

# MATEMATICKÁ OLYMPIÁDA NA STREDNÝCH ŠKOLÁCH

53. ROČNÍK, školský rok 2003/2004

Zadania úloh II. kola kategórie P

Druhé kolo 53. ročníka MO kategórie P sa koná 6. januára 2004 v dopoludňajších hodinách. Na riešenie úloh máte 4 hodiny čistého času. Riešenie každého príkladu musí obsahovať (pokiaľ nie je v zadaní uvedené ináč):

- **Popis riešenia**, to znamená slovný popis použitého algoritmu, argumenty zdôvodňujúce jeho správnosť (prípadne dôkaz správnosti algoritmu), diskusiu o efektivite vášho riešenia (časová a pamäťová zložitosť). Slovný popis riešenia musí byť jasný a zrozumiteľný i bez nahliadnutia do samotného zápisu algoritmu (do programu).
- **Program**. V úlohách **P-II-1**, **P-II-2** a **P-II-3** treba uviesť dostatočne podrobný zápis algoritmu, najlepšie v tvare zdrojového textu najdôležitejších častí programu v jazyku Pascal alebo C. Zo zápisu môžete vynechať jednoduché operácie ako vstupy, výstupy, implementáciu jednoduchých matematických vzťahov a pod. Na druhej strane triedenie **nie je** jednoduchá operácia, preto ak používate triedenie, nezabudnite uviesť aj jeho algoritmus. (V jazyku C máte k dispozícii funkciu `qsort`, ak ju chcete použiť, stačí uviesť porovnávaciu funkciu.) V úlohe **P-II-4** zostrojte príslušný registrový počítač.

Hodnotí sa nielen správnosť programu, ale tiež kvalita popisu riešenia a efektivita zvoleného algoritmu.

## P-II-1

Firma Truhlík a syn má v meste  $N$  budov a chce svoje budovy pripojiť počítačovou sieťou. Vedenie rozhodlo, že v  $K$  ( $1 \leq K \leq N$ ) budovách zakúpia vysokorýchlostné pripojenie na internet. Okrem toho, niektoré dvojice budov budú prepojené optickým káblom.

Dve budovy sú v tom istom komponente siete, ak sa medzi nimi dá komunikovať cez niekoľko optických káblov (teda prípadne cez niekoľko iných budov). Aby mohli spolu komunikovať dve budovy v rôznych komponentoch siete, musí každý z týchto komponentov obsahovať aspoň jeden počítač pripojený na internet.

### Súťažná úloha

Na vstupe sú dané čísla  $N$  a  $K$  a pre každú dvojicu budov jedno kladné celé číslo – cena vybudovania optického kábla medzi týmito dvoma budovami. Navrhnite efektívny algoritmus, ktorý vyberie, ktorých  $K$  budov pripojiť na internet a ktoré dvojice budov spojiť optickým káblom tak, aby každé dve budovy mohli spolu komunikovať a aby celková cena optických káblov bola najmenšia možná.

#### Príklad:

##### Vstup:

$N = 4, K = 2$

Ceny spojení:

(1,2): 100

(1,3): 10

(1,4): 100

(1,5): 300

(2,3): 100

(2,4): 10

(2,5): 300

(3,4): 47

(3,5): 27

(4,5): 74

##### Výstup:

Na internet pripojíme budovy 1 a 2, káblom spojíme dvojice (1, 3), (2, 4) a (3, 5). Cena kábla bude 47.

## P-II-2

Jankovej programátorskej firme AttoSoft sa podarilo získať druhého klienta. Ten mu dal opäť za úlohu naprogramovať  $N$  jednoduchých programov.

Janko chce tentokrát ušetriť, preto namiesto programátorov zamestnal  $N$  študentov, na každý program jedného. Janko má stále iba jeden počítač,

a na ňom v každom okamihu môže robiť len jeden študent. Hlavný problém je ale v tom, že študenti môžu pracovať len medzi prednáškami.

