



## Korešpondenčný seminár z programovania XXI. ročník, 2003/04

Katedra vyučovania informatiky FMFI UK,  
Mlynská Dolina, 842 48 Bratislava

*KSP finančne podporuje MICROSTEP-MIS spol. s r.o.*

### Vzorové riešenia 1. kola zimnej časti, kat. Z

Milí riešitelia!

Trvalo dlho, ale dočkali ste sa. Vytúžené vzorové riešenia sú už vo vašich rukách a nik vám ich nevezme. Ospravedľňujeme sa za chybu, ktorú nám tlačiarenský škriatok prepašoval do príkladu v zadaní „Zlavy na železničiach“.

Čítaj tieto odkazy, môžu sa ti hodiť na sústreďení.

KSPáci

### O ideálnom riešení

Na úvod by sme vám radi ešte raz pripomenuli, ako si predstavujeme ideálne riešenie.

Nechceme, aby ste sa zbytočne hrali so vstupom a výstupom. Samozrejme, zakazovať vám to nebudeme, ale bodíky navyše za to nedostanete a ešte môžete znechutiť opravovateľa, ktorý musí čítať množstvo kódu nesúvisiaceho s riešením samotnej úlohy. Vo svojom riešení sa nemusíte zaoberať kontrolovaním správnosti vstupných údajov.

Pod pojmom **popis algoritmu** nerozumieme preloženie programu z Pascalu, prípadne C-čka do slovenčiny. Popis by mal zhruba obsahovať: **Popis myšlienky**, na ktorej je založený algoritmus (bez detailov ako sú názvy premenných, procedúr...). **Popis dátových štruktúr**. Čo si vlastne chceme počas výpočtu pamätať a aké štruktúry sú na to vhodné. **Popis samotného algoritmu**. Tu môžeme využívať už popísané dátové štruktúry a spomenúť dôležité procedúry a funkcie, ako aj najdôležitejšie premenné. **Zdôvodnenie správnosti**. Nemusí to byť formálne presný ani detailný dôkaz, avšak mal by obsahovať argumenty zdôvodňujúce každý dôležitý aspekt programu, ktorý nie je na prvý pohľad zrejmý. Sem zaradíme aj dôkaz konečnosti v tých prípadoch, keď nie je zrejmé, že sa program nezacyklí. Na koniec príde **odhad** časovej a pamäťovej zložitosti programu.

Takáto štruktúra popisu nie je univerzálna. Niektoré časti možno vynechať, niektoré zlúčiť, prípadne nejaké pridať. Poradie by však malo byť zhruba zachované. Popis by sa v žiadnom prípade nemal detailne venovať spracovávaniu vstupu, ani výpisu výstupu.

**Niekoľko slov o odhade zložitosti.** Pod odhadom časovej zložitosti rozumieme odhad času, ktorý bude program trvať, v závislosti od vstupných premenných. Tento čas je priamo úmerný počtu elementárnych operácií, ktoré program vykonáva – priradenie jednoduchovej premennej, porovnanie jednoduchých premenných, aritmetické operácie, atď. Väčšinou sa zaujímate o to, ako dlho beží program v priemernom, alebo najhoršom prípade (čiže si všímame priemerný, alebo maximálny čas vykonávania programu). Analogicky odhad pamäťovej zložitosti je odhad veľkosti použitej pamäti v závislosti od vstupných premenných.

Nezaujímajú nás presné hodnoty (ktoré je hlavne u časovej zložitosti navyše aj problém určiť), iba ich rádová veľkosť. Za týmto účelom bola zavedená tzv. *O*-notácia. Hovoríme, že funkcia  $f(n)$  je  $O(g(n))$ , ak funkciu  $f(n)$  vieme zhora odhadnúť nejakým násobkom funkcie  $g(n)$ .<sup>1</sup> Túto notáciu budeme používať na jednoduchý zápis zložitosti.

<sup>1</sup>Keby sme chceli byť formálni:  $f(n) = O(g(n)) \Leftrightarrow \exists n_0 \in \mathbb{N}, c \in \mathbb{R}^+; \forall n > n_0; f(n) \leq c \cdot g(n)$

Napríklad ak program pre vstup veľkosti  $n$  nebeží nikdy viac ako  $T(n) = 0.5n^3 + 5n \log n + 83$  mikrosekúnd, vieme čas jeho behu zhora odhadnúť vhodným násobkom funkcie  $n^3$ . V takomto prípade hovoríme, že časová zložitosť nášho programu je  $O(n^3)$ . (Tento zápis môžeme tiež čítať nasledovne: Čas behu nášho programu je **(v najhoršom prípade) rádovo  $n^3$** .) Všimnite si, že ak je nejaká funkcia  $O(n^2)$ , tak je aj  $O(n^3)$ , atď. Vždy sa však snažíme nájsť čo najmenšie horné ohraničenie, t.j. funkciu, ktorá čo najpomalšie rastie.

Takto odhadnúť časovú a pamäťovú zložitosť programu nie je väčšinou nič ťažké. Ak napr. obsahuje program dva vnorené cykly, z ktorých sa každý vykonáva rádovo  $n$ -krát, vykoná rádovo  $n^2$  krokov. Teda jeho časová zložitosť bude  $O(n^2)$ . Zložitosti  $O(n)$  zvykneme hovoriť lineárna,  $O(n^2)$  kvadratická a pod. Konštantnú (časovú alebo pamäťovú) zložitosť môžeme značiť  $O(1)$ . Viacero príkladov odhadu časovej i pamätevej zložitosti nájdete, ak si pozorne prečítate tieto vzorové riešenia.

A ešte jedno upozornenie pre **vás** – riešiteľov: uvedením nepostačujúceho popisu prípadne jeho vynechaním riskujete, že opravovateľ vaše riešenie nepochopí a dostanete napr. 0 (slovom **nula**) bodov.

opravoval Martin  
(max. 15 bodov)

## 1. Zľavy na železničiach

Tento príklad vás poslalo ozaj veľa, ale bolo smutné, že len traja alebo štyria z vás vymysleli lineárne riešenie a mali ho naozaj správne. Najsmutnejšie však bolo že aj z tých troch či štyroch riešiteľov mal len jediný riešiteľ riešenie, v ktorom si dal pozor na všetky okrajové prípady a nebol k nemu lenivý napísať aj kvalitný popis. Teda nečudujte sa, že tak málo z vás má za tento príklad viac než 12 bodov.

