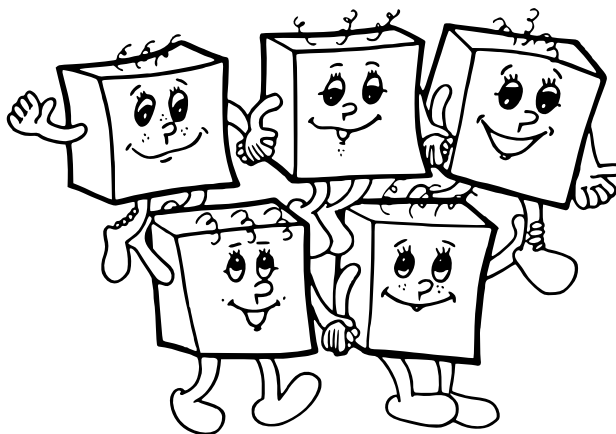


# OLYMPIÁDA V INFORMATIKE NA STREDNÝCH ŠKOLÁCH

<http://oi.sk/>



**dvadsiaty šiesty ročník**  
školský rok 2010/11

**zadania celoštátneho kola**  
**kategória A**

**2. súťažný deň**

## **Priebeh celoštátneho kola**

Celoštátne kolo 26. ročníka Olympiády v informatike, kategórie A, sa koná v dňoch 30. 3. – 2. 4. 2011. 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 jazyku Pascal, C, alebo C++. Ak sa váš program nepodarí na našom testovacom počítači skompilovať, bude automaticky hodnotený 0 bodmi.

## **Hodnotenie riešení druhého (praktického) dňa**

Za každú úlohu môžete získať od 0 do 15 bodov.

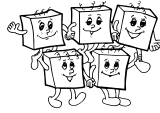
Po ukončení súťaže zoberieme pre každú úlohu váš posledný odovzdaný program a ten automaticky otestujeme na vopred pripravených testovacích vstupoch.

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.

Ku každej úlohe máme pripravených 15 sad testovacích vstupov. Sada vstupov pozostáva z jedného alebo viacerých testovacích vstupov. Za každú sadu vstupov, ktorej všetky vstupy (každý zvlášť) váš program celú správne vyrieši, získate jeden bod.

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 Asfaltistan

V Asfaltistane nedávno objavili jedno veľké územie zatiaľ nedotknuté asfaltom. Rozhodli sa to napraviť tak, že krížom cez neho postavia diaľnicu. Minister financií si ale kladie podmienky: diaľnica musí byť naozaj zjazdná automobilmi a zároveň čo najlacnejšia, pretože je kríza a nikto nemá peniaze na rozhadzovanie.

Projektanti si celé územie rozdelili na políčka (štvorce so stranou dlhou 1 km) a pre každé políčko si spočítali jeho priemernú nadmorskú výšku. V našej úlohe tieto nadmorské výšky udávame v takých jednotkách, aby platilo: diaľnica vedúca medzi dvoma susednými políčkami je zjazdná vtedy a len vtedy, ak sa ich nadmorské výšky líšia **nanajvýš o 1**.

Navyše môžu stavať mosty a tunely – mostom sa dajú prepojiť dve políčka v rovnakom stĺpci alebo riadku, ak je ich nadmorská výška zhodná a nadmorská výška **všetkých** políčok medzi nimi je **ostro** menšia. Tunelom sa tiež dajú prepojiť dve políčka s rovnakou nadmorskou výškou v rovnakom stĺpci alebo riadku, ale v tomto prípade nadmorská výška **všetkých** políčok medzi nimi musí byť **ostro** väčšia.

Je povolené, aby na jednom políčku zároveň končil jeden most či tunel a zároveň tam začínal druhý.

#### Súťažná úloha

Územie má rozmery  $r \times s$  km, jeho mapa má teda  $r$  riadkov po  $s$  stĺpcov. Vašou úlohou je nájsť najlacnejšiu diaľnicu medzi jej ľavým horným a pravým dolným rohom, teda políčkami  $(0, 0)$  a  $(r - 1, s - 1)$ . Na týchto dvoch políčkach musí viesť diaľnica po povrchu, teda nesmie byť niekde uprostred mosta alebo tunelu.

