadok vstupu obsahuje N -znakový reťazec popisujúci atrakcie.

Vypíšte jeden riadok a v ňom jedno celé číslo – najmenší počet dní, ktoré musí Peter stráviť na dovolenke.

Príklad

vstup	výstup
<pre>10 abaacazaba</pre>	<pre>4</pre> <p><i>Od 2. do 5. dňa zažije všetky tri udalosti. Žiadny kratší úsek nevyhovuje.</i></p>



Hodnotenie

- Riešenia fungujúce pre $N \leq 10\,000\,000$ môžu získať maximálne 10 bodov.
- Riešenia fungujúce pre $N \leq 1\,000\,000$ môžu získať maximálne 9 bodov.
- Riešenia fungujúce pre $N \leq 5\,000$ môžu získať maximálne 7 bodov.
- Riešenia fungujúce pre $N \leq 500$ môžu získať maximálne 5 bodov.

B-II-3 Sánkovanie

Malý Kleofáš našiel pod stromčekom nové sánky. A odvtedy vyzerajú všetky jeho dni rovnako – akonáhle môže, vytiahne sánky na kopec za domom, spustí sa dole, zase vytiahne sánky na kopec... a tak dokola, až kým ho mama nezavolá domov.

Po čase však aj Kleofáša omrzelo spúšťať sa len tak, a tak si začal plánovať rôzne trasy: Z vrcholu sa napríklad pustí k osamelej jabloni, od nej k odstavenému traktorovi, odtiaľ k mostíku, a od mostíka rovno dole.

Kleofáš si na papieri nakreslil schému kopca. Na nej vyznačil N zaujímavých miest a očísloval ich od 1 do N . Miesto číslo 1 je vrchol kopca, kde Kleofáš každú jazdu začína. Miesto číslo N je úpätie kopca, kde každú jazdu končí. (Pozor, ostatné miesta **nemusia byť** očíslované podľa nadmorskej výšky!)

Medzi niektorými dvojicami zaujímavých miest sa dá priamo pustiť na saniach, medzi inými nie. Kleofáš si vypísal všetkých M dvojíc miest, medzi ktorými sa pustiť dá. Teraz sedí nad svojim papierom a snaží sa porátať, koľkými rôznymi cestami sa vlastne môže z kopca spustiť.

Súťažná úloha

Napište program, ktorý načíta údaje z Kleofášovho papiera a spočíta hľadaný počet ciest.

Formát vstupu a výstupu

Prvý riadok vstupu obsahuje celé číslo N ($2 \leq N \leq 100\,000$), udávajúce počet zaujímavých miest. Druhý riadok vstupu obsahuje celé číslo M ($1 \leq M \leq 1\,000\,000$), udávajúce počet dvojíc miest, medzi ktorými sa dá priamo spustiť. Nasleduje M riadkov, v každom z nich sú dve čísla a a b hovoriace, že z miesta a sa dá priamo spustiť na miesto b .

Popis vstupu skutočne zodpovedá miestam na kopci. Ak sa vieme spustiť z a do b , znamená to, že a má väčšiu nadmorskú výšku ako b .

Pri písaní programu môžete predpokladať, že sa hľadaný počet ciest zmestí do bežnej celočíselnej premennej.

Vypíšte jeden riadok a v ňom jediné celé číslo, udávajúce počet spôsobov, ako sa dá spustiť z kopca popísaného na vstupe.

Príklad

vstup

```
7
9
1 5
3 2
5 2
5 4
1 3
3 6
6 7
6 2
2 7
```

Možné cesty vyzerajú nasledovne:

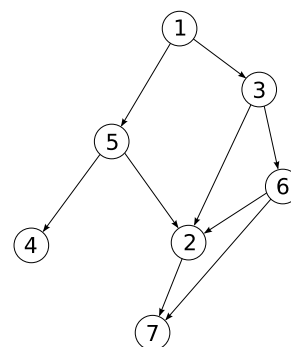
$1 \rightarrow 5 \rightarrow 2 \rightarrow 7$, $1 \rightarrow 3 \rightarrow 2 \rightarrow 7$, $1 \rightarrow 3 \rightarrow 6 \rightarrow 2 \rightarrow 7$

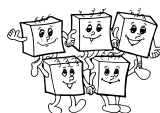
a $1 \rightarrow 3 \rightarrow 6 \rightarrow 7$.

output

```
4
```

Takto vyzerá zadaný kopec:





Hodnotenie

- Riešenie zvládajúce limity uvedené pri formáte vstupu môže získať 10 bodov.
- Nanajvýš 8 bodov dostanete za algoritmus, ktorý na bežnom počítači do sekundy skončí pre $N \leq 2000$.
- Nanajvýš 6 bodov dostanete za algoritmus, ktorý vyrieši pôvodnú úlohu za predpokladu, že zaujímavé miesta sú očíslované podľa nadmorskej výšky. (Teda že pre každú dvojicu „ a b “ na vstupe platí $a < b$.)
- Nanajvýš 5 bodov dostanete za algoritmus, ktorý vyrieši túto zjednodušenú úlohu pre $N \leq 2000$.
- Aspoň 3 body môžete získať za ľubovoľné funkčné riešenie.

B-II-4 Banka

V mestečku Kocúrkovo onedlho otvoria novú banku. Čísla účtov v tejto banke budú mať práve N cifier. Číslo účtu môže začínať aj nulami, teda napríklad 0047 je platné číslo účtu pre $N = 4$.

Banka by chcela pridelovať čísla účtov zákazníkom tak, aby ich ochránila pred následkom preklepu pri zadávaní čísla účtu. Presnejšie, ak napíšeme ľubovoľné číslo existujúceho účtu a pomýlime sa pri tom v jednej cifre, nesmieme dostať číslo iného existujúceho účtu.

- (1 bod)** Ukážte, že pre $N = 2$ môže mať Kocúrkovská banka 10 zákazníkov.
- (2 body)** Dokážte, že pre žiadne N nemôže mať Kocúrkovská banka viac ako 10^{N-1} zákazníkov.
- (4 body)** Navrhnite, ako má Kocúrkovská banka vyberať čísla účtov, aby mohla mať čo najviac zákazníkov. (Plný počet bodov za túto podúlohu dostanete, ak sa vám podarí nájsť spôsob ako zostrojiť 10^{N-1} vyhovujúcich čísel účtov.)
- (3 body)** Aj v susedných dvoch dedinách, Popletenej Vieske a Bezpečnej Trieske, otvárajú nové banky, a tie majú ešte lepšie ponuky.

Banka v Popletenej Vieske bude voliť čísla účtov tak, že keď zákazník napíše svoje číslo účtu a pomýli sa **v nanajvýš 4 cifrách**, nikdy nedostane číslo iného existujúceho účtu.

Banka v Bezpečnej Trieske bude voliť čísla účtov tak, že keď zákazník napíše svoje číslo účtu a pomýli sa **v nanajvýš 2 cifrách**, nielenže určite nedostane číslo iného existujúceho účtu, ale navyše bude banka vždy vedieť jednoznačne určiť jeho skutočné číslo účtu.

Ktorá banka môže mať viac zákazníkov, ak obe používajú 10-ciferné čísla účtov a volia ich najlepšie ako sa dá?

(Čísla účtov v rôznych bankách sú navzájom nezávislé, môže sa stať, že nejaký zákazník prvej banky bude mať trebárs aj rovnaké číslo účtu ako iný zákazník druhej banky.)