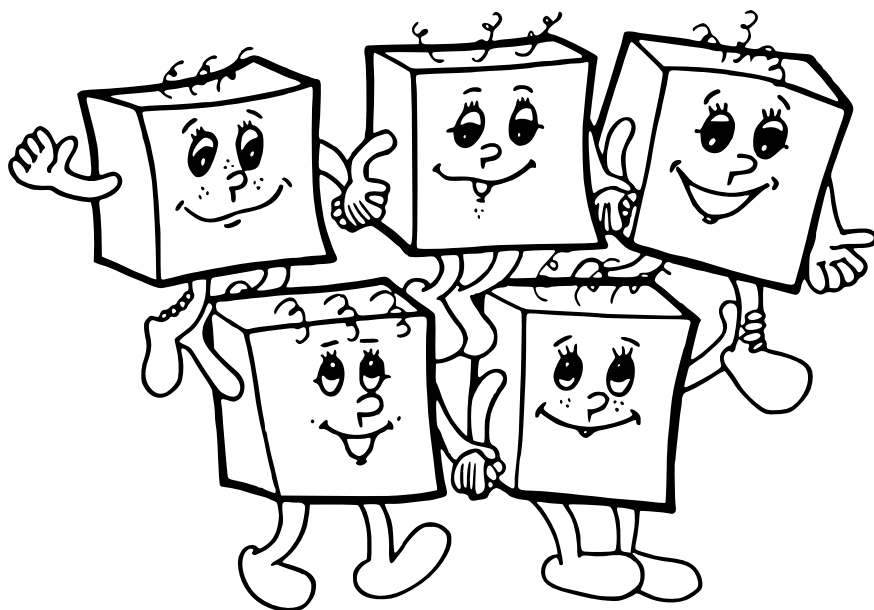
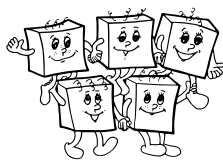


# OLYMPIÁDA V INFORMATIKE NA STREDNÝCH ŠKOLÁCH



**23. ročník**  
školský rok 2007/08  
**zadania domáceho 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://www.ksp.sk/oi/>.



## Informácie a pravidlá

### Pre koho je súťaž určená?

OI sa uskutočňuje v týchto kategóriách:

- a) **kategória A** je určená pre žiakov tretieho a štvrtého ročníka stredných škôl a príslušných ročníkov viacročných gymnázií a má tri kolá: domáce, krajské a celoštátne
- b) **kategória B** je určená pre žiakov prvého a druhého ročníka stredných škôl a príslušných ročníkov viacročných gymnázií a má dve kolá: domáce a krajské

Do každej kategórie sa môžu zapojiť aj žiaci nižších ročníkov ako tí, pre ktorých je určená.

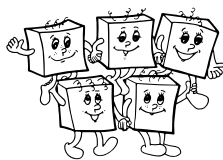
### Ako bude súťaž prebiehať?

V *domácom kole* každej kategórie účastníci riešia štyri úlohy. Riešenia odozdajú svojmu učiteľovi informatiky do **15. novembra 2007**. Učitelia informatiky odošlú riešenia v tomto termíne zo školy na adresu príslušnej krajskej komisie OI. Adresy krajských komisií sú uvedené na poslednej strane tohto letáku.

Najúspešnejší riešitelia domáceho kola sú pozvaní do *krajského kola*, kde riešia štyri teoretické úlohy.

V kategórii A sú do *celoštátneho kola* pozývaní najúspešnejší riešitelia krajských kôl. Presnejšie, po vyhodnotení krajských kôl prebehne koordinácia bodovacích škál, spoja sa výsledkové listiny do jednej celoštátnej, a do celoštátneho kola sú pozvaní najlepší riešitelia podľa tejto výsledkovej listiny.

V celoštátnom kole účastníci prvý deň riešia tri teoretické úlohy, druhý deň dve praktické úlohy (pri počítači). Z najlepších riešiteľov tohto kola SK OI vyberie družstvá pre Medzinárodnú informatickú olympiádu (IOI) a Stredoeurópsku informatickú olympiádu (CEOI).



### Ako majú vyzerieť riešenia domáceho kola?

Riešenia súťažných úloh domáceho kola pozostávajú z dvoch častí:

**Popis riešenia.** Riešenia musia obsahovať podrobný popis použitého algoritmu, **zdôvodnenie jeho správnosti** a diskusiu o efektívnosti zvoleného riešenia (t.j. posúdenie časových a pamäťových nárokov programu). Algoritmus by mal byť jasný už z popisu riešenia, teda bez toho, aby bolo potrebné nahliadnuť do programu.

