



## 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: <https://oi.sk/>.

Riešenia kategórie A je potrebné odovzdať najneskôr **15. 11. 2023**.

Riešenia kategórie B je potrebné odovzdať najneskôr **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 sústredenie. Podľa jeho výsledkov SK OI vyberie družstvá pre Medzinárodnú olympiádu v informatike (IOI) a Stredoeurópsku olympiádu v informatike (CEOI).

### 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 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 NIVAM v tesnej súčinnosti so Slovenskou komisiou OI.



## B-I-1 Volebný bazén

*Toto je praktická úloha. Riešiť sa dá v ľubovoľnom programovacom jazyku, odovzdávajú sa len správne riešenia pre vopred pripravené testovacie dáta. Detailnejšie pokyny k odovzdávaniu riešení sú uvedené nižšie.*

V Kocúrkove sa blížia voľby, a tak kandidáti dumajú, ako by oslovili svojich voličov. Hlavnou predvolebnou témou sa stala diskusia o výstavbe nového mestského bazéna. Obyvatelia Kocúrkova sú však nároční. Síce sa všetci zhodli na tom, aký má byť bazén veľký, každý z nich má však inú predstavu o tom, v ktorej oblasti by ho bolo vhodné postaviť.

Ivan je jedným z kandidátov vo voľbách. Rozhodol sa, že ak osloví niekoľko najdôležitejších obyvateľov a podarí sa mu naplánovať miesto výstavby bazéna tak, aby boli všetci spokojní, vyhrať voľby už bude hračka. Podarilo sa mu zistiť rozmery bazéna, ktorý sa bude v meste stavať, a preferencie niekoľkých obyvateľov. Teraz by rád zistil, kde bazén postaviť tak, aby boli všetci spokojní.

Pre účely úlohy si môžeme Kocúrkovo predstaviť ako rovinu so súradnicovou sústavou. Bazén ako aj preferované oblasti budeme reprezentovať obdĺžnikmi, ktorých vrcholy majú celočíselné súradnice a ktorých strany sú rovnobežné so súradnicovými osami.

Ivan pozná hodnotu  $s$  udávajúcu požadovaný obsah bazéna. Taktiež získal preferencie na polohu bazéna od  $n$  obyvateľov. Teraz by ho zaujímalo, či a kde v meste je možné postaviť bazén s daným obsahom tak, aby ležal v preferovanej oblasti každého obyvateľa.

Obyvateľom záleží len na správnom obsahu bazéna. Jeho presné rozmery si Ivan môže zvoliť, aké chce.

### Formát vstupu a výstupu

V prvom riadku vstupu sa nachádzajú dve kladné celé čísla:  $s$  a  $n$  – plocha bazéna a počet obyvateľov s názorom na jeho polohu.

Nasleduje  $n$  riadkov. V každom z nich sa nachádza názor jedného obyvateľa: štyri medzerou oddelené nezáporné celé čísla  $x_1, y_1, x_2, y_2$ . Tie reprezentujú preferovanú oblasť zadanú ako obdĺžnik, ktorého ľavým dolným rohom je bod  $(x_1, y_1)$  a pravým horným rohom bod  $(x_2, y_2)$ .

Na výstup vypíšete jeden riadok. Ak sa dá postaviť bazén s požadovanými vlastnosťami, vypíšete jeho popis v rovnakom formáte ako na vstupe – teda najskôr súradnice jeho ľavého dolného a potom súradnice jeho pravého horného rohu. Ak existuje viac správnych riešení, môžete vypísať ľubovoľné z nich. Ak hľadaný obdĺžnik neexistuje, na výstup namiesto toho vypíšete slovo „Neexistuje“.

Nezabudnite, že súradnice nájdeného bazéna musia byť celočíselné.

### Obmedzenia a hodnotenie

Táto úloha má 10 hodnotených vstupov. Za každý správne vyriešený vstup dostanete jeden bod.

Vo všetkých vstupoch platí, že  $1 \leq n \leq 100\,000$  a  $1 \leq s \leq 10^{15}$ .