Za správne riešenie s lineárnou časovou zložitosťou ste dostali **12 bodov**, s kvadratickou **9 bodov**, s kubickou **7 bodov** a s exponenciálnou **4 body**. Za nesprávne riešenie ste dostali jeden až dva body. Mnohým z vás som z týchto bodov niekoľko strhol za rôzne chyby (napríklad, ak napísaný program nefungoval pre malé počty zliav, alebo ak ste neošetrili prípad, keď všetky ponúkané zľavy boli nevýhodné, alebo ste zabudli inicializovať premenné, alebo za rôzne iné chyby). Ďalej som strhol po jednom bode, ak ste neuviedli odhad časovej zložitosti vášho riešenia. K týmto bodom ste mohli navyše ešte získať **3 body** za popis vášho riešenia.

Máme postupnosť  $N$  kladných čísel – zliav očíslovaných postupne číslami od 1 po  $N$ . Chceme nájsť taký úsek zliav, v ktorom bude súčin zliav najmenší možný. Takýto úsek budeme nazývať najlepší. Pre začiatok predpokladajme, že poznáme najlepší úsek zliav, ktorý končí v  $k$ -tej zľave ( $1 \leq k < N$ ). Zamyslime sa, ako nájsť najlepší úsek zliav, ktorý končí v  $k+1$ -vej zľave. Súčin zliav v tomto úseku je súčinom  $k+1$ -vej zľavy a zliav z nejakého (možno prázdneho) úseku zliav končiacého  $k$ -tou zľavou. Nech  $s$  je súčin zliav z najlepšieho úseku končiacého  $k$ -tou zľavou. Potom súčin zliav z akéhokoľvek iného úseku končiacého  $k$ -tou zľavou je aspoň  $s$ , preto ak  $s \geq 1$ , tak úsek obsahujúci iba  $k+1$ -vú zľavu bude najlepší úsek končiacim  $k+1$ -vou zľavou. Naopak ak  $s < 1$ , tak najlepší úsek končiaci  $k$ -tou zľavou predĺžený o  $k+1$ -vú zľavu bude aj najlepší úsek končiacim v  $k+1$ -vej zľave.

Kedže poznáme najlepší úsek končiaci prvou zľavou – je to samotná prvá zľava, popísaným postupom vieme jednoducho nájsť najlepšie úseky končiace postupne druhou, tretou, ... až  $N$ -tou zľavou. Zrejme najlepší z týchto úsekov je aj hľadaným najlepší úsek. (Každý úsek končí nejakou zľavou.) Dôležité je si ešte uvedomiť<sup>2</sup>, že ak súčin zliav v najlepšom úseku je väčší ako 1, najvýhodnejšie je nezobrať žiadnu zľavu.

Nie je ťažké nahliadnúť, že takéto riešenie bude mať lineárnu časovú a konštantnú pamäťovú zložitosť. (Všimnite si, že si nepotrebujeme pamätať všetky zľavy v poli, ale môžeme ich spracovávať priebežne počas nacítavania.)

<sup>2</sup>A mnohí z vás na to zabudli.

**Listing programu:**

```

var N, i, j, od, po: Integer;
    z, s, min: Double;

begin
  Read(N);
  min := 1; s := 1; j := 1;

  for i := 1 to N do begin
    Read(z);
    s := s * z;
    if (s < min) then begin min := s; od := j; po := i; end;
    if (s >= 1) then begin s := 1; j := i + 1; end;
  end;

  if (min < 1) then
    Writeln('Najlepsi je usek od ', od, ' po ', po, ' (plati sa ', (100 * min) :0 :2, '%)')
  else
    Writeln('Neoplati sa ziadna zlava');
end.

```

**2. Zapálenie druidského ohňa**

opravovala Evka  
(max. 15 bodov)

Vašou úlohou v tomto príklade bolo zistiť, či zadaných  $N$  bodov v rovine tvorí pravidelný  $N$ -uholník. Našli sa takí, ktorí si neuvedomili, že  $N$  je zadané na vstupe a aj takí, ktorí si nevšimli, že body vôbec nemusia byť (a teda, potvory, skoro určite nebudú ;-)) zadané v správnom poradí.

$N$ -uholník je pravidelný práve vtedy, keď všetky jeho strany sú rovnaké a všetky body sú rovnako vzdialené od ťažiska, teda v prípade pravidelného útvaru – od stredu opísanej kružnice. Tá druhá podmienka je len jedna z možných ekvivalentných (napr. všetky vnútorné uhly sú rovnaké, ...). Prišli riešenia, ktoré kontrolovali vzdialenosť od stredu alebo uhly, a potom vzdialenosti bodov od seba. Platí, že keď nájdete minimum vzdialeností prvého bodu od všetkých ostatných, potom ak útvar je pravidelný  $N$ -uholník, toto minimum bude jeho stranou. Potom ste počítali vzdialenosti bodov štýlom každý s každým, a zisťovali, či každý má ďalšieho suseda so vzdialenosťou nájseného minima. Tento algoritmus má časovú zložitosť  $O(N^2)$ , kvôli dvom vnoreným cyklom. Za takéto riešenia, ktoré ožaj fungovali, som udeľovala 12 bodov. Ak chýbal odhad zložitosti alebo popis, samozrejme išli body dolu. Za jedno funkčné riešenie v čase  $O(N^N)$  dostal riešiteľ 7 bodov.

**Vzorové riešenie.** Označme dané body  $B_1, \dots, B_N$ . Začneme tým, že spočítame súradnice ťažiska  $S$  ako priemer súradníc zadaných bodov. Teraz overíme, či všetky zadané body sú rovnako ďaleko od  $S$ . Ak nie, pravidelný  $N$ -uholník to nie je, ak áno, pokračujeme. Zadané body ležia na kružnici so stredom  $S$ . Ešte potrebujeme zistiť, či sú na nej rozmiestnené pravidelne. Najjednoduchšie by bolo utriediť ich podľa uhlu, ktorý zvierajú polpriamka  $SB_i$  s kladnou poloosou osi  $x$ . Tak zistíme, ktoré dvojice bodov susedia. Uhol medzi každou dvojicou susedných bodov musí byť presne  $\frac{360^\circ}{N}$ . Najpomalšou časťou tohto riešenia je triedenie, ktoré vieme spraviť v čase  $O(N \log N)$ .

