

MATEMATICKÁ OLYMPIÁDA NA STŘEDNÍCH ŠKOLÁCH

kategorie A, B, C a P

61. ROČNÍK, 2011/2012

<http://math.muni.cz/mo>

Studenti středních škol,

zveme vás k účasti v matematické olympiádě, jejíž soutěžní kategorie A, B, C a P pořádáme právě pro vás.

Kategorie A je určena žákům maturitních a předmaturitních ročníků,

kategorie B žákům, kterým do maturity zbývá více než 2 roky,

kategorie C žákům, kterým do maturity zbývá více než 3 roky,

kategorie P je zaměřena na programování a je určena žákům všech ročníků.

Podrobnější rozdělení uvádí následující tabulka:

| Předpokládaný školní rok ukončení studia maturitou | kategorie MO |
|---|--------------|
| 2011/2012 | A |
| 2012/2013 | A |
| 2013/2014 | B |
| 2014/2015 | C |

Žáci nižších ročníků víceletých gymnázií soutěží v MO společně s žáky základních škol v kategoriích Z6 až Z9. Jim je věnován zvláštní leták.

Průběh soutěže v kategoriích A, B, C: V kategorii A probíhá soutěž ve třech kolech (školním, krajském a ústředním), v kategoriích B a C probíhá ve dvou kolech (školním a krajském).

Školní kolo má dvě části – domácí a klauzurní. V **domácí části** na vás čeká šest úloh, které najdete v tomto letáku. Jejich řešení (ne nutně všech) odevzdejte svému učiteli matematiky do **28. listopadu 2011** (kategorie A) a do **16. ledna 2012** (kategorie B, C). Ten je opraví, ohodnotí podle stupnice *1 – výborně, 2 – dobře, 3 – nevyhovuje*. Pak je s vámi rozebere, vysvětlí vám případné nedostatky a seznámí vás se správným řešením, které také najdete na našich internetových stránkách. Jestliže budou vaše řešení alespoň čtyř úloh ohodnocena jako výborná nebo dobrá, budete pozváni do **klauzurní části** školního kola. Tam budete ve stanoveném čase samostatně řešit další tři úlohy.

Nejlepší účastníci školního kola budou pozváni do **krajského kola**. Tam budou během čtyř hodin samostatně řešit čtyři úlohy. Podle rozhodnutí

Ústřední komise MO z března 2011 nebudou od školního roku 2011/2012 v klauzurních kolech MO povoleny kalkulačky, notebooky ani žádné jiné elektronické pomůcky. O pořadí v krajských kolech soutěže rozhoduje součet bodů získaných za jednotlivé úlohy, a to 0 až 6 bodů za každou z nich. Pokud prvních n žáků dosáhne stejného počtu bodů, je jejich pořadí označeno shodně 1.– n . místem. Podobně se určí pořadí na dalších místech. Žádná jiná kritéria nejsou přípustná.

V kategorii A budou ještě nejlepší řešitelé krajského kola z celé republiky soutěžit v **ústředním kole**, a to za podmínek podobných jako na **mezinárodní matematické olympiádě**, kde během soutěže lze používat pouze psací a rýsovací potřeby. Právě pro ni se z vítězů ústředního kola vybere družstvo České republiky.

Průběh soutěže v kategorii P: Ve **školním kole** řešíte jen čtyři úlohy uvedené v tomto letáku spolu s pokyny, jak máte naložit s řešeními, která nebudete odevzdávat ve škole.

V kategorii P se nekoná klauzurní část školního kola, takže úspěšní řešitelé domácích úloh budou pozváni přímo do **krajského kola**. Stejně jako v kategorii A se i v kategorii P koná **ústřední kolo**, jehož vítězové se zúčastní každoroční **mezinárodní olympiády v informatice**.

Termíny soutěžních kol 61. ročníku MO jsou stanoveny takto:

| | I. kolo (školní část) | II. kolo (krajské) | III. kolo (ústřední) |
|----------------|--------------------------|-----------------------|-------------------------|
| Kategorie A | 6.12. 2011 | 17.1. 2012 | 25.3.–28.3. 2012 |
| Kategorie B, C | 26.1. 2012 | 17.4. 2012 | — |
| Kategorie P | — | 24.1. 2012 | 28.3.–31.3. 2012 |

MO pořádají *Ministerstvo školství, mládeže a tělovýchovy ČR, Jednota českých matematiků a fyziků a Matematický ústav Akademie věd České republiky*. Soutěž organizuje *ústřední komise MO* a v krajích ji řídí *krajské komise MO* při pobočkách JČMF. Na jednotlivých školách ji zajišťují pověřeni učitelé matematiky, na které se můžete s otázkami kolem MO kdykoli obracet.

