Metody programowania 2016

Lista zadań na pracownię nr 4+5

Termin zgłaszania w KNO: 6 czerwca 2016, godzina 6:00 AM CET

W zamierzchłej przeszłości (lata dziewięćdziesiąte dwudziestego wieku) płatności można było regulować wystawiając tak zwane „czeki". Kwotę, na jaką był wystawiony czek, wpisywało się zwykle ręcznie. Aby uniknąć oszustw, kwotę tę wpisywano dwukrotnie: w zapisie dziesiętnym oraz słownie, dodając na końcu nazwę waluty (co zabezpieczało przed możliwością dopisania dodatkowych słów na końcu kwoty). Zadanie polega na napisaniu w Haskellu programu `slownie`, który przedstawia słownie podaną kwotę waluty, np.:

```
shell> slownie 1 PLN
jeden złoty
shell> slownie 222 PLN
dwieście dwadzieścia dwa złote
shell> slownie 127 PLN
sto dwadzieścia siedem złotych
shell> slownie 83496523 PLN
osiemdziesiąt trzy miliony czterysta dziewięćdziesiąt sześć tysięcy pięćset dwadzieścia trzy złote
shell> slownie 1 UAH
jedna hrywna
shell> slownie 222 UAH
dwieście dwadzieścia dwie hrywny
shell> slownie 127 UAH
sto dwadzieścia siedem hrywien
```

Do zastosowań bankowych wystarczy, by program umiał wypisywać jedynie liczby dodatnie, ale nie ma powodu, by nie rozszerzyć go na dowolne liczby całkowite:

```
shell> slownie 0 PLN
zero złotych
shell> slownie -25 UAH
minus dwadzieścia pięć hrywien
```

Liczebniki główne mniejsze od 1000 tworzymy w języku polskim wymieniając osobno liczby setek, dziesiątek i jedności, np. liczbę 453 nazywamy po polsku „czterysta pięćdziesiąt trzy". Liczby od 11 do 19 mają swoje odrębne nazwy. Mówimy zatem np. „siedemnaście", a nie „dziesięć siedem":

1	jeden	11	jedenaście	10	dziesięć	100	sto
2	dwa	12	dwanaście	20	dwadzieścia	200	dwieście
3	trzy	13	trzynaście	30	trzydzieści	300	trzysta
4	cztery	14	czternaście	40	czterdzieści	400	czterysta
5	pięć	15	piętnaście	50	pięćdziesiąt	500	pięćset
6	sześć	16	szesnaście	60	sześćdziesiąt	600	sześćset
7	siedem	17	siedemnaście	70	siedemdziesiąt	700	siedemset
8	osiem	18	osiemnaście	80	osiemdziesiąt	800	osiemset
9	dziewięć	19	dziewiętnaście	90	dziewięćdziesiąt	900	dziewięćset

W połączeniach z rzeczownikiem liczebniki główne mają specjalną formę dla rzeczowników męskoosobowych (np. „czterech chłopców", „dwunastu studentów"). Na szczęście nazwy walut są rzeczownikami rodzaju męskorzeczowego („złoty"), żeńskiego („hrywna") lub nijakiego („peso") i łączą się zwykle z formą podstawową liczebników. Większość liczebników możemy więc tworzyć jedynie w formie podstawowej. Wyjątkiem jest osobna forma liczebników „jeden" i „dwa" w połączeniach z rzeczownikami rodzaju żeńskiego (mówimy „jedna, dwie hrywny", a nie „jeden, dwa hrywny") i nijakiego (mówimy „jedno peso" a nie „jeden peso"). W połączeniach rzeczowników z liczebnikami odmienia się natomiast rzeczownik. Liczebnik „jeden" łączymy z mianownikiem liczby pojedynczej („jeden złoty", „jedna hrywna"), liczebniki od „dwa" do „cztery" — z mianownikiem liczby mnogiej („trzy złote", „cztery hrywny") a liczebniki „zero" i od „pięć" do „dziewięć" — z dopełniaczem liczby mnogiej rzeczownika („zero złotych", „siedem hrywien").