Avšak existuje aj lineárne riešenie. Začneme tým, že spočítame uhol  $\beta$ , ktorý zvierajú polpriamka  $SB_1$  a kladná poloos osi  $x$ . (V Paspale na to existuje funkcia  $\arctan2(y, x)$ <sup>3</sup>, ktorá vráti uhol medzi polpriamkou z  $[0, 0]$  cez  $[x, y]$  a kladnou poloosou osi  $x$ .) Teraz vieme

<sup>3</sup>Presnejšie vo FreePaspale, starý borlandovský má asi len funkciu  $\arctan$ , ktorá vždy vracia uhol z 1. kvadrantu.

povedať, že keby sme spočítali uhly pre polpriamky cez všetky body, mali by sme dostať práve uhly  $\beta, \beta + \frac{360^\circ}{N}, \beta + 2 \cdot \frac{360^\circ}{N}, \dots, \beta + (N-1) \cdot \frac{360^\circ}{N}$ .

Budeme postupne rátať uhly pre polpriamky  $SB_i$ , pričom v poli si budeme pamätať, ktoré z týchto uhlov sme už videli. Ak naozaj každý z nich (práve raz) nájdeme, na vstupe bol naozaj pravidelný  $N$ -uholník, inak nie.

### Listing programu:

```
program Druidi;
uses math;

const EPS = 0.00000001;
type bod = record x,y:double; end;
var nuholnik : array [1..100] of bod;
    uhly : array [0..100] of boolean;
    i,N : integer;
    diff, stredovy_uhol, referencny_uhol, uhol, polomer : double;
    stred : bod;

function Vzd(a,b:bod):double;
begin Vzd:=sqrt(sqr(a.x-b.x) + sqr(a.y-b.y)); end;

begin
  assign(input,'z2.in'); reset(input); read(N);

  stred.x:=0; stred.y:=0;
  for i:=1 to N do begin
    read(nuholnik[i].x, nuholnik[i].y);
    stred.x:=stred.x + nuholnik[i].x; stred.y:=stred.y + nuholnik[i].y;
  end;
  stred.x:=stred.x/N; stred.y:=stred.y/N; polomer:=Vzd(stred,nuholnik[1]);

  for i:=2 to N do { vsimnite si, ako testujeme rovnost realnych cisel! }
    if (abs(Vzd(stred,nuholnik[i])-polomer))>EPS then begin writeln('Nie.');exit;end;
  {ak body nie su na kruznici, program sa ukonci}

  for i:=1 to N-1 do uhly[i]:=false;
  stredovy_uhol:=2*pi/N;
  referencny_uhol:=arctan2(nuholnik[1].y,nuholnik[1].x);
  for i:=1 to N do begin
    uhol:=arctan2(nuholnik[i].y-stred.y, nuholnik[i].x-stred.x);
    uhol:=(uhol-referencny_uhol)/stredovy_uhol;
    if (uhol<0) then uhol:=uhol+N;

    diff:=uhol-ceil(uhol);
    if (diff<EPS) then uhly[round(ceil(uhol))]:=true;
    if (1-diff<EPS) then uhly[round(ceil(uhol))+1]:=true;
  end;

  for i:=0 to N-1 do if uhly[i]=false then begin writeln('Nie.');exit; end;
  writeln('Ano.');
```

## 3. Zadania písomky

opravovala Julka  
(max. 15 bodov)

Tento príklad bol zjavne veľmi jednoduchý, preto(že) ho riešil skoro každý (ak nie každý). V zadaní sme požadovali čo najmenšiu pamäťovú zložitosť, a preto bodové hodnotie vyzerá nasledovne:

- Pamäťová zložitosť  $O(M)$  a časová zložitosť  $O(N)$  – max 15 bodov
- Pamäťová zložitosť  $O(M)$  a časová zložitosť  $O(MN)$  – max 12 bodov
- Pamäťová zložitosť  $O(N)$  a časová zložitosť  $O(N)$  – max 8 bodov
- Chýbajúci popis –3 body
- Chýbajúce alebo zlé odhady časovej a pamätej zložitosti –1 bod

**Vzorové riešenie** Kedže sme nevedeli, koľko čísel profesor zadá a chceli sme  $M$  posledných čísel, je zrejmé, že nám stačí zapamätať si vždy  $M$  čísel, pretože tie predtým už potrebovať nebudeme. Takže vzorové riešenie bude zapamätanie si posledných  $M$  čísel. To môžeme tak, že vždy, keď načítame ďalšie číslo, tak všetky prvky v poli posunieme. Toto by nám ale zhoršilo časovú zložitosť na  $O(NM)$ , pretože pre každé nové číslo znova posunieme pole, v ktorom je  $M$  čísel. Avšak v skutočnosti toto pole nemusíme posúvať, stačí, ak použijeme tzv. cyklické pole. To znamená, že budeme mať pole veľkosti  $M$ , do ňho budeme postupne zapisovať prichádzajúce čísla a keď sa dostaneme na koniec poľa, tak prepíšeme znova začiatok a takto budeme pokračovať ďalej. Čiže vždy prepíšeme číslo, ktoré už potrebovať nebudeme. Keď príde 0, tak ju už nezapišeme, pretože by sme si prepísali práve to číslo, ktoré potrebujeme, t.j.  $M$ -té odzadu.

### Listing programu:

**program** ZadaniaPisomky;

```
var A:array [1..1000] of integer;
    M,N,i,c:integer;
```

**begin**

```
  readln(M);
```

```
  N:=0; i:=0;
```

**repeat**

```
  read(c);
```

```
  if i>=M then i:=1 else inc(i);
```

```
  if c<>0 then begin A[i]:=c; inc(n); end;
```

**until** c=0;

```
  if N>=M then writeln('Na pisomke bude priklad cislo ',A[i])
```

```
  else writeln('Nebude ziadna pisomka, lebo nezadal dost prikladov ;-');;
```

**end.**

## 4. Záhada víly Amálky

opravovala Ňaňka  
(max. 15 bodov)

Tento príklad riešilo pomerne veľa z vás a poväčšinou boli tieto riešenia aj správne. Myšlienky, že treba kontrolovať iba tie onblká, ktorých farbu ešte nevieme a zastaviť sa, keď je jednej farby viac ako polovica asi nikomu z tých, čo tento príklad riešili netreba vysvetľovať. A tí, čo ste tento príklad neriešili, tiež po krátkom zamyslení sa pochopíte, že je to vlastne jasné. Zaujímavejšie je, akým spôsobom sa táto myšlienka dá implementovať. Tu už sa vyskytli rôzne riešenia, ktoré získali rôzny počet bodov.

