

# 43. ročník matematické olympiády na středních školách

---

## 6. Mezinárodní olympiáda v informatice

In: Leo Boček (editor); Karel Horák (editor); Pavel Leischner (editor); Jozef Moravčík (editor); Jaromír Šimša (editor); Pavel Töpfer (editor): 43. ročník matematické olympiády na středních školách. Zpráva o řešení úloh ze soutěže konané ve školním roce 1993/1994. 35. mezinárodní matematická olympiáda. 6. mezinárodní olympiáda v informatice. (Czech). Olomouc: Univerzita Palackého v Olomouci, 2016. pp. 92–99.

Persistent URL: <http://dml.cz/dmlcz/405241>

### Terms of use:

Institute of Mathematics of the Czech Academy of Sciences provides access to digitized documents strictly for personal use. Each copy of any part of this document must contain these *Terms of use*.



This document has been digitized, optimized for electronic delivery and stamped with digital signature within the project *DML-CZ: The Czech Digital Mathematics Library* <http://dml.cz>

## 6. mezinárodní olympiáda v informatice



Šestá mezinárodní olympiáda v informatice IOI'94 (International Olympiad in Informatics) se konala od 3. do 10. července 1994 ve městě Haninge nedaleko hlavního města Švédska Stockholmu. Olympiády se zúčastnilo celkem 189 soutěžících (z toho 6 dívek) ze 49 zemí.

Mezinárodní olympiáda v informatice je organizována jako soutěž jednotlivců, každá země na ni může vyslat delegaci tvořenou dvěma vedoucími a nejvýše čtyřmi soutěžícími. Vedoucí delegace se automaticky stává členem mezinárodní jury, jeho zástupce se po dobu soutěže stará o soutěžní družstvo. Soutěžícími jsou studenti středních škol, případně čerství absolventi v příslušném školním roce, ale ve věku do 19 let. Soutěž je řízena mezinárodním výborem IOI a je pořádána pod patronací UNESCO.

Naši soutěžící se zúčastnili všech předchozích ročníků soutěže a po každé dosáhli velmi dobrých výsledků. Společné československé družstvo nás reprezentovalo na prvních čtyřech ročnících IOI v letech 1989–92, loni jelo na IOI'93 do Argentiny poprvé samostatné české a samostatné slovenské družstvo. Reprezentační družstvo pro mezinárodní olympiádu v informatice je každoročně vybíráno na základě výsledků, kterých studenti dosáhli v kategorii P příslušného ročníku matematické olympiády. Pro účast na IOI může být vybrán pouze úspěšný řešitel celostátního kola. Při výběru ovšem nerozhoduje pouze dosažené absolutní pořadí v celostátním kole. Ústřední výbor matematické olympiády přihlíží i k tomu, jakých výsledků jednotliví soutěžící dosáhli při řešení těch úloh 2. a 3. kola kategorie P, které jsou zaměřeny na tvorbu programů a jsou tedy bližší charakteru úloh, jaké se zadávají na IOI. Dále se při výběru berou v úvahu také výsledky dosažené v loňském ročníku MO kategorie P, případně na loňské IOI.

České reprezentační družstvo pro IOI'94 odcestovalo do Švédska v následujícím složení: *Martin Mareš*, student 3. ročníku gymnázia U libeňského zámku v Praze 8, *Daniel Škarda*, absolvent gymnázia Zborovská v Praze 5, *Robert Špalek*, student 2. ročníku gymnázia na tř. kpt. Jaroše v Brně, a *Petr Novák*, absolvent gymnázia na tř. kpt. Jaroše v Brně.

Vedoucím delegace byl jmenován doc. RNDr. *Václav Sedláček*, CSc., z přírodovědecké fakulty Masarykovy univerzity v Brně, jeho zástupcem RNDr. *Pavel Töpfer*, CSc., z matematicko-fyzikální fakulty Univerzity Karlovy v Praze.