**Program.** V úlohách 1, 2 a 3 je potrebné k riešeniu pripojiť odladený program napísaný v jazyku Pascal, C alebo C++. (Musí sa dať skompilovať kompilátorom *FreePascal*, resp. *gcc*.) Program sa odovzdáva v písomnej forme (vytlačенý) aj v elektronickej forme (na CD, prípadne diskete).

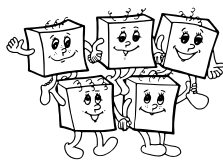
Svoje programy pomenujte *k-i-x.pas/.c/.cpp*, kde *k* je kategória a *x* je číslo súťažnej úlohy. Z jednej školy možno poslať všetky riešenia na jednom CD. V takomto prípade pre každého riešiteľa vytvorte podadresár označený jeho priezviskom.

V úlohe 4 odovzdávate program len v písomnej podobe.

Popis riešenia vypracujte čitateľne na listy formátu A4. **Každú úlohu začnite na novom liste** a v záhlaví uveďte vaše meno, ročník, adresu školy a označenie príkladu podľa tohto letáku. Zadania úloh nemusíte opisovať. Ak sa vám riešenie úlohy nezmestí na jeden list, uveďte na ďalších listoch vľavo hore svoje meno a označenie úlohy, listy očísľujte a zopnite.

### Usporiadateľ súťaže

Olympiádu v informatike (OI) vyhlasuje *Ministerstvo školstva SR* v spolupráci so *Slovenskou informatickou spoločnosťou* 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.



## Zadania kategórie B

Do **15. novembra 2007** odovzdajte riešenia týchto úloh svojmu učiteľovi informatiky. Ten ich dňa 15. novembra 2007 odošle na adresu príslušnej krajskej komisie OI. Adresy krajských komisií sú uvedené na poslednej strane tohto letáku.

### B-I-1 O spájaní polí

Máme dve polia celých čísel. V nich máme uložených dokopy  $M + N$  **kladných** celých čísel.

Prvé pole sa volá  $A$  a obsahuje  $M + N$  prvkov. Prvých  $M$  čísel v ňom je kladných, na ostatných  $N$  pozíciách sú nuly (ktoré predstavujú voľné miesto). Prvých  $M$  čísel je navyše utriedených vzostupne.

Druhé pole sa volá  $B$  a obsahuje  $N$  kladných čísel utriedených vzostupne.

Vašou úlohou bude naprogramovať čo najefektívnejší algoritmus, ktorý presunie všetkých  $M + N$  **kladných** čísel do poľa  $A$ , a to navyše tak, aby pole  $A$  bolo na konci vzostupne utriedené.

#### Príklad

*na začiatku*

pole  $A$ : 1, 4, 5, 7, 0, 0, 0

pole  $B$ : 2, 5, 10

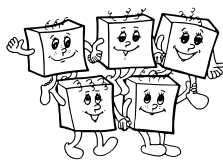
*na konci*

pole  $A$ : 1, 2, 4, 5, 5, 7, 10

#### Súťažná úloha

Napište vo vami zvolenom programovacom jazyku procedúru/funkciu, ktorá dostane ako parametre polia  $A$  a  $B$  (a prípadne aj čísla  $M$  a  $N$ ), a upraví obsah poľa  $A$  podľa zadania.

**Dôležité obmedzenie:** Okrem týchto dvoch polí už smiete použiť len konečný počet celočíselných premenných. Inými slovami, je zakázané používať pomocné polia.



## B-I-2 Krtkovou norou

Kdesi v okolí Brna žijú prúdoví krtkovia. Žijú pod zemou, majú tam svoje brlôžky a medzi nimi vyrazené chodbičky. Chodbičky sú rôzne dlhé, priamo úmerne tomu, koľko dní danú chodbičku krtkovia razili. A že sú krtkovia prúdoví, chodbičky razia rýchlo, a teda im to nikdy dlho netrvá.

V jednom z brlôžkov býva šéf, krtek Vítek. Ten rád chodí na návštevy k ostatným krtkom. Zaujímalo by ho teraz, ku komu to vlastne má najďalej.

### Súťažná úloha

Napište program, ktorý načíta popis siete brlôžkov, a zistí, kam to má krtek Vítek zo svojho brlôžka najďalej.