#### Formát vstupu

Na vstupe dostanete v prvom riadku celé čísla  $r, s, c, m$  a  $t$ .

Čísla  $c, m$  a  $t$  udávajú cenu za postavenie jedného políčka obyčajnej diaľnice, mosta a tunela. Celkovú cenu cesty teda zistíme tak, že za každé vnútorné políčko mosta (t.j. všetky okrem začiatku a konca) zarátame  $m$ , za každé vnútorné políčko tunela zarátame  $t$  a za každé iné políčko, ktorým cesta vedie, zarátame  $c$ .

V každom z ďalších  $r$  riadkov dostanete  $s$  čísel  $h_{i,j}$  oddelených medzerou. Tieto čísla predstavujú nadmorské výšky jednotlivých políčok.

Môžete predpokladať, že platia nasledujúce obmedzenia:

$$1 \leq r, s \leq 1\,000, 1 \leq c, m, t \leq 1\,000\,000 \text{ a } 0 \leq h_{i,j} \leq 1\,000\,000.$$

Niektoré zo vstupov sú **malé**: platí v nich  $r, s \leq 50$ . Ak váš program správne vyrieši všetky malé vstupy, dostane aspoň 5 bodov.

Niektoré zo vstupov sú **lahšie**: platí v nich, že existuje najlacnejšia cesta, ktorá neobsahuje žiaden most ani tunel. Ak váš program správne vyrieši všetky ľahšie vstupy, dostane aspoň 5 bodov.

(Niektoré vstupy môžu byť súčasne malé aj ľahšie.)

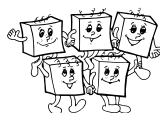
#### Formát výstupu

V prvom riadku výstupu vypíšete najmenšiu sumu, za ktorú sa dá postaviť celá diaľnica. V nasledujúcich riadkoch vypíšete jednu možnú trasu, ktorá sa dá za dané peniaze postaviť. V každom riadku vypíšete dvojicu celých čísel  $r_i, s_i$ : riadok a stĺpec políčka, ktorým cesta prechádza. Vynechajte políčka vo vnútri mostov a tunelov.

Ak žiadne riešenie neexistuje, vypíšete namiesto toho jediný riadok výstupu a v ňom slovo „NEEXISTUJE“. V prípade, že existuje viacero optimálnych riešení, vypíšete ľubovoľné jedno z nich.

Všimnite si, že celková cena cesty môže byť veľmi **veľká**. Na uloženie celkovej ceny cesty vám nemusí stačiť 32-bitová celočíselná premenná.

Tiež si všimnite, že je teoreticky možné, že sa na optimálnej ceste budú dva mosty alebo tunely krížovať. Toto je povolené, naši šikovní inžinieri to zvládnu vyriešiť tak, aby sa autá nezrážali.



**Príklady**

vstup

```
2 7 1 20 20
5 1 1 5 5 5
5 5 5 5 9 9 5
```

*Stavať mosty a tunely je drahé. Najlacnejšia cesta ide po políčkach s výškou 5, obíde jamu aj kopec.*

výstup

```
10
0 0
1 0
1 1
1 2
1 3
0 3
0 4
0 5
0 6
1 6
```

vstup

```
2 7 1 1 20
5 1 1 5 5 5
5 5 5 5 9 9 5
```

*Stavať mosty je teraz rovnako lacné ako stavať obyčajnú cestu. Je teda lacnejšie postaviť most z (0,0) na (0,3) ponad jamu ako ju obchádzať.*

výstup

```
8
0 0
0 3
0 4
0 5
0 6
1 6
```

vstup

```
2 7 1 20 1
5 1 1 5 5 5
5 5 5 5 9 9 5
```

*V tomto príklade je zase lacné stavať tunely, preto optimálna cesta vedie tunelom z (1,3) na (1,6).*

výstup

```
8
0 0
1 0
1 1
1 2
1 3
1 6
```

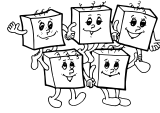
vstup

```
3 3 1 1 1
42 32 32
32 22 12
22 12 2
```

*Nevieme sa ani len pohnúť z políčka (0,0).*

výstup

NEEXISTUJE



### A-III-5 Romantické básničky II

Po tom, ako včera Hermi (aj vďaka vášmu programu!) poobľetoval všetky huslistky z orchestra, je naňho jeho priateľka Milka poriadne nasrdená (a teraz radí, Hviezdoslav).