Řešení soutěžních úloh vypracujte čitelně na listy formátu A4. Každou úlohu začněte na novém listě a uveďte vlevo nahoře záhlaví podle vzoru:

Karel Smutný
2. D, gymnázium
Kulaté nám. 9, 629 79 Lužany
B–I–4

Poslední údaj je označení úlohy podle tohoto letáku. Znění úloh nemusíte opisovat. Nevejde-li se vám řešení na jeden list, uveďte na dalších listech vlevo nahoře své jméno a označení úlohy a očísľujte stránky. **Řešení pište jako výklad, v kterém jsou uvedeny všechny podstatné úvahy tak, aby bylo možno sledovat váš myšlenkový postup.**

Kategorie A

1. Označme n součet všech desetimístných čísel, která mají ve svém dekadickém zápise každou z číslic $0, 1, \dots, 9$. Zjistěte zbytek po dělení čísla n sedmdesáti sedmi. (Pavel Novotný)
2. Na setkání bylo několik lidí. Každí dva, kteří se neznali, měli mezi ostatními přítomnými právě dva společné známé. Účastníci A a B se znali, ale neměli ani jednoho společného známého. Dokažte, že A i B měli mezi přítomnými stejný počet známých. Ukažte rovněž, že na setkání mohlo být právě šest osob. (Vojtech Bálint)
3. Označme S střed kružnice vepsané, T těžiště a V průsečík výšek daného rovnoramenného trojúhelníku, který není rovnostranný.
 - a) Dokažte, že bod S je vnitřním bodem úsečky TV .
 - b) Určete poměr délek stran daného trojúhelníku, je-li bod S středem úsečky TV . (Jaromír Šimša)

4. Nechtě p, q jsou dvě různá prvočísla, m, n přirozená čísla a součet

$$\frac{mp - 1}{q} + \frac{nq - 1}{p}$$

je celé číslo. Dokažte, že platí nerovnost

$$\frac{m}{q} + \frac{n}{p} > 1.$$

(Jaromír Šimša)

5. Jsou dány dvě shodné kružnice k_1, k_2 o poloměru rovném vzdálenosti jejich středů. Jejich průsečíky označme A a B . Na kružnici k_2 zvolme bod C tak, že úsečka BC protne kružnici k_1 v bodě různém od B , který označme L . Přímka AC protne kružnici k_1 v bodě různém od A , který označme K . Dokažte, že přímka, na níž leží těžnice z vrcholu C trojúhelníku KLC , prochází pevným bodem nezávislým na poloze bodu C . (Tomáš Jurík)

6. Najděte největší reálné číslo k takové, že nerovnost

$$\frac{2(a^2 + kab + b^2)}{(k + 2)(a + b)} \geq \sqrt{ab}$$

platí pro všechny dvojice kladných reálných čísel a, b . (Ján Mazák)

Kategorie B

1. Mezi všemi desetimístnými čísly dělitelnými jedenáctí, v nichž se žádná číslice neopakuje, najděte nejmenší a největší. (*Jaroslav Zhouf*)
2. Je dán pravoúhlý trojúhelník ABC s pravým úhlem při vrcholu C , jehož obsah označme P . Nechť F je pata výšky z vrcholu C na přeponu AB . Na kolmicích k příince AB , které procházejí vrcholy A a B , v polorovině opačné k polorovině ABC uvažujme po řadě body D a E , pro něž platí $|AF| = |AD|$ a $|BF| = |BE|$. Obsah trojúhelníku DEF označme Q . Dokažte, že platí $P \geq Q$, a zjistěte, kdy nastane rovnost. (*Jaroslav Švrček*)

3. Najděte všechny dvojice reálných čísel x, y , které vyhovují soustavě rovnic

$$\begin{aligned}x \cdot \lfloor y \rfloor &= 7, \\ y \cdot \lfloor x \rfloor &= 8.\end{aligned}$$

(Zápis $\lfloor a \rfloor$ značí *dolní celou část* čísla a , tj. největší celé číslo, které nepřevyšuje a .) (*Pavel Novotný*)

