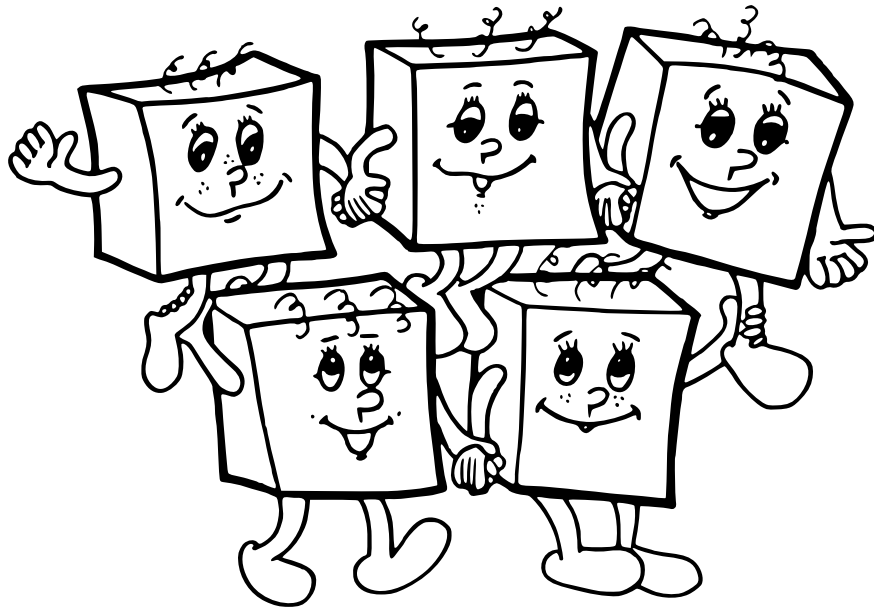


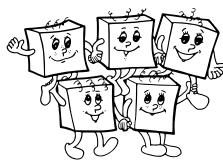
OLYMPIÁDA V INFORMATIKE NA STREDNÝCH ŠKOLÁCH



dvadsiaty druhý (a zároveň prvý) ročník
školský rok 2006/07

zadania domáceho kola
kategória B

- **Olympiáda v informatike** je od tohto školského roku 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/>.
- Novinkou je zavedenie **dvoch kategórií**. Nová kategória B je určená pre mladších riešiteľov. Podrobnosti nájdete v pravidlách.



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.

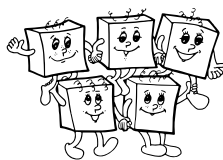
Ako bude súťaž prebiehať?

V *domácom kole* každej kategórie účastníci riešia štyri úlohy. Riešenia odovzdajú svojmu učiteľovi informatiky do **15. novembra 2006**. 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 vyzerat' 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 efektivite 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 nahliadnúť 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).

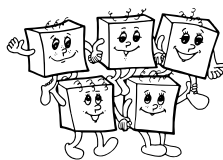
Svoje programy pomenujte *k-i-x.pas/.c/.cpp*, kde *k* je kategória a *x* je číslo súťažnej úlohy. Z jednej školy možno poslať všetky riešenia na jednom CD. V takomto prípade pre každého riešiteľa vytvorte podadresár označený jeho priezviskom.

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 2006** odovzdajte riešenia týchto úloh svojmu učiteľovi informatiky. Ten ich dňa 15. novembra 2006 odošle na adresu príslušnej krajskej komisie OI. Adresy krajských komisií sú uvedené na poslednej strane tohto letáku.

B-I-1 Pestovanie mrkvičiek

Janko je majiteľom veľmi úspešnej firmy, zaoberajúcej sa pestovaním mrkvičiek. Jeho pole, po ktorom sa rád prechádza, je veľká rovina, ktorá sa tiahne od vidím do nevidím celou obrovskou krajinou. Aby Janko presne vedel, kde sú mrkvičky nasadené, rozhodol sa ich nasadiť práve do bodov s celočíselnými súradnicami.