Ale poďme pekne od začiatku. A doslova od začiatku, pretože od spôsobu, ako ste si uložili vstup veľmi veľa záviselo. Možno ste už niektorí počuli, že onbloň je vlastne graf. Je to vlastne niekoľko vrcholov (v našom prípade onblík), z ktorých niektoré sú pospájané hranami (v našom prípade vetvičkami). Takýto graf si môžeme uložiť viacerými spôsobmi. Mnohí z vás si uložili zoznam vetvičiek. Vo vzorovom riešení si budeme do dvojrozmerného poľa (**A**) ukladať, či medzi dvomi onblkami je alebo nie je vetvička. Neskôr ukážem, prečo to je výhodnejšie. Pre každé onblko si tiež budem pamätať (v poli **O**) jeho farbu (1 alebo

2) alebo, že ešte jeho farbu ešte neviem (označím 0). Dôležité je ešte pole **Stack** v ktorom si budem pamätať onblka, ktorých farbu už viem, ale je možné, že som ešte nepreskúmala ich susedov. Ostatné premenné predstavovať nejdem, pri prečítaní programu ich význam hravo pochopíte.

Začnem z onblka 1 (alebo vlastne z ľubovoľného sympatického onblka) a budem pokračovať, kým nebudem vedieť odpoveď. Samozrejme prvému onblku môžem priradiť ľubovoľnú farbu, napríklad 1. Keďže viem jeho farbu a je možné (ak nie je  $N = 1$ , tak dokonca isté), že farbu jeho susedov nepoznám, zaradím ho do poľa **Stack**. Ďalej budem postupovať takto: Vyberieme onblko z poľa **Stack**. Pozrieme sa na jeho susedov, ktorých farbu ešte nevieme. Pre každého z nich jednou otázkou zistím jeho farbu a zaradím ho do poľa **Stack** na spracovanie. V prípade, že je jednej farby viac ako polovica, vyberiem si onblko, ktoré má vhodnú farbu.<sup>4</sup>

Za zamyslenie stojí, že vždy viem skončiť, že sa mi nestane, že o niektorom onblku jeho farbu nemám ako zistiť. Na riešenie tohto problému bola v zadaní veverička (v reči grafov sa tejto podmienke hovorí **súvislosť**). Keďže sa veverička vie medzi ľubovoľnými onblkami dostať, prečo by ste s tým mali mať problémy vy?

A teraz slúbené vysvetlenie, že prečo je vzorové riešenie lepšie ako pamätať si vetvičky len tak halabala v poli. Samozrejme, sú prípady, kedy sú vaše riešenia rýchlejšie, ale pri rátaní časovej zložitosti, berieme do úvahy priemerný alebo najhorší prípad. Vzorové riešenie samozrejme rovnako ako vaše spracuje každé onblko najviac raz. Zaujímave je, ako z tohto onblka získavame ďalšie, ktoré treba spracovať. Vo vzorovom riešení prejdeme všetky onblka (ktorých je  $N$ ), nájdeme tie, ktoré s spracúvaným susedia a nevieme ich farbu. Pre každé z  $N$  onblkov prejdeme všetkých  $N$  onblkov, časová zložitosť nášho algoritmu je teda  $O(N^2)$ . Vaše riešenia musia zakaždým prejsť všetkých  $M$  vetvičiek. Časová zložitosť vášho obľúbeného riešenia je teda  $O(MN)$ . Priemerná onbloň však má až rádovo  $N^2$  vetvičiek, preto takéto riešenie má v priemernom (aj v najhoršom) prípade časovú zložitosť až  $O(N^3)$ . Niektorí z vás použité vetvičky vymazávali, ale to v našom prípade príliš nepomôže. Nezabudneme dodať, že naše riešenie potrebuje v najhoršom prípade  $N - 1$  otázok.

Na záver zhrnutie a bodovanie. Za popis boli **3 body**. Ostáva teda ešte 12. Správne riešenia so zložitou  $O(N^2)$  dostali **12 bodov**, riešenia so zložitou  $O(NM)$  mohli dostať **9 bodov**. Riešenia správne, ale s ešte väčšou zložitou maximálne **6 bodov**. Nefunkčné riešenia mohli získať najviac **1–2 bodíky**.

### Listing programu:

```
const Nmax=100;
ekv=1; {su rovnake-potom odpoved=1}
var A:array[1..Nmax,1..Nmax] of boolean;
O:array[1..Nmax] of 0..2;
Stack:array[1..Nmax] of integer;

Nfarba:array[1..2] of integer;
Nstack,N,M,farba,i,x,y:integer;

procedure Koniec(f,i:integer);
{pocet onblk farby F sa zvacsil, posledne onblko je I, viem skonict?}
begin
  if (Nfarba[f]>=N/2) then begin
    writeln('odpoved: ',i);
    readln;
    halt;
  end;
end;
```

<sup>4</sup>Napríklad onblko, ktorého farbu sme práve zistili. Skúste sa zamyslieť prečo!

```

procedure Pop;
begin
  x:=Stack[Nstack]; dec(Nstack); {vyberiem X -- prvok pola Stack}
  farba:=O[x];

  for i:=1 to N do
    {hladam vrchol, ktoreho farbu este neviem, ale viem ju zistit}
    if (A[i,x]) and (O[i]=0) then begin
      writeln('otazka: ',i,', ',x);
      readln(y);
      inc(Nstack); Stack[Nstack]:=i;

      if (y=ekv) then {I a X su rovnake}
        if (farba=1) then begin O[i]:=1; inc(Nfarba[1]); Koniec(1,i); end
          else begin O[i]:=2; inc(Nfarba[2]); Koniec(2,i); end
        else {I a X su odlisne}
          if (farba=2) then begin O[i]:=1; inc(Nfarba[1]); Koniec(1,i); end
            else begin O[i]:=2; inc(Nfarba[2]); Koniec(2,i); end;
        end;
    end;

begin
  {nacistavanie}
  readln(N,M);
  for i:=1 to M do begin
    readln(x,y);
    A[x,y]:=true; A[y,x]:=true;
  end;

  {inicializacia}
  for i:=1 to N do O[i]:=0;
  Stack[1]:=1; Nstack:=1;
  O[1]:=1; Nfarba[1]:=1; Nfarba[2]:=0;

  while (Nstack>=1) do Pop;
end.

