

**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.*

## Príklady 2. kola zimnej časti, kat. Z

Milé naše riešiteľky, milí naši riešitelia,

Zima sa pomaly blíži, hoci to tak teraz nevyzerá. Pre istotu pomaly už brúste hrany na svojich zjazdovkách, kúpte si vosky a teplé spodky z Makyty Púchov. Snáď aj napadne sneh a bude sa dať lyžovať. Keď vás už budú bolieť z lyžovania nohy, sadnite si a pustite sa do riešenia týchto príkladov. Svoje riešenie nám posielajte do **7. januára 2004**. A ako už býva zvykom, nezabudnite na známky v hodnote 15 Sk.

Prajeme vám pekného Mikuláša, Vianoce a Nový rok.

KSPáci

### 1. Zachariáš bezdomovec

15 bodov

Zachariáš je bezdomovec a navyše nezamestnaný. Dlhú chvíľu si kráti tým, že na zastávke zbiera špaky, robí si z nich cigarety a fajčí. (Uznávame, že tento príbeh nie je práve mravný a poučný, ale taký je život. Snáď chápete.)

**Úloha:** Zachariáš nazbieral  $N$  špakov. Keďže je šikovný, z  $K$  špakov ( $K$  je najviac 10) si vie spraviť jednu cigaretu. Napíšte program, ktorý z čísel  $N$  a  $K$  spočíta, koľko cigariet Zachariáš vyfajčí. Pokúste sa, aby váš program fungoval aj pre veľmi veľké hodnoty  $N$ .

**Príklad:**

**Vstup:**

$N = 9, K = 3$

$N = 5, K = 2$

$N = 3, K = 1$

$N = 123456789123456789, K = 5$

**Výstup:**

4 cigarety

4 cigarety

nekonečne veľa cigariet

30864197280864197 cigariet

Vysvetlíme prvý výstup: Z 9 špakov si Zachariáš vie spraviť 3 cigarety. Ale tým, že ich vyfajčí, dostane 3 nové špaky a z tých si vie spraviť ďalšiu cigaretu.

**Prémia za 1 bod.** Keď si Zachariáš prečítal toto zadanie, vysvetlil nám, že u neho  $K = 3$ . A ešte nám povedal, že dnes ráno mal 10 špakov, žiadne už nenašiel a napriek tomu vyfajčil 5 cigariet. Ako je to možné?

### 2. Zábavka pred obedom

15 bodov

Na jednom zo sústreďení KSP hrali účastníci nasledujúcu hru. Bolo už tesne pred obedom, keď sa všetci postavili vedľa seba na štartovú čiaru. Každý držal v ruke zalepenú obálku a netrpezlivo čakal na štartovací povel. Vtom okolím zaznelo mocné „Štáárrtt!“ a všetci vybehli k cieľovej čiare umiestnenej opodiaľ. Ich úlohou bolo bežať čo najrýchlejšie k cieľu a cestou splniť úlohu popísanú v liste v obálke. Bežia, bežia, trhajú obálku, dolujú z nej papier, keď vtom prvý z nich zastal...

Zastali aj ostatní a rozmýšľajú ostošesť. Na papieri mal každý správu tohoto typu: „Do cieľa môžeš prísť až po Katke, Paľkovi, Jožkovi, Monike a Rastovi“. Každý z účastníkov mal zoznam ľudí, ktorí ho musia predbehnúť (ostatní môžu, ale nemusia). Zoznam mohol byť aj prázdny.

Super hra, čo? Možno, ale nie tesne pred obedom. Všetci sú už hladní a nevedia, v akom poradí majú dobehnúť. Pomôžte im!<sup>1</sup>

**Úloha:** Na vstupe je dané  $N$  – počet účastníkov, ktorí hrali hru. Účastníkov pre jednoduchosť očísľujeme od 1 po  $N$ . Na každom z ďalších  $N$  riadkoch je úloha pre jedného účastníka. Konkrétne v  $i + 1$ -om riadku je zoznam ľudí, ktorí majú predbehnúť účastníka  $i$ , ukončený nulou.