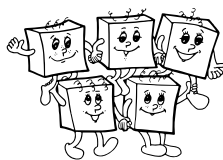
Teraz sa rozhodol svoje podnikanie rozšíriť a poskytovať mrkvičky aj cez internet. Objednal si preto pripojenie svojej farmy optickým káblom. Dozvedel sa ale hroznú novinu. Aby sa kábel dostal až ku jeho farme, musí prechádzať popod mrkvičkové pole. Na jednom mieste musí vystúpiť na povrch, aby sa vyhol zavlažovaciemu zariadeniu, a za ním opäť zostúpiť pod zem. Spoločnosť, ktorá mu má kábel inštalovať, mu oznámila súradnice, kde na jeho poli kábel vystúpi a kde sa vráti opäť pod zem. Janko teraz počíta, o koľko mrkvičiek kvôli káblu nad zemou príde. Skúste mu pomôcť.

Súťažná úloha

Napište program, ktorý pre zadané dva body s celočíselnými súradnicami zistí, koľko bodov s celočíselnými súradnicami leží na úsečke, ktorá zadané body spája.

Formát vstupu

Prvý riadok vstupu obsahuje súradnice prvého bodu (dve celé čísla x_1, y_1 oddelené medzerou). Druhý riadok obsahuje súradnice druhého bodu x_2, y_2 . Môžete predpokladať, že $-10^9 \leq x_1, x_2, y_1, y_2 \leq 10^9$ a že sú to dva rôzne body.



Formát výstupu

Vypíšte jediný riadok a v ňom jediné číslo – počet bodov s celočíselnými súradnicami na úsečke spájajúcej body zo vstupu (vrátane jej koncových bodov).

Príklad

vstup

1 1
5 3

výstup

3

V tomto príklade ide o body $[1, 1]$, $[3, 2]$ a $[5, 3]$.

B-I-2 Hladný Samo

Hladný Samo sa konečne dočkal. Jeho rodičia práve odišli z domu a on sa konečne môže poriadne najesť. Vklzol do bytu, otvoril chladničku a už už sa chcel pustiť do všetkého, čo v nej našiel. Potom sa však pozrel bližšie a pochopil, že keby sa všetky tieto veci skombinovali v jeho žalúdku, asi by to nemalo najlepšie následky. Určite by naraz nemal zjesť čokoládu s kečupom, vyprážené kurča s mliekom ani kapustu s lekvárom. Narýchlo si preto vytvoril dvojice jedál, ktoré teraz určite nechce naraz zjesť. Samo je však veľmi hladný a preto by toho chcel zjesť čo najviac.

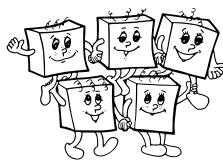
Teraz však len smutne pozerá do chladničky a rozmýšľa, čo z toho má zjesť. Pretože ste jeho dobrý kamarát/ka, napíšte mu program, ktorý mu pomôže.

Súťažná úloha

Napíšte program, ktorý zistí, koľko najviac toho Samo môže zjesť bez toho, aby zjedol obe zo zakázanej dvojice.

Formát vstupu

Prvý riadok vstupu obsahuje počet potravín v chladničke N ($1 \leq N \leq 25$). Na nasledujúcich riadkoch sú dvojice čísel i, j ($1 \leq i, j \leq N$) reprezentujúce



dvojice čísel potravín, ktoré sa nedajú zjesť spolu. Vstup je ukončený dvojicou čísel „0 0“.

Formát výstupu

Prvý riadok výstupu má obsahovať číslo M – maximálny počet potravín, ktoré môže Samo zjesť. V ďalšom riadku má byť M čísel týchto potravín. Ak existuje viacero možností s rovnakým počtom potravín, vypíšte ľubovoľnú jednu z nich.

Príklad

vstup

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

výstup

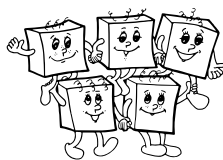
```
4
1 3 4 5
```

B-I-3 Družstvá

Na ihrisku sa zišlo niekoľko detí, ktoré si medzi sebou chcú zahrať bližšie nešpecifikovanú loptovú hru. Ich jediný problém je rozdelenie sa do dvoch družstiev. Vlastne ich ani veľmi nezaujíma, aby proti sebe hrali družstvá s rovnakým počtom hráčov, dôležité je len, aby rovnako silní hráči boli zastúpení v oboch družstvách.