```

## 5. Získajte body zadarmo!

opravovala Janka  
(max. 15 bodov)

Taká jedinečná šanca získať body, a tak málo riešiteľov ju využilo! Každý riešiteľ tohoto príkladu získal aspoň 2 body. Tí, čo len náhodne zvolili vstup pre bodovaciu funkciu, získali väčšinou 2 alebo 3 body. Najšťastnejším tipom bolo číslo 7645, za ktoré dala funkcia až 10 bodov. Niektorí skúšali v cykle spúšťať bodovaciu funkciu pre rôzne vstupy, tento prístup však nebol veľmi úspešný, kvôli veľkému počtu rekurzívnych volaní pomocných funkcií im pretiekol zásobník. Ďalšiu skupinu riešiteľov tvorili tí, ktorí zistili čo počítajú jednotlivé funkcie, a potom už bez problémov určili jediný vstup, ktorý bodovacia funkcia ohodnotila 15-imi bodmi.

A ako určiť správny vstup? Postupne si povieme, čo robili jednotlivé funkcie.

### Funkcia lala(a,b : longint) : longint

Funkcia sa rekurzívne volá s o jedno menšími parametrami, až kým nie je jeden z parametrov rovný nule. (Ak si pozorne pozrieme celý program, zistíme, že funkcia lala je vždy volaná s nezápornými parametrami.) Rozoberieme dva prípady:

- $a \leq b$ : Po  $a$  rekurzívnych vnorení bude  $a = 0$ ,  $b = b - a$ , a teda  $lala = b - a$ .

- $a > b$ : Po  $b$  rekurzívnych vnoreniach bude  $a = a - b$ ,  $b = 0$ ,  $lala = 0 - (a - b) = b - a$ . Funkciu  $lala(a, b)$  budeme v ďalších funkciách nahrádzať výrazom  $b - a$ .

### Funkcia $huhu(a, b : \text{longint}) : \text{longint}$

Stačí jednoduchá úprava:  $lala(b + 1042, a + 1047) - 5 = a + 1047 - b - 1042 - 5 = a - b$ . Funkciu  $huhu$  teda zjednodušíme na: **if** ( $a \leq b$ ) **then**  $huhu := 0$  **else**  $huhu := a - b$ ; (Alebo inými slovami  $huhu(a, b) = \max(a - b, 0)$ .)

### Funkcia $dodo(a, b : \text{longint}) : \text{boolean}$

Opäť, keď si pozrieme program, zistíme, že pokiaľ  $dodo$  nevolá sám seba, platí  $a > 0$ . Rozoberieme tieto prípady:

- $a < b$ :  $huhu(b, a)$  bude určite rôzna od nuly, teda  $dodo(a, b) = \text{false}$
- $a \geq b$ : Vieme, že  $huhu(b, a) = 0$ ,  $huhu(a, b) = a - b$  a  $lala(a, a) = 0$ . Potom ale dostávame, že  $lala(huhu(b, a) + a, huhu(a, b) + b) = lala(0 + a, (a - b) + b) = 0$ . Tým pádom príkaz  $dodo := (lala(huhu(b, a) + a, huhu(a, b) + b) < 0)$ ; je jednoducho  $dodo := \text{false}$ ; Zatiaľ necháme bokom zisťovanie, kedy sa toto priradenie zavolá a pozrieme sa na posledný riadok funkcie.

Kedy môže získať  $dodo(a, b)$  hodnotu **true**? Jedine ak sa niekedy rekurzívne zavolá s parametrom  $a = 0$ . Predstavme si, že sa  $dodo$   $k$  krát rekurzívne zavolala. Vtedy má prvý parameter hodnotu  $a - kb$ .  $dodo$  bude **true** ak je  $a - kb = 0$ , teda  $a = kb$ . Inými slovami ak  $b$  je deliteľom  $a$ . Pozrime sa ešte na podmienku vo for cykle a zistíme, že  $dodo$  tu môže získať hodnotu **false** len ak  $b$  delí niektoré z čísel  $a - 1, a - 2, \dots, a - b + 1$ . Ak však  $b$  delí niektoré z týchto čísel, určite nedelí  $a$ .

Zistili sme, že  $dodo(a, b) = \text{true}$ , ak  $b$  je deliteľom  $a$ , inak je  $dodo(a, b) = \text{false}$ .

### Funkcia $zuzu(a : \text{longint}) : \text{boolean}$

Treba si uvedomiť, že  $huhu(a, a) = 0$ , a že číslo  $4 + 7y$  nie je deliteľné číslom 7, teda  $dodo(4 + 7 * (10000 \text{ div } a), 7) = \text{false}$ . Funkcia  $zuzu$  teda robí nasledovné: v prvom kroku for cyklu zistí, že  $a$  je deliteľné samé sebou a premennej  $j$  priradí hodnotu  $a$ . V nasledujúcich cykloch sa zisťuje, či má  $a$  aj ďalšieho deliteľa, ak áno,  $j$  je už väčšie ako 0 a  $zuzu$  bude **false**.  $zuzu$  bude **true**, ak sa nenašiel žiadny ďalší deliteľ  $a$ , teda ak  $a$  je prvočíslo.

### Funkcia $BODY(VSTUP : \text{longint}) : \text{longint}$

Najprv ohodnotí vstupy menšie ako 2 alebo väčšie ako  $2 \cdot 10^9$  0 bodmi. V prvom cykle sa do  $a$  priradí najmenšie prvočíslo, ktoré je deliteľom  $VSTUPu$  ( $a$  je najprv 2, potom sa v repeat cykle zvyšuje, kým nie je ďalším prvočíslom. V podmienke while cyklu sa overuje, či  $a$  nie je deliteľom  $VSTUPu$ ).

V ďalšom cykle sa  $VSTUP$  delí číslom  $a$ , kým  $a$  delí  $VSTUP$  a zároveň sa pri každom delení zvyšuje premenná  $b$  (Keď sa pozrieme na posledný riadok funkcie  $BODY$ , vidíme, že by bolo dobre, ak by  $b$  bolo čo najvyššie). Zapišme si teraz  $VSTUP$  ako:  $VSTUP = x \cdot p^k$ , pričom  $p$  je najmenšie prvočíslo, ktoré delí  $VSTUP$  a  $k$  je najväčšie číslo také, že  $p^k$  je ešte deliteľom vstupu. Vidíme, že po ukončení tohoto while cyklu budú mať premenné  $a$ ,  $b$  a  $VSTUP$  takéto hodnoty:  $a = p$ ,  $b = k$ ,  $VSTUP = x$ .