Vašou úlohou je vypísať poradie, v akom majú účastníci dobehnúť do cieľa. Ak je takýchto poradí viac, vypíšte ľubovoľné z nich. Ak poradie neexistuje, vypíšte o tom príslušnú správu.

**Vstup:**

$N = 5$

2 5 0

5 0

4 0

0

4 0

**Výstup:**

Môžu dobehnúť v poradí: 4, 3, 5, 2, 1.

$N = 3$

2 0

3 0

1 0

Poradie neexistuje, umrú od hladu.

### 3. Zima je tu!

12 bodov

Sneh, nesneh, určite si už zistil, že zima sa blíži. Preto treba zobrať lyže a hor sa na lyžovačku. Len ktoré lyže si vybrať, aby si mal tie najideálnejšie a kamaráti tiež?

**Úloha:** Na vstupe je  $N$  – počet párov lyží a zároveň počet lyžiarov. Potom nasleduje  $N$  čísel zodpovedajúcich výškam lyžiarov a  $N$  čísel, zodpovedajúcich dĺžkam lyží. Tvojou úlohou je napísať program, ktorý každému lyžiari priradí práve jedny lyže (dva lyžiar nemôžu mať tie isté lyže) a to tak, že súčet hodnôt  $|\text{výška lyžiara} - \text{dĺžka lyží}|$  je minimálny.

**Príklad:**

**Vstup:**

$N = 5$

158 169 163 179 182

147 191 179 181 160

**Výstup:**

Prvý lyžiar dostane lyže dĺžky 147, druhý 179, tretí 160, štvrtý 181 a piaty 191.

### 4. Za rohom je kôň

15 bodov

Na dlážke bol koberec a na koberci rozhádzaná kopa šachových figúrok. Na šachovnici stálo 12 koní, vedľa nej sedel malý Gary a smutne hľadel na neohrozené políčko A8. Nech sa snažil ako chcel, nedarilo sa mu rozmiestniť kone tak, aby každé prázdne políčko aspoň jeden z nich ohrozoval. A nech sa snažil ako chcel, viac ako 12 koní doma nenašiel. A ide to vôbec? Asi sa rozplače. Rodičia mu kúpia ďalšiu šachovnicu a bude po probléme.

Rodičom sa ale nechce vyhadzovať peniaze zbytočne, preto by potrebovali vedieť, koľko koní vlastne malý Gary na svoju hru potrebuje.

**Úloha:** Zistite, koľko najmenej koní stačí umiestniť na šachovnicu  $N \times N$  tak, aby každé neobsadené políčko bolo ohrozené aspoň jedným koňom. Pošlite nám správne odpovede pre čo najviac  $N$ , pre každé  $N$  aj jedno vhodné rozostavenie koní. Nezabudnite na program, ktorým ste tieto hodnoty získali.

<sup>1</sup> Keď už nie zjesť obed, tak aspoň zistiť, v akom poradí majú dobehnúť, resp. či vôbec existuje vhodné poradie.

**Poznámka.** Kôň (alebo oficiálne jazdec) je šachová figúrka, ktorá sa hýbe „skokom za roh“ – pohne sa najskôr o 2 políčka vodorovne alebo zvislo, potom o jedno políčko smerom kolmým na ten, ktorým sa hýbal doteraz. Kôň ohrozuje tie políčka, na ktoré sa môže jedným ťahom dostať. Kôň stojaci v strede šachovnice  $5 \times 5$  teda ohrozuje 8 políčok.

**Príklad:**

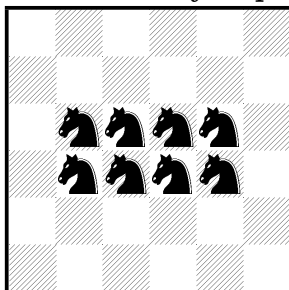
**Vstup:**

$N = 6$

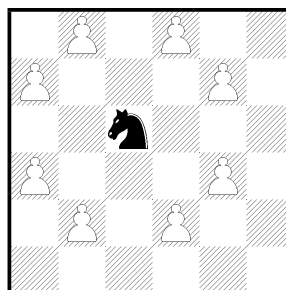
**Výstup:**

Stačí 8 koní.

