

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.

Odovzdávanie 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. 11. 2023**, riešenia kategórie B do **30. 11. 2023**.

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).

Najlepšie štyri riešiteľky kategórie A (podľa výberového sústreďenia alebo poradia najvyššieho kola, z ktorého ďalej postúpia menej ako štyri) postúpia na Medzinárodnú dievčenskú olympiádu v informatike (EGOI).

Ako majú vyzerat' 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 jednom z podporovaných jazykov (napr. C++, Pascal alebo Java). 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 vyzerat' 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 efektívite 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 stručný 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 NIVAM v tesnej súčinnosti so Slovenskou komisiou OI.



A-I-1 Potrubná pošta

Toto je **praktická úloha**. Pomocou webového rozhrania odovzdajte **funkčný, odladený program**.

V budove úradu pre zbytočné otázky je dokopy n kancelárií. Majú na dverách čísla od 0 po $n - 1$. V každej kancelárii sedí jeden úradník, ktorý sa špecializuje na zbytočné otázky nejakého konkrétneho typu.

V každej miestnosti je terminál pre potrubnú poštu. Každý terminál má na sebe k koncoviek pre potrubia. Keď dve takéto koncovky prepojíme potrubím, vedú si potom prepojené kancelárie preposielať kapsuly so správami. To je veľmi praktické – umožní to úradníkom konzultovať s kolegami, ak dostanú klienta s neznámou témou zbytočnej otázky. Každé potrubie sa dá používať oboma smermi.

Niektoré koncovky už sú prepojené potrubiami. Týchto už existujúcich potrubí je p .

Hovoríme, že dve kancelárie spolu vedú komunikovať, ak existuje spôsob, ako dostať poštu potrubiami z jednej z nich do druhej. Toto nemusí ísť priamo: správa môže postupne prejsť cez viacero potrubí a viacero kancelárií, v ktorých ju vždy tamojší úradník prepošle ďalej iným potrubím.

Žiadne z už existujúcich potrubí nie je zbytočné. Inými slovami, pre každé z týchto potrubí platí, že keby sme ho odstránili, kancelárie, ktoré spája, už spolu nebudú vedieť komunikovať (ani nepriamo).

Úrad práve dostal nového riaditeľa. Ten sa rozhodol, že keďže je úplne zbytočné, aby úradníci medzi sebou vedeli komunikovať, je v duchu poslania úradu, aby mohli spolu komunikovať úplne všetci. Riaditeľ preto rozhodol, že je treba pridať nejaké nové potrubia – a to tak, aby po ich pridaní už vedeli komunikovať každé dve kancelárie.

Súťažná úloha

Tvojou úlohou je pridať nové potrubia tak, aby vo výslednej sieti potrubí vedela každá kancelária komunikovať s každou inou. Nové potrubia môžeš pridávať úplne ľubovoľne – niektoré z nich môžu byť aj zbytočné. Musíš si len dať pozor na to, že v každej miestnosti máš k dispozícii len obmedzený počet koncoviek na potrubia. (Ak pridáš zbytočné potrubie, ktoré má oba konce v tej istej kancelárii, spotrebuješ tým v nej dve koncovky.)

Formát vstupu a výstupu

V prvom riadku vstupu sú tri medzerou oddelené celé čísla: n , k a p .

Zvyšok vstupu tvorí p riadkov popisujúcich už existujúce potrubia. V každom z týchto riadkov sú dve celé čísla: čísla dvoch kancelárií prepojených potrubím. Je zaručené, že žiadne potrubie na vstupe nie je zbytočné a že v každej miestnosti je použitých najviac k koncoviek.

Ak žiadne riešenie neexistuje, vypíš na výstup jeden riadok a v ňom číslo -1 .

Ak riešenie existuje, ľubovoľne jedno zostroj a vypíš v nasledovnom formáte: v prvom riadku vypíš počet q nových potrubí, ktoré chceš pridať, a následne vypíš q riadkov popisujúcich tieto potrubia. V každom z týchto riadkov vypíš čísla dvoch kancelárií, ktoré chceš prepojiť.

