



### Priebeh celoštátneho kola

Celoštátne kolo 37. ročníka Olympiády v informatike, kategórie A, sa koná v dňoch 23.-26. 3. 2022. Na riešenie úloh druhého, praktického dňa majú súťažiaci 4,5 hodiny čistého času. Akékoľvek pomôcky okrem písacích potrieb (napr. knihy, výpisy programov, kalkulačky) sú zakázané.

### Čo má obsahovať riešenie úlohy?

- Skompilovateľný program v podporovanom programovacom jazyku. Ak sa váš program nepodarí na našom testovacom počítači skompilovať a spustiť, bude automaticky hodnotený 0 bodmi.

### Hodnotenie riešení druhého (praktického) dňa

Sú tri úlohy. Ku každej úlohe máme pripravených 10 sád testovacích vstupov. Sada vstupov pozostáva z jedného alebo viacerých testovacích vstupov. Za každú sadu vstupov, ktorej všetky vstupy (každý zvlášť) váš program správne vyrieši, získate jeden bod.

Testovanie na každom vstupe prebieha samostatne. Spustíme váš program a na štandardný vstup mu dáme konkrétne vstupné údaje. Hovoríme, že váš program daný vstup vyriešil, ak splní nasledujúce kritériá:

- Skončí skôr ako uplynie stanovený časový limit.
- Neprekročí stanovený pamäťový limit.
- Skončí korektne, nie chybou počas behu.
- Dáta, ktoré vypíše na štandardný výstup, tvoria korektný výstup, zodpovedajúci danému vstupu.
- Nebude používať žiadne funkcie zakázané kvôli bezpečnosti testovacieho systému.

Počas súťaže môžete priebežne odovzdávať svoje riešenia. Odovzdané riešenie bude otestované a dozviete sa svoj bodový zisk. (V prípade preťaženia testovača môžu organizátori obmedziť toto priebežné testovanie na vhodnú podmnožinu všetkých testovacích dát.)

Po ukončení súťaže zoberieme pre každú úlohu váš posledný odovzdaný program a ten otestujeme na všetkých testovacích vstupoch. Vaše výsledné body za úlohu budú body získané týmto programom.

Sady vstupov sú navrhované tak, aby každé korektné riešenie získalo nejaké body, bez ohľadu na to, ako pomalé je. Bližšie informácie o testovacích dátach nájdete na konci zadania každej úlohy.



### A-III-4 Hyperjazdec

*Jazdec* je klasická šachová figúrka. Pohybuje sa tak, že ho zdvihneme, posunieme o dve políčka niektorým smerom (v riadku alebo v stĺpci), potom o jedno políčko v kolmom smere a tam ho položíme.

*Hyperjazdec* sa pohybuje podobne ako jazdec, len rýchlejšie: najskôr ho posunieme niektorým smerom (v riadku alebo v stĺpci), potom ho posunieme o **iný počet políčok** v kolmom smere a tam ho položíme. V oboch smeroch sa hyperjazdec musí naozaj pohnúť. Najkratšie platné pohyby hyperjazdca teda zodpovedajú pohybom obyčajného jazdca.

Máme obriu šachovnicu s  $r$  riadkami a  $s$  stĺpcami. Na každom políčku je napísané jedno písmeno. Chceme po nej preskákať hyperjazdcom tak, aby postupne navštívil písmenká daného slova  $w$ . Zčať môžeme na ľubovoľnom políčku obsahujúcom prvé písmeno  $w$ , každým ďalším ťahom sa musíme presunúť na jeho nasledujúce písmeno.

Koľkými rôznymi spôsobmi sa to dá spraviť? (Dva spôsoby považujeme za rôzne, ak im zodpovedajú rôzne postupnosti navštívených políčok.)

Správna odpoveď môže byť veľmi veľká. Preto nám stačí, keď vypočítate, aký zvyšok dáva po delení  $10^9 + 7$ .