Vlastní soutěž byla již tradičně soustředěna do dvou dnů. V každém soutěžním dnu byly studentům zadány k řešení tři úlohy. Soutěžní úlohy byly zvoleny vždy v den jejich řešení mezinárodní porotou složenou z vedoucích delegací všech zúčastněných států. Soutěžící pracovali samostatně u přidělených osobních počítačů typu PC 486, v každém soutěžním dnu měli na práci 5 hodin čistého času. Mohli používat některý ze čtyř předem oznámených programovacích jazyků, jimiž byly Turbo Pascal, Borland C++, LCN Logo a Quick Basic. Výsledné programy pak byly za přítomnosti studenta a vedoucího delegace testovány koordinátory. Hodnotila se jak jejich správnost, tak také rychlost výpočtu pro různá testovací data. Příliš pomalé programy, které nedokázaly vyřešit úlohu ve stanoveném a předem známém časovém limitu, byly hodnoceny jako nevyhovující. Letos bylo poprvé využito automatické testování a hodnocení studentských programů pomocí testovacích a vyhodnocovacích programů připravených předem organizátory soutěže. Na základě výsledků těchto testů byla řešení úloh obodována. Každý den mohl soutěžící získat maximálně 100 bodů. Celkové výsledky byly stanoveny jako součet bodových zisků získaných v obou soutěžních dnech.

Letošní soutěžní úlohy se ukázaly být velmi obtížné a stanovený způsob hodnocení vytvořených programů byl značně přísný. Průměrná dosažená bodová ohodnocení jednotlivých úloh byla proto o dost nižší než v minulých letech. Prvních 101 soutěžících z přítomných 189 bylo oceněno některou z medailí. Celkově bylo uděleno 16 zlatých medailí (za bodový zisk 195–148 bodů), 35 stříbrných medailí (za 145–96 bodů) a 50 bronzových medailí (za 95–65 bodů). Naši studenti navázali na dobrou tradici a opět dosáhli vynikajících výsledků. *Martin Mareš* získal zlatou medaili (162 bodů), *Robert Špalek* (121 bodů) a *Daniel Škarda* (101 bodů) získali medaile stříbrné a *Petr Novák* medaili bronzovou (72 bodů). Soutěž družstev nebyla na IOI vyhlášena a ani žádné pořadí družstev nebylo publikováno. Naše výsledky v soutěži nás ale opět řadí mezi několik nejlepších družstev na světě. Všichni naši reprezentanti byli oceněni některou z medailí.

Mimo vlastní soutěž připravili organizátoři pro všechny účastníky bohatý doprovodný program. K nejzajímavějším akcím patřila vyhlídková

plavba lodí Stockholmem, prohlídka muzea historické lodi Vasa a celodenní výlet parníkem na ostrov Utö.

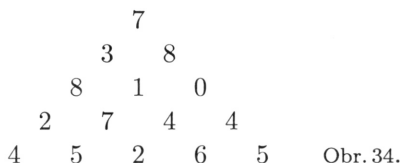
Celá olympiáda byla švédskými organizátory velmi pečlivě připravena, počínaje jejím slavnostním zahájením ve výukovém středisku stockholmské university Riksäppet a konče slavnostním předáváním medailí vítězům soutěže na stockholmské radnici v sále, ve kterém jsou každoročně udělovány Nobelovy ceny nejlepším vědcům světa.

Příští, v pořadí sedmá mezinárodní olympiáda v informatice se bude konat ve dnech 26.6.–3.7. 1995 v Nizozemí ve městě Eindhoven. Přítomní zástupci organizátorů příštího ročníku IOI pozvali na tuto olympiádu všechny země zúčastněné v letošním ročníku. Pořadatelé počítají s rozšířením počtu členů soutěžních družstev na pět studentů a dva vedoucí. Podmínkou účasti pátého soutěžícího je nominace alespoň jedné dívky do družstva. Další ročníky IOI uspořádají postupně Maďarsko v roce 1996, Jihoafrická republika v roce 1997, Portugalsko v roce 1998, Turecko v roce 1999, Čína v roce 2000, USA nebo Thajsko nebo Irsko v roce 2001 a Korea v roce 2002. Nejbližší ročník středoevropské regionální olympiády v informatice uspořádá v květnu 1995 Maďarsko.