Ľubovoľné platné riešenie s $q \leq 500\,000$ bude akceptované. (Pre každý vstup, pre ktorý existuje riešenie, existujú riešenia s výrazne menším q .) Nové potrubia môžeš vypisovať v ľubovoľnom poradí. Ich počet nemusí byť minimálny.

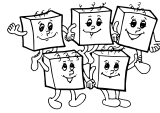
Obmedzenia a hodnotenie

Vstupy sú rozdelené do piatich sád. Za každú sadu, ktorú tvoje riešenie celú správne vyrieši, dostaneš príslušný počet bodov.

V každom vstupe platí $1 \leq n \leq 100\,000$, $0 \leq k \leq 100\,000$ a $0 \leq p \leq n - 1$.

V každej sade platia iné dodatočné obmedzenia pre veľkosť vstupu. Nájdeš ich v nasledujúcej tabuľke.

číslo sady	1	2	3	4	5
body	1	3	2	2	2
obmedzenia	$n = 5$	$2 \leq n \leq 10$	$2 \leq n \leq 100$	$2 \leq n \leq 3\,000$	$1 \leq n \leq 100\,000$
	$k = 4$	$2 \leq k \leq 10$	$1 \leq k \leq 100$	$2 \leq k \leq 3\,000$	$0 \leq k \leq 100\,000$



Príklady

vstup

```
5 2 2
0 3
1 4
```

výstup

```
3
0 2
2 1
4 3
```

Vstup popisuje päť kancelárií. V každej sú dve koncovky na potrubia. Máme dve už existujúce potrubia: jedno medzi kancelárkami 0 a 3, druhé medzi 1 a 4.

Vo vyššie ukázanom výstupe sme pridali tri nové potrubia, čím sme vyrobili „kruh“ potrubí vedúci cez všetky kancelárie. Tým sú zjavne všetky prepojené.

Existuje aj veľa iných platných riešení. Napr. ak by sme vynechali ľubovoľné jedno z týchto troch potrubí, ešte stále by sme mali platné riešenie.

vstup

```
6 4 4
0 1
1 2
3 4
4 5
```

výstup

```
2
1 4
5 5
```

V tomto príklade stačí na prepojenie všetkých kancelárií pridať jedno nové potrubie, ktoré bude mať na jednom konci niektorú z kancelárií 0, 1, 2 a na druhom konci niektorú z 3, 4, 5.

Na hociktoré z týchto potrubí máme v miestnostiach dosť voľných koncoviek, každé z nich teda predstavuje platné riešenie.

V našom príklade výstupe sme pridali potrubie 1 – 4 a okrem neho ešte jedno zbytočné potrubie, ktoré má oba konce v kancelárii 5. Toto potrubie ničomu nepomôže, ale ani nič nepokazí.

vstup

```
9 1 1
4 7
```

výstup

```
-1
```



A-I-2 Farebný plot

Toto je **praktická úloha**. Pomocou webového rozhrania odovzdajte **funkčný, odladený program**.

Nedávno sme dostali oznam, že do nášho mestečka príde na návštevu pani prezidentka. Zavládla mierna panika: čo jej tu ukážeme? Nakoniec sa všetci zhodli na tom, že najzaujímavejšia vec, čo tu máme, je krásny dlhý a farebný plot. Pôjdeme ju previesť okolo neho. Hm, ale možno by bolo lepšie ten plot predtým premaľovať, nech vyzerá reprezentatívnejšie.

V celom mestečku si jediným vlastníkom štetca, preto niet divu, že ťa zvolili do jednočlenného výboru na revitalizáciu plota.

Súťažná úloha

Existuje 10^9 rôznych farieb. Majú priradené čísla od 0 po $10^9 - 1$.

Plot je tvorený n blokmi. Tie sú za radom očíslované od 0 do $n - 1$. Blok číslo i má momentálne farbu f_i .

Do návštevy prezidentky ostáva d dní. Za každý deň stíhaš premaľovať presne jeden blok plota. A aby to nevyzeralo, že prípravu flákaš, každý deň si **musíš vybrať** jeden blok plota a premaľovať ho – teda daný blok **musí zmeniť farbu** z tej, ktorú práve mal, na nejakú inú.