Na ďalšom riadku funkcie  $BODY$  zistíme, že ak  $a$  bude menšie ako 5, získame nanajvýš 3 body (4 nie je prvočíslo). Teda, ak chceme získať viac bodov, náš pôvodný  $VSTUP$  nesmie byť deliteľný ani dvomi, ani tromi.

Keď sa pozrieme na posledný if funkcie  $BODY$ , zistíme, že ak  $x \neq 1$ , tak sa spustí while cyklus, v ktorom:

- ak  $((x+3) \text{ div } 2)$  je deliteľné 2 (teda while sa vykoná párny počet krát), získame 25 mod 15, teda 10 bodov,
- ak  $((x+3) \text{ div } 2)$  nie je deliteľné 2 (teda while sa vykoná nepárny počet krát), získame 5 mod 15, resp. 50 mod 15, teda 5 bodov.

Nám však 5 alebo 10 bodov nestačí, teda musí platiť  $x = 1$ .  $VSTUP$  teda musí byť mocninou nejakého prvočísla  $p$  iného ako 2 a 3. V takomto prípade dostaneme  $k + 2$  bo-



dov. Teda  $k$  má byť čo najväčšie, ale zároveň  $p^k$  musí byť menšie ako  $2 \cdot 10^9$ . Zistíme, že ak  $VSTUP = 5^{13}$ , uvedená podmienka ešte bude splnená, ale pre  $VSTUP = 5^{14}$  alebo  $VSTUP = 7^{13}$  už neplatí. Funkcii BODY teda zadáme číslo

$$5^{13} = 1\,220\,703\,125$$

a dostaneme 15 bodov.

## BONUS

V týchto vzorových riešeniach nám ešte ostal kopec voľného miesta, preto ho využijeme aspoň trochu užitočne. Budeme pokračovať na tému piateho príkladu a ukážeme si niekoľko zaujímavých programov, väčšinou v jazyku C. Začneme zľahka. Čo robí tento pochybný program?

```
#include <stdio.h>
int x,y;
int main(void){scanf("%d %d",&x,&y); x^=y^=x^=y; printf("%d %d\n",x,y); return 0;}
```

A tento?

```
const a='begin writeln(const a=#39,a,#39,#59);writeln(copy(a,1,14),#39,copy(a,15,8),#39,copy(a,23,79));end.';
begin writeln('const a=#39,a,#39,#59); writeln(copy(a,1,14),#39,copy(a,15,8),#39,copy(a,23,79)); end.
```

No dobre. Doteraz to bolo ľahké. Ale čo preboha má byť toto? Tvar mnohé napovedá, aj názvy premenných pomôžu...<sup>5</sup>

```
char*M,A,Z,E=40,J[40],T[40];main(C){for(*J=A=scanf(M="%d",&C);
-- E; J[E]=T[E]; for(;(A-=Z!=Z) || (printf("\n| "
), A = 39, C == 0); Z || printf(M))M[Z]=Z[A-(E=A[J-Z])&&!C
& A == T[A]; |6<<27<rand() || !C&!Z?J[T[E]=T[A]]=E,J[T[A]=A-Z]=A,"_.";" |"];}
```

Jemne zákerná otázka: V akom jazyku je písaný nasledujúci program?

```
(*haluz/**)begin writeln('pas');end.(**/);int main(){printf("c\n");return 0;}/**)
```

A čo ešte tento?

```
(*haluz/* ) (PostScript)
/Times-Roman findfont 40 scalefont setfont
newpath 100 400 moveto show showpage % */ ); /*
<~ */
#include <stdio.h>
int main() /*~>( /* { /* )<~*/ printf("C\n"); return 0;
/* ~> ( *) {
% } begin writeln('pascal'); (*
} % *) end. {*/}
```

Na záver ešte exkurzia do krás jazyka C. Napovieme, že `'-'-'-'` je nula – je to (znak mínus) mínus (znak mínus). So zvyškom sa môžete pohrať sami.

```
int i;main(){for(;i["<i;++i]{--i;}"];read('-'-'-'',i+++hell\
o, world!\n",','/'/'/')));read(j,i,p){write(j/p+p,i---j,i/i);}
```

<sup>5</sup>Technický detail: gcc potrebuje parameter `-fwritable-strings`, aby tento program skompilovalo správne.