Študent  $i$  potrebuje  $p_i$  hodín času na napísanie prideleného programu, príde do firmy v čase  $s_i$  a musí odísť najneskôr v čase  $t_i$ . Program nemusí napísať naraz, môže občas prácu prerušiť, počítač uvoľniť iným študentom a vypíť si kofolu.

### Súťažná úloha

Napište program, ktorý zistí, ktorý študent má kedy používať počítač, tak aby všetci stihli napísať svoje programy za jeden deň, alebo rozhodne, že to nie je možné.

#### Príklad:

##### Vstup

$N = 3$   
 $p_1 = 1, s_1 = 3, t_1 = 4$   
 $p_2 = 2, s_2 = 2, t_2 = 5$   
 $p_3 = 5, s_1 = 1, t_1 = 10$

##### Výstup

Študent 1 pracuje od tretej do štvrtej, študent 2 pracuje od druhej do tretej a od štvrtej do piatej, študent 3 pracuje od piatej do desiatej.

##### Vstup

$N = 2$   
 $p_1 = 200, s_1 = 300, t_1 = 500$   
 $p_2 = 200, s_2 = 400, t_2 = 600$

##### Výstup

Nedá sa.

### P-II-3

Kleofáš dostal dnes zrána hlad, a tak sa rozhodol spraviť si syrovú bagetu. Bagetu si môžeme predstaviť ako úsečku dlhú  $N$  cm, alebo presnejšie ako uzavretý interval  $\langle 0, N \rangle$ . Každý kúsok syru tvorí taktiež uzavretý interval. Keďže Kleofáš je pedant, kladie kúsky syru tak, aby súradnice ich začiatkov aj koncov boli celé čísla. Kleofáš by chcel, aby jeho bageta bola pokrytá syrom presne podľa jeho predstáv. Na to ale potrebuje jednoduchý počítačový program, ktorý by mu pomohol.

### Súťažná úloha

Na vstupe je veľkosť bagety  $N$  ( $1 \leq N \leq 10^6$ ). Nasleduje  $P$  príkazov ( $1 \leq P \leq 10^9$ ), každý z nich je jedného z nasledujúcich tvarov:

PRIDAJ  $a$   $b$  (Kleofáš pridal kúsok syra, siahajúci od  $a$  po  $b$ .)  
KOLKO  $c$  (Program by mal vypísať, koľko kusov syra leží na pozícii  $c$ .)

Váš program musí spracúvať príkazy v poradí, v akom sú uvedené na vstupe. Pre každý príkaz KOLKO vypíšete jedno číslo – počet dovedy pridanych kúskov syru, ktoré ležia nad súradnicou  $c$ . (Keďže kúsky syra sú uzavreté intervaly, rátajú sa aj tie kúsky, pre ktoré je  $c$  súradnica jeho začiatku alebo konca.)

### Príklad:

Vstup	Výstup
N=20	
PRIDAJ 1 10	
PRIDAJ 6 12	
KOLKO 5	1
KOLKO 6	2
PRIDAJ 4 14	
KOLKO 5	2
KOLKO 16	0

## P-II-4

V tomto kole sa budeme zaoberať tzv. *dvojsmernými* registrovými počítačmi. Od registrových počítačov z domáceho kola sa odlišujú tým, že sa pri čítaní vstupného slova vedia vracať späť. Ich formálnu definíciu nájdete v študijnom texte, ktorý nasleduje za zadáním súťažnej úlohy. Zmena oproti pôvodnej definícií je zvýraznená.

### Súťažná úloha

Napište program pre dvojsmerný registrový počítač, ktorý bude riešiť nasledujúcu úlohu: Vstupom bude reťazec písmen  $a, b, c, d$ . Počítač ho má vyhlásiť za dobrý práve vtedy, keď je to palindróm, t.j. je rovnaký, keď ho čítame odpredu aj odzadu. Formálne: slovo  $a_1a_2a_3 \dots a_{n-1}a_n$  je palindróm, ak  $a_1 = a_n, a_2 = a_{n-1}, \dots, a_{\lfloor n/2 \rfloor} = a_{\lceil n/2 \rceil}$ . Teda napríklad vstupy *bacab* a *dd* sú palindrómy, vstupy *baacdbb* a *bacabdccc* nie sú.