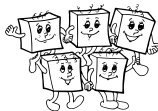
Pre súradnice každého obdĺžnika na vstupe platí  $0 \leq x_1, y_1, x_2, y_2 \leq 10^{15}$  a navyše  $x_1 < x_2$  a  $y_1 < y_2$ .

Pre niektoré vstupy platia prísnejšie obmedzenia, aby ich šlo ľahšie vyriešiť:

– Pre prvých päť vstupov platí, že  $1 \leq n \leq 1\,000$  a  $1 \leq s, x_1, y_1, x_2, y_2 \leq 1\,000\,000$ .

– Pre vstupy číslo 6, 7 a 8 platí, že  $1 \leq s, x_1, y_1, x_2, y_2 \leq 10^9$ .

Upozorňujeme, že niektoré čísla použité vo vstupoch 9 a 10 sa nezmestia bežnej 32-bitovej celočíselnej premennej (napr. int v C/C++/Java). Odporúčame preto použiť 64-bitové celočíselné premenné (napr. long long v C/C++ alebo long v Java).



### Príklady

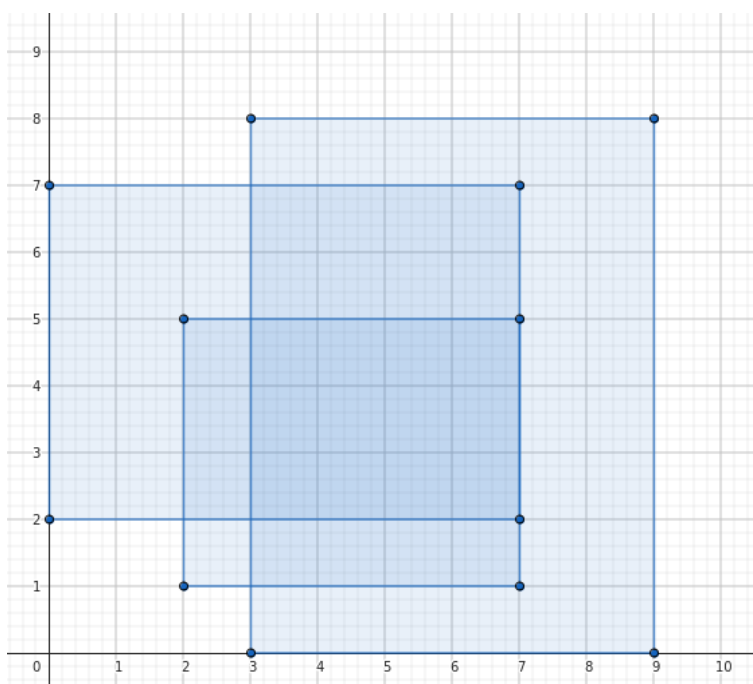
vstup

```
8 3
0 2 7 7
2 1 7 5
3 0 9 8
```

výstup

```
3 2 7 4
```

Vyššie uvedený vstup zodpovedá nasledovnému obrázku:



Ukázkový výstup popisuje obdĺžnik s ľavým spodným bodom (3, 2) a pravým vrchným bodom (7, 4).

Lahko overíme, že tento obdĺžnik je platným riešením: má obsah 8 (jeho šírka je  $7 - 3 = 4$  a jeho výška je  $4 - 2 = 2$ ) a celý leží v každom z troch vstupných obdĺžnikov.

Pre tento vstup existuje ešte jeden správny výstup: „3 3 7 5“.

vstup

```
9 3
3 1 8 5
4 2 8 6
6 1 10 7
```

výstup

```
Neexistuje
```

Pre tento vstup neexistuje žiaden bazén, ktorý spĺňa všetky požiadavky na polohu a má obsah väčší ako 6.

### Odvzdávanie riešení

Zo stránky <https://oi.sk/?d=rocnik> si stiahnite ZIP archív obsahujúci 10 testovacích vstupov.

Vstupy sú nazvané 01.txt až 10.txt.