#### Formát vstupu a výstupu

V prvom riadku vstupu sú rozmery šachovnice:  $r$  a  $s$ . V druhom riadku je slovo  $w$ . Zvyšok vstupu tvorí popis písmen na šachovnici:  $r$  riadkov a v každom z nich  $s$  písmen. Môžete predpokladať, že všetky písmená na vstupe sú veľké písmená anglickej abecedy.

Na výstup vypíšete jeden riadok so správnou odpoveďou.

#### Obmedzenia a hodnotenie

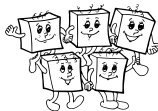
V prvej sade vstupov (1 bod) platí  $r, s, |w| \leq 5$ .

V druhej sade vstupov (2 body) platí  $r, s, |w| \leq 30$  a navyše v každom vstupe existuje dokopy nanajvýš  $10^5$  možností ako môže hyperjazdec preskákať nejaký prefix slova  $w$ .

V tretej sade vstupov (2 body) platí  $r, s, |w| \leq 40$ .

Vo štvrtej sade vstupov (2 body) platí  $r, s, |w| \leq 100$ .

V piatej sade vstupov (3 body) platí  $r, s \leq 800$  a  $|w| \leq 100$ .



### Príklady

vstup

```
3 4
OI
KIVI
MALO
OBAL
```

výstup

```
3
```

Z písmena *O* v ľavom dolnom rohu môžeme skočiť na ľubovoľné *I*. Z písmena *O* vpravo vieme skočiť len na jedno z dvoch *I*: to v druhom stĺpci.

vstup

```
3 5
BRAT
BANAN
NEMAL
SUPKU
```

výstup

```
0
```

Keďže na šachovnici nemáme žiadne *R* ani *T*, je zjavné, že po písmenách slova *BRAT* sa preskákať nedá.

vstup

```
3 5
UUUUUU
UUKUU
HESLO
UUPUU
```

výstup

```
868
```

Z písmena *U* v rohu šachovnice vieme skočiť na tri iné *U*. Z písmena *U*, ktoré nie je v rohu, vieme skočiť len na dve iné *U*: tie v protilahlých rohoch.

vstup

```
6 8
KSP
KSPPSKSP
PPSKSKSP
SKKSKSKP
PPPSKSKK
KKSKSKPP
PPPSKSKP
```

výstup

```
1276
```

Jedno prípustné riešenie je začať na *K* úplne vľavo hore, odtiaľ skočiť na *S* zhruba v strede dole a odtiaľ na *P* úplne vpravo hore.



### A-III-5 Kváder

Samko našiel na povale program, ktorý vypisuje obsah trojrozmerného poľa  $pole[0..a-1][0..b-1][0..c-1]$ . Každé jeho políčko vypíše práve raz. Samkov program vyzerá nasledovne:

```
for s = 0 to a+b+c-3:
  for i = 0 to a-1:
    for j = 0 to b-1:
      for k = 0 to c-1:
        if i+j+k == s:
          print( pole[i][j][k] )
```

#### Súťažná úloha

Dané sú rozmery poľa: čísla  $a, b, c$ . Ďalej je daných niekoľko otázok  $n_1, \dots, n_q$ . Pre každú otázku  $n_i$  zistite súradnice políčka, ktoré Samkov program vypíše ako  $n_i$ -te v poradí.

#### Formát vstupu a výstupu

V prvom riadku vstupu sú kladné celé čísla  $a, b, c$ . V druhom riadku je kladné celé číslo  $q$ : počet otázok. V treťom riadku sú kladné celé čísla  $n_1, \dots, n_q$ : jednotlivé otázky.

Pre každú otázku vypíšte jeden riadok tvaru „ $i j k$ “ s hľadanými súradnicami.

#### Obmedzenia a hodnotenie

Vo všetkých sádach vstupov platí  $q \leq 10^5$  a  $\forall i: 1 \leq n_i \leq (a \cdot b \cdot c) \leq 10^{18}$ .

(Všimnite si, že na hodnoty  $a, b, c$  aj  $n_i$  môže byť potrebné použiť 64-bitové celočíselné premenné.)