Preto medzi sebou našli dvojice rovnako silných hráčov, ktorí musia hrať proti sebe. Napr. nech na ihrisko prišli Janko, Ferko, Jožko, Peťko a Mirko. Navyše vieme, že Janko určite musí hrať proti Jožkovi a Ferko musí hrať proti Peťkovi. Potom jedno možné rozdelenie do družstiev je:

1. družstvo: Janko, Ferko, Mirko
2. družstvo: Jožko, Peťko



Súťažná úloha

Na vstupe je počet hráčov a zoznam dvojíc hráčov, ktorí nemôžu hrať v jednom družstve. Napíšte program, ktorý zistí, či je možné vytvoriť dve družstvá s požadovanými vlastnosťami. Ak áno, tak jedno také rozdelenie vypíše.

Formát vstupu

Prvý riadok vstupu obsahuje celé čísla N a M , kde N je počet hráčov a M je počet dvojíc hráčov, ktorí nemôžu hrať spolu ($N \leq 10\,000$). Hráči sú očíslovaní číslami od 1 do N . Na každom z nasledujúcich M riadkov sa nachádzajú dve čísla, určujúce dvojicu hráčov, ktorí nemôžu hrať spolu v jednom družstve.

Formát výstupu

Prvý riadok výstupu má obsahovať slovo „ANO“, ak aspoň jedno vhodné rozdelenie existuje, a slovo „NIE“ v opačnom prípade. V prípade, že nejaké rozdelenie existuje, v druhom riadku výstupu majú byť čísla hráčov hrajúcich v jednom z družstiev. Ak existuje viacero správnych rozdelení, vypíšte ľubovoľné jedno z nich.

Príklady

vstup

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

výstup

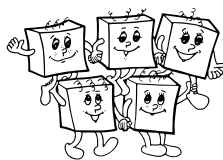
```
ANO
1 2 5
```

vstup

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

výstup

```
NIE
```



B-I-4 Assembler

Mnoho mladých programátorov túži naučiť sa programovať v asembleri. Prichystali sme pre vás príklad, ktorý vám dáva možnosť si to vyskúšať.

Budeme uvažovať zjednodušený assembler. K dispozícii je 8 registrov („pre-menných“) označených R_0, \dots, R_7 . Okrem nich už nie je k dispozícii žiadna ďalšia pamäť. Registre vedia uchovávať ľubovoľne veľké nezáporné celé číslo. Na prácu s nimi máte 6 inštrukcií:

1. **inc** R_i (increment)

Zvýši hodnotu registra R_i o 1.

2. **dec** R_i (decrement)

Ak je $R_i > 0$, zníži hodnotu registra R_i o 1. Ak $R_i = 0$, tak nespraví nič. Ak je po vykonaní inštrukcie $R_i = 0$, tak sa nastaví príznak **Z** na 1. Inak sa **Z** nastaví na 0.

3. **jmp** návěstie (jump)

Skočí na inštrukciu napísanú za daným návěstím (niečo ako **goto** v Pascale/C).

4. **test** R_i, R_j

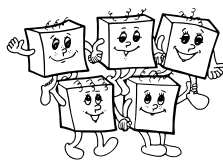
Vypočíta bitový súčin (bitwise and) registrov R_i a R_j . Výsledok však ignoruje a len nastaví príznak **Z**, podľa toho, či je daný súčin 0 alebo nie. Zvyčajne budete túto inštrukciu používať len na overenie, či je register R_i nulový a to príkazom **test** R_i, R_i , čo nastaví príznak **Z** na 1 ak $R_i = 0$, inak ho nastaví na 0.

5. **jz** návěstie (jump if zero)

V prípade, že je príznak **Z** nastavený na 1, skočí na dané návěstie. V opačnom prípade pokračuje vo vykonávaní programu.

6. **jnz** návěstie (jump if not zero)

V prípade, že je príznak **Z** nastavený na 0, skočí na dané návěstie.



Pre jednoduchosť budeme do jedného riadku programu písať len jednu inštrukciu. Pred ľubovoľnou inštrukciou môže byť napísané návěstie (oddelené od inštrukcie dvojbodkou). Je to značka v mieste programu, na ktorú sa môžeme odvolávať inštrukciami **jmp**, **jz** a **jnz**.