4. Jsou dány dvě různoběžky a, c procházející bodem P a bod B , který na nich neleží. Sestrojte pravoúhelník $ABCD$ s vrcholy A, C a D po řadě na přímkách a, c a PB . (*Jaromír Šimša*)
5. V jistém městě mají vybudovanou síť na šíření pomluv, v níž si každý pomlouvač vyměňuje informace se třemi pomlouvačkami a každá pomlouvačka si vyměňuje informace se třemi pomlouvači. Jinak se pomluvy nešíří.
 - a) Dokažte, že pomlouvačů a pomlouvaček je stejný počet.
 - b) Předpokládejme, že síť na pomlouvání je souvislá (pomluvy od libovolného pomlouvače a libovolné pomlouvačky se mohou dostat ke všem ostatním). Dokažte, že i když jeden pomlouvač zemře, zůstane síť souvislá. (*Ján Mazák*)
6. Anna a Bedřich hrají karetní hru. Každý z nich má pět karet s hodnotami 1 až 5 (od každé jednu). V každém z pěti kol oba vyloží jednu kartu a kdo má vyšší číslo, získá bod. V případě karet se stejnými čísly nezíská bod nikdo. Použité karty se do hry nevracejí. Kdo získá na konci více bodů, vyhrál. Kolik procent ze všech možných průběhů takové hry skončí výhrou Anny? (*Tomáš Jurík*)

Kategorie C

1. Najděte všechny trojčleny $p(x) = ax^2 + bx + c$, které dávají při dělení dvojitčlenem $x+1$ zbytek 2, při dělení dvojitčlenem $x+2$ zbytek 1, přičemž $p(1) = 61$.
(Jaromír Šimša)

2. Délky stran trojúhelníku jsou v metrech vyjádřeny celými čísly. Určete je, má-li trojúhelník obvod 72 m a je-li nejdelší strana trojúhelníku rozdělena bodem dotyku vepsané kružnice v poměru 3 : 4.
(Pavel Leischner)

3. Najděte všechny trojice přirozených čísel a, b, c , pro něž platí množinová rovnost

$$\{(a, b), (a, c), (b, c), [a, b], [a, c], [b, c]\} = \{2, 3, 5, 60, 90, 180\},$$

kde (x, y) a $[x, y]$ značí po řadě největší společný dělitel a nejmenší společný násobek čísel x a y .
(Tomáš Jurík)

4. Reálná čísla a, b, c, d vyhovují rovnici $ab + bc + cd + da = 16$.
a) Dokažte, že mezi čísly a, b, c, d se najdou dvě se součtem nejvýše 4.
b) Jakou nejmenší hodnotu může mít součet $a^2 + b^2 + c^2 + d^2$?
(Ján Mazák)

5. Je dán rovnoramenný trojúhelník se základnou délkou a a rameny délky b . Pomocí nich vyjádřete poloměr R kružnice opsané a poloměr r kružnice vepsané tomuto trojúhelníku. Pak ukažte, že platí $R \geq 2r$, a zjistěte, kdy nastane rovnost.
(Leo Boček)

6. Na hrací desce $n \times n$ tvořené bílými čtvercovými poli se Markéta a Tereza střídají v tazích jedním kamenem při následující hře. Nejprve Markéta umístí kámen na libovolné pole a toto pole obarví modře. Dále vždy hráčka, která je na tahu, provede s kamenem *skok* na pole, které je dosud bílé, a toto pole obarví modře. Přitom *skokem* rozumíme obvyklý tah šachovým jezdcem, tj. přesun kamene o dvě pole svisle nebo vodorovně a současně o jedno pole v druhém směru. Hráčka, která je na řadě a již nemůže táhnout, prohrává. Postupně pro $n = 4, 5, 6$ rozhodněte, která z hráček může hrát tak, že vyhraje nezávisle na tazích druhé hráčky.
(Pavel Calábek)

KATEGORIE P

Úlohy P–I–1 a P–I–2 jsou praktické, vaším úkolem v nich je vytvořit a odladit efektivní program v jazyce Pascal, C nebo C++. Řešení těchto dvou úloh odevzdávejte ve formě zdrojového kódu přes webové rozhraní přístupné na stránce <http://mo.mff.cuni.cz/submit/>, kde také naleznete další informace. Odevzdaná řešení budou automaticky vyhodnocena pomocí připravených vstupních dat a výsledky vyhodnocení se dozvíte krátce po odevzdání. Pokud váš program nezíská plný počet bodů, můžete své řešení opravit a znovu odevzdat.