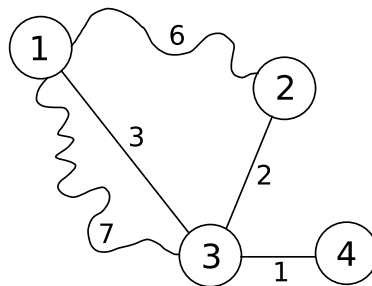
### Príklad

vstup

4	5	
1	2	6
1	3	7
1	3	3
2	3	2
3	4	1

výstup

2
---

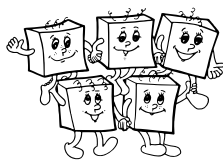


*Najkratšia cesta do brlôžkov 2 aj 4 vedie cez brlôžok 3.*

### Formát vstupu a výstupu

V prvom riadku vstupu sú dve celé čísla  $N$  a  $M$  ( $1 \leq N \leq 1\,000$ ,  $0 \leq M \leq 100\,000$ ), kde  $N$  je počet brlôžkov a  $M$  počet chodbičiek medzi nimi. Brlôžky sú očíslované od 1 do  $N$ . Krtek Vítek býva (ako správny šéf) v brlôžku číslo 1.

Nasleduje  $M$  riadkov. V každom z nich sú tri kladné celé čísla  $a$ ,  $b$  a  $d$ , kde  $a$  a  $b$  sú čísla brlôžkov spojených chodbičkou a  $d$  udáva, koľko dní krtkovia chodbičku razili.



Všimnite si, že  $d$  je celé číslo. Keďže krtkovia sú prúdoví a razia chodby rýchlo, môžete predpokladať, že  $d \leq 7$ . Navyše môžete predpokladať, že sieť chodbičiek je súvislá – teda že sa krtiek Vítek vie dostať do hociktorého brlôžka.

Na výstup vypíšte jeden riadok a v ňom medzerami oddelené čísla všetkých brlôžkov, ktoré sú od brlôžka číslo 1 najďalej.

### B-I-3 Kontrola XML

So skratkou XML (eXtensible Markup Language) ste sa už pravdepodobne stretli. Zjednodušene môžeme povedať, že ide o akýsi univerzálny formát na uloženie a prenos dát akéhokoľvek druhu.

Dáta v XML dokumente môžu byť hierarchicky rozčlenené. Na toto slúžia tzv. *tagy* – značky, ktorými vyznačíme časť XML dokumentu.

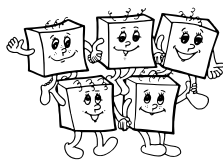
Ukážeme si teraz príklad XML dokumentu:

```
<kviz>
  toto je text len tak
  <zaznam>
    <otazka>Kolky ročník OI prave prebieha?</otazka>
    <odpoved>23</odpoved>
  </zaznam>
  <zaznam><otazka>2+2?</otazka><odpoved>4</odpoved></zaznam>
</kviz>
```

Tagy sú uzavreté v špicatých zátvorkách. Tento dokument teda obsahuje napr. tagy `<kviz>`, `<otazka>` a `</kviz>`.

Všimnite si, že tagy musia byť *párové* – jeden (otvárací) na začiatku vyznačenej oblasti, druhý (zatvárací) na jej konci. Zatvárací tag má vždy rovnaké meno ako otvárací a navyše začína znakom `/`.

Tiež si všimnite, že tagy sú v sebe vnorené – keď napríklad vo vnútri tagu `<zaznam>` otvoríme tag `<otazka>`, musíme najskôr použiť zatvárací tag `</otazka>`, až potom môžeme použiť tag `</zaznam>`.



### Súťažná úloha

Vašou úlohou bude napísať program, ktorý dostane na vstupe text a skontroluje, či je to korektný XML dokument.

Formálnejšie si definujeme, ako (pre účely tejto úlohy) vyzerá korektný XML dokument. Dokument sa skladá z tagov a textu medzi nimi. Text medzi tagmi neobsahuje znaky `<` a `>`. Názov každého tagu je tvorený len malými písmenami, a to aspoň jedným a nanajvýš ôsmimi.

Za každým otváracím tagom musí nasledovať jemu zodpovedajúci zatvárací, pričom musí byť dodržané vnáranie tagov do seba. (Inými slovami, pre ľubovoľný otvárací tag musí platiť, že keď zoberieme všetky tagy a text za ním, až po jemu zodpovedajúci zatvárací tag, dostaneme opäť korektný XML dokument.)

### Formát vstupu a výstupu

Vstup obsahuje niekoľko (nanajvýš 1000) riadkov textu, každý z nich obsahuje najviac 100 znakov.

