



Informácie a pravidlá

Pre koho je súťaž určená?

Do **kategórie B** sa smú zapojiť len tí žiaci základných a stredných škôl, ktorí ešte ani v tomto, ani v nasledujúcom školskom roku nebudú končiť strednú školu.

Do **kategórie A** sa môžu zapojiť všetci žiaci (základných aj) stredných škôl.

Odvzdávanie riešení domáceho kola

Riešitelia domáceho kola odovzdávajú riešenia sami, v elektronickej podobe, a to priamo na stránke olympiády: <http://oi.sk/>. Odovzdávanie riešení bude spustené niekedy v septembri.

Riešenia kategórie A je potrebné odovzdať najneskôr **15. novembra 2015**.

Riešenia kategórie B je potrebné odovzdať najneskôr **30. novembra 2015**.

Priebeh súťaže

Za každú úlohu domáceho kola sa dá získať od 0 do 10 bodov. Na základe bodov domáceho kola stanoví Slovenská komisia OI (SK OI) pre každú kategóriu bodovú hranicu potrebnú na postup do **krajského kola**. Očakávame, že táto hranica bude približne rovná **tretine maximálneho počtu bodov**.

V krajskom kole riešitelia riešia štyri teoretické úlohy, ktoré môžu tematicky nadväzovať na úlohy domáceho kola. V kategórii B súťaž týmto kolom končí.

V kategórii A je približne najlepších 30 riešiteľov krajského kola (podľa počtu bodov, bez ohľadu na kraj, v ktorom súťažili) pozvaných do **celoštátneho kola**. V celoštátnom kole účastníci prvý deň riešia teoretické a druhý deň praktické úlohy. Najlepší riešitelia sú vyhlásení za víťazov. Približne desať najlepších riešiteľov následne SK OI pozve na týždňové výberové sústreďenie. Podľa jeho výsledkov SK OI vyberie družstvá pre Medzinárodnú olympiádu v informatike (IOI) a Stredoeurópsku olympiádu v informatike (CEOI).

Ako majú vyzeráť riešenia úloh?

V praktických úlohách je vašou úlohou vytvoriť program, ktorý bude riešiť zadanú úlohu. Program musí byť v prvom rade korektný a funkčný, v druhom rade sa snažte aby bol čo najefektívnejší.

V kategórii B môžete použiť ľubovoľný programovací jazyk.

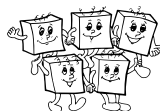
V kategórii A musíte riešenia praktických úloh písať v jazyku Pascal, C, alebo C++. Odovzdaný program bude automaticky otestovaný na viacerých vopred pripravených testovacích vstupoch. Podľa toho, na koľko z nich dá správnu odpoveď, vám budú pridelené body. Výsledok testovania sa dozviete krátko po odovzdaní. Ak váš program nezíska plný počet bodov, budete ho môcť vylepšiť a odovzdať znova, až do uplynutia termínu na odovzdávanie.

Presný popis, ako majú vyzeráť riešenia praktických úloh (napr. realizáciu vstupu a výstupu), nájdete na webstránke, kde ich budete odovzdávať.

Ak nie je v zadaní povedané ináč, riešenia teoretických úloh musia v prvom rade obsahovať **podrobný slovný popis použitého algoritmu, zdôvodnenie jeho správnosti** a diskusiu o efektivite zvoleného riešenia (t. j. posúdenie časových a pamäťových nárokov programu). Na záver riešenia uveďte program. Ak používate v programe netriviálne algoritmy alebo dátové štruktúry (napr. rôzne súčasti STL v C++), súčasťou popisu algoritmu musí byť dostatočný popis ich implementácie.

Usporiadateľ súťaže

Olympiádu v informatike (OI) vyhlasuje *Ministerstvo školstva SR* v spolupráci so *Slovenskou informatickou spoločnosťou* (odborným garantom súťaže) a *Slovenskou komisiou Olympiády v informatike*. Súťaž organizuje *Slovenská komisia OI* a v jednotlivých krajoch ju riadia *krajské komisie OI*. Na jednotlivých školách ju zaisťujú učitelia informatiky. Celoštátne kolo OI, tlač materiálov a ich distribúciu po organizačnej stránke zabezpečuje IUVENTA v tesnej súčinnosti so Slovenskou komisiou OI.