Je desať sád vstupov, za každú je 1 bod. Dodatočné obmedzenia pre jednotlivé sady sú v tabuľke.

číslo sady	1	2	3	4	5	6	7	8	9	10
max $a$	20	180	2500	1	1	$10^6$	$10^6$	$10^6$	$10^6$	
max $b, \max c$	20	180	2500	$10^6$	$10^9$			$10^6$	$10^6$	
max $q$	20	1	1			100		1		
iné						kocka	kocka			

V sádach 6 a 7 („kocka“) v každom vstupe platí  $a = b = c$ .

#### Príklady

vstup	výstup
<pre>3 3 3 3 1 2 10</pre>	<pre>0 0 0 0 0 1 2 0 0</pre>

Máme pole rozmerov  $3 \times 3 \times 3$ . Prvých 10 políčok, ktoré program zo zadania vypíše:  $(0,0,0)$ ,  $(0,0,1)$ ,  $(0,1,0)$ ,  $(1,0,0)$ ,  $(0,0,2)$ ,  $(0,1,1)$ ,  $(0,2,0)$ ,  $(1,0,1)$ ,  $(1,1,0)$  a  $(2,0,0)$ .

vstup	výstup
<pre>13 2 5 4 70 130 24 108</pre>	<pre>6 1 2 12 1 4 3 1 0 9 1 3</pre>



### A-III-6 Firma

Firma MixérLabs má striktnu binárnu hierarchiu hĺbky  $k$ . Každý zamestnanec firmy má nejakú celočíselnú úroveň. Úrovne majú hodnoty od 0 (stážisti) po  $k$  (šéfka celej firmy). Šéfka je práve jedna. Stážisti nemajú žiadnych priamych podriadených. Každý iný zamestnanec (vrátane šéfkky) má práve dvoch priamych podriadených. Každý zamestnanec (okrem šéfkky) má práve jedného priameho nadriadeného.

Po firme si ľudia posielajú správy. Správy vždy putujú len cez priamych nad- a podriadených, a to najkratšou možnou cestou. Inými slovami: vždy, keď má zamestnanec  $x$  správu určenú pre iného zamestnanca  $y$ , postupuje nasledovne: ak  $y$  je podriadeným  $x$ , prepošle  $x$  správu tomu svojmu *priamemu* podriadenému, v ktorého oddelení  $y$  pracuje. Vo všetkých ostatných prípadoch prepošle  $x$  správu svojmu *priamemu* nadriadenému. Hovoríme, že zamestnanec  $b$  je v hierarchii medzi zamestnancami  $a$  a  $c$ , ak správa od  $a$  určená pre  $c$  bude prechádzať cez  $b$ .

Zamestnanci vo firme majú svoje navzájom rôzne identifikačné čísla od 1 po  $n = 2^{k+1} - 1$ . Chceli by sme o celej firme zistiť, kto má aké číslo.

Hanka pozná celú firmu. Je ale ochotná nám len odpovedať na otázky o tom, ako po firme chodia správy. Viete pomocou nich zrekonštruovať celú hierarchiu firmy?

#### Interakcia

Vašou úlohou je napísať program, ktorý bude komunikovať s „Hankou“ (naším programom). Váš program bude striedavo čítať údaje zo štandardného vstupu a písať ich na štandardný výstup.

Na začiatku behu programu prečítajte zo vstupu dve čísla  $k$  a  $\ell$ : hĺbku hierarchie vo firme a limit na počet otázok, ktoré je Hanka ochotná zodpovedať.

Následne môžete klásť otázky. Každú otázku položíte tak, že vypíšete riadok tvaru „0 a b c“ a následne prečítate odpoveď na ňu: jedno číslo, ktoré bude mať hodnotu 1 ak  $b$  je v hierarchii medzi  $a$  a  $c$ , resp. hodnotu 0 ak nie. Po každej otázke je potrebné explicitne *flushnúť* štandardný výstup vášho programu, aby sa ním vypísané dáta ihneď dostali ku Hanke. Toto viete napr. v C++ spraviť príkazom `cout << flush;` (`iostream`), resp. `fflush(stdout);` (`cstdio`).