Podobny sposób tworzenia liczebników głównych występuje w większości języków europejskich. Zwykle mamy pozostałość systemu dwudziestkowego[1] w postaci osobnych nazw dla liczb z zakresu 11–19, choć zdarzają się dodatkowe nieregularności (np. w języku francuskim). Nam przydadzą się jeszcze liczebniki łacińskie:

1	ūnus	11	ūndecim	10	decem	100	centum
2	duo	12	duodecim	20	vīgintī	200	ducentī
3	trēs	13	tredecim	30	trīgintā	300	trecentī
4	quattuor	14	quattuordecim	40	quadrāgintā	400	quadringentī
5	quīnque	15	quīndecim	50	quīnquāgintā	500	quīngentī
6	sex	16	sēdecim	60	sexāgintā	600	sescentī
7	septem	17	septendecim	70	septuāgintā	700	septingentī
8	octō	18	duodēvīgintī	80	octōgintā	800	octingentī
9	novem	19	ūndēvīgintī	90	nōnāgintā	900	nōngentī

Liczebniki oznaczające liczby $n \geq 1000$ budujemy w następujący sposób: przedstawiamy wpierw liczbę w zapisie pozycyjnym przy podstawie 1000, tj. $n = n_k \cdot 1000^k + n_{k-1} \cdot 1000^{k-1} + \ldots + n_1 \cdot 1000 + n_0$, gdzie $n_i < 1000$ dla $i = 0, \ldots, k$. Każdą z liczb n_i przedstawiamy za pomocą liczebnika złożonego w opisany wyżej sposób. Liczebniki odpowiadające liczbom n_i łączymy z opisanymi niżej liczebnikami oznaczającymi potęgi liczby 1000 i tak otrzymane ciągi wyrazów ustawiamy jeden po drugim w kolejności malejących potęg.

Pozostaje opisać, w jaki sposób tworzymy liczebniki główne oznaczające liczby 1000^k. Dla $k = 1$, tj. na oznaczenie liczby 1000 Rzymianie używali liczebnika „mīlle”. Słowianie mają własne słowo „tysiąc” (ukr. тисяча, ros. тысяча) etymologicznie związane z takimi słowami, jak „tyć” bądź „tuczyć” i oznaczające coś silnego, wielkiego, potężnego. Nazwy wyższych potęg są spolszczonymi i zniekształconymi liczebnikami zapożyczonymi z łaciny. Na oznaczenie „tysiąca tysięcy” używamy słowa „milion” (ukr. мільйон, ros. миллион) od łacińskiego „mīlle”. Na tym jednak podobieństwa między językami polskim oraz ukraińskim i rosyjskim się kończą. Istnieją bowiem na świecie dwie konwencje nazywania potęg liczby 1000: tzw. krótka i długa skala. W Polsce i krajach kontynentalnej Europy Zachodniej używa się długiej skali, zaś w Wielkiej Brytanii, USA, Rosji i krajach posowieckich — krótkiej. Według długiej skali liczebnik oznaczający 1000^{2k+i} budujemy na podstawie łacińskiego wyrażenia oznaczającego k i dodajemy końcówkę „-lion”, jeśli $i = 0$ oraz „-liard”, jeśli $i = 1$. W krótkiej skali liczebnik oznaczający 1000^{k+1} budujemy na podstawie takiego samego łacińskiego wyrażenia oznaczającego liczbę k i dodajemy końcówkę „-lion”. Prowadzi to do wielu nieporozumień. Na przykład „bilion” w długiej skali oznacza $1000^{2 \cdot 2} = 10^{12}$, a w krótkiej $1000^{2+1} = 10^9$. Aby zamieszanie było większe, to w krajach, w których używa się krótkiej skali, liczebnik „miliard” jest synonimem „biliona”.

Znormalizowane, notowane w słownikach językowych nazwy potęg liczby 1000 występują wyłącznie dla małych k, przeważnie od 1 do 9. Mamy więc:

k	polski	skala długa	ukraiński	angielski	skala krótka
1	milion	$1000^{2\cdot1} = 10^6$	мільйон	million	$1000^{1+1} = 10^6$
	miliard	$1000^{2\cdot1+1} = 10^9$	мільярд	milliard	
2	bilion	$1000^{2\cdot2} = 10^{12}$	більйон	billion	$1000^{2+1} = 10^9$
	biliard	$1000^{2\cdot2+1} = 10^{15}$			
3	trylion	$1000^{2\cdot3} = 10^{18}$	трильйон	trillion	$1000^{3+1} = 10^{12}$
	tryliard	$1000^{2\cdot3+1} = 10^{21}$			
4	kwadrylion	$1000^{2\cdot4} = 10^{24}$	квадрильйон	quadrillion	$1000^{4+1} = 10^{15}$
	kwadryliard	$1000^{2\cdot4+1} = 10^{27}$			
5	kwintylion	$1000^{2\cdot5} = 10^{30}$	квінтильйон	quintillion	$1000^{5+1} = 10^{18}$
	kwintyliard	$1000^{2\cdot5+1} = 10^{33}$			
6	sekstylion	$1000^{2\cdot6} = 10^{36}$	секстильйон	sextillion	$1000^{6+1} = 10^{21}$
	sekstyliard	$1000^{2\cdot6+1} = 10^{39}$			
7	septylion	$1000^{2\cdot7} = 10^{42}$	септильйон	septillion	$1000^{7+1} = 10^{24}$
	septyliard	$1000^{2\cdot7+1} = 10^{45}$			
8	oktylion	$1000^{2\cdot8} = 10^{48}$	октильйон	octillion	$1000^{8+1} = 10^{27}$
	oktyliard	$1000^{2\cdot8+1} = 10^{51}$			
9	nonilion	$1000^{2\cdot9} = 10^{54}$	нонільйон	nonillion	$1000^{9+1} = 10^{30}$
	noniliard	$1000^{2\cdot9+1} = 10^{57}$			

Dla $k = 9$ niektóre źródła podają alternatywną polską pisownię „nonylion” i „nonyliard”.

[1] System dwudziestkowy został porzucony, gdy ludzie zaczęli nosić buty.

Można się zastanawiać, po co komu tak wielkie liczby, ale przedstawione poniżej przykłady banknotów pokazują, że kwoty, które chcemy niekiedy przedstawiać, mogą być całkiem duże[2]:

Banknot o najwyższym nominale, który został kiedykolwiek wydrukowany, to 10^{21} pengő, wprowadzony do obiegu na Węgrzech w 1946 roku[3]. Niedługo później dokonano wymiany waluty na forinty po kursie $4 \cdot 10^{29}$ pengő za 1 forinta…

W języku polskim nazwy liczb 1000^{2k+i} to „p_k-lion” dla $i = 0$ i „p_k-liard” dla $i = 1$. Przyjmijmy[4], że przedrostki p_k dla $k < 1000$ tworzymy zgodnie z poniższą tabelą:

$k < 10$

k	p_k
1	mi-
2	bi-
3	try-
4	kwadry-
5	kwinty-
6	seksty-
7	septy-
8	okty-
9	noni-

$k = 100c + 10b + a$, $p_k = q_a r_b s_c$

a	q_a	b	r_b	c	s_c
1	un-	1	decy-	1	centy-
2	do-	2	wicy-	2	ducenty-
3	tri-	3	trycy-	3	trycenty-
4	kwatuor-	4	kwadragi-	4	kwadryge-
5	kwin-	5	kwintagi-	5	kwinge-
6	seks-	6	seksginty-	6	sescenty-
7	septen-	7	septagi-	7	septynge-
8	okto-	8	oktagi-	8	oktynge-
9	nowem-	9	nonagi-	9	nonge-

Mamy więc na przykład:

$$\begin{aligned}
10^{12} &= 10^{6\cdot 2} = \text{bilion} & 10^{282} &= 10^{6\cdot 47} = \text{septenkwadragilion}\\
10^{15} &= 10^{6\cdot 2+3} = \text{biliard} & 10^{333} &= 10^{6\cdot 55+3} = \text{kwinkwintagiliard}\\
10^{18} &= 10^{6\cdot 3} = \text{trylion} & 10^{369} &= 10^{6\cdot 61+3} = \text{unseksgintyliard}\\
10^{24} &= 10^{6\cdot 4} = \text{kwadrylion} & 10^{378} &= 10^{6\cdot 63} = \text{triseksgintylion}\\
10^{48} &= 10^{6\cdot 8} = \text{oktylion} & 10^{399} &= 10^{6\cdot 66+3} = \text{seksseksgintyliard}\\
10^{69} &= 10^{6\cdot 11+3} = \text{undecyliard} & 10^{450} &= 10^{6\cdot 75} = \text{kwinseptagilion}\\
10^{99} &= 10^{6\cdot 16+3} = \text{seksdecyliard} & 10^{510} &= 10^{6\cdot 85} = \text{kwinoktagilion}\\
10^{120} &= 10^{6\cdot 20} = \text{wicylion} & 10^{600} &= 10^{6\cdot 100} = \text{centylion}\\
10^{150} &= 10^{6\cdot 25} = \text{kwinwicylion} & 10^{666} &= 10^{6\cdot 111} = \text{undecycentylion}\\
10^{201} &= 10^{6\cdot 33+3} = \text{tritrycyliard} & 10^{999} &= 10^{6\cdot 166+3} = \text{seksseksgintycentyliard}\\
10^{216} &= 10^{6\cdot 36} = \text{sekstrycylion} & 10^{1197} &= 10^{6\cdot 199+3} = \text{nowemnonagicentyliard}\\
10^{240} &= 10^{6\cdot 40} = \text{kwadragilion} & 10^{5997} &= 10^{6\cdot 999+3} = \text{nowemnonaginongeliard}
\end{aligned}$$

