Meno: Tomáš Belan
Príklad 5**, kat.** O
Škola: ŠpMNDaG Teplická, Bratislava
Strana 1 **z** 2

Toto riešenie má lineárnu časovú a pamäťovú náročnosť.

Z veľkej časti je to o matike. Najprv ukážem prípad, keď N nie je deliteľné tromi (je jednoduchší, pri N deliteľnom tromi je viacero riešení).

Zo zadania vieme, že

$$a_n = x_{n-1} + x_n + x_{n+1}$$

a keď miesto n dosadíme n+1,

$$a_{n+1} = x_n + x_{n+1} + x_{n+2}$$

Keď druhú rovnicu odčítam od prvej,

$$a_n - a_{n+1} = x_{n-1} - x_{n+2}$$

čiže po menších úpravách (a dosadení n+1 miesto n) „hlavný vzorec"

$$a_{n+1} - a_{n+2} + x_{n+3} = x_n$$

čo sa hodí: Keďže a-čka všetky poznáme, môžeme vyjadriť

$$x_3 = \text{nejaká konštanta} + x_0$$

$$x_6 = \text{nejaká konštanta} + x_3$$

$$\text{takže } x_6 = \text{konštanta} + \text{druhá konštanta} + x_0$$

atď

a keďže N nie je deliteľné tromi, týmto spôsobom všetky x-ká vyjadríme ako číslo + x_0. Tieto konštanty budeme volať k, kde platí

$$x_n = k_n + x_0$$

z čoho vyplýva aj $k_0 = 0$. S pomocou hlavného vzorca dostaneme

$$k_n = a_{n+1} - a_{n+2} + k_{n+3}$$

a keďže k_0 poznáme, môžeme vyrátať všetky k, v cykle s krokom -3. Zadanie hovorí, že

$$a_0 = x_{-1} + x_0 + x_1$$

$$a_0 = (k_{-1} + x_0) + x_0 + (k_1 + x_0)$$

$$a_0 - k_{-1} - k_1 = 3x_0$$

$$x_0 = (a_0 - k_{-1} - k_1) / 3$$

a keď už poznám x_0, ľahko dorátam všetky ostatné x. (A je to falzifikát vtedy, keď x_0 nie je celé, alebo je nejaké x záporné.)

Druhý prípad – N je deliteľné tromi. Hlavný vzorec stále platí, ale problém pôsobí to, že už nemôžem všetky x-ká vyjadriť ako konštanta + x_0 – môžem síce takto určiť x_3, x_6, x_9 atď, ale tie zvyšné sú izolované, cyklus s krokom -3 sa k nim nikdy nedostane. Ale pozor – x_4, x_7, x_{10} atď sa dajú vyjadriť ako konštanta + x_1, a x_5, x_8, x_{11} sa dajú vyjadriť ako konštanta + x_2. Takže v tejto variante význam k-čiek trochu pozmením:

$$x_n = k_n + x_{(n \bmod 3)}$$

Toto má tú výhodu, že keď zoberieme $k_0 = k_1 = k_2 = 0$, tak sa aj zachová funkčnosť vzorca na rátanie k-čiek. Vzniknú tri samostatné „reťaze" x-iek, ktoré od seba závisia. Keďže hlavný vzorec je taký, aký je, tieto reťaze sú navzájom nezávislé, dokonca až tak, že keď máme hodnoty x_0, x_1, x_2 ktorých súčet je a_1, tak budú sedieť všetky a-čka. Neviem k tomuto spraviť veľký a formálny dôkaz, ako sa patrí, ale vyplýva to z toho, ako sa rátajú k-čka, ako je postavený hlavný vzorec, a z toho, že každé a-čko je súčet jedného čísla z reťaze 0, jedného z reťaze 1 a jedného z reťaze 2.

Meno: Tomáš Belan
Príklad 5, **kat.** O
Škola: ŠpMNDaG Teplická, Bratislava
Strana 2 **z** 2

Istý problém predstavuje zvoliť x_0, x_1 a x_2 tak, aby žiadne x nebolo záporné. To spravím tak, že v každej reťazi nájdem najnižšie x (čiže také s najnižším k-čkom), a tomu priradím hodnotu 1 (najnižší možný kladný lesk):

$$1 = x_n$$
$$x_n = k_n + x_{(n \bmod 3)}$$
$$x_{(n \bmod 3)} = 1 - k_n$$

Takže x_0 bude $1 - k_n$ kde n je číslo najnižšieho prvku reťaze 0, a podobne pre x_1 a x_2. Toto sú ešte len pracovné minimálne možné hodnoty. Ak sú ale ešte stále priveľké (teda $x_0 + x_1 + x_2 > a_1$), tak je to falzifikát, nejde to spraviť bez nejakého záporného x. Ak sú primalé, patrične zvýšim x_0 na hodnotu $x_0 = a_1 - x_1 - x_2$. Potom dorátam zvyšné x-ká. Hotovo.

Tu je moja implementácia popísaného algoritmu v Ruby.[1]

```
readline  # prvy riadok odignorujem, miesto tohto pouzijem a.length
a = readline.split.collect{ |cislo| cislo.to_i }

$N = a.length   # $ znamena globalna premenna, nech ju mozem pouzit v nrm()
x = Array.new($N)
k = Array.new($N, 0)  # naplnene nulami

def nejde() raise("JE TO FALZIFIKAT") end   # raise vyhodi vynimku
nejde if $N < 3  # to nie je nahrdelnik ale nejaky prsten alebo co

def nrm(i)   # normalizuje index pola do intervalu 0..N
  i += $N while i < 0
  i -= $N while i >= $N
  return i
end

if $N % 3 != 0      # nie je to delitelne tromi
  # zaciatok.step(koniec, jedenkrok) je ako 'for' cyklus
  i = -3; $N.times do
    k[nrm(i)] = a[nrm(i+1)] - a[nrm(i+2)] + k[nrm(i+3)]
    i -= 3
  end
  triX0 = a[0] - k[1] - k[nrm(-1)]; nejde if triX0 % 3 != 0
  x[0] = triX0 / 3
  # naplnime pole x (a overime ze nic neni zaporne)
  $N.times { |i| x[i] = k[i] + x[0]; nejde if x[i] < 0 }
else     # je to delitelne tromi
  ($N-1).downto(3) do |i|
    k[i] = a[nrm(i+1)] - a[nrm(i+2)] + k[nrm(i+3)]
  end
  min = 0; 0.step($N-1, 3) { |i| min = i if k[i] < k[min] }
  x[0] = 1 - k[min]
  min = 1; 1.step($N-1, 3) { |i| min = i if k[i] < k[min] }
  x[1] = 1 - k[min]
  min = 2; 2.step($N-1, 3) { |i| min = i if k[i] < k[min] }
  x[2] = 1 - k[min]
  nejde if x[0] + x[1] + x[2] > a[1]
  x[0] = a[1] - x[1] - x[2] if x[0] + x[1] + x[2] < a[1]
  $N.times { |i| x[i] = k[i] + x[i % 3] }
end

puts x.join(" ")   # vypise vysledok
```

1 http://www.ruby.ch/interpreter/rubyinterpreter.shtml je online interpreter, s ktorým si môžete program vyskúšať. Nepodporuje vstup z konzoly, preto miesto prvého a druhého riadku (čo obsahujú 'readline') treba napísať priamo inicializáciu poľa a, napr. „a = [7, 6, 9, 8]“