Keď už si ste istí hierarchiou firmy, váš program by mal vypísať  $n + 1$  riadkov a skončiť. V prvom z týchto riadkov by malo byť číslo 1. Následne postupne pre každého zamestnanca v poradí od čísla 1 po  $n$  vypíšete jeden riadok v ktorom bude najskôr počet jeho priamych zamestnancov a potom ich čísla (v ľubovoľnom poradí).

#### Obmedzenia a hodnotenie

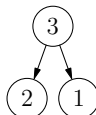
Program, s ktorým budete interagovať („Hanka“) sa nesnaží byť zákerný. V každom teste je celá hierarchia firmy vopred zvolená a Hanka vám naozaj len odpovedá na otázky o nej. Môžete predpokladať, že na každú otázku vie odpovedať v zanedbateľnom čase.

Vo všetkých sadách platí  $1 \leq k$ . V každej sade majú všetky testy tú istú hodnotu  $\ell$ .

sada	body	max $k$	$\ell$
1	3	5	250 000
2	1	5	50 000
3	2	6	100 000
4	3	10	800 000
5	1	11	500 000

#### Príklady

Na obrázku je štruktúra firmy s  $k = 1$ . (Táto firma je s  $\ell = 250\,000$  použitá ako prvý ukázkový vstup.)

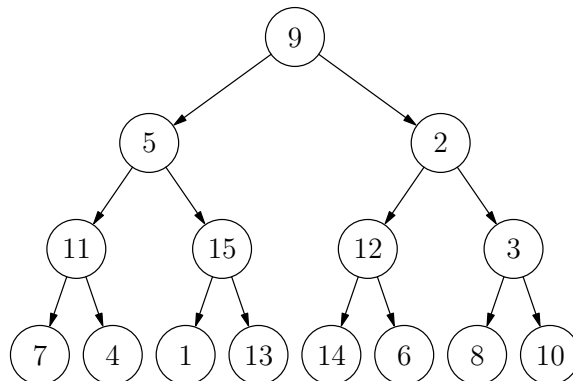




Nasleduje príklad kompletnej komunikácie Hanky s vaším programom. Čítať ho treba zhora dole. Prázdne riadky sú tam len kvôli lepšej čitateľnosti, v skutočnej komunikácii sa neobjavia.

Hanka	váš program	komentár
1 250000		k=1, l=250000
	0 1 2 3	leží 2 medzi 1 a 3?
0		nie
	0 1 3 2	leží 3 medzi 1 a 2?
1		áno
	1	už vieme riešenie
	0	zamestnanec 1 nemá podriadených
	0	zamestnanec 2 nemá podriadených
	2 1 2	zamestnanec 3 má dvoch priamych podriadených

Na ďalšom obrázku je príklad väčšej hierarchie firmy s  $k = 3$ . (Táto firma je s  $\ell = 250\,000$  použitá ako druhý ukázkový vstup.)



Pre túto firmu by odpoveď „áno“ dostali napríklad otázky „4 5 9“, „3 2 11“ a „12 12 6“. Odpoveď „nie“ by dostali napríklad otázky „7 13 2“, „12 11 6“ a „3 9 8“.

### Ukázkový testovač

V systéme na odovzdávanie riešení bude k dispozícii ukázková implementácia „Hanky“, ktorú môžete použiť pri testovaní svojich programov. Tento program najskôr prečíta zo súboru `firma.in` popis firmy a potom vyššie popísaným spôsobom komunikuje pomocou svojho štandardného vstupu a výstupu.

Súbor `firma.in` má obsahovať dva riadky. V prvom sú čísla  $k$  a  $\ell$ . V druhom je  $n = 2^{k+1} - 1$  čísel: postupne pre každého zamestnanca (od 1 po  $n$ ) číslo jeho priameho nadriadeného, resp. 0 ak ide o šéfkku.

Pre oba vyššie uvedené príklady firmy si môžete stiahnuť súbory, ktoré ich v tomto formáte popisujú.