Účast na takovéto světové soutěži představuje pro každého nejen náročný úkol reprezentovat co nejlépe naši zemi v zahraničí, ale je také příležitostí podívat se někam do světa, poznat mnoho zajímavých míst i lidí a třeba také vyhrát některou z hodnotných cen. Každý ze současných i budoucích studentů středních škol, kdo se více zajímá o programování, má možnost dostat se za rok nebo v některém z dalších let na IOI. Stačí jenom začít řešit úlohy kategorie P matematické olympiády, věnovat přípravě na olympiádu potřebné úsilí a část svého volného času a pak také mít trochu štěstí, jako ostatně v každé soutěži. Zcela bez šance je ale jenom ten, kdo to vůbec nezkusí.

### Texty soutěžních úloh

**1. Trojúhelník** Obrázek 34 znázorňuje trojúhelník čísel. Napište program, který spočítá maximální součet čísel podél cesty začínající v horním vrcholu trojúhelníku a končící někde na jeho spodní základně.



- ▷ V každém kroku cesty můžete jít šikmo dolů směrem doleva nebo šikmo dolů směrem doprava.
- ▷ Počet řádků v trojúhelníku je větší než 1 a menší nebo roven 100.
- ▷ Všechna čísla v trojúhelníku jsou celá z rozmezí od 0 do 99.

**Vstupní data.** Vstupní soubor `INPUT.TXT` obsahuje v prvním řádku počet řádků trojúhelníku. Na dalších řádcích souboru jsou uvedena čísla tvořící jednotlivé řádky trojúhelníku. Pro výše uvedený příklad trojúhelníku má vstupní soubor `INPUT.TXT` následující tvar:

```

5
7
3 8
8 1 0
2 7 4 4
4 5 2 6 5

```

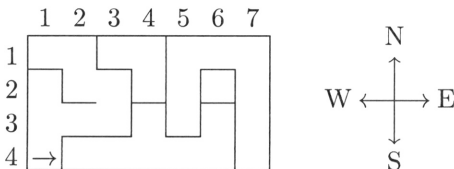
**Výstupní data.** Do výstupního souboru `OUTPUT.TXT` je zapsáno jedno celé číslo udávající nalezený maximální součet. Pro náš příklad bude výstupní soubor `OUTPUT.TXT` obsahovat:

30

**2. Hrad** Obrázek 35 znázorňuje plánec hradu. Napište program, který určí

- 1) kolik má hrad místností,
- 2) jak velká je největší místnost,
- 3) která jedna zeď hradu se má odstranit, aby vznikla co největší místnost.

Hrad se skládá ze sítě  $mn$  ( $m \leq 50, n \leq 50$ ) jednotkových čtvercových modulů. Každý z těchto modulů může mít žádnou, jednu, dvě, tři nebo čtyři zdi.



Obr. 35.

**Vstupní data.** Plánek je uložen ve vstupním souboru `INPUT.TXT` a je zakódován pomocí čísel. Každému modulu odpovídá jedno číslo.

- ▷ V souboru je uložen v prvním řádku počet modulů ve směru sever-jih a ve druhém počet modulů ve směru západ-východ.

- ▷ V následujících řádcích je každý modul zakódován jedním číslem z rozmezí od 0 do 15. Toto číslo je rovno součtu kódů stěn ohraničujících modul:

1 = západní stěna  
 2 = severní stěna  
 4 = východní stěna  
 8 = jižní stěna.

Vnitřní zdi jsou tedy uvedeny dvakrát; například jižní stěna modulu 1,1 je zároveň severní stěnou modulu 2,1.

- ▷ Zadaný hrad má vždy nejméně dvě místnosti.

Vstupní soubor `INPUT.TXT` pro příklad z obr. 2:

```

4
7
11 6 11 6 3 10 6
7 9 6 13 5 15 5
1 10 12 7 13 7 5
13 11 10 8 10 12 13

```

**Výstupní data.** Výstupní soubor `OUTPUT.TXT` obsahuje tři řádky. Na prvním řádku je uveden počet místností, na druhém je velikost největší místnosti (měřena počtem jednotkových modulů) a na třetím je návrh, kterou zeď je třeba odstranit. Odstraňovaná zeď je určena takto: nejprve řádek modulu, potom sloupec modulu a nakonec světová strana udávající stěnu v modulu pomocí anglické zkratky světové strany (viz obr. 2).