Chceš dosiahnuť, aby pri návšteve prezidentky bol celý plot jednej farby (je jedno ktorej). Ak to nejde, chceš aspoň dosiahnuť, aby bol na plote čo najdlhší súvislý úsek ktorý má celý jednu farbu.

Formát vstupu a výstupu

V prvom riadku vstupu je číslo $t \in \{1, 2\}$ udávajúce typ testu. V druhom riadku vstupu sú celé čísla n a d . V treťom riadku vstupu je n medzerami oddelených celých čísel f_0, \dots, f_{n-1} : začiatkové farby jednotlivých blokov plota.

Prvý riadok výstupu má obsahovať dĺžku najdlhšieho jednofarebného úseku, ktorý vieš vyrobiť.

Pre testy s $t = 1$ je dôležitý len tento riadok. Za ním môže vo výstupe nasledovať čokoľvek.

Pre testy s $t = 2$ je potrebné následne vypísať jeden konkrétny postup maľovania. Tento má tvoriť presne d ďalších riadkov. V j -tom z nich majú byť dve celé čísla: poradové číslo bloku b_j , ktorý chceš prefarbiť v j -ty deň, a číslo g_j novej farby pre tento blok. (Pripomíname, že nová farba sa vždy musí líšiť od aktuálnej farby daného bloku.)

Vypísaná postupnosť akcií musí viesť k tomu, že po poslednej z nich bude niekde na plote najdlhší možný jednofarebný úsek blokov. Ak existuje viacero optimálnych postupností akcií, môžeš vypísať ľubovoľnú z nich.

Obmedzenia a hodnotenie

Vo všetkých vstupoch platí $1 \leq n \leq 250\,000$ a $0 \leq d \leq 250\,000$.

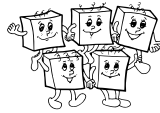
Testovacie vstupy sú rozdelené do desiatich sád v ktorých platia rôzne ďalšie obmedzenia. Za každú sadu, ktorej všetky vstupy tvoj program správne a dostatočne efektívne vyrieši, dostaneš bod.

Sady s číslom 1 až 5 obsahujú len vstupy s $t = 1$. Sady s číslom 6 až 10 obsahujú presne tie isté vstupy ale s $t = 2$. Dodatočné obmedzenia, ktoré platia v jednotlivých sádach, uvádzame v nasledujúcej tabuľke.

číslo sady	obmedzenia
1, 6	malé čísla farieb, $d = 0$
2, 7	malé čísla farieb, $2 \leq n \leq 100$, $1 \leq d \leq 100$
3, 8	malé čísla farieb, $2 \leq n \leq 5\,000$, $1 \leq d \leq 5\,000$
4, 9	malé čísla farieb, primálo času, $2 \leq n$
5, 10	

Obmedzenie „malé čísla farieb“ znamená, že všetky farby použité vo vstupe majú čísla menšie ako 250 000. (Tvoje riešenie stále môže používať aj farby mimo tohto rozsahu.)

Obmedzenie „primálo času“ znamená, že v každom vstupe tejto sady je primálo času na to natrieť celý plot jednou farbou.



Príklady

vstup

```
2
5 2
7 3 7 7 9
```

výstup

```
5
4 7
1 7
```

Jediné optimálne riešenie je prefarbiť celý plot na farbu 7. Na to treba prefarbiť bloky s číslom 1 a 4. Tieto dve akcie môžeme spraviť v ľubovoľnom poradí.

vstup

```
2
5 1
10 20 30 40 50
```

výstup

```
2
1 30
```

Najlepšie, čo vieme tu dosiahnuť, je vyrobiť dva bloky rovnakej farby vedľa seba. Je veľa spôsobov, ako toto dosiahnuť. Po tom z ukázkového výstupu budú mať bloky 1 a 2 oba farbu číslo 30.

vstup

```
1
5 3
7 7 7 7 7
```

výstup

```
5
trikrat prefarbim
ten isty blok
```

Keďže ide o vstup s $t = 1$, dôležitý je len prvý riadok výstupu. Za ním môže nasledovať čokoľvek.



