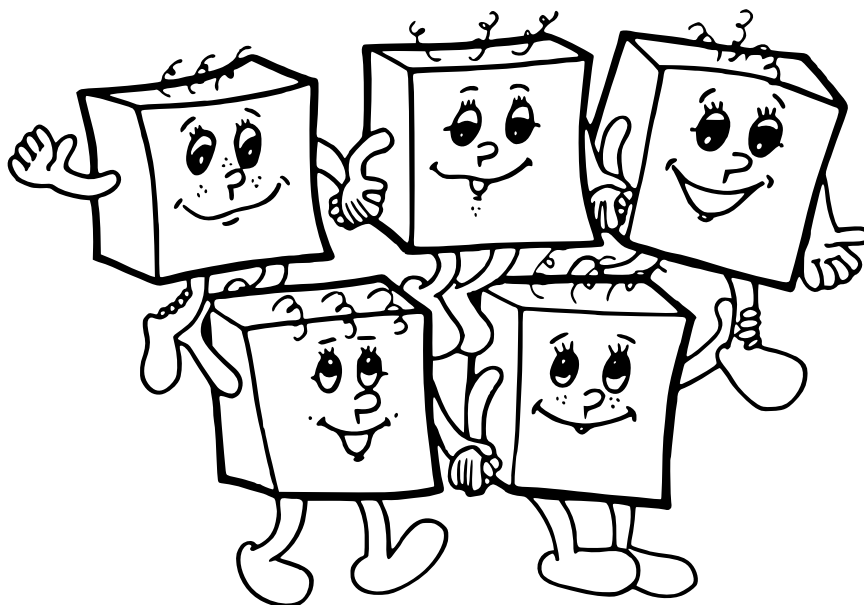


# OLYMPIÁDA V INFORMATIKE NA STREDNÝCH ŠKOLÁCH

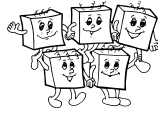


dvadsiaty štvrtý ročník  
školský rok 2008/09

zadania celoštátneho kola  
kategória A

2. súťažný deň

- **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://oi.sk/>.



## Informácie a pravidlá

### Priebeh celoštátneho kola

Celoštátne kolo 24. ročníka Olympiády v informatike, kategórie A, sa koná v dňoch 25. – 28. marca 2009. 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?

V každej z úloh **A-III-4** a **A-III-5** odovzdávate program v jednom z jazykov Pascal, C alebo C++.

Tento program bude automaticky otestovaný na niekoľkých sadách vstupných údajov. Počet udelených bodov závisí od toho, koľko z nich správne vyrieši (pri dodržaní časového a pamäťového limitu).

Testovacie vstupy budú rôzne veľké, takže aj menej efektívne ale korektné programy získajú kladný počet bodov.

## Zadania kategórie A

### A-III-4 Výlet do Švajčiarska

Tibor je vášnivý cyklista. Počas najbližších letných prázdnin by sa rád vybral bicyklovať do Švajčiarska. Ideálny výlet by sa podľa neho mal skladať zo štyroch jednodenných etáp. Po každej etape prespí v mestečku, kde skončila, a na druhý deň ráno odtiaľ vyrazí na ďalšiu etapu. Aby si mohol kúpiť spätný lístok na vlak do Švajčiarska (a tak ušetril), potreboval by začínať prvú etapu v tom istom meste ako končiť poslednú.

Tibor si na internete našiel zoznam všetkých možných jednodenných etáp po Švajčiarsku. Ku každej už dokonca iní cyklisti uviedli aj hodnotenie, aká je pekná.

### Súťažná úloha

Pre jednoduchosť očísľujeme mestá v Švajčiarsku číslami od 1 po  $N$ . Počet etáp, ktoré Tibor našiel, označíme  $M$ . Každá etapa spája dve rôzne mestá. Žiadne dve etapy nespájajú tú istú dvojicu miest. V každom meste končí najvyšš 100 etáp.

Každú etapu je možné ísť oboma smermi. Každá etapa má číselné ohodnotenie, čo je celé číslo medzi 1 a 256. Ohodnotenie je rovnaké pre oba smery.

Vašou úlohou je nájsť postupnosť štyroch **rôznych** etáp, ktoré budú na seba nadväzovať, a navyše posledná bude končiť v meste, kde prvá začína. Ak je viac možností, vyberte tú, v ktorej je súčet ohodnotení etáp najväčší. Ak je ešte stále viac možností, vyberte ľubovoľnú z nich.

### Formát vstupu

V prvom riadku vstupu sú dve medzerou oddelené celé čísla  $N$  a  $M$  ( $4 \leq N \leq 10\,000$  a  $4 \leq M \leq 1\,000\,000$ ).

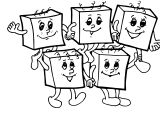
Nasleduje  $M$  riadkov, každý z nich popisuje jednu etapu. Presnejšie, v  $i$ -tom riadku sú tri medzerami oddelené prirodzené čísla  $x_i, y_i, h_i$  ( $1 \leq x_i, y_i \leq N, x_i \neq y_i$  a  $1 \leq h_i \leq 256$ ), kde  $x_i$  a  $y_i$  sú čísla miest, ktoré etapa spája, a  $h_i$  je jej ohodnotenie.