V našem příkladě bude výstupní soubor `OUTPUT.TXT` obsahovat:

```

5
9
4 1 E

```

*Poznámka:* „4 1 E“ je jednou z více možností správného řešení.

**3. Prvočísla** Na obr. 36 je znázorněn čtverec složený z pěti řádků a pěti sloupců, přičemž v každém políčku čtverce je zapsána jedna dekadická číslice. Každý řádek, každý sloupec a každou z hlavních diagonál čtverce budeme číst jako pěticiferné prvočíslu. Řádky čtverce se čtou zleva doprava. Sloupce se čtou shora dolů. Obě diagonály se čtou zleva doprava.

1	1	3	5	1
3	3	2	0	3
3	0	3	2	3
1	4	0	3	3
3	3	3	1	1

Obr. 36

Napište program, který na základě dat obsažených ve vstupním souboru `INPUT.TXT` vytváří takové čtverce.

- ▷ Všechna prvočísla musí mít stejný ciferný součet (v našem příkladě je to 11).
- ▷ Číslice v levém horním rohu čtverce je předem pevně dána (v našem příkladě je to 1).
- ▷ Stejně prvočíslo může být v jednom čtverci obsaženo vícekrát.
- ▷ Má-li úloha více řešení, program musí nalézt všechna řešení.
- ▷ Pěticiferné prvočíslo nemůže začínat vedoucími nulami, tzn. 00003 není pěticiferné prvočíslo.

**Vstupní data.** Program čte data ze vstupního souboru `INPUT.TXT`, který obsahuje dva řádky. Na prvním je uveden ciferný součet prvočísel, na druhém řádku je číslice z levého horního rohu čtverce. Můžete předpokládat, že pro daná testovací data vždy existuje řešení. Pro náš příklad vypadá vstupní soubor `INPUT.TXT` takto:

```
11
1
```

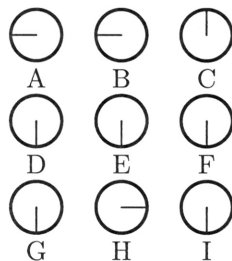
**Výstupní data.** Program zapíše pro každé nalezené řešení pět řádků do výstupního souboru `OUTPUT.TXT`. Každý řádek řešení je tvořen pěti ciframi prvočísla. Výše uvedený příklad má tři řešení, takže soubor `OUTPUT.TXT` bude obsahovat (prázdné řádky mezi jednotlivými řešeními jsou nepovinné):

```
1 1 3 5 1
1 4 0 3 3
3 0 3 2 3
5 3 2 0 1
1 3 3 1 3

1 1 3 5 1
3 3 2 0 3
3 0 3 2 3
1 4 0 3 3
3 3 3 1 1

1 3 3 1 3
1 3 0 4 3
3 2 3 0 3
5 0 2 3 1
1 3 3 3 1
```

**4. Hodiny** Na obr. 37 je devět ciferníků hodin uspořádaných ve tvaru tabulky  $3 \times 3$ . Cílem je, aby všechny ukazovaly 12 hodin. Máte 9 dovolených způsobů (budeme je nazývat tahy) na změnu nastavení polohy ručiček. V každém kroku si zvolíte tah určený číslem 1 až 9. Pokud si ciferníky po řádcích zleva doprava označíme A, B, C, D, E, F, G, H, I, pak tahem 1 až 9 přiřadíme tato otočení:



Obr. 37

- |         |          |         |
|---------|----------|---------|
| 1: ABDE | 2: ADG   | 3: DEGH |
| 4: ABC  | 5: BDEFH | 6: GHI  |
| 7: BCEF | 8: CFI   | 9: EFHI |

Podle takto zvoleného čísla pootočíte o 90 stupňů (ve směru otáčení hodinových ručiček) ručičky těch ciferníků, které jsou označeny.

**Vstupní údaje.** Ze souboru `INPUT.TXT` přečtete 9 čísel. Tato čísla udávají výchozí pozici na cifernících: 0 znamená 12 hodin, 1 znamená 3 hodiny, 2 znamená 6 hodin a 3 znamená 9 hodin.