A-I-3 Policajti a zlodej

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

Peter a Zuzka sa hrajú jednoduchú stolnú hru.

Jej hrací plán je obdĺžnikového tvaru. Je rozdelený na políčka, ktoré tvoria mriežku s r riadkami a s stĺpcami. Riadky sú očíslované od 0 po $r - 1$ zhora dole, stĺpce od 0 po $s - 1$ zľava doprava. Na hracom pláne sú tri figúrky: dvaja policajti a jeden zlodej. Peter posúva po pláne policajtov, Zuzka zase zlodeja.

Hráči ťahajú striedavo, Peter začína. Hráč, ktorý je na ťahu, si pre každú svoju figúrku vyberie jeden z piatich možných pohybov: buď ju nechá na mieste alebo ju pohne na jedno zo štyroch susedných políčok. Pohnúť figúrku mimo plánu je zakázané. Na každom políčku môže stáť ľubovoľne veľa figúrok naraz.

Peter hru vyhrá, ak niekedy aspoň jeden z policajtov príde na políčko, kde práve stojí zlodej. Zuzka hru vyhrá, ak to nikdy nedopustí – teda ak vie zlodejom hýbať tak, aby hra bežala do nekonečna.

Súťažná úloha

Napiš program, ktorý bude hrať hru za Petra. Po spustení by mal najskôr načítať zo vstupu rozmery hracieho plánu a súradnice všetkých figúrok. Potom by mal v cykle striedavo vypísať na výstup nasledujúci ťah, ktorý má spraviť Peter, a následne načítať zo vstupu popis ťahu, ktorý spravila Zuzka. (Detaily formátu vstupu a výstupu si zvoľ, ako uznáš za vhodné.)

Ak existuje v začiatkovej pozícii vyhrávajúca stratégia, tvoj program musí eventuálne hru vyhrať. Ak nie, musí korektne hrať do nekonečna.

Obmedzenia a hodnotenie

Plných 10 bodov môžu získať riešenia efektívne pre $r, s \leq 100\,000$. Takéto riešenie by malo obsahovať dostatočne podrobný popis stratégie pre Petra (program alebo pseudokód + slovný popis) a taktiež dostatočný **dôkaz správnosti** implementovanej stratégie.

Nanajvýš 6 bodov môžu získať riešenia efektívne pre $r, s \leq 8$.

Čiastočné body môžete získať aj za riešenie, ktoré len načíta popis konkrétnej začiatkovej pozície a zistí, ktorý z hráčov v nej má vyhrávajúcu stratégiu.

Príklad priebehu hry

Hrací plán má $r = 3$ riadky a $s = 4$ stĺpce. Policajti začínajú na pozíciách $(2, 0)$ a $(0, 1)$, zlodej na pozícii $(1, 2)$. Situácia teda vyzerá nasledovne:

```
.P..  
..Z.  
P...
```

Z tejto začiatkovej pozície mohla prebehnúť napríklad takáto partia hry:

- Peter: Pohnem prvého policajta doprava na $(2, 1)$ a druhého tiež doprava na $(0, 2)$.
- Zuzka: Jediný pohyb, po ktorom ma okamžite nechytí, je ten, čo spravím: pohnem zlodeja doprava na $(1, 3)$.
- Peter: Pohnem prvého policajta doprava na $(2, 2)$ a druhého dodola na $(1, 2)$.
- Zuzka: Mohla by som hru ešte o jeden môj ťah natiahnuť tým, že teraz pohnem zlodeja dohora, ale to by bol tento príklad zbytočne dlhý. Radšej už nechám zlodeja na mieste na políčku $(1, 3)$.
- Peter: Prvého policajta nechám na mieste a druhého pohnem na políčko $(1, 3)$, čím som chytil zlodeja.



A-I-4 O Vekslábotovi a Pokladničke

Toto je teoretická úloha. Pomocou webového rozhrania odovzdajte súbor vo formáte PDF, obsahujúci riešenie, spĺňajúce požiadavky uvedené v pravidlách. K tejto úlohe patrí študijný text uvedený na nasledujúcich stranách. Odporúčame najskôr prečítať ten a až potom sa vrátiť k samotným súťažným úlohám.