Na výstup vypíšete text „ano“ alebo „nie“ podľa toho, či text na vstupe predstavuje korektný XML dokument.

### Príklady

vstup

```
<srnka><paroh>hnedy</paroh>  
</srnka><jelen></jelen>
```

výstup

ano

vstup

```
<a> <b> </a> </b>
```

výstup

nie

*Tagy nie sú v sebe vnorené.*

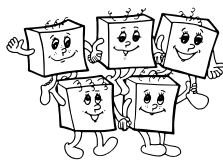
vstup

```
<body> <zly/zly tag> </body>
```

výstup

nie

*Nekorektný tag.*



vstup

```
<p><data>47</data> 94 <b><q>
</q> <q>q</q> <a><cc></cc>
</a></b> nejaky-text
</p> a este aj na konci
```

výstup

ano

vstup

a < b + c

výstup

nie

*Znak < nesmie byť použitý.*

vstup

```
<a> <b> <b> </b>
</b> </a> </a>
```

výstup

nie

*Posledný zatvárací tag nemá zodpovedajúci otvárací.*

vstup

```
<srnka><paroh></paroh>
```

výstup

nie

*Tag <srnka> zostal otvorený.*

## B-I-4 Rákosie

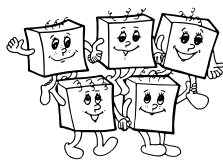
V tomto ročníku olympiády sa budeme v teoretickej úlohe stretávať so zvláštnym druhom rákosia. V študijnom texte uvedenom za zadaním tejto úlohy je popísané, ako toto rákosie vyzerá.

### Súťažná úloha

Nájdite nejakú odrodu rákosia, ktorej mŕtve steblá sú tvorené len bunkami  $a$ , a navyše počet týchto buniek je mocnina dvoch.

Vo vašej odrode teda musia ísť „vyrobiť“ steblá  $a$ ,  $aa$ ,  $aaaa$ ,  $a^8$ ,  $a^{16}$ , ... Naopak, nesmú sa dať vyrobiť steblá  $aaa$  ani  $abc$ .





## Študijný text

**Programátorské rákosie** je zvláštna rastlina. Každý exemplár sa skladá z niekoľkých buniek, ktoré rastú v rade za sebou. Každá bunka je buď živá, alebo mŕtva. Živé bunky budeme značiť veľkými písmenami abecedy, mŕtve bunky malými. Keďže bunky rastú v rade, každé steblo rákosia vieme jednoducho popísať tak, že vypíšeme zaradom (od koreňa) typy jeho buniek.

Každé steblo rákosia sa počas svojho života vyvíja, až do okamihu, keď už obsahuje samé mŕtve bunky. Počas tohto vývoja sa môže meniť počet aj druh buniek, ktoré obsahuje.

Existuje veľa rôznych odrôd rákosia. Každá odroda má v génoch „naprogramovanú“ svoju sadu pravidiel, podľa ktorých sa steblá danej odrody môžu meniť.

Mŕtve bunky sú pekné farebné. Pestovatelia preto často šľachtia rôzne odrody, ktoré by vytvárali z mŕtvych buniek rôzne zaujímavé vzory.

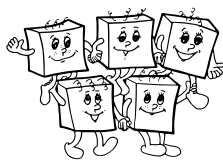
### Formálnejšie definície

**Steblo rákosia** je ľubovoľná (aj prázdna) postupnosť veľkých a malých písmen, predstavujúcich živé a mŕtve bunky. Ak steblo neobsahuje žiadne živé bunky, hovoríme, že je to mŕtve steblo.

Aby sme prázdnu postupnosť buniek videli, budeme ju značiť  $\varepsilon$ . Ak ide v zápise stebľa  $k$  rovnakých písmen  $x$  za sebou, môžeme to skrátene zapísať  $x^k$ . Teda napríklad  $baCDGk^3v^{47}XX$  je steblo rákosia, ktoré je tvorené 57 bunkami, z nich je 5 živých.

Všimnite si, že steblo vypisujeme vždy od koreňa nahor. Teda steblo  $abc$  je iné ako steblo  $cba$ .

**Genetické pravidlo** je pravidlo, ktoré určuje jednu možnú zmenu stebľa. Každé pravidlo je tvaru: „ak sa na stebli vyskytne postupnosť buniek *vstup*, môže sa zmeniť na postupnosť buniek *výstup*“. Postupnosti *vstup* aj *výstup* môžu obsahovať ľubovoľne veľa (aj nula) živých aj mŕtvych buniek. Jediné obmedzenie je, že *vstup* musí obsahovať aspoň jednu živú bunku.



