

1 Riešené úlohy

2 Úlohy pre základné programovanie

2.1 Jednoduché cykly

Vo všetkých nasledujúcich úlohách budeme predpokladať, že máme dostatok pamäte na výpočet uvedených hodnôt.

2.1.1

Je dané prirodzené číslo n . Je potrebné vypočítať:

a) 2^n

```
f_2_1_1_a n = 2^n
```

b) $n!$

```
f_2_1_1_b n = product [1..n]
```

c) $(1 + 1^{-2}) \cdot (1 + 2^{-2}) \cdot \dots \cdot (1 + n^{-2})$

```
f_2_1_1_c n = product [ 1+i**(-2) | i <- [1..n] ]
```

d) $\sin^{-1} 1 + (\sin 1 + \sin 2)^{-1} + \dots + (\sin 1 + \sin n)^{-1}$

```
f_2_1_1_d n = sum [(sin 1 + sin i)**(-1) | i <- [1..n]]
```

e) $\sqrt{2 + \sqrt{2 + \dots + \sqrt{2}}}$ (n druhých odmocnín)

```
f_2_1_1_e n = last $ take n $ iterate sqrt 2
```

f) $\cos(1) \cdot \sin^{-1} 1 \cdot (\cos 1 + \cos 2) \cdot (\sin 1 + \sin 2)^{-1} \cdot \dots \cdot (\cos 1 + \dots + \cos n) \cdot (\sin 1 + \sin n)^{-1}$

g) $\sqrt{3 + \sqrt{6 + \dots + \sqrt{3(n-1) + \sqrt{3n}}}}$

Vstup: prirodzené číslo n ;

Výstup: vypočítaná hodnota podľa uvedeného vzťahu.

2.1.3

Je dané reálne číslo a . Je potrebné nájsť:

1. medzi číslami $1, 1 + 2^{-1}, 1 + 2^{-1} + 3^{-1}, \dots$ prvé číslo také, ktoré je väčšie než a ;

Stačí si uvedomiť, že ide o zápis čísiel $1, 1 + \frac{1}{2}, 1 + \frac{1}{2} + \frac{1}{3}, \dots$. Každý prvok tejto postupnosti je $\sum_i^n \frac{1}{i}$.

```
f_2_1_3_sum n = sum [1 / i | i <- [1..n]]  
f_2_1_3_a a = head $ dropWhile (<=a) [ f_2_1_3_sum n | n <- [1..] ]
```

2. najmenšie prirodzené číslo n také, že $1 + 2^{-1} + 3^{-1} + \dots + n^{-1} > a$

Generujeme postupnosti súčtov dovtedy, kým je prvok postupnosti (teda jedna suma), menšia než a .

```
f_2_1_3_b a = length $ takeWhile (<a) [ f_2_1_3_sum n | n <- [1..]
```

Vstup: reálne číslo a ;

Výstup: číslo vyhovujúce podmienke

2.1.5

Je dané prirodzené číslo n .

a) Koľko je cifier v čísle n ?

```
pocetCifier n
  | n < 0 = error "Zaporne_cislo"
  | n < 10 = 1
  | otherwise = 1 + pocetCifier (n 'div' 10)

f_2_1_5_a = pocetCifier
```

b) Koľko rôznych cifier je v čísle n ?

Použijeme pomocnú funkciu, ktorá pre dané číslo vráti zoznam jeho cifier. Popri tom využijeme pomocnú funkciu `Data.List.nub`, ktorá vráti unikátne prvky zo zoznamu.

```
cifier n
  | n < 10 = [n]
  | otherwise = cifier (n 'div' 10) ++ [cifra]
  where cifra = n 'mod' 10
cisloNaCifry = cifier

pocetRoznychCifier n = length $ nub $ cifier n
f_2_1_5_b = pocetRoznychCifier
```

c) Čomu sa rovná súčet všetkých jeho cifier?

```
sucetCifier n = sum $ cifier n
f_2_1_5_c n = sucetCifier
```

d) Čomu sa rovná súčet všetkých jeho cifier na párnych pozíciách?

```
sucetCifierNaParnychPoziciach n =
  sum [cifier n !! (index - 1) | index <- [2,4..pocetCifier n] ]
f_2_1_5_d = sucetCifierNaParnychPoziciach
```

Vstup: prirodzené číslo n ;

Výstup: výsledná hodnota ako odpoveď na danú otázku

2.1.6

Je dané prirodzené číslo n . Je potrebné

- a) určiť počet výskytov číslice 3 v zápise čísla n^2 ;

```
-- pocetVyskytovCifry cifra n = length
--           $ filter (\c -> c == cifra) (cisloNaCifry n)
pocetVyskytovCifry cifra n = length
           $ filter (== cifra) (cisloNaCifry n)
pocetVyskytovTrojkyVNNaDruhu n = pocetVyskytovCifry 3 (n^2)
f_2_1_6_a = pocetVyskytovTrojkyVNNaDruhu
```

- b) zameniť poradie číslic v čísle n na opačné;

```
cifryNaCislo cifra = foldl (\acc cifra -> acc * 10 + cifra) 0 cifra
zamenitPoradieCifierNaOpacne n = cifryNaCislo $ reverse $ cifier n
f_2_1_6_b = zamenitPoradieCifierNaOpacne
```

- c) vymeniť prvú a poslednú číslicu v čísle n ;

```
vymenitPrvuAPoslednuCifru n
  | n < 0 = error "Zaporne cislo"
  | n < 10 = n
  | otherwise = cifryNaCislo $ [posledna] ++ stred ++ [prva]
  where prva = head cifra
        posledna = last cifra
        stred = init ( tail cifra )
        cifra = cifier n
f_2_1_6_c = vymenitPrvuAPoslednuCifru
```

- d) pripísať dvojku na začiatok a koniec čísla.

```
pripisatDvojkuNaZaciatokAKoniec n = cifryNaCislo $
  [2] ++ cisloNaCifry n ++ [2]
f_2_1_6_d = pripisatDvojkuNaZaciatokAKoniec
```

Vstup: prirodzené číslo n ;

Výstup: hodnoty vyhovujúce podmienkam.