# Výsledková listina po 1. kole kategórie KSP-Z

|    | Meno a priezvisko  | Škola                         | Ročník | 11 | 12 | 13 | 14 | 15 | Σ  |
|----|--------------------|-------------------------------|--------|----|----|----|----|----|----|
| 1  | Morihladko Peter   | Gym. Školská Spiš. Nová Ves   | 3      | 9  | 12 | 15 | 15 | 15 | 66 |
| 2  | Ilavský Milan      | SPŠE Liptovský Hrádok         | 4      | 3  | 11 | 14 | 12 | 15 | 55 |
| 3  | Hrinčár Peter      | ZŠ Jarná Poprad               | 0      | 9  | 6  | 14 | 10 | 15 | 54 |
| 3  | Trnovcová Zuzana   | Gym. Jura Hronca BA           | 3      | 14 |    | 11 | 14 | 15 | 54 |
| 5  | Mazák Tomáš        | Gym. Jura Hronca BA           | 1      | 5  | 13 | 14 | 6  | 15 | 53 |
| 6  | Ďudák Juraj        | Gym. Golianova Nitra          | 2      | 10 | 12 | 15 | 12 | 2  | 51 |
| 7  | Mindek Peter       | Škola pre mim. nadané deti BA | 2      | 11 | 12 | 12 | 12 | 3  | 50 |
| 8  | Ambrož Peter       | Gym. Jura Hronca BA           | 4      | 6  |    | 15 | 12 | 15 | 48 |
| 8  | Ľudvík Martin      | Gym. Školská Spiš. Nová Ves   | 3      | 15 | 4  | 14 | 12 | 3  | 48 |
| 10 | Herman Peter       | Gym. Jura Hronca BA           | 1      | 8  |    | 11 | 12 | 15 | 46 |
| 11 | Varga Peter        | Gym. Jura Hronca BA           | 4      | 11 | 11 | 11 | 12 |    | 45 |
| 12 | Jerguš Ján         | Gym. Alejová Košice           | 1      | 10 | 7  | 15 | 12 |    | 44 |
| 12 | Kottman Michal     | Gym. Jura Hronca BA           | 3      | 4  | 11 | 14 |    | 15 | 44 |
| 14 | Jirásek Jozef      | Gym. Zbrojničná Košice        | 1      |    | 13 | 15 |    | 15 | 43 |
| 15 | Dubás Jakub        | Gym. Humenné                  | -1     | 4  | 12 | 14 | 9  | 3  | 42 |
| 16 | Horník Jozef       | Gym. SNP Galanta              | 4      | 3  | 11 | 12 | 12 | 3  | 41 |
| 17 | Buštor Ivan        | Gym. Košická BA               | 1      | 11 | 15 | 14 |    |    | 40 |
| 17 | Zápotocký Radoslav | Gym. Alejová Košice           | 4      | 4  | 6  | 14 | 13 | 3  | 40 |
| 19 | Dobiaš Michal      | Gym. Jura Hronca BA           | 2      | 10 | 15 | 14 |    |    | 39 |
| 20 | Belan Tomáš        | Škola pre mim. nadané deti BA | ???    | 2  |    | 11 | 9  | 15 | 37 |
| 20 | Dzurenko Miroslav  | Gym. Ľ. Štúra Zvolen          | 3      | 7  | 6  | 14 | 10 |    | 37 |
| 22 | Tomori Rudolf      | Gymnázium                     | 2      | 4  | 4  | 11 | 14 | 3  | 36 |
| 22 | Špalek Lukáš       | Gym. Čadca                    | 3      | 10 | 3  | 15 | 8  |    | 36 |
| 24 | Vanderka Peter     | Gym. Trstená                  | 3      | 10 |    | 15 | 10 |    | 35 |
| 24 | Černo Lukáš        | Gym. Ľudovíta Štúra Trenčín   | ???    | 8  | 12 | 15 |    |    | 35 |
| 26 | Domány Dušan       | Gym. P. Horova Michalovce     | 3      | 4  | 5  | 8  | 12 | 5  | 34 |
| 26 | Koman Pavol        | Gym. Daxnera V.n. Topľou      | 4      | 8  |    | 14 | 12 |    | 34 |
| 26 | Villaris Vojtech   | Gym. Jura Hronca BA           | 3      | 8  |    | 15 | 11 |    | 34 |
| 29 | Zagora Martin      | SPŠ Komenského Trnava         | 3      | 8  |    | 11 | 12 | 2  | 33 |
| 30 | Balga Jozef        | Gym. maďarské Šahy            | 4      | 3  |    | 14 |    | 15 | 32 |
| 31 | Kajan Peter        | Gym. Jura Hronca BA           | 1      | 3  | 8  | 11 | 9  |    | 31 |
| 32 | Fedák Matúš        | Gym. Stará Ľubovňa            | 2      | 4  |    | 12 | 12 | 2  | 30 |
| 32 | Sedlák Štefan      | Gym. Daxnera V.n. Topľou      | 3      | 3  |    | 12 | 15 |    | 30 |
| 32 | Sábo Jozef         | Gymnázium                     | 1      | 3  | 4  | 11 | 12 |    | 30 |
| 35 | Bundala Daniel     | Gym. Jura Hronca BA           | 2      | 14 | 15 |    |    |    | 29 |
| 35 | Halamiček Radovan  | Gym. Jura Hronca BA           | 2      | 10 | 4  | 12 |    | 3  | 29 |
| 35 | Hlavačiková Jana   | Gym. Einsteinova BA           | 3      | 10 | 12 | 7  |    |    | 29 |
| 35 | Holub Michal       | Gym. Jura Hronca BA           | ???    | 4  |    | 15 | 10 |    | 29 |
| 35 | Hrubý Martin       | Gym. P. Horova Michalovce     | 3      | 4  |    | 14 | 11 |    | 29 |
| 35 | Ragány Peter       | SPŠE Prešov                   | ???    | 11 | 3  | 15 |    |    | 29 |
| 35 | Sivák Michal       | Gym. Ľudovíta Štúra Trenčín   | 2      |    | 14 | 15 |    |    | 29 |
| 42 | Varga Ľubomír      | SPŠ Nitra                     | 4      | 8  |    | 11 |    | 8  | 27 |
| 43 | Golier Richard     | Gym. Alejová Košice           | 4      | 1  | 12 | 13 |    |    | 26 |
| 43 | Pagáč Matej        | Gym. Školská Spiš. Nová Ves   | 3      | 12 |    | 14 |    |    | 26 |
| 45 | Barna Jozef        | Gym. Humenné                  | 3      | 4  | 11 | 8  |    | 2  | 25 |
| 45 | Korcsok Peter      | Gym. Trstená                  | 0      | 10 |    | 15 |    |    | 25 |
| 47 | Pančík Andrej      | Gym. Š. Moyzesa B. Bystrica   | 1      | 7  |    | 12 |    | 5  | 24 |
| 48 | Sopoliga Radoslav  | Gym. Svidník                  | 4      | 9  |    | 14 |    |    | 23 |
| 49 | Goga Peter         | Gym. 17. novembra Topoľčany   | 4      | 6  | 8  |    |    | 8  | 22 |
| 50 | Fzeši Štefan       | Gym. maďarské Šahy            | 4      | 8  |    | 13 |    |    | 21 |