Aby si ju udobril, potrebuje Hermi zložiť najkrajšiu báseň všetkých čias. Preto sa rozhodol, že nenechá nič na náhodu a zloží básničku s pôsobivou štruktúrou – cyklickým združeným rýmom. Báseň je písaná cyklickým združeným rýmom, ak majú rýmy schému  $ab\ bc\ cd\ de\ \dots\ yz\ za$ ; t.j. každá sloha má dva verše, pričom sa vždy ten druhý rýmuje s prvým veršom nasledujúcej slohy; navyše druhý verš poslednej slohy sa rýmuje s prvým veršom prvej slohy.

Hermi má skutočne bohatú fantáziu (a tiež váš včerajší program; a tiež sa bojí, že ho Milka zbije), takže už stihol vymyslieť celkový obsah básne aj jej jednotlivé slohy. Pre niektoré slohy už dokonca vymyslel niekoľko rôznych variantov, ktoré síce obsahujú iné dvojice rýmov, ale vyjadrujú v podstate tú istú myšlienku. Stále sa mu však akosi nedarí zo slôh poskladať celú básničku. Ten cyklický rým mu robí problémy. Hermi by potreboval s výberom jednotlivých variantov do výslednej básne pomôcť.

Aby ste si nemuseli lámať hlavu nad tým, ktoré verše se dostatočne rýmujú a ktoré nie, očísloval Hermi všetky možné rýmy prirodzenými číslami. Každý variant slohy potom popísal dvojicou  $(x, y)$  obsahujúcou čísla rýmov v oboch veršoch. Za slohou  $(x, y)$  se teda môže vyskytovať len sloha tvaru  $(y, z)$  pre nejaké  $z$ .

#### Súťažná úloha

Daný je počet slôh  $n$ , pre každú slohu počet variantov  $s_i$ , ktoré Hermi vymyslel, a pre každý z nich popis rýmov, ktoré ho tvoria. Zistite, či sa dá výberom vhodných variantov zostrojiť báseň, ktorá bude mať cyklický združený rým. Ak áno, jednu takúto báseň zostrojte.

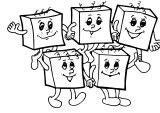
#### Formát vstupu

Prvý riadok vstupu obsahuje prirodzené číslo  $n$ , ktoré udáva počet slôh básničky. Nasleduje  $n$  skupín riadkov. Skupina  $i$  začína riadkom obsahujúcim číslo  $s_i$  udávajúce počet variantov pre  $i$ -tu slohu. Nasleduje  $s_i$  riadkov, každý s dvoma číslami  $a_{i,j}$  a  $b_{i,j}$ , popisujúcimi prvý a druhý verš jednotlivých variantov. Môžete predpokladať, že  $1 \leq n \leq 1\,000$ ,  $1 \leq s_i \leq 1\,000$  a  $1 \leq a_{i,j}, b_{i,j} \leq 10^9$ .

Niektoré zo vstupov sú **malé**: platí v nich  $n \leq 10$  a  $s_i \leq 10$ . Ak váš program správne vyrieši všetky malé vstupy, dostane aspoň 5 bodov.

#### Formát výstupu

Program vypíše na výstup buď jeden riadok s refazcom „NEEXISTUJE“, ak sa básnička z daných variantov nedá zostaviť, alebo  $n$  riadkov, pričom  $i$ -ty riadok bude obsahovať číslo variantu  $v_i$  ( $1 \leq v_i \leq s_i$ ), ktorý bol vybratý do  $i$ -tej slohy. Ak existuje viacero riešení, vypíšte ľubovoľné jedno z nich.



**Príklady**

vstup

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

výstup

```
1
2
2
1
2
```

*Jednotlivé slohy básničky budú mať rýmy:  
11, 16, 63, 38, 81.*

vstup

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

výstup

```
NEEXISTUJE
```