Formálne vyzerá program nasledovne:

$$\begin{aligned}\langle \text{program} \rangle &::= \langle \text{riadok} \rangle \mid \langle \text{riadok} \rangle \langle \text{program} \rangle \\ \langle \text{riadok} \rangle &::= \langle \text{instrukcia} \rangle \mid \langle \text{návěstie} \rangle : \langle \text{instrukcia} \rangle \\ \langle \text{instrukcia} \rangle &::= \mathbf{inc} \langle \text{register} \rangle \mid \\ &\quad \mathbf{dec} \langle \text{register} \rangle \mid \\ &\quad \mathbf{jmp} \langle \text{návěstie} \rangle \mid \\ &\quad \mathbf{test} \langle \text{register} \rangle, \langle \text{register} \rangle \mid \\ &\quad \mathbf{jz} \langle \text{návěstie} \rangle \mid \\ &\quad \mathbf{jnz} \langle \text{návěstie} \rangle \\ \langle \text{register} \rangle &::= R_0 \mid R_1 \mid \dots \mid R_7\end{aligned}$$

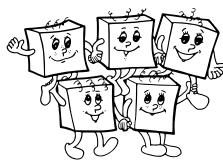
Pričom $\langle \text{návěstie} \rangle$ je ľubovoľný reťazec neobsahujúci ':' (dvojbodku).

Inštrukcie programu sa pri jeho spustení vykonávajú zaradom, podľa toho, ako boli zapísané v programe, začínajúc od prvej inštrukcie. Jedinou výnimkou sú inštrukcie **jmp**, **jz** a **jnz** po ktorých môže program pokračovať na inom mieste.

Príklad: V registroch R_1 a R_2 sú zapísané nejaké čísla (ostatné registre sú vynulované). Napíšte program, ktorý do registra R_0 zapíše súčet čísel v registroch R_1 a R_2 .

Riešenie: Najskôr budeme postupne register R_1 znižovať o 1 a súčasne zvyšovať hodnotu registra R_0 , až kým R_1 nebude nula. Tým vlastne nastavíme R_0 na hodnotu R_1 . Potom to isté spravíme aj s registrom R_2 , čím k R_0 pripočítame hodnotu R_2 .

```
c1: test R1,R1
    jz c2
    dec R1
    inc R0
    jmp c1
```



```
c2: test R2,R2
    jz koniec
    dec R2
    inc R0
    jmp c2
```

koniec:

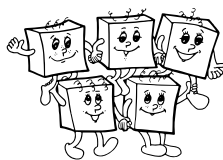
Súťažná úloha

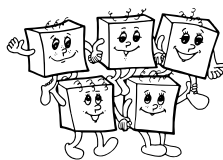
- a) Napíšte program, ktorý zo súboru `program.asm` načíta program, zo súboru `registre.in` načíta počiatočný obsah registrov a odsimuluje výpočet programu. V prípade, že výpočet skončí, vypíše do súboru `registre.out` obsah registrov po ukončení výpočtu.

Príklad

program.asm	registre.in	registre.out
<pre>c1: test R1,R1 jz koniec dec R1 inc R0 inc R0 jmp c1 koniec:</pre>	<pre>0 47 0 0 0 0 0 0</pre>	<pre>94 0 0 0 0 0 0 0</pre>

- b) V registroch R_1 a R_2 sú zapísané nejaké čísla, pričom $R_2 \neq 0$ (ostatné registre sú vynulované). Napíšte assemblerovský program, ktorý do registra R_0 zapíše zvyšok po delení čísla R_1 číslom R_2 .





Adresy krajských komisií OI

Bratislavský kraj

KSP, KZVI FMFI UK, Mlynská dolina, 842 48 Bratislava

Košický a Prešovský kraj

RNDr. Rastislav Krivoš-Belluš, Ústav informatiky PF UPJŠ,
Park Angelinum 9, 040 01 Košice

Žilinský kraj

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

Banskobystrický kraj

PaedDr. L. Huraj, KI FPV UMB, Tajovského 40, 974 01 Banská Bystrica

Trnavský, Trenčiansky a Nitriansky kraj

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

SLOVENSKÁ KOMISIA OLYMPIÁDY V INFORMATIKE
DVADSIATY DRUHÝ ROČNÍK OLYMPIÁDY V INFORMATIKE

Vydala IUVENTA s finančnou podporou Ministerstva školstva SR

Náklad: 400 výtlačkov

Zodpovedný redaktor: Michal Forišek

Sadzba programom L^AT_EX

© Slovenská komisia Olympiády v informatike, 2006