Vyrobte k čo najviac vstupom správne výstupy a uložte ich do súborov sol01.txt až sol10.txt.

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

Za každý správny výstupný súbor získate 1 bod.



## B-I-2 Kopa kníh

*Toto je praktická úloha. Riešiť sa dá v ľubovoľnom programovacom jazyku, odovzdávajú sa len správne riešenia pre vopred pripravené testovacie dáta. Detailnejšie pokyny k odovzdávaniu riešení sú uvedené nižšie.*

Existuje knižnica. Občas sa do nej vracajú knihy, ktoré z nej boli požičané. Knižničnica je lenivá, nechce sa jej nosiť ich na správne miesta, a tak pred ňou na stole postupne vzniká čoraz vyšší stĺpik kníh. Vždy, keď sa nejaká kniha vráti do knižnice, ju knižničnica jednoducho položí na vrch tejto kopy. Občas ju ale príliš hryzie svedomie. Vtedy vezme jednu knihu z vrchu kopy a vráti ju na jej miesto v knižnici.

Keďže sa popri svojej robote nesmierne nudí, zaoberala si robotické rameno a občas ho použije. Rameno je schopné otočiť súvislý úsek tvorený  $k$  knihami na vrchu kopy. Presnejšie, použitie ramena spôsobí, že sa poradie vrchných  $k$  kníh obráti. (Ak v je kope menej ako  $k$  kníh, tak sa obráti poradie všetkých kníh v kope.)

Hodnota  $k$  je určená konštrukciou ramena a teda je pri všetkých jeho použitíach rovnaká.

Vašou úlohou je simulovať vyššie popísané typy udalostí: príchod knihy, upratanie knihy a použitie ramena.

### Formát vstupu a výstupu

Na prvom riadku vstupu je číslo  $k$ : veľkosť robotického ramena. Nasleduje niekoľko riadkov popisujúcich jednotlivé udalosti v poradí, v akom sa udiali. Každá udalosť je popísaná jedným číslom:

- Kladné číslo  $x$  znamená, že na vrch kopy bola položená kniha s identifikačným číslom  $x$ .  
(Čísla kníh sú kladné a neprevyšujú  $10^9$ . Čísla nie sú nutne rôzne – vstup môže obsahovať viacero kópií tej istej knihy.)
- Číslo  $-1$  aj číslo  $-10$  znamená, že bola odstránená vrchná kniha kopy.  
(Môžete predpokladať, že ak je v nejakom momente kopa kníh prázdna, tak vtedy určite nebudete požiadaní o odstránenie vrchnej knihy.)
- Číslo  $-2$  znamená, že bolo spustené robotické rameno.
- Číslo  $0$  značí koniec vstupu (žiadne ďalšie udalosti).
- Žiadne iné číslo sa vo vstupe neobjaví.

Pre každú udalosť typu  $-10$  vypíšte na samostatný riadok číslo knihy, ktorá bola z kopy odstránená.

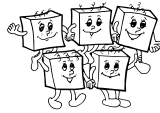
### Obmedzenia a hodnotenie

Táto úloha má 10 hodnotených vstupov. Za každý správne vyriešený vstup dostanete jeden bod.

Vo vstupe číslo 2 je nanajvýš 10 000 udalostí. V každom inom vstupe je nanajvýš 1 000 000 udalostí. V tabuľke nižšie je pre každý vstup uvedená veľkosť ramena.

číslo vstupu	1	2	3	4	5	6	7	8	9	10
$k$	1	10	10	$10^6$	$10^6$	$10^6$	$10^4$	$2 \cdot 10^4$	$5 \cdot 10^4$	$10^5$

Všimnite si, že vo vstupoch 4, 5 a 6 platí  $k \geq n$ . V týchto vstupoch teda rameno vždy otočí celú kopy kníh.