B-I-1 Fľaše



V rade stojí n fliaš s vodou. Rôzne fľaše obsahujú rôzne množstvo vody: v i -tej fľaši zľava je vody v_i mililitrov.

Súťažná úloha

Jakub by chcel vodu poprelievat' medzi fľašami tak, aby bolo na konci v každej fľaši presne rovnaké množstvo vody. Jakub je skúsený chemik, a tak dokáže z jednej fľaše do druhej preliať ľubovoľný celočíselný počet mililitrov vody. Je však lenivý, a tak nechal na vás, aby ste mu poradili, odkiaľ, kam a koľko vody má preliať.

Formát vstupu

Dostanete od nás 5 vstupných súborov, označených `1.txt` až `5.txt`. Každý obsahuje jednu postupnosť fliaš.

V prvom riadku vstupného súboru je uvedené číslo n : počet fliaš. V jednotlivých vstupných súboroch má počet fliaš hodnotu 10, 100, 1000, 100 000 a 300 000.

V druhom riadku je uvedená postupnosť celých čísel v_1, \dots, v_n oddelených medzerami: počty mililitrov vody v jednotlivých fľašiach. Môžete predpokladať, že všetky v_i sú z rozsahu od 0 po 1 000 000. Taktiež môžete predpokladať, že súčet všetkých hodnôt v_i je deliteľný číslom n . (Inými slovami, na konci prelievania bude v každej fľaši celočíselný počet mililitrov vody.) Dajte si pozor na to, že vo vstupných súboroch 4 a 5 bude súčet všetkých v_i veľký. Použite na jeho uloženie premennú s dostatočným rozsahom!

Formát výstupu

Prelievanie je postupnosť inštrukcií tvaru „prelej x mililitrov vody z fľaše y do fľaše z “. Číslo y a z musia byť platné čísla dvoch rôznych fliaš. Fľaša y musí v okamihu, kedy Jakub inštrukciu vykoná, obsahovať aspoň x mililitrov vody. Môžete predpokladať, že sa prilievaná voda do fľaše z vždy zmestí.

Nájdite ľubovoľný postup prelievania, ktorý bude tvorený **nanajvýš** n inštrukciami a po ktorom bude v každej fľaši rovnako veľa vody. V prvom riadku výstupu vypíšte číslo k udávajúce počet inštrukcií vo vašom postupe. Následne vypíšte k riadkov a v každom z nich čísla x , y a z popisujúce jednu inštrukciu.

Odvzdávanie riešení

Toto je praktická úloha. Napíšte v **ľubovoľnom programovacom jazyku** program, ktorý ju rieši.

Zo stránky <http://oi.sk/> stiahnite ZIP archív obsahujúci 5 testovacích vstupov, nazvaných `1.txt` až `5.txt`.

Vyrobte k čo najviac vstupom správne výstupy a uložte ich do súborov `sol1.txt` až `sol5.txt`.

Odvzdajte ZIP archív obsahujúci **zdrojový kód vášho programu** a tieto výstupné súbory.

Za každý správny výstupný súbor získate 2 body.

Príklad

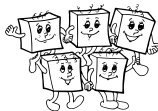
vstup

```
5
10 8 9 10 13
```

výstup

```
3
2 5 3
1 5 2
1 3 2
```

Všimnite si, že vypísaný postup nemusí byť najkratší možný. Iným správnym riešením tohto príkladu by bolo najskôr preliať 2 mililitre vody z fľaše 5 do fľaše 2 a potom preliať 1 mililiter vody z fľaše 5 do fľaše 3.



B-I-2 Disc golf

Disc golf je zábavný šport. Hrá sa podobne ako klasický golf, len namiesto toho, aby hráč palicou odpaľoval loptičku, hádže rukou špeciálny lietajúci tanier (disk). No a na konci nie je cieľom trafiť sa do jamky, ale do špeciálneho koša, ktorý vidíte na obrázku vpravo. (Samotný kôš je približne v strede výšky. Dobre hodený disk väčšinou narazí do reťazí, tie ho zastavia, a následne spadne do koša.) Hodnotí sa, koľko hodov hráč potreboval na to, aby dostal disk do koša – samozrejme, čím menej, tým lepšie.