Môžete predpokladať obmedzenia uvedené v časti „Súťažná úloha“: Žiadne dve etapy nespájajú tú istú dvojicu miest a v každom meste končí najvyšš 100 etáp.

### Formát výstupu

Ak žiadna vyhovujúca trasa neexistuje, vypíšte jeden riadok a v ňom reťazec „NEEXISTUJE“ (bez úvodzoviek).

V opačnom prípade vypíšte dva riadky. V prvom z nich uveďte najväčší možný súčet ohodnotení etáp. V druhom vypíšte 5 čísel miest v poradí, v akom ich Tibor má navštíviť. (Prvé mesto musí byť rovnaké ako piate.)



**Príklad 1**

vstup

```
6 9
1 2 10
2 5 11
3 1 10
6 3 7
1 4 3
2 6 15
5 3 10
4 5 5
4 6 9
```

výstup

```
43
2 6 3 5 2
```

**Príklad 2**

vstup

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

výstup

```
NEEXISTUJE
```

**A-III-5 Robot**

Zuzka si nedávno postavila malého robota. Ovládať ho vie jedine tak, mu zadá postupnosť príkazov. Tú robot potom dokola opakuje, až kým ho Zuzka nevypne. Robot pozná len štyri príkazy: otočenie o 90° doľava, doprava, krok vpred a krok vzad.

Zuzka si vymyslela program, ktorý zadala robotovi. Potom na stole nakreslila štvorcovú sieť tak, aby veľkosť políčka zodpovedala dĺžke kroku robota. Robot je maličký, celý sa zmestí na jedno políčko. Na niektoré políčka umiestnila Zuzka prekážky. Ak sa robot pokúsi vykonať krok na políčko s prekážkou, tá ho zastaví, takže zostane na políčku, na ktorom bol. (A samozrejme pokračuje vykonaním ďalšieho príkazu v postupnosti.)

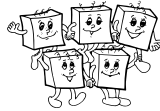
Teraz by ju zaujímalo, kam môže postaviť robota tak, aby zo stola nikdy nespadol.

**Súťažná úloha**

Predpokladajme, že stôl otočíme tak, aby boli jeho strany rovnobežné so svetovými stranami. Zuzka ukladá robota na stôl vždy tak, aby sa pozeral na sever. Robota môže samozrejme položiť len na prázdne políčko. Ak robota položí na políčko, zapne ho, a ten časom zo stola spadne, hovoríme, že jeho štartovacie políčko bolo *nebezpečné*.

Napište program, ktorý načíta postupnosť Zuzkiných príkazov a popis stola a spočíta, koľko je na stole nebezpečných políčok.

Navyše by váš program mal pre každé nebezpečné políčko spočítať, koľko príkazov vykoná robot, ktorý na ňom začína, kým spadne zo stola. (Do tohto počtu rátame aj príkazy „krok vpred/vzad“ pri vykonaní ktorých sa robot nepohol.)



### Formát vstupu

V prvom riadku vstupu je celé číslo  $L$  ( $1 \leq L \leq 500$ ) udávajúce počet príkazov, ktoré Zuzka robotovi zadala.

V druhom riadku je reťazec  $L$  znakov popisujúci túto postupnosť. Znak  $L$  a  $R$  predstavujú otočenie doľava a doprava, znaky  $+$  a  $-$  predstavujú krok o políčko vpred a vzad. (To, ktorým smerom je „vpred“, samozrejme závisí od aktuálneho natočenia robota.)

V treťom riadku sú dve medzerou oddelené celé čísla  $W$  a  $H$  ( $1 \leq W, H \leq 500$ ) udávajúce rozmery stola. V smere zo západu na východ (resp. vo vstupe vodorovne) je na stole  $W$  políčok, v smere zo severu na juh (vo vstupe zvisle) ich je  $H$ .

V posledných  $H$  riadkoch vstupu je mapa stola. Každý riadok obsahuje  $W$  znakov, a každý znak je buď  $.$  (bodka, voľné políčko), alebo  $\#$  (mriežka, obsadené políčko). Prvý znak v prvom riadku predstavuje severozápadný roh mapy.

### Formát výstupu

V prvom riadku vypíšte jedno celé číslo – počet nebezpečných políčok.

Následne vypíšte  $H$  riadkov a v každom z nich  $W$  medzerami oddelených čísel. Presnejšie,  $j$ -te číslo v  $i$ -tom z týchto riadkov má byť počet príkazov, ktoré robot začínajúci na tomto políčku vykoná, kým spadne zo stola. Ak políčko nie je nebezpečné alebo obsahuje prekážku, vypíšte namiesto toho nulu.

#### Príklad 1

vstup

```
16
+R++R+L+++R+R++R
5 3
..#..
....#
...#.
```

výstup

```
7
1 1 0 1 1
0 0 0 4 0
12 0 0 0 3
```

Grafické znázornenie plánu stola (ktorý je v oboch príkladoch rovnaký):

[1; 1]				
				[5; 3]

#### Príklad 2

vstup

```
8
+R++LL+R
5 3
..#..
....#
...#.
```

výstup

```
9
1 1 0 1 1
9 9 0 4 0
17 0 0 0 3
```

Nezabudnite, že robot danú postupnosť príkazov dokola opakuje.