Pravidlo zapisujeme  $vstup \rightarrow výstup$ . Ak máme viac pravidiel, ktoré majú rovnaký vstup, môžeme ich skráteno zapísať ako  $vstup \rightarrow výstup_1 \mid výstup_2 \mid výstup_3$ .

Napríklad zo stebľa  $aBacBd$  sa použitím pravidla  $aB \rightarrow X$  môže stať buď steblo  $XacBd$  alebo steblo  $aBacXd$ .

Použitie pravidla budeme značiť  $\Rightarrow$ . (Všimnite si, že ide o dvojité šípku, zatiaľ čo na zápis pravidla používame jednoduchú.) Teda napríklad druhú zmenu z predchádzajúceho príkladu by sme zapísali  $aBacBd \Rightarrow aBacXd$ .

Keď vznikne nové steblo rákosia, je vždy tvorené práve jednou bunkou  $Z$ . **Odroda rákosia** je jednoznačne určená *konečnou* sadou genetických pravidiel. Tá jednoznačne popisuje, aké všetky druhy stebiel môžu zo začiatočnej bunky  $Z$  narásť.

**Súbor vzorov**, ktoré vie odroda rákosia vyrobiť, je množina *všetkých mŕtvych* stebiel, ktoré môžu v rákosi danej odrody narásť.

### Príklad 1

Pestovateľ Ferko chcel, aby mu na rybníku vyrástlo rákosie, ktoré bude mať pekné strakaté stebľá. Presnejšie, chcel, aby súbor vzorov jeho rákosia tvorili práve *všetky* stebľá, na ktorých sa striedajú bunky  $a$  a  $b$ .

Ferko teda chce odrodu, v ktorej môžu narásť mŕtve stebľá  $ababa$ ,  $abab$  aj  $babab$ , ale nie  $aab$  ani  $klmn$ .

### Riešenie príkladu 1

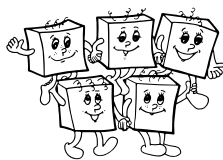
Existuje veľa rôznych odrôd, ktoré majú túto vlastnosť. Ferko vyšľachtil odrodu, ktorá mala nasledujúce genetické pravidlá:

$$Z \rightarrow X, \quad Z \rightarrow bX, \quad X \rightarrow abX, \quad X \rightarrow \varepsilon \quad \text{a} \quad X \rightarrow a.$$

Ukážeme, ako môže narásť niekoľko dobrých stebiel. Ku každému použitiu pravidla  $\Rightarrow$  napíšeme pre názornosť poradové číslo pravidla, ktoré sme použili. Teda napríklad  $\Rightarrow_3$  znamená, že bolo použité pravidlo  $X \rightarrow abX$ .

$$\begin{array}{llllll} Z & \Rightarrow_1 & X & \Rightarrow_3 & abX & \Rightarrow_3 & ababX & \Rightarrow_3 & ababa \\ Z & \Rightarrow_1 & X & \Rightarrow_3 & abX & \Rightarrow_3 & ababX & \Rightarrow_4 & abab \\ Z & \Rightarrow_2 & bX & \Rightarrow_3 & babX & \Rightarrow_3 & bababX & \Rightarrow_4 & babab \end{array}$$

Všimnite si, že použitie pravidla 4 zmaže zo stebľa bunku  $X$ .



## Príklad 2

Pestovatelia Jožko a Mirko chceli, aby ich rákosie „vyrábalo“ mŕtve steblá, ktoré budú mať rovnako veľa buniek  $a$  aj  $b$  (a žiadne iné), ale na rozdiel od Ferka nech tieto bunky môžu byť ľubovoľne rozmiestnené.

Chcú teda odrodu, kde môžu narásť steblá  $ab$ ,  $abba$  aj  $ba^{47}b^{46}$ , ale nie  $ababa$  ani  $bac$ .

## Riešenie príkladu 2

Jožko vyšľachtil rákosie s pravidlami:

$$Z \rightarrow aZbZ \mid bZaZ \mid \varepsilon.$$

Mirko vyšľachtil rákosie s pravidlami:

$$Z \rightarrow ABZ \mid \varepsilon, \quad AB \rightarrow BA, \quad BA \rightarrow AB, \quad A \rightarrow a, \quad B \rightarrow b.$$

Hlavný rozdiel medzi týmito dvoma odrodami je v tom, že Mirkovi sa bude ľahšie presvedčať návštevy, že jeho rákosie naozaj „funguje“.

