

Programovanie (1) v C/C++ 2024/25

Cvičenia 4, príklad 3

Faktoriál

V tomto príklade dopisujte vaše riešenie do poskytnutej kostry programu, pričom naprogramujte požadované funkcie a hotovú časť programu nemeňte.

Ako sme videli na prednáške 2, funkcia faktoriál veľmi rýchlo rastie a už pre $n = 13$ sa $n!$ nezместí do premennej typu `int`. Zopár ďalších hodnôt vieme spočítať pomocou typu `long long int`, ale v tejto úlohe naprogramujeme výpočet faktoriálu pre n až po 1000, kde výsledok má tisícky cifier.

Čísla si budeme ukladať do poľa `int`-ov, každú cifru ako jeden prvok poľa. V priloženej kostre je na tento účel zadefinovaný `struct cislo`, ktorý obsahuje pole cifier a počet cifier daného čísla. Na indexe 0 v poli `cifry` je najmenej významná cifra čísla, teda jednotky, na indexe 1 sú desiatky, na indexe 2 stovky atď. Ďalšie položky za poslednou cifrou nemusai byť inicializované, ich hodnotu nepoužívajte.

V kostre je už hotová aj funkcia `main`, ktorá načíta tri celé kladné čísla a , b a n , každé zadané na jednom riadku. Prvé číslo a sa potenciálne nemusí zmestiť do premennej typu `int`, dve čísla b a n sa do takejto premennej dajú uložiť. Program potom spočíta a vypíše čísla a , $a \cdot b$ a $n!$ a ku každému vypíše aj počet jeho cifier. Aby však program fungoval, je treba doňho doprogramovať funkcie `vypis`, `nasob` a `faktorial`. Riadte sa pokynmi v komentároch pri jednotlivých funkciách. Ak ich budete programovať v tomto poradí, mal by vám najskôr správne fungovať prvý výpis, potom druhý a napokon tretí.

Na ukážku je už hotová funkcia `vytvorCislo`, ktorá rozloží číslo typu `int` na cifry do poľa v štruktúre typu `cislo`.

Návod pre funkciu `nasob`. Predstavme si, že v premennej `a` typu `cislo` máme uloženú hodnotu 432. V premennej `a.pocet_cifier` teda bude hodnota 3 a v poli `a.cifry` budú hodnoty 2,3,4. Keď toto číslo násobíme hodnotou $b = 40$, môžeme postupovať takto: prenásobíme najmenej významnú cifru, teda $2 \cdot 40 = 80$. Najmenej významnú cifru výsledku, teda 0, uložíme späť do `a.cifry[0]`, hodnota 8 nám zostane ako zvyšok. Potom násobíme číslom b ďalšiu cifru a pripočítame zvyšok: $3 \cdot 40 + 8 = 128$. Do `a.cifry[1]` uložíme poslednú cifru 8, 12 bude zvyšok. Tento postup ešte opakujeme pre poslednú cifru pôvodnej hodnoty a : $4 \cdot 40 + 12 = 172$. Do `a.cifry[2]` uložíme cifru 2, 17 bude zvyšok. Teraz už nemáme ďalšie cifry vo vstupnom čísle, takže stačí zvyšok rozložiť na cifry a uložiť do poľa `a.cifry` podobne ako vo funkcii `vytvorCislo`. Vo výsledku máme v `a.pocet_cifier` hodnotu 5 a v poli `a.cifry` hodnoty 0, 8, 2, 7, 1.

Príklad vstupu:

```
432
40
5
```

Príklad výstupu:

```
a: 432
  pocet cifier: 3
a*40: 17280
  pocet cifier: 5
5!: 120
  pocet cifier: 3
```

Príklad vstupu:

1112221112229
4
1000

Príklad výstupu:

a: 1112221112229
 pocet cifier: 13
a*4: 4448884448916
 pocet cifier: 13
1000!: 40238726007709377354370243392300398571937486421071463254379...
 pocet cifier: 2568

V tomto príklade kvôli prehľadnosti zobrazujeme iba zopár najvýznamnejších cifier čísla 1000!, číslo však treba vypísať celé, so všetkými 2568 ciframi (veľa cifier na konci sú nuly)