

## Cvícenie 6

### Instrukcie:

- Vypracujte všetky ulohy z tohto cvícenia. Na cvícení sa pokúste vypracovať čo najviac uloh a ulohy, ktoré nestihnúte na cvícení, potom vypracujte doma. Uloha 11 je podľa mojho názoru veľmi pekná. Dufam, že ju niektorí z vás ocenia :)
- Ulohy riešte bez použitia zložených datových typov (zoznamy, reťazce, množiny, slovníky, atď.). Zložené datové typy sme ešte nepreberali.

1. Definujte funkciu, ktorá bude načítavať čísla z klavesnice, kým užívateľ nezadá záporné číslo alebo nulu. Po zadaní záporného čísla alebo nuly funkcia vráti súčet kladných načítaných čísel. Príklad činnosti funkcie: ak budú načítané čísla 10,4,0 funkcia **vráti** 14. Ak budú načítané čísla 1,2,3,-1 funkcia **vráti** 6. Ak budú načítané čísla 1,5,-5 funkcia **vráti** 6.

2. Definujte funkciu s parametrom  $n$ , ktorá vypíše všetky stvorce menšie ako číslo  $n$ . (Prírodné číslo nazývame stvorcom, ak je druhou mocninou nejakého prírodného čísla. Napríklad čísla 0,1,4,9,16 sú stvorce.

Príklad činnosti funkcie: pre hodnotu parametra  $n=10$  funkcia **vypíše** čísla 0,1,4,9. Pre hodnotu parametra  $n=16$  funkcia **vypíše** čísla 0,1,4,9. Pre hodnotu parametra  $n=17$  funkcia **vypíše** čísla 0,1,4,9,16.

3. Definujte funkciu s parametrom  $n$ , ktorá vráti najväčšie prírodné číslo  $x$  také, že  $2^x$  je menšie ako  $n$ . Skúste úlohu vyriešiť bez použitia operácie **\*\***.

Príklad činnosti funkcie: pre hodnotu parametra  $n=10$  funkcia **vráti** číslo 3. Pre hodnotu parametra  $n=33$  funkcia **vráti** číslo 5. Pre hodnotu parametra  $n=32$  funkcia **vráti** číslo 4.

4. Definujte funkciu, ktorá bude načítavať čísla z klavesnice, kým užívateľ nezadá nulu. Po zadaní nuly funkcia vráti poradové číslo najväčšieho načítaného čísla. Napríklad pre postupnosť čísel -5, -4, -3, 0 funkcia **vráti** číslo 4, lebo najväčšie číslo (v tomto prípade je to číslo 0) bolo načítané ako štvrté. Pre postupnosť -5, 2, 8, 1, 0 zase funkcia **vráti** číslo 3.

5. Definujte funkciu, ktorá bude načítavať čísla z klavesnice, kým užívateľ nezadá nulu. Po zadaní nuly funkcia vráti počet kolkokrat sa stalo, že načítané číslo bolo väčšie ako číslo načítané pred ním. Napríklad pre postupnosť čísel 1,8,9,2,4,-1,0 funkcia **vráti** číslo 4, pretože  $8 > 1$ ,  $9 > 8$ ,  $4 > 2$  a  $0 > -1$ .

6. Definujte funkciu, ktorá bude načítavať čísla z klavesnice, kým užívateľ nezadá nulu. Po zadaní nuly funkcia vráti kolko z načítaných čísel je rovných najväčšiemu načítanému číslu. Napríklad pre postupnosť čísel 2,8,3,3,8,0 funkcia **vráti** číslo 2. Otestujte vašu funkciu pre postupnosť 2,2,2,2,8,1,8,1,8,0.

7. Definujte funkciu, ktorá bude načítavať čísla z klavesnice, kým užívateľ nezadá nulu. Po zadaní nuly funkcia vráti dĺžku najdlhšej podpostupnosti po sebe idúcich rovnakých čísel z postupnosti načítaných čísel. Napríklad pre postupnosť čísel 2,2,3,3,3,3,1,3,3,0 funkcia **vráti** číslo 4.

8. V kapitole 6 v knihe ste sa oboznámili s Fibonacciho postupnosťou. Definujte teraz funkciu s parametrom  $x$ , ktorá vráti index čísla  $x$  vo Fibonacciho postupnosti. To znamená, že funkcia vráti  $i$  také, že  $a_i = x$ . Ak sa číslo  $x$  nenachádza vo Fibonacciho postupnosti, funkcia vráti číslo -1.

Využite pri tom funkciu fibonacci( $n$ ) z knihy. Napríklad pre parameter  $x=3$  **vráti** funkcia číslo 4, pretože číslo 3 má index 4 vo Fibonacciho postupnosti. Pre parameter  $x=6$  **vráti** funkcia číslo -1, lebo číslo 6 sa vo Fibonacciho postupnosti nenachádza.

9. Vyrieste cvicenie 7.1 na strane 69 v knihe.

10. Vyrieste cvicenie 7.2 na strane 69 v knihe.

11. Vyrieste cvicenie 7.3 na strane 70 v knihe.

12. Definujte funkciu *kvocient(a, b)*, ktora ma 2 vstupy: nezaporne cele cisla *a* a *b*. Funkcia vrati dolnu celu cast po deleni *a/b*, t.j. rovnaku hodnotu, ako by vratil operator *a // b*. Funkciu implementujte **bez pouzitia** operatorov *//, /, %*. (t.j. bez celociselneho delenia, klasickeho delenia a operatora modulo). V pripade, ze  $b == 0$ , funkcia vrati hodnotu *None*.  
Prklady navratovych hodnot: *kvocient(6,3)* vrati 2, *kvocient(1,4)* vrati 0, *kvocient(3,0)* vrati *None*, *kvocient(11,3)* vrati 3.

13. Definujte funkciu *modulo(a, b)*, ktora ma 2 vstupy: nezaporne cele cisla *a* a *b*. Funkcia vrati zvysok *a* po deleni cislom *b*, t.j. rovnaku hodnotu, ako by vratil operator *a % b*. Funkciu implementujte **bez pouzitia** operatorov *//, /, %*. (t.j. bez celociselneho delenia, klasickeho delenia a operatora modulo). V pripade, ze  $b == 0$ , funkcia vrati hodnotu *None*.  
Prklady navratovych hodnot: *modulo(6,3)* vrati 0, *modulo(1,4)* vrati 1, *modulo(3,0)* vrati *None*, *modulo(11,3)* vrati 2.

14. Definujte funkciu *faktorizacia(n)*, ktora pre argument *n*, ktorym je kladne cele cislo vacsie ako 1, **vypise** jeho faktorizaciju, t.j. rozklad cisla *n* na sucin prvocisiel. Pre jednoduchost uvazujte, ze program **vypise** kazdy faktor na novy riadok. Na poradi vypisanych faktorov nezalezi.

Prklady cinnosti funkcie:

*faktorizacia(10)* postupne vypise na obrazovku:

2

5

*faktorizacia(100)* postupne vypise na obrazovku:

2

2

5

5

*faktorizacia(13)* postupne vypise na obrazovku:

13

*faktorizacia(80850)* postupne vypise na obrazovku:

2

3

5

5

7

7

11