Úlohy P–I–3 a P–I–4 jsou teoretické, vaším úkolem je nalézt efektivní algoritmus řešící zadaný problém. Řešení úlohy se skládá z popisu navrženého algoritmu, zdůvodnění jeho správnosti (funkčnosti) a odhadu časové a paměťové složitosti. Součástí řešení je i zápis algoritmu ve formě zdrojového kódu nebo pseudokódu v úloze P–I–3 a výsledný zlomkový program v úloze P–I–4. Řešení těchto dvou úloh odevzdávejte ve formě souboru typu PDF přes výše uvedené webové rozhraní.

Řešení všech úloh můžete odevzdávat do 15. listopadu 2011. Opravená řešení úloh a seznam postupujících do krajského kola najdete na webových stránkách olympiády na adrese <http://mo.mff.cuni.cz/>, kde jsou také k dispozici další informace o kategorii P.

P–I–1 Bezpečná planeta

Ve vesmíru existují tři typy planet:

- ▷ *typ 0: neobyvatelné* planety, na nichž člověk okamžitě zahyne,
- ▷ *typ 1: nehostinné* planety, na nichž člověk může chvíli přežít, ale nemůže tam žít dlouhodobě,
- ▷ *typ 2: přívětivé* planety, na nichž člověk může žít dlouhodobě.

Mezi jednotlivými planetami cestujeme vesmírem pomocí *jednosměrných* teleportů.

Planetu nazveme *bezpečnou*, jestliže je přívětivá a zároveň jsou přívětivé také všechny planety, na které se z ní můžeme dostat pomocí teleportů (třeba i několika po sobě). Pokud tedy vysadíte člověka na bezpečné planetě, máte jistotu, že může spokojeně žít všude, kam z ní docestuje.

Planetu označíme jako *snositelnou*, jestliže sice není bezpečná, ale můžeme se z ní pomocí teleportů (třeba i několika po sobě) dostat na bezpečnou planetu, aniž abychom museli cestou navštívit nějakou neobyvatelnou planetu. Pokud tedy vysadíte člověka na snositelné planetě a dáte mu vhodné instrukce, jak má cestovat, po čase dorazí živý a zdravý na nějakou bezpečnou planetu. Všimněte si, že žádná neobyvatelná planeta nemůže být snositelná.

Soutěžní úloha. Je dán počet planet n , počet teleportů m , pro každou planetu její typ a pro každý teleport odkud kam vede. Určete všechny bezpečné a všechny snositelné planety.

Formát vstupu: První řádek standardního vstupu obsahuje dvě celá čísla n (počet planet) a m (počet teleportů). Všechny planety si očíslováme od 1 do n . Druhý řádek vstupu obsahuje n mezerami oddělených celých čísel z množiny $\{0, 1, 2\}$ — postupně pro každé i je to typ planety číslo i .

Zbytek vstupu tvoří m řádků, z nichž každý popisuje jeden teleport. Popis teleportu je tvořen dvěma čísly planet — odkud a kam tento teleport vede.

Ve vstupech, za které lze získat celkem 5 bodů, bude $1 \leq n \leq 500$ a $0 \leq m \leq 10\,000$.

V ostatních vstupech bude $1 \leq n \leq 100\,000$ a $0 \leq m \leq 200\,000$.

Formát výstupu: Program vypíše na standardní výstup dva řádky. První z nich začíná řetězcem „**bezpecne:**“ a za ním následuje posloupnost čísel všech bezpečných planet. Čísla planet vypíše v rostoucím pořadí a před každým z nich vypíše jednu mezeru. Druhý řádek výstupu začíná řetězcem „**snesitelne:**“, za kterým následuje posloupnost čísel všech snesitelných planet ve stejném formátu, jako na prvním řádku. Je-li některá z posloupností prázdná, pak bezprostředně za dvojtečkou bude následovat konec řádku.

Příklad:

Vstup:

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

Výstup:

```
bezpecne: 3 4 5
snesitelne: 1 2
```

P-I-2 Naložená loď

Na planety, které ještě nemají vybudovány teleportové terminály, létají meziplanetární kosmické lodě s lidskou posádkou. A ta potřebuje jíst. Vaší úlohou bude přesně naplnit celý potravinový nákladní prostor kosmické lodě balíčky s jídlem.

