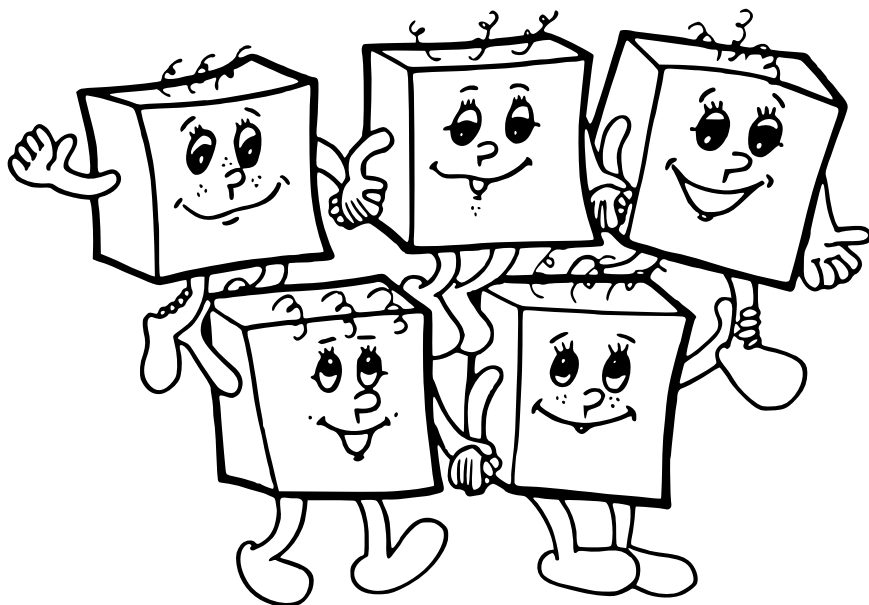


# OLYMPIÁDA V INFORMATIKE

## NA STREDNÝCH ŠKOLÁCH

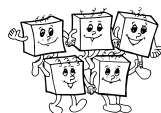


dvadsiaty tretí ročník  
školský rok 2007/08

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://www.ksp.sk/oi/>.



## Informácie a pravidlá

### Priebeh celošťatneho kola

Celošťatne kolo 23. ročníka Olympiády v informatike, kategórie A, sa koná v dňoch 20.–23. apríla 2008. Na riešenie úloh druhého, praktického dňa majú súťažiaci 4 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 druhého súťažného dňa

### A-III-4 O lenivých prasiatkach

Na lúke si polihovalo  $P$  lenivých prasiatok. Až sa priblížil večer a prasiatka sa rozhodli, že keďže sa schýľuje k búrke, pôjdu prespať do senníkov. Na lúke je práve  $P$  senníkov. Je jedno, v ktorom senníku bude ktoré prasiatko spať, ale do každého sa zmestí len jedno.

Prasiatka sú ale lenivé, a tak by sa chceli schovať s čo najmenšou celkovou námahou.

Lúku si môžeme predstaviť ako veľký štvorec, zložený z  $N \times N$  jednotkových štvorcov, ktoré budeme volať políčka. Na niektorých políčkach rastú stromy, tadiaľ prasiatka ísť nemôžu. Ostatné políčka sú voľné. Na niektorých  $P$  voľných políčkach stoja senníky. Na niektorých  $P$  voľných políčkach stoja prasiatka. (Je možné, že niektoré prasiatka už sú na políčkach, kde sú senníky.) V každom okamihu môže byť na každom políčku najviac jedno prasiatko.

Námahu presunu budeme merať na kroky. V jednom kroku sa môže jedno prasiatko presunúť na jedno zo štyroch políčok, ktoré stranou susedia s políčkom, kde práve stojí. (Samozrejme, na políčku nesmie byť strom ani iné prasiatko, a prasiatko nesmie opustiť lúku.)

Navyše vedia naše prasiatka skákať cez seba. Ak by prasiatko  $A$  chcelo ísť smerom, kde stojí prasiatko  $B$ , môže v jednom kroku prasiatko  $B$  preskočiť a dopadnúť až na políčko za ním. (Takto sa vlastne prasiatko  $A$  v jednom kroku posunulo až o dve políčka.)

### Súťažná úloha

Na vstupe je mapa lúky, polohy prasiatok a senníkov.

Zistite, s akou najmenšou námahou sa vedia všetky prasiatka schovať do senníkov.

### Formát vstupu

V prvom riadku vstupu je celé číslo  $N$  ( $1 \leq N \leq 20$ ), udávajúce rozmer lúky.

Nasleduje  $N$  riadkov, ktoré obsahujú mapu lúky. Každý z nich obsahuje  $N$  znakov, a každý znak je buď „.“ (bodka) alebo veľké písmeno „X“. Bodky predstavujú voľné políčka a písmená sú stromy.

V nasledujúcom riadku je celé číslo  $P$  ( $1 \leq P \leq 4$ ), udávajúce počet prasiatok.



Nasleduje  $P$  riadkov, ktoré popisujú polohu prasiatok. Každý z nich obsahuje dve celé čísla od 1 do  $N$  – riadok a stĺpec políčka, kde prasiatko stojí. (Riadky aj stĺpce sú číslované v poradí v akom sú na vstupe.) Všetkých  $P$  riadkov je navzájom rôznych, a všetky popísané políčka sú voľné.

Posledných  $P$  riadkov rovnako popisuje polohu senníkov.

### Formát výstupu

Vypíšte jediný riadok a v ňom jedno celé číslo – najmenšiu námahu, ktorá stačí na to, aby sa všetky prasiatka schovali do senníkov.

Môžete predpokladať, že pre každý použitý testovací vstup bude existovať riešenie.