### Príklad

vstup

```
3
12345
23456
34567
45678
-2
-10
98765
-2
-1
-10
-2
-10
0
```

výstup

```
23456
34567
12345
```

Ukážkový vstup zodpovedá nasledovnej postupnosti udalostí:

- Postupne boli vrátené knihy 12345, 23456, 34567, 45678. V tomto poradí (zdola hore) sú teraz v kope.
- -2: Rameno otočilo tri horné knihy. Zdola hore teraz kopa vyzerá nasledovne: 12345, 45678, 34567, 23456.
- -10: Knihovníčka upratala hornú knihu (23456), vypíšeme jej číslo.
- Bola vrátená nová kniha. Aktuálna kopa zdola hore: 12345, 45678, 34567, 98765.
- -2: Rameno otočilo tri horné knihy. Kopa zdola hore: 12345, 98765, 34567, 45678.
- -1: Knihovníčka upratala hornú knihu (45678), nevypisujeme nič.
- -10: Knihovníčka upratala hornú knihu (34567), vypíšeme jej číslo.
- -2: Keďže sú v kope len dve knihy, rameno ju otočilo celú. Aktuálna kopa zdola hore: 98765, 12345.
- -10: Knihovníčka upratala hornú knihu (12345), vypíšeme jej číslo.
- 0: Hotovo.

### Odovzdávanie riešení

Zo stránky <https://oi.sk/?d=rocnik> si stiahnite ZIP archív obsahujúci 10 testovacích vstupov.

Vstupy sú nazvané 01.txt až 10.txt.

Vyrobte k čo najviac vstupom správne výstupy a uložte ich do súborov sol01.txt až sol10.txt.

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

Za každý správny výstupný súbor získate 1 bod.



### B-I-3 Pyramída z kociek

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

Klára sa rada hrá s drevenými kockami. Na narodeniny dostala od svojej kamarátky Lenky veľkú kopu rovnako veľkých červených a modrých kociek. Na oplátku chcela Lenku potešiť a niečo pekné jej postaviť, a keďže Lenka sa vášnivo zaujíma o staroveký Egypt, Klára sa rozhodla pre pyramídu. Nebola by to však Klára, keby si k stavaniu nevymyslela špeciálne pravidlá.

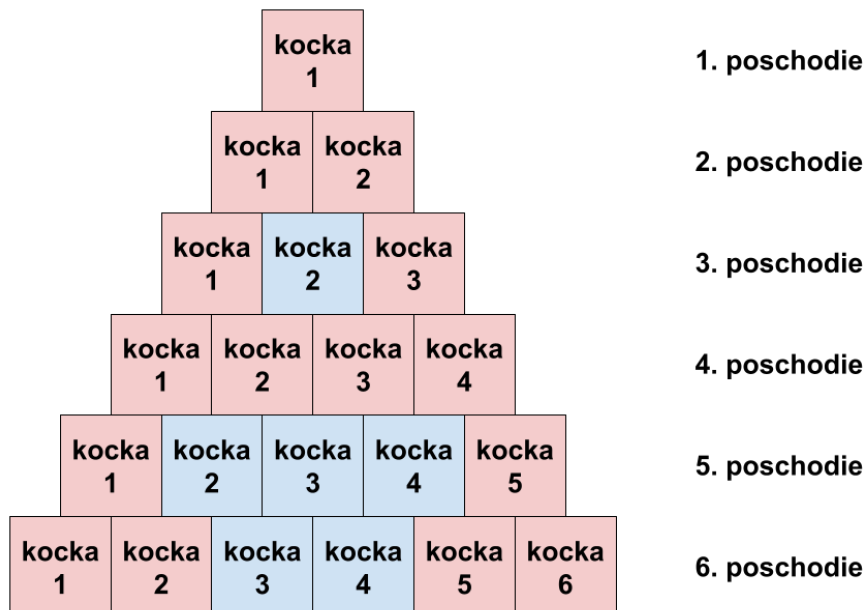
Pyramída sa bude skladať z niekoľkých poschodí poukladaných na sebe. Každé poschodie sa skladá z **jedného riadku** kociek. Ak si poschodia pyramídy očísľujeme od vrchu postupne začínúc číslom 1, tak na poschodí  $k$  bude v rade práve  $k$  kociek. Kocky na  $k$ -tom poschodí si očísľujeme od 1 po  $k$  zľava doprava.