**Základným kritériom hodnotenia bude počet registrov, ktoré váš počítač potrebuje.** Snažte sa zostrojiť počítač, ktorý ich používa čo najmenej!

### Študijný text

*Register* je niečo podobné ako premenná. Presnejšie, v registri môže byť uložené *ľubovoľne veľké* nezáporné celé číslo. Na rozdiel od premenných, ktoré vieme medzi sebou sčítať, odčítať a násobiť, s registrom vieme robiť

len tri jednoduché operácie: zväčšiť jeho obsah o 1, zmenšiť jeho obsah o 1 (ak v ňom bola 0, ostane v ňom 0) a pozrieť sa, či je v ňom 0.

*Registrový počítač* má k dispozícii neobmedzene veľa registrov, označených  $R_0, R_1, R_2$ , atď. Okrem registrov má registrový počítač k dispozícii len **konečne veľkú** pomocnú pamäť.

*Program* pre náš registrový počítač budeme písať v jazyku podobnom jazyku Pascal. Náš programovací jazyk bude rozšírený napr. o príkazy na prácu s registrami, na druhej strane niektoré príkazy z Pascalu budeme musieť zakázať.

Náš počítač bude riešiť úlohy nasledovného typu: dostane na vstupe slovo (reťazec písmen) a po nejakom čase odpovie, či je to slovo dobré alebo nie. Aby nám vedel odpovedať, zavedieme špeciálne príkazy **Accept** a **Reject**. Akonáhle sa vykoná príkaz **Accept**, slovo je dobré a výpočet končí. Akonáhle sa vykoná príkaz **Reject**, slovo je zlé a výpočet končí. Ak sa výpočet zacyklí, prípadne ak skončí a počas neho sa nevykoná ani **Accept**, ani **Reject**, slovo je tiež zlé.

Príkaz „prirátaj 1 k obsahu registra  $R$ “ budeme značiť **Inc**( $R$ ), „odrátaj 1 od obsahu registra  $R$ “ budeme značiť **Dec**( $R$ ). Výraz **Zero**( $R$ ) bude pravdivý, ak je v registri  $R$  nula a nepravdivý inak. Na začiatku výpočtu sú vo všetkých registroch nuly.

V programe môžeme použiť len konečne veľa registrov. Okrem nich môžeme použiť už iba konštantný počet pomocných premenných typu *byte*<sup>1</sup> (teda nemôžeme použiť polia!) a špeciálnu premennú **vstup** typu *char*.

*Premenná vstup vždy obsahuje hodnotu aktuálneho písmena vstupu. Na začiatku je aktuálnym písmenom prvé písmeno vstupu. Aktuálne písmeno vstupu vieme zmeniť zavolaním príkazov **Left** (aktuálnym sa stane predchádzajúce písmeno) a **Right** (aktuálnym sa stane nasledujúce písmeno). Ak by sa aktuálne písmeno malo nachádzať mimo vstupného slova, premenná vstup bude obsahovať špeciálny znak \$. (Môžeme si predstaviť, že pred aj za vstupným slovom sa nachádza dostatočne veľa znakov \$.)*

Keďže registrový počítač má okrem registrov len konečne veľa pamäte, nemôže si dovoliť používať ani rekurziu (nemal by si kde pamätať návratové adresy). My pre istotu úplne zakážeme definovať a používať v našom programe procedúry. Takisto je zakázané volanie štandardných procedúr a funkcií jazyka Pascal. V aritmetických výrazoch sa smú používať iba premenné (teda nie registre), konštanty, celočíselné operátory  $+$ ,  $-$ ,  $*$ ,

---

<sup>1</sup>obsah takejto premennej je celé číslo od 0 do 255

div, mod a zátvorky. V podmienkach sa smú používať výrazy  $\mathbf{Zero}(R_i)$ , bežné relačné operátory ( $<$ ,  $<=$ ,  $\dots$ ), logické spojky a zátvorky.