### Horšie riešenia

V 12 z 15 sád testovacích vstupov bude  $N \leq 10$ . V 10 z týchto 12 sád bude dokonca  $N \leq 8$ .

V aspoň 5 testovacích sádach bude  $P \leq 2$ . V aspoň 2 testovacích sádach bude existovať optimálne riešenie, pri ktorom sa ani raz neskočí.

### Príklady

vstup

```
3
X.X
...
X.X
2
2 1
2 2
3 2
1 2
```

výstup

3

*Najskôr prejde druhé prasiatko z [2, 2] do jedného senníka, napríklad na [3, 2], následne môže prvé prasiatko prejsť na dva kroky do druhého senníka.*

vstup

```
6
.....
.X.X..
.X.XX.
.X..X.
.XXXX.
.....
1
6 5
4 4
```

výstup

13

*Toto prasiatko je na lúke samo. Nemá ako skákať, preto sa proste vyberie najkratšou cestou do senníka. V tomto prípade je výhodnejšie stenu obísť sprava.*

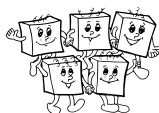
vstup

```
5
X.XXX
X.XXX
.....
X.XXX
X.XXX
2
1 2
3 1
3 5
5 2
```

výstup

7

*Najlepšie riešenie je, aby jedno prasiatko prišlo na križovatku na súradniciach [3, 2] a nechalo druhé cez seba preskočiť.*



vstup

```
7
.....
.....
.....
.....
.....
.XX.XX.
.....
2
1 1
1 7
7 3
7 5
```

výstup

15

Prasiatka sa v hornom riadku vyberú k sebe, stredným stĺpcom sa presunú dole (pričom striedavo cez seba skáču), a tam sa opäť rozídu každé na inú stranu.

vstup

```
7
.....
.....
xxxxx..
....X..
....X..
....X..
....X..
4
1 1
1 2
2 1
2 2
6 6
6 7
7 6
7 7
```

výstup

20

Prasiatka začínajú v ľavom hornom rohu mapy, rozostavené do štvorca. Pri optimálnom presune v každom kroku niektoré prasiatko skáče – kým to ide, skáču voodorovne, keď už to nejde, tak zvisle. Po prvých 10 krokoch budú prasiatka na štyroch políčkach v pravom hornom rohu, po ďalších 10 sa dostanú všetky do senníkov.

### A-III-5 Posledná dobrota

Na stole je obrovská kopa dobrôt: koláčov aj ovocia. Julka a Monika sa s nimi hrajú nasledovnú hru.

Hra prebieha v ťahoch. Prvá ťahá Julka, potom sa striedajú. Hráčka na ťahu si buď môže vziať nejaké koláče, alebo niekoľko kusov ovocia, alebo oboje. Ale ak chce brať oboje, musí si koláčov zobrať toľko isto ako kusov ovocia.

Aby sa neprejedli, dohodli sa navyše na tom, že v jednom ťahu si môže každá z nich vziať najviac  $K$  koláčov a najviac  $K$  kusov ovocia.

Hru vyhrá tá, ktorá zoberie zo stola úplne poslednú dobrotu.

### Súťažná úloha

Vašou úlohou bude napísať knižnicu, ktorá bude túto hru hrať namiesto Julky, proti nášmu programu, ktorý bude hrať namiesto Moniky. Body dostanete za každú hru, ktorú vaša knižnica celú korektne odohrá a vyhrá.

Dostanete od nás k dispozícii kostru knižnice, ktorú máte napísať, a ukázkový program protihráča. Tento protihráč však na rozdiel od toho, ktorého použijeme pri testovaní, nebude hrať optimálne.



### Popis knižnice

Vaša knižnica má obsahovať nasledujúce dve procedúry / funkcie:

```
procedure zaciatok(A, B, K: longint);  
void zaciatok(int A, int B, int K);
```

V tejto funkcii si vaša knižnica môže inicializovať premenné a štruktúry, ktoré potom bude využívať pri hre. Náš program túto funkciu zavolá práve raz, na začiatku celej hry. Pomocou parametrov oznámi náš program vašej knižnici počet  $A$  koláčov na stole, počet  $B$  kusov ovocia na stole a limit na ťah  $K$ .

Vo všetkých testovacích vstupoch bude  $1 \leq A, B \leq 10\,000\,000$  a  $1 \leq K \leq 1\,000\,000$ .

```
procedure tahaj(A, B: longint; var berA, berB: longint);  
void tahaj(int A, int B, int *berA, int *berB);
```

Túto funkciu náš program zavolá vždy, keď je vaša knižnica na ťahu. (Prvýkrát teda hneď po zavolaní funkcie `zaciatok`.) V premenných  $A$  a  $B$  vám oznámi aktuálne počty koláčov a ovocia. Pred skončením musí vaša funkcia nastaviť hodnoty premenných `berA` a `berB` na počty koláčov a ovocia, ktoré chcete v tomto ťahu zo stola zobrať.

### Príklad hry

volanie funkcie

```
zaciatok(7,2,3)  
  
tahaj(7,2,berA,berB)  
  
tahaj(3,1,berA,berB)  
  
tahaj(1,0,berA,berB)
```

popis

Na stole je 7 koláčov, 2 kusy ovocia,  
v jednom ťahu sa dajú vziať max. 3 rovnaké veci.  
Prvý ťah. Julka nastaví `berA` aj `berB` na 1.  
Monika berie 3 koláče, ostávajú 3 koláče a 1 ovocie.  
Julka sa dozvie aktuálny stav hry, zoberie 1 koláč.  
Monika berie po jednom koláči aj ovocí.  
Julka berie posledný koláč a vyhráva.