Problém spočívá v tom, že balíčky s jídlem se připravují na Zemi, zatímco loď čeká na oběžné dráze. Balíčky je pak třeba dopravit ze Země na kosmickou loď. Vezme se vždy několik balíčků, naloží se do dopravní kapsle, ta se vystřelí ze Země na oběžnou dráhu a tam už si ji posádka lodě odchyťí.

Existuje n typů dopravních kapslí. Kapsle i -tého typu má kapacitu přesně a_i balíčků. Nevejde se do ní více balíčků, ale zároveň jich nesmí být ani méně kvůli správnému vyvážení a doletu. Máme k dispozici dostatečné množství kapslí všech typů.

Soutěžní úloha. Napište program, který na základě zadaných typů kapslí a jejich kapacit spočítá *nejmenší možný počet* kapslí potřebný k úplnému naložení kosmické lodě. Součet kapacit použitých kapslí musí tedy být *přesně roven* kapacitě k nákladního prostoru lodě.

Formát vstupu: První řádek standardního vstupu obsahuje jedno celé číslo n — počet typů kapslí. Druhý řádek obsahuje n celých čísel a_1, \dots, a_n oddělených mezerami — kapacity kapslí jednotlivých typů. Třetí řádek obsahuje jedno celé číslo k — kapacitu nákladního prostoru lodě.

Můžete předpokládat, že $a_1 < \dots < a_n$ a že $a_1 = 1$ (tedy vždy existuje způsob, jak lze přesně celou loď naložit).

- ▷ Ve vstupech za 3 body bude platit $n \leq 5$, $a_n \leq 2\,000$ a $k \leq 30$.
- ▷ Ve vstupech za další 4 body bude platit $n \leq 30$, $a_n \leq 2\,000$ a $k \leq \leq 1\,000\,000$.
- ▷ Ve vstupech za zbývající 3 body bude platit $n \leq 30$, $a_n \leq 2\,000$ a $k \leq \leq 10^{18}$.

Všimněte si, že pro uložení hodnoty k musíte použít 64-bitovou proměnnou: `long long` v C/C++, `int64` v Pascalu.

Formát výstupu: Na první řádek standardního výstupu program vypíše nejmenší počet kapslí potřebný k přesnému naložení naší kosmické lodě.

Na druhý řádek vypíše jedno libovolné řešení, které využívá tento počet kapslí. Řádek bude obsahovat n celých čísel b_1, \dots, b_n oddělených mezerami, přičemž b_i udává počet použitých kapslí velikosti a_i .

Příklady:

| | |
|---------------|----------------|
| <i>Vstup:</i> | <i>Výstup:</i> |
| 4 | 12 |
| 1 10 100 1000 | 7 4 1 0 |
| 147 | |

Použijeme jednu kapsli s kapacitou 100, čtyři kapsle s kapacitou 10 a sedm kapslí s kapacitou 1. To je cekem $1 + 4 + 7 = 12$ kapslí.

| | |
|---------------|----------------|
| <i>Vstup:</i> | <i>Výstup:</i> |
| 3 | 2 |
| 1 8 10 | 0 2 0 |
| 16 | |

Optimální řešení využívá dvě kapsle s kapacitou 8.

P–I–3 Skladník

Ignác byl skladníkem. Pil alkohol, sprostě nadával a v pracovní době většinou spal. Není divu, že mu šéf už dlouhá léta vyhrožoval, že ho nahradí počítačem. Až jednoho dne ta chvíle opravdu nastala. Šéf koupil počítač,

posadil ho na vrátnici skladu a Ignáce propustil. Tím dosáhl přesně stejného stavu jako dosud, jenom počítači už nemusel platit žádnou mzdu.

Netrvalo dlouho a šéf si uvědomil, že počítač dokonce může zapnout a používat. Potřebuje k tomu ale od vás napsat vhodný program.

a) (*4 body*) Navrhněte co nejefektivnější algoritmus, který bude spravovat inventář skladu. Na vstup algoritmu budou přicházet informace o tom, kolik kusů jakého druhu zboží přibýlo nebo ubylo. Po každém načtení nové informace musí váš algoritmus vypsat dva údaje: celkový počet kusů zboží ve skladu a počet různých druhů zboží ve skladu.

b) (*6 bodů*) Napište ještě druhý algoritmus, který bude rovněž spravovat inventář skladu. Jeho vstup bude vypadat stejně jako vstup prvního algoritmu. Po každém načtení nové informace musí tento algoritmus vypsat dva údaje: název druhu zboží, od kterého je momentálně ve skladu největší počet kusů, a tento počet kusů. Pokud existuje více takových druhů zboží, algoritmus vypíše ten z nich, jehož název je první v abecedním pořadí.