|    | Meno a priezvisko     | Škola                         | Ročník | 11 | 12 | 13 | 14 | 15 | Σ  |
|----|-----------------------|-------------------------------|--------|----|----|----|----|----|----|
| 50 | Mikuláš Ondrej        | Gym. Haličská Lučenec         | 1      | 10 | 11 |    |    |    | 21 |
| 50 | Urminský Tomáš        | SPŠ Nové Mesto nad Váhom      | 2      | 4  | 4  | 11 | 2  |    | 21 |
| 50 | Zachar Lukáš          | Gym. A.Sládkoviča B. Bystrica | 3      | 7  |    | 14 |    |    | 21 |
| 54 | Macík Tomáš           | Gym. A.Sládkoviča B. Bystrica | 3      | 8  | 3  | 7  | 2  |    | 20 |
| 55 | Baumann Martin        | Gym. Jura Hronca BA           | 1      |    |    | 4  |    | 15 | 19 |
| 55 | Dzurňák Tomáš         | Gym. Školská Spiš. Nová Ves   | 2      | 8  |    | 11 |    |    | 19 |
| 57 | Kuchár Richard        | SPŠE Prešov                   | 3      | 10 |    | 8  |    |    | 18 |
| 58 | Herich Andrej         | Gym. Golianova Nitra          | 2      | 10 |    | 7  |    |    | 17 |
| 58 | Žubrietovský Tomáš    | Gym. L. Štúra Zvolen          | 3      | 6  | 3  | 6  | 2  |    | 17 |
| 60 | Hrabčák Ján           | Gym. Alejová Košice           | 4      |    | 8  | 8  |    |    | 16 |
| 60 | Kubek Vojtech         | Gym. Jura Hronca BA           | 1      | 3  | 2  | 11 |    |    | 16 |
| 62 | Kováč Ivo             | Gym. Alejová Košice           | ???    | 7  |    | 8  |    |    | 15 |
| 62 | Lukáč Matej           | Gym. Šrobárova Košice         | ???    |    | 2  | 13 |    |    | 15 |
| 62 | Škrovinová Katarína   | Gym. Párovská Nitra           | 2      | 1  | 4  | 9  | 1  |    | 15 |
| 65 | Kompuš Patrik         | Gym. Šrobárova Košice         | 1      | 1  | 2  | 11 |    |    | 14 |
| 65 | Sivák Vladimír        | Gym. Ľudovíta Štúra Trenčín   | 2      |    |    | 14 |    |    | 14 |
| 65 | Svonava Daniel        | Gym. Jura Hronca BA           | 2      |    |    | 14 |    |    | 14 |
| 68 | Buday Tomáš           | Gym. Golianova Nitra          | 3      | 2  | 4  | 7  |    |    | 13 |
| 69 | Mižáková Katarína     | Gym. Alejová Košice           | ???    | 4  |    | 8  |    |    | 12 |
| 70 | Bálint Farkaš         | Gym. Mládežnícka Šahy         | 3      |    |    | 11 |    |    | 11 |
| 70 | Gálik Maroš           | Gym. Daxnera V.n. Topľou      | 2      | 11 |    |    |    |    | 11 |
| 70 | Mikuláš Ján           | Gym. Haličská Lučenec         | 2      | 7  | 4  |    |    |    | 11 |
| 70 | Nižňanský             | Gym. Alejová Košice           | ???    | 7  |    | 4  |    |    | 11 |
| 70 | Šimják Patrik         | Gym. Párovská Nitra           | 2      | 1  | 2  | 8  |    |    | 11 |
| 75 | Kramárik Matúš        | Gym. Ľudovíta Štúra Trenčín   | 2      |    | 3  | 7  |    |    | 10 |
| 76 | Godány Robert         | Gym. Grösslingová BA          | 2      | 2  |    | 4  |    | 3  | 9  |
| 76 | Hornáková Anna        | Gym. Alejová Košice           | ???    | 1  |    | 8  |    |    | 9  |
| 76 | Kolesár Štefan        | Gym. Alejová Košice           | 2      | 1  |    | 8  |    |    | 9  |
| 76 | Kováč Peter           | Gym. Alejová Košice           | ???    | 1  |    | 8  |    |    | 9  |
| 80 | Hegedušová Monika     | Gym. P. Horova Michalovce     | 2      | 2  |    | 6  |    |    | 8  |
| 80 | Jacko Vladimír        | Gym. Alejová Košice           | ???    |    |    | 8  |    |    | 8  |
| 80 | Petreszél Tibor       | Gym. maďarské Šahy            | 4      | 1  |    | 7  |    |    | 8  |
| 83 | Hulitka Edmund        | Gym. Filakovo                 | 2      | 1  | 2  | 4  |    |    | 7  |
| 83 | Kertész Tomáš         | Gym. Filakovo                 | 2      | 1  | 2  | 4  |    |    | 7  |
| 83 | Varga Marek           | SPŠ Nitra                     | 4      |    |    | 7  |    |    | 7  |
| 86 | Kollár Martin         | Gym. Alejová Košice           | ???    | 2  |    | 4  |    |    | 6  |
| 86 | Račko Tibor           | Gym. Mládežnícka Šahy         | 3      |    |    | 6  |    |    | 6  |
| 88 | Lopata Ladislav       | Gym. Alejová Košice           | ???    |    | 5  | 0  |    |    | 5  |
| 89 | Kostolanský Rastislav | SPŠ Nitra                     | 4      |    |    | 4  |    |    | 4  |
| 89 | Mačaj Martin          | SPŠE Prešov                   | 3      |    |    | 4  |    |    | 4  |
| 89 | Tinajová Andrea       | Gym. Malá Hora Martin         | 3      | 4  |    |    |    |    | 4  |
| 89 | Vida Jozef            | SPŠ Nitra                     | 4      |    |    | 4  |    |    | 4  |
| 93 | Búdová Lucia          | Gym. Šrobárova Košice         | ???    |    | 1  | 2  |    |    | 3  |
| 93 | Hopková Dominika      | Gym. Šrobárova Košice         | 1      |    | 1  | 2  |    |    | 3  |
| 93 | Kráľová Diana         | Gym. Šrobárova Košice         | 1      |    | 1  | 2  |    |    | 3  |
| 96 | Buch Stanislav        | SPŠ Nitra                     | 4      |    |    | 2  |    |    | 2  |
| 96 | Jokel Tomáš           | Gym. Šrobárova Košice         | 1      |    | 2  |    |    |    | 2  |
| 96 | Kučera Dmitrij        | Gym. Šrobárova Košice         | 1      |    |    | 2  |    |    | 2  |
| 96 | Mondoková Denisa      | Gym. Šrobárova Košice         | 1      |    |    | 2  |    |    | 2  |
| 96 | Sobota Viktor         | Gym. Šrobárova Košice         | 1      |    | 2  | 0  |    |    | 2  |
| 96 | Stanovič Štefan       | SPŠ Nitra                     | 4      |    |    | 2  |    |    | 2  |
| 96 | Tokolyova Zuzana      | Gym. Šrobárova Košice         | 1      |    |    | 2  |    |    | 2  |