Discgolfové ihrisko je tvorené j „jamkami“ (teda košmi), očíslovanými od 1 po j . Disc golf hrá naraz skupinka hráčov, pričom jednotlivé jamky hrajú po poradí: keď všetci odohrajú jamku x , presunú sa k jamke $x + 1$.

My budeme sledovať skupinku n hráčov (pričom $2 \leq n \leq 26$). Pre potreby našej úlohy budú títo hráči označení n prvými veľkými písmenami anglickej abecedy. Teda napríklad pre $n = 3$ sa hráči volajú A, B a C.

Skôr, než začnú hráči hrať konkrétnu jamku, je potrebné určiť, v akom poradí budú hádzať svoj prvý hod. Tu platí nasledovné pravidlo: „komu sa práve darí lepšie, ten pôjde hádzať skôr“. Ak by napríklad predchádzajúcu jamku A zahral na 4 hody, B na 7 a C len na 3 hody, tak túto jamku pôjdu hádzať v poradí CAB.

Formálne si toto pravidlo môžeme definovať nasledovne. Majme dvoch hráčov: X a Y . Kto z nich pôjde skôr hrať aktuálnu jamku? Ten, kto potreboval menej hodov na predchádzajúcej. Ak zahrali predchádzajúcu jamku obaja rovnako, pozrieme sa na o jedno skoršiu jamku, a tak ďalej, až kým sa nerozhodne alebo neprídeme na začiatok hry. A ak doteraz všetky jamky zahrali obaja rovnako dobre, skôr pôjde hádzať ten, koho meno je skôr v abecede.



Súťažná úloha

Daný je počet hráčov n , počet jamiek j , a následne pre každého hráča počet hodov, ktoré potreboval na každej z jamiek. Napíšte program, ktorý pre každú jamku zistí, v akom poradí ju hráči začali hrať.

Formát vstupu

Dostanete od nás 5 vstupných súborov, označených `1.txt` až `5.txt`. Každý z nich obsahuje popis jednej hry. Popis každej hry začína riadkom obsahujúcim čísla n a j . Nasleduje n riadkov: jeden pre každého hráča (v abecednom poradí). Popis hráča je tvorený reťazcom j cifier. Všetky cifry sú z rozsahu od 1 po 9. Platí, že i -ta z týchto cifier je počet hodov, ktoré tento hráč potreboval na jamke i .

Veľkosti vstupov

Ako sme už povedali, pre počet hráčov vždy platí $2 \leq n \leq 26$.

V jednotlivých vstupných súboroch bude počet jamiek v hre nasledovný: 10, 100, 1000, 250 000 a 500 000.

Formát výstupu

Do výstupného súboru vypíšte j riadkov. V i -tom z nich vypíšte reťazec n písmen: mená hráčov v poradí, v ktorom hádzali na jamke i .

Odvzdávanie riešení

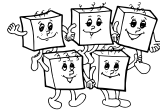
Toto je praktická úloha. Napíšte v ľubovoľnom programovacom jazyku program, ktorý ju rieši.

Zo stránky <http://oi.sk/> stiahnite ZIP archív obsahujúci 5 testovacích vstupov, nazvaných `1.txt` až `5.txt`.

Vyrobte k čo najviac vstupom správne výstupy a uložte ich do súborov `sol1.txt` až `sol5.txt`.

Odvzdajte ZIP archív obsahujúci zdrojový kód vášho programu a tieto výstupné súbory.

Za každý správny výstupný súbor získate 2 body.



Príklady

vstup

```
3 2
43
45
44
```

výstup

```
ABC
ABC
```

Na prvej jamke hádžu hráči v abecednom poradí. Keďže ju zahrali všetci rovnako dobre, aj na druhej hádžu všetci v abecednom poradí. Na výsledkoch na druhej jamke nezáleží, keďže žiadna ďalšia jamka už nenasleduje.

vstup

```
4 5
31249
23345
33327
23345
```

výstup

```
ABCD
BDAC
ABDC
ABDC
CABD
```

Opäť, na prvej jamke hrajú hráči v abecednom poradí.