Googol (10^{100}) to inaczej dziesięć seksdecyliardów, a 10^{1000} to dziesięć seksseksgintycentyliardów.

Literatura

[1] John H. Conway, Richard K. Guy, *The Book of Numbers*, Copernicus (Springer), 1996.

[2] *Codes for the representation of currencies*, ISO 4217:2015.

[3] Georges Ifrah, *Dzieje liczby czyli historia wielkiego wynalazku*, Ossolineum, 1990.

[4] Georges Ifrah, *Historia powszechna cyfr*, W.A.B., 2006.

[2] Źródło zdjęć: Wikimedia Commons.

[3] Zob. hasło *Hungarian pengő hyperinflation* w Wikipedii.

[4] Nie wszystkie przedrostki są ustalone lub notowane w literaturze. Autor zadania wziął je częściowo z Wikipedii, a częściowo z głowy, czyli z niczego. Niektóre źródła proponują np. „oktogi-” zamiast „oktagi-”, „duo-” zamiast „do-” itp. Najczęściej cytowana wersja angielska została zaproponowana w książce [1], str. 13–16.

Zadanie. Zaprogramuj w Haskellu moduł `Slownie`:

```
module Slownie (Rodzaj(..), Waluta(..), slownie) where
   data Rodzaj = Meski | Zenski | Nijaki deriving Show

   data Waluta = Waluta {
      mianownik_poj :: String,
      mianownik_mn  :: String,
      dopelniacz_mn :: String,
      rodzaj        :: Rodzaj
   } deriving Show

   slownie :: Waluta -> Integer -> String
```

implementujący funkcję `slownie` wyznaczającą słowną reprezentację podanej waluty. Implementacja może dotyczyć języka polskiego lub innego języka europejskiego (wówczas moduł powinien nazywać się np. `Verbally`, `Usno` itp.). Wybór języka innego niż polski należy skonsultować z prowadzącym pracownię.

Funkcja `slownie` powinna działać poprawnie dla liczb o wartości bezwzględnej mniejszej niż $10^{6000} - 1$ (dla długiej skali; w przypadku języka używającego skali krótkiej: 10^{3003}). Dla większych liczb funkcja powinna zwracać napis `mnóstwo` (lub jego odpowiednik w innym języku).

Następnie napisz w Haskellu program `slownie` wykorzystujący moduł `Slownie`, który wywołany z argumentami:

`shell> slownie` *n waluta*

wypisze do standardowego strumienia wyjściowego słowną reprezentację podanej kwoty w podanej walucie. Program powinien obsługiwać co najmniej następujące waluty[5]:

AUD BGN BRL BYR CAD CHF CNY CZK DKK EUR GBP HKD HRK HUF IDR ISK JPY KRW MXN MYR NOK NZD PHP PLN RON RUB SDR SEK SGD THB TRY UAH USD ZAR

Problemy, które mogą powstać podczas rozwiązywania zadania[6] skonsultuj z prowadzącym pracownię.

Niniejszy plik jest dostępny w serwisie KNO na stronie zajęć.

[5] Kody walut według normy ISO 4217, patrz [2].

[6] Np. warianty pisowni liczebników, pytania: czy 1000000 to „milion", czy „jeden milion" itp. Również sposób kodowania znaków spoza zbioru ASCII może stwarzać problemy.