Ako pomôcku k riešeniu úlohy sme zverejnili interpreter programov pre Pokladničku. Aspoň do termínu odovzdania domáceho kola ho nájdete tu: <https://oi.sk/apps/pokladnicka/>.

Jednotlivé podúlohy sú hodnotené nezávisle, môžete ich riešiť v ľubovoľnom poradí. Pri hodnotení úloh domáceho kola budeme prihliadať len na korektnosť vašich programov – na ich efektívnosti (časovej zložitosti) nám nebude záležať.

Vo všetkých troch podúlohách je na začiatku v Pokladničke $c > 0$ červených žetónov, $m > 0$ modrých žetónov a nič iné.

Podúloha A (3 body).

Napište program, po skončení ktorého bude v Pokladničke opäť $c + m$ žetónov, ale inej farby: ak bolo na začiatku viac červených ako modrých, majú byť všetky žetóny ružové, inak majú byť všetky žetóny tyrkysové.

Podúloha B (3 body).

Napište program, po ktorého skončení bude v Pokladničke presne c červených, m modrých a $c + m$ fialových žetónov. (Navyše v nej môže byť aj ľubovoľne veľa žetónov iných farieb.)

Upozorňujeme, že váš program musí skončiť – v záverečnej situácii sa už nesmie dať vykonať žiadna z jeho inštrukcií.

Podúloha C (4 body).

Napište program, po ktorého skončení bude v pokladničke presne $c \cdot m$ fialových žetónov a nič iné.

(Ak vyrobíte presne $c \cdot m$ fialových žetónov ale ešte vám ostanú aj žetóny iných farieb, dostanete 3 body. Ak navyše potrebujete mať pred začiatkom výpočtu okrem červených a modrých žetónov aj vami presne špecifikovanú sadu žetónov iných farieb, dostanete 2 body.)



Študijný text: O Vekslákbotovi a Pokladničke

Za 111 horami a 111 dolinami leží jedna prazvláštna krajina. V tejto krajine žijú sami roboti a ako platidlá používajú žetóny všetkých možných farieb od výmyslu sveta. V domčeku na úpätí ďalšej hory si tam spolu nažívajú dvaja hrdinovia nášho príbehu: roboti Vekslákbót a Pokladnička.

Ako možno tušíte z jej mena, Pokladnička v sebe rada uskladňuje všetky možné žetóny. Ako zrejme netušíte z jej mena, Pokladnička tiež veľmi rada vykonáva rôzne programy. A ako ste si po prečítaní predchádzajúcej vety úplne istí (koniec koncov, toto je študijný text Olympiády v informatike a nie len taká rozprávka), práve tieto programy pre ňu budete písať vy ako riešenia súťažných úloh.

Vekslákbót je majster vo vymieňaní žetónov: či už chcete vymeniť dva červené za tri modré alebo žltý a čierny za miliónpäť sivých, Vekslákbót určite pozná robota, ktorý pozná robota, ktorý s ním práve takú výmenu rád spraví. Vekslákbót má veľmi rád Pokladničku, a tak keď ho ona o hocijakú výmenu poprosí, okamžite (alebo, ako hovoríme my, v konštantnom čase) jej ju zabezpečí.

Podme sa teda pozrieť na to, ako vlastne budú vyzerat programy pre našu Pokladničku.

Základy: vstup, výstup, program

„Vstupom“ pre Pokladničku budú jednoducho žetóny, ktoré má na začiatku v sebe. Neexistuje nijaký výstup. V zadaní rôznych úloh budeme rôzne definovať ciele, ktoré majú vaše programy dosiahnuť.

Program pre Pokladničku sa skladá z dvoch častí: *obmedzení*, ktoré musí dodržať, a *pokynov*, ktoré má vykonávať.

Obmedzenia

Prvou časťou programu pre Pokladničku je **konečná množina** obmedzení. (Táto množina môže byť aj prázdna.) Obmedzenia pre Pokladničku hovoria, najviac koľko žetónov konkrétnej farby, prípadne kombinácií žetónov rôznych farieb, smie mať naraz v sebe.