Pozrime sa na druhú jamku. Tú najskôr pôjdu hádzať hráči B a D, keďže obaja zvládli prvú jamku na 2 hody, a až potom hráči A a C, ktorí potrebovali po 3 hody. Keďže žiadne iné skôr odohraté jamky v tejto chvíli nemáme, o tom, že B hádže skôr ako D (a tiež o tom, že A je skôr ako C), rozhodlo abecedné poradie.

Tretiu jamku pôjde ako prvý hádzať hráč A, lebo sa mu najlepšie darilo na druhej jamke.

Tiež si môžeme všimnúť, že na štvrtej jamke bude hráč B hrať skôr ako hráč C: mali síce rovnaký výsledok na tretej aj na druhej jamke, na prvej jamke však hráč B potreboval len 2 hody, zatiaľ čo hráč C až 3.



B-I-3 Pokazené kalkulačky

Toto je teoretická úloha. Pomocou webového rozhrania odovzdajte súbor vo formáte PDF, obsahujúci riešenie, spĺňajúce požiadavky uvedené v pravidlách.

V tejto úlohe sa budeme hrať s obyčajnými jednoduchými kalkulačkami, ako napríklad tá, ktorú vidíte na obrázku napravo. Pre nás bude dôležité, že tieto kalkulačky nepoznajú prioritu operátorov a jednoducho ich vyhodnocujú v tom poradí, v akom ich zadáte. Keď teda napíšete napríklad $6+3/3=$, výsledkom bude 3, nie 7.

Hlavnými hrdinami nášho príbehu sú kamarátky Janka a Zuzka. Každá má svoju kalkulačku. Obe kalkulačky sú však už staré a takmer nefunkčné.

Na Jankinej kalkulačke už funguje len tlačidlo AC (ktoré všetko zmaže a na displeji nechá nulu) a tlačidlá $46+-*/=$. Vie teda písať cifry 4 a 6, sčítať, odčítať, násobiť a deliť. Môže napríklad do kalkulačky zadať výraz $66/4=$ a na displeji si prečítať, že výsledok je 16.5.

Podúloha A (2 body). Ukážte, ako vie Janka na displeji svojej kalkulačky vyrobiť číslo 13. (Existuje viacero spôsobov. Stačí, ak nájdete ľubovoľný jeden z nich.)

Podúloha B (2 body). Janka na tom vôbec nie je tak zle, ako by sa možno zdalo. Ukážte, že vie na displeji vyrobiť ľubovoľné celé číslo od 1 do 1 000 000.

Na Zuzkinej kalkulačke okrem tlačidla AC už fungujú len tlačidlá $68-/*=$.

Podúloha C (1 bod). Ukážte, ako vie Zuzka na displeji svojej kalkulačky vyrobiť číslo -13 . (Pozor, chceme číslo mínus trinásť, nie číslo trinásť!) Opäť, stačí nájsť jeden ľubovoľný spôsob.

Podúloha D (3 body). Ukážte, že Zuzka vie tiež na displeji vyrobiť ľubovoľné celé číslo od 1 do 1 000 000.

Podúloha E (2 body). Vyberte si ľubovoľnú z našich dievčín a ukážte, že čísla od 1 po milión vie dokonca na displeji vyrobiť *rozumne rýchlo*. Presnejšie, pre vami zvolenú dievčinu nájdite postup, ktorým ľubovoľné celé číslo od 1 po milión vyrobí na menej ako tisíc stlačení kláves.
(Úloha má riešenie pre obe dievčiny, nám ale stačí keď nájdete riešenie pre ľubovoľnú jednu z nich.)

Pri riešení úlohy vám ešte trochu pomôžeme: Na adrese <http://oi.sk/archiv/2015/kalkulacky/> sa môžete pohrať s oboma pokazenými kalkulačkami.





B-I-4 Regulárne výrazy

Každý z nás pri práci s počítačom občas hľadá nejaký text – či už je to na webstránke alebo v dokumente. Častokrát stačí, ak stlačíme **Ctrl+F**, napíšeme hľadaný výraz a program nám nájde všetky jeho výskyty. Ak však potrebujeme vyhľadať viacero synonym toho istého slova, všetky mená začínajúce veľkým písmenom a končiacie na „ka“ alebo platnú e-mailovú adresu, bežné nástroje prestávajú stačiť. Vtedy môžeme siahnuť po *regulárnych výrazoch*.