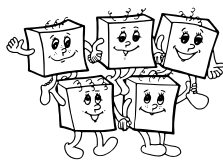
V prípade Mirkovho rákosia je to naozaj jednoduché. Použitím prvého pravidla  $k$ -krát dostaneme steblo, na ktorom je  $k$  živých buniek  $A$  a  $k$  živých buniek  $B$ . Použitím pravidiel 2 a 3 sa tieto bunky môžu ľubovoľne premiešať a keď následne použijeme pravidlá 4 a 5, vyhrali sme.

U Jožkovho rákosia je zjavné, že počty  $a$  a  $b$  budú rovnaké, ale vieme naozaj vyrobiť všetky možné dobré steblá? Matematickou indukciou podľa dĺžky stebľa dokážeme, že áno.

Steblá dĺžky 0 ( $\varepsilon$ ) aj 2 ( $ab$  a  $ba$ ) vyrobiť vieme. Nech teraz vieme vyrobiť všetky dobré steblá dĺžky  $\leq 2n$ . Zoberme si ľubovoľné dobré steblo dĺžky  $2n+2$ . Bez ujmy na všeobecnosti nech začína bunkou  $a$ . Budeme postupne čítať steblo a počítať si, o koľko viac  $a$  ako  $b$  sme videli. Na začiatku je táto hodnota 1 (začínáme bunkou  $a$ ), na konci bude 0 (lebo  $a$  a  $b$  v celom stebly je rovnako).

Všimnime si okamih, kedy prvýkrát klesne táto hodnota na 0. Muselo sa to stať tak, že sme práve prečítali nejaké  $b$ . Napríklad pre steblo  $aabaabbabbbaab$  sa tak stane po prečítaní desiateho znaku, pre  $aabb$  po štvrtom.

Teraz naše slovo môžeme schematicky zapísať ako  $a\beta b\gamma$ , kde  $b$  je to  $b$ , ktoré sme si práve našli. Vieme, že v časti  $a\beta b$  je rovnako veľa  $a$  aj  $b$ . Preto musí byť



rovnako veľa  $a$  aj  $b$  v časti  $\beta$ , a teda aj v časti  $\gamma$ . Podľa indukčného predpokladu vieme teda zo  $Z$  vyrobiť aj  $\beta$ , aj  $\gamma$ . No a celé naše steblo teda vyrobíme tak, že najskôr použijeme pravidlo  $Z \rightarrow aZbZ$ , potom z prvého  $Z$  vyrobíme  $\beta$ , a nakoniec z druhého  $Z$  vyrobíme  $\gamma$ .

---

### Adresy krajských komisií OI

#### *Banskobystrický kraj*

PaedDr. Miloslava Sudolská, KI FPV UMB, Tajovského 40, 974 01 B. Bystrica

#### *Bratislavský kraj*

RNDr. Eva Hanulová, Gym. J. Hronca, Novohradská 3, 821 09 Bratislava.

#### *Košický kraj*

RNDr. Rastislav Krivoš-Belluš, Ústav informatiky PF UPJŠ,  
Jesenná 5, 041 54 Košice

#### *Nitriansky kraj*

prof. Ing. Veronika Stoffová, CSc., KI PF UJS, Roľníckej školy 1519,  
945 01 Komárno

#### *Prešovský kraj*

Mgr. Mária Majherová, Katedra matematiky, FHPV, Prešovská univerzita,  
Ul. 17 novembra č. 1, 081 16 Prešov

#### *Trenčiansky kraj*

Ing. Andrea Julény, Katedra informatiky, Fakulta mechatroniky TnUAD,  
Pri parku 19, 911 06 Trenčín

#### *Trnavský kraj*

Mgr. Blanka Thomková, Gym. Jána Hollého, Na hlinách 30, 917 01 Trnava

#### *Žilinský kraj*

RNDr. Peter Varša, Ph.D., KI FRI ŽU, Moyzesova 20, 010 26 Žilina

---

## SLOVENSKÁ KOMISIA OLYMPIÁDY V INFORMATIKE

### DVADSIATY TRETÍ ROČNÍK OLYMPIÁDY V INFORMATIKE

Vydala IUVENTA s finančnou podporou Ministerstva školstva SR

Náklad: 400 výtlačkov

Zodpovedný redaktor: Michal Forišek

Sadzba programom L<sup>A</sup>T<sub>E</sub>X

© Slovenská komisia Olympiády v informatike, 2007