V takto postavenej pyramíde na každej kocke stoja (polovicou svojej podstavy) *práve dve iné kocky*. Výnimkou sú len najvyššia kocka (nad ktorou už nie je žiadna iná) a kocky na ľavom a pravom okraji pyramídy (na ktorých stojí len jedna iná kocka).

Klára sa pri určovaní farieb kociek v pyramíde riadi nasledovnými pravidlami:

- Kocky na ľavom a pravom okraji pyramídy (vrátane tej na úplnom vrchu) sú všetky *červené*.
- Ak na kocke stoja dve kocky **rôznych farieb**, bude táto kocka *červená*.
- Ak na kocke stoja dve kocky **rovnakých farieb**, bude táto kocka *modrá*.

Tieto pravidlá jednoznačne určujú, ako vyzerá celá pyramída.



Na obrázku vidíme pyramídu so šiestimi poschodiami. Všimnite si napríklad, že:

- Kocka 2 na 3. poschodí je modrá, pretože na nej stoja dve červené kocky.
- Kocky 2 a 3 na 4. poschodí sú červené, na oboch totiž stojí práve jedna červená a jedna modrá kocka.
- Kocky 3 a 4 na 6. poschodí sú modré preto, že na každej z nich stoja dve modré kocky.



## Úloha

- a) (3 body) Klára by chcela, aby ňou postavená pyramída mala spodný riadok poskladaný zo samých červených kociek. To by napríklad platilo pre pyramídu so 4 poschodiami. Určite všetky hodnoty  $n$ , pre ktoré bude mať pyramída s  $n$  poschodiami spodný ( $n$ -tý) riadok zložený zo samých červených kociek. Nezabudnite na dôkaz správnosti.
- b) (2 body) Keď Klára postavila pyramídu, podarovala ju nadšenej Lenke. Lenkin pes Bobi však do pyramídy nešťastne narazil, vďaka čomu z nej tri kocky odpadli a on ich následne zjedol. Lenka si zapísala súradnice týchto kociek. Viete z nich zistiť, akej farby sú kocky v Bobim?  
Ide o tieto kocky: kocka 10 na 20. poschodí; kocka 266 na 276. poschodí; kocka 157 na 2023. poschodí.
- c) (5 bodov) Nehoda sa Lenke môže ľahko stať aj v budúcnosti, Klára by preto chcela algoritmus, ktorý by jej pomohol **čo najefektívnejšie** zistiť, kocka akej farby patrí na zadanú pozíciu. Popíšte algoritmus, ktorý dostane na vstupe dve čísla  $x$  a  $y$  a zistí, akú farbu má mať kocka  $x$  na  $y$ -tom poschodí.  
Nezabudnite zdôvodniť, prečo je váš algoritmus správny a odhadnite jeho časovú zložitosť.

## B-I-4 Tabuľkový počítač

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

*Toto zadanie je o čosi dlhšie, odporúčame ho ale aj tak prečítať a pochopiť, keďže **na krajskom kole** sa s tabuľkovými počítačmi opäť stretne. Súťažná úloha sa nachádza na konci tohto študijného textu. Jej riešenie by po pochopení tabuľkových počítačov nemalo byť náročné.*

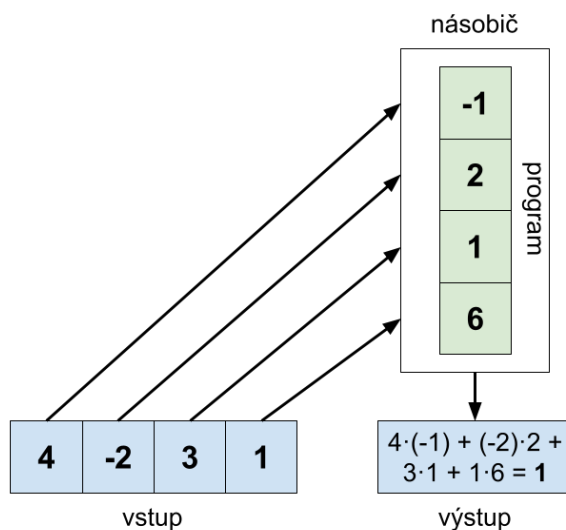
Počas prázdnin u starých rodičov našiel Krištof na povale zvláštny mechanický stroj, okolo ktorého sa povalovalo množstvo pásov tvrdého papiera. Opýtal sa starkého, na čo ten stroj slúži, a starký mu prezradil, že je to jeho starý *tabuľkový počítač* a ponúkol sa Krištofovi, že mu vysvetlí ako funguje.