Regulárne výrazy patria medzi najpoužívanejšie nástroje na vyhľadávanie v texte. V tomto zadaní si popíšeme základy ich tvorby postačujúce na riešenie súťažných úloh. Pokojne si toho ale o regulárnych výrazoch prečítajte aj viac, takmer určite sa vám získané vedomosti ešte budú hodiť v praxi.

Základy použitia regulárnych výrazov

Pre jednoduchosť sa dohodneme, že všetky objekty, ktoré budeme hľadať, budú *reťazce* tvorené obmedzenou množinou znakov. Povolené znaky budú len písmená anglickej abecedy a cifry („a“-„z“, „A“-„Z“, „0“-„9“).¹ Vyhľadávať budeme tak, že napíšeme regulárny výraz – teda nejakú *vzorčku* (pattern). Vzorka popisuje, ako vyzerajú reťazce, ktoré nás zaujímajú. Presnejšie, o každom reťazci vieme povedať, či vzorke zodpovedá (matches the pattern) a našim cieľom pri vyhľadávaní bude napísať vzorku tak, aby jej zodpovedali práve tie reťazce, ktoré chceme nájsť, a žiadne iné. Ako uvidíme nižšie, vo vzorke sa budú môcť vyskytovať aj niektoré iné znaky ako v reťazcoch.

Teraz si popíšeme základy tvorby vzoriek – teda povieme si, z čoho sa taká vzorka môže skladať a ktoré reťazce jej potom zodpovedajú.

- Základným stavebným kameňom vzoriek sú samotné znaky z vyššie uvedeného zoznamu. Vzorke tvorenej jediným znakom zodpovedá reťazec tvorený dotýčným znakom a nič iné.
- Základná operácia so vzorkami je zretazenie. Keď napíšeme dve vzorky za seba, dostaneme novú vzorku. Tej zodpovedajú reťazce zložené z dvoch častí, pričom prvá zodpovedá prvej vzorke a druhá druhej. Zretaziť samozrejme môžeme ľubovoľne veľa vzoriek. Napr. vzorka „jablko“ zodpovedá len reťazec „jablko“.
- Vzorky môžeme uzatvárať do obyčajných zátvoriek. Napr. vzorka „(jablko)“ zodpovedá len reťazec „jablko“ a nič iné.
- Logický or (alebo) je označovaný symbolom „|“. Ak ním spojíme dve vzorky, dostaneme novú, ktorej zodpovedajú aj reťazce zodpovedajúce prvej, aj reťazce zodpovedajúce druhej vzorke. Napr. vzorka „jabl(k|c)k“ zodpovedajú reťazce „jablko“ a „jablcko“.
Zretazenie má vyššiu prioritu ako or. Napr. vzorka „aaa|bbb“ zodpovedajú len reťazce „aaa“ a „bbb“.
- Inú možnosť, ako dať vo vzorke na výber, predstavuje množina znakov, ktoré sa na danom mieste môžu vyskytnúť. Tú zapisujeme „[znaky]“. Napríklad vzorka „jablk[oa]“ zodpovedajú reťazce „jablko“ a „jablka“.
Množina všetkých dostupných znakov sa skrátene zapisuje „.“. Teda vzorka „j...a“ zodpovedá každý 6-znakový reťazec začínajúci na j a končiaci na a.
Zložitejšie množiny znakov môžeme zapísať pomocou rozsahov, napr. „[a-z0-9]“ je ľubovoľné malé písmeno alebo číslica. Ak popis množiny znakov začína znakom „^“, znamená to negáciu: danej množine zodpovedajú všetky znaky okrem vymenovaných. Napríklad vzorka „[a-z][^a-zA-Z]“ zodpovedajú všetky dvojznakové reťazce, ktorých prvý znak je malé písmeno a druhý znak nie je ani malé, ani veľké písmeno.
- Ak chceme, aby sa mohla časť vzorky v reťazci zopakovať, použijeme *kvantifikátor*. Základné kvantifikátory sú „?“ (nulakrát alebo jedenkrát) a „*“ (ľubovoľne veľakrát, vrátane nulakrát).