**Výstupní údaje.** Do výstupního souboru `OUTPUT.TXT` napište nějakou nejkratší posloupnost tahů, která nastaví všechny ciferníky na 12 hodin. Pokud má úloha více řešení, vypište pouze jedno z nich.

**5. Autobusy** Na autobusové zastávce staví několik autobusových linek. Člověk přijel na zastávku přesně ve 12:00 hod. Zůstal zde od 12:00 do 12:59 hod. a zaznamenával si časy příjezdů všech autobusů na zastávku.

- ▷ Autobusy každé linky přijíždějí na zastávku po celou dobu od 12:00 do 12:59 v pravidelných intervalech.
- ▷ Časy příjezdů jsou dány v celých minutách jako celá čísla od 0 do 59.
- ▷ Autobusy každé linky staví na zastávce v uvažovaném časovém intervalu alespoň dvakrát.
- ▷ Počet autobusových linek v testovacích příkladech bude nejvýše 17.
- ▷ Autobusy různých linek mohou přijet na zastávku současně.
- ▷ Více autobusových linek může mít stejný čas prvního příjezdu na zastávku anebo stejný interval. I když mají dvě autobusové linky stejný čas prvního příjezdu a stejný interval, jedná se o odlišné linky a obě musí být uvedeny.

Určete nejmenší počet autobusových linek, které musí zastavovat na autobusové zastávce, aby řešení vyhovovalo vstupním datům. Pro každou autobusovou linku vypište čas prvního příjezdu autobusu na zastávku a interval mezi příjezdy.



**Vstupní data.** Vstupní soubor `INPUT.TXT` obsahuje nejprve číslo  $N$  ( $N \leq 300$ ), které udává, kolik autobusů přijelo na zastávku. Následují časy příjezdů všech  $N$  autobusů ve vzestupném pořadí. Příklad:

```
17
0 3 5 13 13 15 21 26 27 29 37 39 39 45 51 52 53
```

**Výstupní data.** Výstupní soubor `OUTPUT.TXT` obsahuje údaje o každé autobusové lince. Na jednom řádku jsou uvedeny údaje vždy o jedné lince. Každý z těchto řádků obsahuje čas příjezdu prvního autobusu na zastávku a interval mezi příjezdy autobusů této linky v minutách. Nezáleží na tom, v jakém pořadí budou jednotlivé autobusové linky na výstupu uvedeny. Pokud existuje více řešení, uveďte pouze jedno z nich. Pro výše uvedený příklad vstupních dat bude výstupní soubor obsahovat:

```
0 13
3 12
5 8
```

**6. Kruh** Máte kruh rozdělen do sektorů. Jsou dány tři čísla:  $n$  ( $n \leq 6$ ),  $m$  ( $m \leq 20$ ) a  $k$  ( $k \leq 20$ ), kde  $n$  je počet sektorů. Napište program, který vybere a umístí celá čísla do každého sektoru; tato čísla mají být větší nebo rovno  $k$ . Po naplnění sektorů můžete vytvářet nová čísla použitím čísel z jednoho sektoru nebo sečtením čísel ze dvou nebo více sousedních sektorů.

Z nově vytvořených čísel máte vytvořit souvislou posloupnost všech celých čísel mezi  $m$  a  $i$  (tj.  $m, m + 1, m + 2, \dots, i$ ). Úkolem programu je vybrat čísla do sektorů tak, aby největší číslo posloupnosti ( $i$ ) bylo největší možné.

**Vstupní údaje.** Vstupní soubor `INPUT.TXT` obsahuje tři celá čísla ( $n, m, k$ ).

**Výstupní údaje.** Výstupní soubor `OUTPUT.TXT` musí obsahovat:

- ▷ největší číslo v posloupnosti ( $i$ ), které může být vygenerované,
- ▷ všechny uspořádání čísel do kruhových sektorů, které vytvářejí posloupnost od  $m$  do  $i$  (do každého řádku jedno). Uspořádání čísel zapište jako seznam začínající nejmenším číslem (které nemusí být pouze jedno).

Uvědomte si, že pokud by (1123) bylo řešením, musíte vypsát i seznamy (1321), (1231) a (1132).