Z kľúčových slov jazyka Pascal sú teda povolené iba nasledovné: **var**, **begin**, **end**, **if**, **then**, **else**, **case**, **of**, **while**, **do**, **repeat**, **until**, **for**, **to**, **downto**, **div**, **mod**, **and**, **or**, **not** a **xor**.

### Príklad 1

Napište program pre dvojsmerný registrový počítač, ktorý bude riešiť nasledujúcu úlohu: Na vstupe bude reťazec písmen  $a$ ,  $b$ ,  $c$ . Nech  $\alpha$  je počet písmen  $a$  vo vstupnom reťazci,  $\beta$  nech je počet  $b$  a  $\gamma$  nech je počet  $c$ . Počítač má vstupný reťazec vyhlásiť za dobrý práve vtedy, keď  $\alpha = \beta = \gamma$ .

**Riešenie.** Prejdeme vstupné slovo zľava doprava, v registroch  $R_1$  a  $R_2$  spočítame hodnoty  $\alpha$  a  $\beta$ , keď vstupné slovo dočítame, porovnáme obsahy registrov. Vrátime sa na začiatok, pri druhom prechode spočítame v  $R_1$  a  $R_2$  hodnoty  $\beta$  a  $\gamma$  a porovnáme ich. Rozmyslite si, že by stačilo použiť jediný register (v ktorom by sme mali hodnotu  $|\alpha - \beta|$ , resp.  $|\beta - \gamma|$ ).

```

var vstup : char;
begin
    while (vstup <> '$') do begin
        if (vstup = 'a') then Inc(R_1);
        if (vstup = 'b') then Inc(R_2);
        Right;
    end;
    while not Zero(R_1) do begin
        Dec(R_1); if Zero(R_2) then Reject else Dec(R_2);
    end;
    if not Zero(R_2) then Reject;
    { bolo rovnako 'a' a 'b', vratime sa na zaciatok }
    Left; while (vstup <> '$') do Left;
    Right;

    while (vstup <> '$') do begin
        if (vstup = 'b') then Inc(R_1);
        if (vstup = 'c') then Inc(R_2);
        Right;
    end;
    while not Zero(R_1) do begin
        Dec(R_1); if Zero(R_2) then Reject else Dec(R_2);
    end;
    if Zero(R_2) then Accept;
end.

```

## Príklad 2

Napíšte program pre dvojsmerný registrový počítač, ktorý bude riešiť nasledujúcu úlohu: Na vstupe bude reťazec písmen  $a$ . Počítač ho má vyhlásiť za dobrý práve vtedy, keď je jeho dĺžka mocninou troch.

**Riešenie.** Riešenie bude vyzeráť rovnako ako na jednosmernom registrovom počítači. Prejdeme vstupné slovo zľava doprava, pričom si do  $R_1$  uložíme jeho dĺžku. Potrebujeme zistiť, či je to mocnina troch. Aby sme to zistili, budeme uloženú hodnotu deliť tromi, kým to pôjde. Ak na konci dostaneme podiel 0 a zvyšok 1, pôvodné číslo bola mocnina 3, inak nie.

```
var vstup : char;
    zvyšok : byte;
begin
  while (vstup <> '$') do begin Inc(R_1); Right; end;
  if Zero(R_1) then Reject;
  while true do begin
    zvyšok := 0;
    while not Zero(R_1) do begin
      Dec(R_1);
      zvyšok := (zvyšok + 1) mod 3;
      if (zvyšok = 0) then Inc(R_2);
    end;
    if (Zero(R_2) and (zvyšok = 1)) then Accept;
    if (zvyšok <> 0) then Reject;
    while not Zero(R_2) do begin Dec(R_2); Inc(R_1); end;
  end;
end.
```

SLOVENSKÁ KOMISIA MATEMATICKEJ OLYMPIÁDY

**53. ROČNÍK MATEMATICKEJ OLYMPIÁDY**

Zadania II. kola kategórie P

Autori príkladov: B. Brejová, M. Forišek, M. Pál, T. Vinař

Zodpovedný redaktor: M. Forišek

Náklad: vopred neznámy

Sadzba programom L<sup>A</sup>T<sub>E</sub>X

© Slovenská komisia Matematickej olympiády 2003