Formát vstupu: Na každém řádku vstupu je uvedeno nejprve jedno nenulové celé číslo δ_i a za ním jeden řetězec s_i tvořený nejvýše l znaky anglické abecedy. Tento řádek znamená, že počet kusů zboží s_i ve skladu se změnil o δ_i .

Při návrhu algoritmů můžete předpokládat, že $l \leq 50$, že počet druhů zboží ve skladu nikdy nepřekročí 100 000 a že celkový počet kusů zboží ve skladu nikdy nepřekročí 10^{18} . Můžete také předpokládat, že nikdy nedostanete pokyn, který by počet kusů nějakého zboží ve skladu změnil na záporný.

Příklad:

| <i>Vstup:</i> | <i>Výstup pro podúlohu a:</i> | <i>Výstup pro podúlohu b:</i> |
|---------------|-------------------------------|-------------------------------|
| +3 koberec | kusů: 3, typů: 1 | koberec 3 |
| +2 buldozer | kusů: 5, typů: 2 | koberec 3 |
| +1 buldozer | kusů: 6, typů: 2 | buldozer 3 |
| +7 zebrik | kusů: 13, typů: 3 | zebrik 7 |
| -3 buldozer | kusů: 10, typů: 2 | zebrik 7 |
| -5 zebrik | kusů: 5, typů: 2 | koberec 3 |

Hodnocení

Každá podúloha se hodnotí samostatně.

Ve svém řešení uveďte odhad, jak *nejdéle* (tj. v nejhorším možném případě) může každému z vašich programů trvat zpracování n -tého pokynu. (V tomto odhadu můžete použít proměnnou t pro aktuální počet druhů zboží ve skladu, proměnnou k pro aktuální počet kusů zboží ve skladu a proměnnou l pro maximální délku názvu zboží.) Tato časová složitost

bude hlavním kritériem hodnocení. Snažte se tedy, aby váš program zpracoval co nejrychleji každý pokyn zadaný na vstupu.

Připomínáme, že pokud používáte ve svém programu netriviální algoritmy nebo datové struktury (např. různé součásti STL v C++), potom váš popis algoritmu musí obsahovat také dostatečný popis jejich implementace, případně popis implementace příbuzné datové struktury se stejnou časovou složitostí operací.

P–I–4 Zlomkové programy

V letošním ročníku olympiády se budeme v každém soutěžním kole setkávat se zlomkovými programy. Ve studijním textu uvedeném za zadáním této úlohy je popsáno, jak zlomkové programy fungují.

Soutěžní úloha.

a) (5 bodů) Na vstupu je číslo n tvaru $2^x 3^y 5$, přičemž $x, y \geq 0$. Napište zlomkový program, který ho převede na číslo 5, jestliže $x = y$, resp. na číslo 7, jestliže $x \neq y$.

b) (5 bodů) Na vstupu je číslo n tvaru $2^x 3$, přičemž $x \geq 0$. Napište zlomkový program, který ho převede na číslo tvaru 2^y , kde $y = \lfloor x/2 \rfloor$.

Interpret

Dříve než svoje řešení odevzdáte, můžete si ho otestovat. Na webových stránkách olympiády (<http://mo.mff.cuni.cz/>) najdete odkaz na interpret zlomkových programů.

Studijní text

Zlomkové programy slouží k výpočtu některých funkcí na přirozených číslech. Zlomkový program má velmi jednoduchý zápis — je to konečná posloupnost zlomků, tedy kladných racionálních čísel (z_1, \dots, z_k) .

Výpočet zlomkového programu probíhá po jednotlivých krocích. Během výpočtu si udržujeme jediné celé číslo a , tzv. *aktuální hodnotu*. Na začátku výpočtu se tato hodnota rovná hodnotě na vstupu. V každém kroku výpočtu najdeme nejmenší i takové, že $a \cdot z_i$ je celé číslo, a změníme aktuální hodnotu na $a \cdot z_i$. Pokud takové i neexistuje, výpočet končí.

Příklad 1:

Ukážeme si program, který pro vstup $n = 2^x$ (kde $x \geq 0$) dává výstup 3^y , kde $y = x \bmod 3$.