Obrázok k výstupu:



Políčka ohrozené koňom:



## 5. Zaopravujte si

15 bodov

Márne píšeme do pravidiel „nezabúdajte na popis algoritmu“, márne je prehováranie do duše „prečítajte si svoje riešenie pred tým, ako ho pošlete, či ho po sebe pochopíte aspoň vy“... So železnou pravidelnosťou nám do KSP prichádzajú záhadné riešenia, ktoré nedávajú zmysel ani po treťom prečítaní a ak vôbec majú nejaký popis, tak jeho úlohou je opravovateľa ešte viac popliesť.

Tentokrát sme sa rozhodli, že to spravíme naopak, aby sme vás o tieto úžasné zážitky nepripravili. Vašou úlohou teda bude opraviť tri (krásne prehľadné a jednoduché) riešenia jednej úlohy. Mali by ste uviesť, koľko bodov (od 0 do 15) si podľa vás každé z týchto riešení zaslúži a poriadne zdôvodniť, prečo. (Rozhodujúca je samozrejme správnosť myšlienky a efektívnosť algoritmu. Ak neviete, čo všetko brať pri opravovaní do úvahy, ešte raz si poriadne prečítajte pravidlá KSP.)

**Úloha:** Na planéte Diks je  $N$  miest, ktoré sa volajú  $1, 2, \dots, N$ . Každé dve mestá sú spojené potrubím, ktorým mimozemšťania cestujú. Pre každú dvojicu miest je známe, ako dlho trvá presun potrubím, ktoré ich spája. Pomôžte mimozemšťanovi Krzysztofovi zistiť, ako najrýchlejšie sa dá dostať z mesta 1 do mesta  $N$ .

**Príklad:** Nech  $N = 3$ , na presun medzi 1 a 2 treba 27 minút, medzi 1 a 3 47 minút a medzi mestami 2 a 3 13 minút. Potom z mesta 1 do mesta 3 sa najrýchlejšie dá dostať za 40 minút (cez mesto 2).

**Riešenie 1:**

```
var A : array[1..100,1..100] of integer;
```

```
    naj : integer;
```

```
    N,i : integer;
```

```
    bol : array[1..100] of boolean;
```

```
procedure load;
```

```
var i,j : integer;
```

```
begin
```

```
    readln(N); for i:=1 to N do A[i,i]:=0;
```

```
    for i:=1 to N do for j:=i+1 to N do begin read(A[i,j]); A[j,i]:=A[i,j]; end;
```

```
end;
```

```
procedure skus(kam,dlzka : integer);
```

```
var i : integer;
```

```

begin
  if (kam=N) then begin
    if (dlzka<naj) then naj:=dlzka;
  end else begin
    bol[kam]:=true;
    for i:=1 to N do if not bol[i] then skus(i,dlzka+A[kam,i]);
    bol[kam]:=false;
  end;
end;

begin
  load;
  naj:=30000; fillchar(bol,sizeof(bol),0);
  skus(1,0);
  writeln(naj);
end.

```

**Riešenie 2:** *Mení sa len procedúra skus.*

```

procedure skus(kam,dlzka : integer);
var i,max,pre : integer;
begin
  if (kam=N) then naj:=dlzka else begin
    max:=30000;
    for i:=1 to N do if (not bol[i]) and (A[kam,i]<max) then
      begin max:=A[kam,i]; pre:=i; end;
    skus(pre,dlzka+A[kam,pre]);
  end;
end;

```

**Riešenie 3:** *Procedúra load zostáva stále rovnaká.*

```

var A : array[1..100,1..100] of integer;
    naj : array[1..100] of integer;
    N,i : integer;

procedure skus;
var i,j : integer;
begin
  for i:=1 to N do for j:=1 to N do
    if (naj[j] > naj[i]+A[i,j]) then naj[j]:=naj[i]+A[i,j];
  end;

begin
  load;
  naj[1]:=0; for i:=2 to N do naj[i]:=30000;
  for i:=1 to N-2 do skus;
  writeln(naj[N]);
end.

```