¹V skutočných regulárnych výrazoch môžeme samozrejme používať aj všetky ostatné znaky vrátane medzier. Naším obmedzením sa však vyhneme vysvetľovaniu mnohých technických detailov.



Napríklad vzorka „jablc?ko“ je ekvivalentná so vzorkou „jabl(k|ck)o“. Vzorke „pe*s“ zodpovedajú okrem iného reťazce „ps“, „pes“ a „peeeeees“. Vzorka „(ab)*c“ nezodpovedá reťazec „aabbc“, len reťazce „c“, „abc“, „ababc“, atď.

Pri opakovaní vzorky jej nemusí zakaždým zodpovedať ten istý reťazec. Napríklad vzorka „j.*a“ zodpovedá nielen reťazec „jxxxa“, ale aj reťazec „jahoda“.

Vzorka „<[^>]*>“ zodpovedá každý XML tag – reťazec začína znakom „<“, za ním nasleduje ľubovoľne veľa znakov iných od „>“ a za nimi nasleduje znak „>“.

Súťažná úloha

Táto úloha má niekoľko podúloh. V každej podúlohe budete mať slovne popísanú nejakú skupinu reťazcov. Vaším cieľom bude napísať vzorku, ktorej zodpovedajú všetky popísané reťazce a žiadne iné!

Podúloha A (1 bod). Napíšte vzorku, ktorej zodpovedajú iba reťazce „zaba“ a „zabka“.

Podúloha B (1 bod). Napíšte vzorku, ktorej zodpovedajú iba reťazce, v ktorých každý znak je „a“ a ktorých dĺžka je aspoň 5.

Podúloha C (1 bod). Napíšte vzorku, ktorej zodpovedajú iba reťazce, v ktorých každý znak je „a“ a ktorých dĺžka je najviac 15.

Podúloha D (3 body). Napíšte vzorku, ktorej zodpovedajú iba reťazce predstavujúce kladné prirodzené čísla deliteľné štvorkou. Zápis čísla nesmie začínať zbytočnými nulami. Vzorky by teda mali zodpovedať napríklad reťazce „4“, „12“ a „400“, ale nie reťazce „0“, „47“, „004“, ani „ahoj“.

Podúloha E (4 body). Napíšte vzorku, ktorej zodpovedajú iba reťazce v ktorých sa nenachádzajú tri „a“ za sebou. Vzorky by teda mali zodpovedať napríklad reťazce „bu“ a „aa7a“, ale nie reťazce „baaaaac“ a „papagaaaaj“.

Odvzdávanie riešení

Toto je teoretická úloha. Ku každej podúlohe napíšte ľubovoľnú jednu správnu vzorku a odôvodnenie, prečo táto vzorka spĺňa zadanie. (Pri jednobodových podúlohách by mala stačiť jedna veta.) Spísané riešenie potom vo formáte PDF odovzdajte pomocou webového rozhrania.

Pomôcka

Existuje množstvo online nástrojov, kde si viete vyskúšať prácu s regulárnymi výrazmi a testovať riešenia tejto úlohy. Medzi ne patrí napríklad aj <https://regex101.com/>. Ak na tejto stránke napíšete do okienka „regular expression“ svoju vzorku a do okienka „test string“ nejaký text, stránka vo vami zadanom texte vyhľadá podreťazec, ktorý zadanej vzorky zodpovedá. Vyskúšajte si napr. zadať vzorku „jabl(k|ck)o“ a text „mak jablko jahoda“. Ak si chcete vynútiť test, či celý zadaný text zodpovedá vašej vzorky, dopíšete na jej začiatok znak „^“ a na jej koniec znak „\$“. Tieto dva špeciálne znaky predstavujú začiatok a koniec celého textu, v ktorom hľadáme.

TRIDSIATY PRVÝ ROČNÍK OLYMPIÁDY V INFORMATIKE

Príprava úloh: Michal Anderle, Michal Forišek, Marián Hornák

Recenzia: Michal Forišek

Slovenská komisia Olympiády v informatike

Vydal: IUVENTA – Slovenský inštitút mládeže, Bratislava 2015