Formálne, obmedzenia sú lineárne nerovnosti nasledovného tvaru:

$$k_1 \cdot \text{farba}_1 + k_2 \cdot \text{farba}_2 + \dots + k_n \cdot \text{farba}_n \leq \text{limit}$$

pričom všetky koeficienty k_i aj limit sú konkrétne kladné celé čísla, zatiaľ čo farba_i sú premenné označujúce počet žetónov príslušnej farby v Pokladničke.

Príkladmi obmedzení sú napríklad nerovnosti „modrá ≤ 7 “ a „modrá + 2 · červená ≤ 3 “.

Ak sa nejakej farby netýka žiadne obmedzenie, môže byť v Pokladničke ľubovoľne veľa žetónov tejto farby.

Jedna inštrukcia

Každá inštrukcia pre Pokladničku má nasledovný tvar:

$$k_1 \cdot \text{farba}_1, \dots, k_n \cdot \text{farba}_n \rightarrow \ell_1 \cdot \text{farba}_{n+1}, \dots, \ell_m \cdot \text{farba}_{n+m}$$

Všetko naľavo od \rightarrow budeme volať ľavá strana inštrukcie, všetko napravo zase pravá strana.

Všetky k_i aj ℓ_j musia byť kladné celé čísla. V rámci ľavej aj v rámci pravej strany inštrukcie musia byť všetky použité farby rôzne. Je povolené použiť tú istú farbu na oboch stranách inštrukcie. Je povolené mať $n = 0$ alebo $m = 0$, teda inštrukciu, ktorá má niektorú stranu prázdnu.¹

Príklady inštrukcií:

- 2 červená \rightarrow 3 modrá
- 1 žltá, 1 čierna \rightarrow 1 000 005 sivá
- 3 červená \rightarrow 333 červená, 334 cyklámenová, 335 purpurová
- 2 zelená \rightarrow \emptyset

¹Povolené je mať prázdne aj obe strany inštrukcie, ale ako sa čoskoro dozvieme, program, v ktorom takúto inštrukciu použijeme, zaručene nikdy neskončí.



Posledná inštrukcia z príkladu má prázdnu pravú stranu. Na zápis prázdnej strany inštrukcie budeme používať symbol prázdnej množiny, aby bolo jasné, že je prázdna úmyselne.²

V našich príkladoch budeme používať reálne názvy farieb. Vo svojich programoch môžete ako názvy farieb používať aj ľubovoľné iné alfanumerické reťazce. Je tiež OK pri zápise vynechať koeficient 1. Druhú z vyššie uvedených inštrukcií teda môžete zapísať aj „žltá, čierna \rightarrow 1 000 005 sivá“.

Pokladnička vykoná inštrukciu tak, že zo seba vyberie sadu žetónov na ľavej strane inštrukcie, dá ich Vekslákbotovi a poprosí ho, aby jej za ne priniesol sadu žetónov z pravej strany. Vekslákbot to samozrejme promptne zabezpečí a Pokladnička do seba vloží žetóny, ktoré jej priniesol.

Ak v sebe Pokladnička nemá všetky žetóny, ktoré si vyžaduje ľavá strana inštrukcie, tak danú inštrukciu v danej chvíli vykonať nevie. Ak by napríklad mala len dva červené žetóny, nevie vykonať inštrukciu „3 červené \rightarrow 333 červených“.

Pokladnička tiež nevie vykonať inštrukciu, ak by po jej vykonaní bolo porušené hociktoré z obmedzení.

Pokyny

Pokyny pre Pokladničku tvorí **konečná postupnosť** inštrukcií vyššie uvedeného typu. Zdôrazňujeme, že ide o postupnosť, a teda **záleží** na poradí inštrukcií.

Vykonávanie programu

Program sa vykonáva v krokoch. V každom kroku Pokladnička začne čítať pokyny od začiatku a číta ich, až kým nenájde prvú inštrukciu, ktorú vie momentálne vykonať. Tú inštrukciu vykoná. (Na zvyšok pokynov sa v tomto kroku už ani nepozrie. V ďalšom kroku začne znova čítať postupnosť inštrukcií od začiatku.)