Ako prvé mu ukázal papierové kartičky, ktoré sa volali *pásiky*. Pásik je úzky kúsok tvrdého papiera rozmerov  $1 \times k$  rozdelený na  $k$  štvorcikov, do ktorých sa vpisujú čísla. Starký mal pásov dostatočné množstvo pre všetky možné hodnoty  $k$ .

Následne starký Krištofovi predstavil hlavnú súčiastku tabuľkového počítača – *násobič*. Aby násobič fungoval, treba doň najprv vložiť *program*. Program je pásik dĺžky  $k$ , na ktorého každé políčko sa vpiše jedno číslo. Keď má násobič pripravený program, môžeme doň vložiť *vstup*. Vstup pre násobič je opäť pásik dĺžky  $k$ , do ktorého štvorcikov sú vpísané čísla.

Po vložení vstupu násobič „vynásobí“ vstup s programom a výslednú hodnotu, ktorú bude tvoriť *jedno číslo*, vypíše ako *výstup*. Počítanie výstupu prebieha tak, že násobič postupne vynásobí prvé číslo vstupu s prvým číslom programu, druhé číslo vstupu s druhým číslom programu, a tak ďalej až po  $k$ -te číslo. Takto získaných  $k$  hodnôt následne sčíta a tento súčet je výstupom násobiča.

Program je v násobiči pevne daný, je preto možné tento násobič použiť opakovane pre rôzne vstupy.



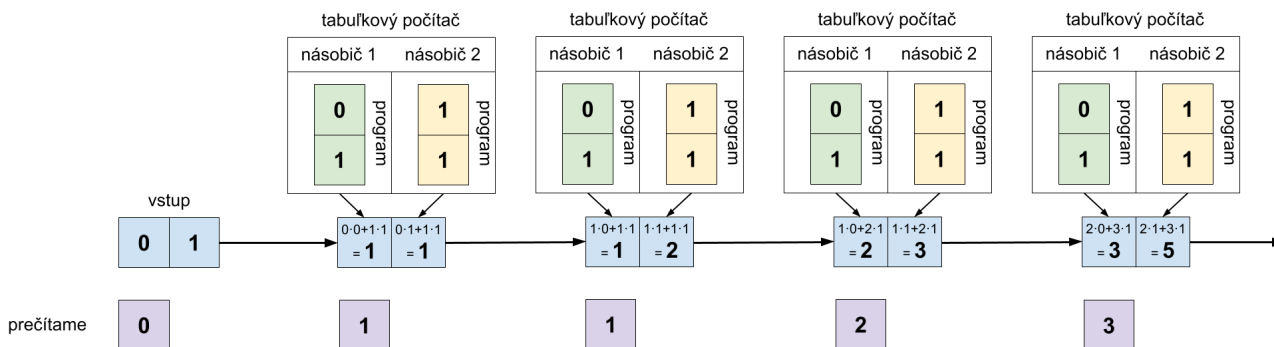
Keď bol starý v Krištofovom veku, veľa sa s násobičmi hral, vždy mu ale prekážalo, že z nich nevie naskladať žiaden zložitejší program. Vstupom pre násobič je totiž pásek s  $k$  číslami, zatiaľ čo jeho výstupom je len jedno číslo a s tým už násobič nevie ďalej nič robiť. Jedného dňa však starý prišiel na nápad, ako tento problém vyriešiť, a tak vznikol *tabulkový počítač*, ktorý vedel po vložení jedného vstupu počítať donekonečna.