Jedním takovým programem je posloupnost $(1/8, 9/4, 3/2)$. Slovně si popíšeme průběh výpočtu tohoto programu: Dokud to jde, zmenšuj x o 3. Když už to nejde, máme v a číslo 1, 2 nebo 4, a převedeme ho tedy snadno na 1, 3 nebo 9.

Například pro $n = 1024 = 2^{10}$ se hodnota a bude měnit takto:

$$2^{10} \xrightarrow{1} 2^7 \xrightarrow{1} 2^4 \xrightarrow{1} 2 \xrightarrow{3} 3.$$

(Číslo nad šipkou je pořadovým číslem zlomku, kterým jsme a v daném kroku násobili.)

Pro přesnost dodejme, že velmi záleží na pořadí jednotlivých zlomků. Například program $(9/4, 3/2, 1/8)$ by z čísla 2^x vyrobil číslo 3^x . Zlomek $1/8$ by se při výpočtech tohoto programu nikdy nepoužil.

Jiné vyhovující programy jsou $(1/8, 3/2)$, $(3/2, 1/27)$ a $(1/27, 3/2)$. Rozmyslete si, proč každý z nich řeší správně Příklad 1.

Příklad 2:

Co spočítá program $(2/2)$ pro vstup $n = 4$? A co spočítá pro vstup $n = 7$?

Častou chybou je odpovědět, že pro vstup $n = 4$ se tento program zacyklí, ale pro vstup $n = 7$ se výpočet programu zastaví, neboť 7 není dělitelné dvěma. Správnou odpovědí ale je, že v *obou* případech poběží výpočet programu do nekonečna. Zajímají nás totiž jenom hodnoty zlomků, nikoli jejich zápis. Zlomek $2/2$ představuje racionální číslo 1, takže $7 \cdot (2/2) = 7 \cdot 1$ je celé číslo.

Abyste se ve svých řešeních takto nespletli, doporučujeme vám psát všechny zlomky v základním tvaru. Pro posloupnost zlomků zapsaných v základním tvaru platí, že v každém kroku výpočtu hledáme nejmenší i takové, že jmenovatel i -tého zlomku dělí beze zbytku aktuální hodnotu.

Příklad 3:

Ukážeme si program, který pro vstup $n = 2^{x+1}$ (kde $x \geq 0$) dává výstup 3^{x+2} .

Číslo zadané na vstupu je určitě sudé a není dělitelné žádným prvočíslem různým od 2. Použijeme prvočíslo 11 jako příznak, že už nejsme na začátku výpočtu. V prvním kroku tedy přepíšeme číslo 2^{x+1} na $2^x 3^2 \cdot 11$. Toho dosáhneme zlomkem $3^2 \cdot 11/2 = 99/2$.

Jakmile se objeví v prvočíselném rozkladu aktuální hodnoty prvočíslo 11, znamená to, že první krok výpočtu máme úspěšně za sebou. Můžeme tedy dále klidně „měnit dvojky na trojky“. To můžeme zajistit například posloupností zlomků $3 \cdot 13/(2 \cdot 11)$ a $11/13$. Všimněte si, že nestačí použít jeden zlomek $33/22$. Není-li vám jasné proč, přečtěte si ještě jednou Příklad 2.

Postupně se takto dostaneme k číslu $3^{x+2} \cdot 11$. V této situaci už stačí jenom vydělit aktuální hodnotu číslem 11 a můžeme skončit.

Musíme dát pozor na to, abychom ve zlomkovém programu výše popsané zlomky správně seřadili: $(39/22, 11/13, 1/11, 99/2)$. V prvním kroku

výpočtu se jistě použije poslední zlomek. Od této chvíle je aktuální hodnota dělitelná číslem 11 nebo 13. Střídavě se proto používají první dva zlomky, dokud se nedostaneme do situace $a = 3^{x+2} \cdot 11$. Pak už se první ani druhý zlomek použít nedá. Použije se proto třetí zlomek, čímž získáme hodnotu $a = 3^{x+2}$ a výpočet zjevně skončí.

Poznámka. V zápisu podobných řešení jako v příkladu 3 není nutné čitatele a jmenovatele zlomků roznásobovat. Své řešení můžete uvést ve tvaru

$$\left(\frac{3 \cdot 13}{2 \cdot 11}, \frac{11}{13}, \frac{1}{11}, \frac{3^2 \cdot 11}{2} \right).$$