Vykonávanie programu skončí, keď sa už žiadna inštrukcia v pokynoch nedá vykonať.

Príklad #1: viac červených

Úloha: Na začiatku sú v Pokladničke nejaké červené a nejaké modré žetóny. Napíšte program, po ktorého skončení bude v Pokladničke práve jeden zlatý žetón a nič iné, ak bolo červených žetónov viac ako modrých. Vo všetkých ostatných prípadoch musí Pokladnička skončiť úplne prázdna.

Riešenie 1: bez obmedzení

Nebudeme mať žiadne obmedzenia. Pokyny budú vyzeráť nasledovne:

1. červená, modrá $\rightarrow \emptyset$
2. červená, zlatá \rightarrow zlatá
3. červená \rightarrow zlatá
4. modrá $\rightarrow \emptyset$

Pri vykonávaní tohto programu bude Pokladnička najskôr používať inštrukciu 1, kým to ide. Keď to prestane ísť, má už v sebe buď len červené alebo len modré žetóny. Ak sú len modré, jediná inštrukcia, ktorá sa dá používať, je inštrukcia 4. Jej opakovaným použitím sa Pokladnička vyprázdni a program skončí.

Ak sú po skončení používania inštrukcie 1 v Pokladničke len červené žetóny, bude to o čosi komplikovanejšie. Jediná inštrukcia, ktoré sa dá v tejto situácii použiť, je inštrukcia 3: vymeníme jeden červený žetón za jeden zlatý. Od tejto chvíle sa už inštrukcia 3 nepoužije, a to preto, že už sa dá používať inštrukcia 2. Tou sa Pokladnička postupne zbaví zostávajúcich červených žetónov. Akonáhle v nej ostane len samotný zlatý žetón, už sa nedá použiť žiadna inštrukcia, a teda výpočet končí.

Riešenie 2: s obmedzením

Budeme mať jedno obmedzenie:

- zlatá ≤ 1

²Formálne môžeme povedať, že aj pravá aj ľavá strana každej inštrukcie obsahuje nejakú multimnožinu farieb. Prázdna strana inštrukcie je teda naozaj prázdna multimnožina.



Pokyny budú vyzerat nasledovne:

1. červená, modrá $\rightarrow \emptyset$
2. červená \rightarrow zlatá
3. červená $\rightarrow \emptyset$
4. modrá $\rightarrow \emptyset$

Tentokrát sa v situácii, kedy sú v Pokladničke len samé červené žetóny, najskôr raz vykoná inštrukcia 2 (jeden vymeníme za zlatý) a potom sa už bude používať inštrukcia 3, až kým sa všetky červené neminú. Obmedzenie, ktoré sme si zvolili, totiž bráni Pokladničke opakovane využiť inštrukciu 2.

Príklad #2: súčet

Úloha: Na začiatku je v Pokladničke $c > 0$ červených žetónov, $m > 0$ modrých a jeden zelený. Napíšte program, po ktorého skončení bude v Pokladničke presne c červených, m modrých a $c + m$ fialových žetónov.

Riešenie. Keby sme len chceli dostať $c + m$ fialových žetónov, stačilo by vymeniť všetky červené aj všetky modré za fialové „s kurzom jedna k jednej“. Ako ale nestratiť pôvodné žetóny?

Naše riešenie nebude mať žiadne obmedzenia a bude mať nasledujúce inštrukcie:

1. červená, zelená \rightarrow tmavočervená, zelená
2. modrá, zelená \rightarrow tmavomodrá, zelená
3. zelená \rightarrow žltá
4. tmavočervená, žltá \rightarrow červená, fialová, žltá
5. tmavomodrá, žltá \rightarrow modrá, fialová, žltá
6. žltá $\rightarrow \emptyset$

Všimnite si, ako náš program používa prítomnosť zeleného a žltého žetónu v Pokladničke: pomocou nich vieme rozlíšiť, či ešte premieňame pôvodné červené a modré žetóny alebo či už naspäť vznikajú nové.