Starého geniálna myšlienka bola vlastne veľmi jednoduchá: zobral  $k$  násobičov a uložil ich do radu. Potom do každého z nich vložil nejaký (nie nutne ten istý) program dĺžky  $k$ . (Tento výtvor nazval *tabulkový počítač*, keďže na  $k$  programov dĺžky  $k$  sa môžeme dívať ako na tabuľku  $k \times k$  čísel.)

Vstupom pre takýto tabulkový počítač bol pásek s  $k$  číslami. Tento vstup sa zadal ako vstup každému z násobičov. Každý násobič tento spoločný vstup vynásobil so svojím konkrétnym programom a tým vyrobil ako svoj výstup jedno číslo. Celkovým výsledkom výpočtu teda bolo dokopy  $k$  čísel: jedno z každého násobiča. Svoje výstupy jednotlivé násobiče zapísali na nový pásek: prvý násobič na jeho prvý štvorček, druhý násobič na druhý štvorček, atď. Pre vstup, ktorým bol pásek dĺžky  $k$ , sme takto dostali aj na výstupe opäť pásek dĺžky  $k$ .

Tento nový pásek, ktorý sme práve získali, je opäť platným vstupom pre tabulkový počítač. Aby sme dostali stroj, ktorý beží donekonečna, stačilo už teda pridať len jednu súčiastku, ktorá vždy výstupný pásek zobrala a vložila ho späť do tabulkového počítača ako jeho nasledujúci vstup.

Ako ale na takomto počítači počítať niečo užitočné, keď vlastne teraz žiaden výstup nemá? Aj s tým si starý poradil: pridal stroju čítačku s reproduktorom. Tá z každého **vstupu**, teda ešte **pred aplikovaním násobičov**, prečítala číslo v jeho **prvom** štvorčeku.



Krištof si všimol, že po spustení vlastne tabulkový počítač pomocou čítačky *generuje nekonečnú postupnosť celých čísel*. A starý mu prezradil, že keď sa správne nastavia programy jednotlivých násobičov a prvý vstupný pásek, výsledné postupnosti vôbec nie sú náhodné.





Zoberme si napríklad Fibonacciho postupnosť. Jej prvé dva členy sú 0 a 1 a každý ďalší sa vypočíta ako súčet dvoch predchádzajúcich členov. Začiatok Fibonacciho postupnosti teda vyzerá nasledovne: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ...

Matematicky túto definíciu vieme zapísať nasledovne: Nech  $F_n$  je  $n$ -tý člen Fibonacciho postupnosti. Pre prvé dva členy platí  $F_0 = 0$  a  $F_1 = 1$ . Následne pre všetky  $n \geq 2$  platí  $F_n = F_{n-1} + F_{n-2}$ .

No a pozrime sa na obrázok vyššie. Pri nastavení prvého násobiča na program (0, 1), druhého násobiča na program (1, 1) a počiatočnom vstupe (0, 1) tento tabuľkový počítač generuje práve Fibonacciho postupnosť.

Prečo je to tak? Vstup, z ktorého začíname, je (0, 1). Ako vidíme z obrázku, po prvom použití násobičov dostaneme na výstupe párik (1, 1). Keď ten dáme na vstup a znova použijeme násobiče, dostaneme na výstupe (1, 2). Po treťom použití násobičov máme výstup (2, 3) a po štvrtom (3, 5). Lahko si všimneme, že zatiaľ platí, že po  $n$  opakovaníach máme na prvom políčku páriku hodnotu  $F_n$  a na druhom hodnotu  $F_{n+1}$ .

Bude toto naozaj platiť donekonečna? Áno: keď dáme tabuľkovému počítaču na vstupe párik  $(F_n, F_{n+1})$ , tak prvý násobič vypočíta hodnotu  $0 \cdot F_n + 1 \cdot F_{n+1} = F_{n+1}$  a druhý vypočíta hodnotu  $1 \cdot F_n + 1 \cdot F_{n+1} = F_{n+2}$ . Ďalším vstupom bude preto párik  $(F_{n+1}, F_{n+2})$  – teda opäť dve po sebe idúce Fibonacciho čísla, akurát s o 1 väčším indexom.

Takýto postup sa dá využiť aj pri iných postupnostiach, stačí si správne zvoliť konštantu  $k$  a potom správne naprogramovať jednotlivé násobiče. Ukážeme si ešte jeden trochu zložitejší príklad.

Majme postupnosť  $T$ , kde  $T_0 = 4$ ,  $T_1 = -2$ ,  $T_2 = 3$  a pre  $n \geq 3$  platí  $T_n = T_{n-1} + 2 \cdot T_{n-2} - 7 \cdot T_{n-3} + 6$ .

Takúto postupnosť by generoval napríklad tabuľkový počítač s násobičmi (0, 1, 0, 0), (0, 0, 1, 0), (-7, 2, 1, 6) a (0, 0, 0, 1) a vstupom (4, -2, 3, 1). Skúste si vypočítat, ako budú vyzerat prvé páriky, ktoré postupne bude vyrábať na výstupe, a rozmyslite si, prečo generuje práve vyššie definovanú postupnosť.

Všimnite si tiež, že rôzne tabuľkové počítače môžu generovať tú istú postupnosť. Ak by sme vo vyššie uvedenom príklade zmenili program tretieho násobiča na (-7, 2, 1, 1) a hodnoty na vstupnom páriku na (4, -2, 3, 6), takýto tabuľkový počítač by tiež generoval postupnosť  $T$ .

### Súťažná úloha

V tejto súťažnej úlohe dostanete popis niekoľkých postupností celých čísel. Vašou úlohou bude ku každej z nich navrhnuť tabuľkový počítač, ktorý ju bude generovať.

- (2 body) Postupnosť  $A$ , kde  $A_n = 2^n$ . Začiatok tejto postupnosti: 1, 2, 4, 8, 16, 32, ...
- (2 body) Postupnosť  $B$ , kde  $B_0 = -1$ ,  $B_1 = 1$ ,  $B_2 = 4$  a pre  $n \geq 3$  platí  $B_n = 3 \cdot B_{n-1} + 8 \cdot B_{n-3}$ . Začiatok tejto postupnosti: -1, 1, 4, 4, 20, 92, 308, ...
- (3 body) Postupnosť  $C$ , kde  $C_n = 5^n - 4^n$ . Začiatok tejto postupnosti: 0, 1, 9, 61, 369, 2101, ...
- (3 body) Postupnosť  $D$ , kde  $D_0 = 15$ ,  $D_1 = 4$  a pre  $n \geq 2$  platí  $D_n = D_{n-1} + D_{n-2} + 2n$ . Začiatok tejto postupnosti: 15, 4, 23, 33, 64, 107, 183, ...

Pri hodnotení vašich riešení bude záležať len na ich správnosti.

Konštantu  $k$  si v každej podúlohe môžete zvoliť ľubovoľne. Ako drobnú pomôcku uvádzame, že každá zo zadaných podúloh sa dá vyriešiť tabuľkovým počítačom, ktorý má nanajvýš päť násobičov.

V riešení každej podúlohy odovzdajte nasledovné veci:

- Program tabuľkového počítača, teda  $k$ , čísla na prvom vstupe a programy jednotlivých násobičov.
- Slovný popis toho, ako ste tieto hodnoty získali, a zdôvodnenie, že váš tabuľkový počítač naozaj generuje zadanú postupnosť.

---

### TRIDSIATY DEVIATY ROČNÍK OLYMPIÁDY V INFORMATIKE

Príprava úloh: Michal Anderle, Truc Lam Bui, Andrej Korman  
Recenzia: Michal Forišek, Paulína Smolárová  
Slovenská komisia Olympiády v informatike  
Vydal: NIVAM – Národný inštitút vzdelávania a mládeže, Bratislava 2023