

Pracovní list 1: Opakování algoritmizace I

Co už máme znát

- elementární algoritmy: extrém, dvojice hodnot, počet vybraných hodnot;
- způsoby zadání a načtení řady hodnot;
- standardní vstup a výstup;
- pole;
- podprogram.

Kontrolní otázky

- 1.1 Jak funguje algoritmus pro určení maximální hodnoty z řady hodnot?
- 1.2 Jakými způsoby lze zadat řadu hodnot na vstupu?
- 1.3 V čem spočívá princip zpracování po sobě jdoucích dvojic hodnot vstupu?
- 1.4 Jakými operacemi se nařídí čtení ze standardního vstupu?
- 1.5 Jakými operacemi se nařídí výpis do standardního výstupu a případné formátování hodnot?
- 1.6 Jak se definuje podprogram? Jaké druhy podprogramů jsou k dispozici?
- 1.7 Jak se definuje datový typ představující pole?
- 1.8 Co je rekurze a k čemu ji lze využít?

Příprava na cvičení

Ve cvičení budeme potřebovat přístup k serveru akela, kde využijeme překladač jazyka C++, editor (joe, případně jiný) a schopnosti příkazového řádku.

Řešené příklady

Příklad 1.1 Na vstupu se nachází řada celočíselných hodnot zakončená nulou. Vypište na výstup počet vstupních hodnot dělitelných beze zbytku deseti.

Řešení: Pro řešení příkladu potřebujeme složit dva elementární algoritmy: čtení hodnot zakončených smlouvenou koncovou hodnotou, která typicky nepatří do zpracovávaných dat, a počet vybraných hodnot (čísel dělitelných deseti). Elementární algoritmy mají známý tvar, připomeneme jej:

```
1 //vstup řady hodnot zakončených smlouvenou
2 //koncovou hodnotou K nepatřící do řady:
3 cin >> hodnota; //přečtení první hodnoty vstupu
4 while (hodnota != K) {
5     ... //zpracování načtené hodnoty
6     cin >> hodnota; //čtení další hodnoty
7 }
8
9 /*počet vybraných hodnot*/
10 int pocet = 0; //počítadlo, počáteční hodnota je nula
11 ... //začátek libovolného cyklu, v němž se získávají hodnoty
12     if (hodnota==vybrana) pocet++; //výběr a započítání počtu
13 ... //konec cyklu
```

Nyní aplikujeme oba algoritmy na danou úlohu a doplníme ostatní náležitosti programu:

```
14 #include <iostream>
15 using namespace std;
16
17 int main() {
18     int hodnota;
19     int pocet=0;
20     cin >> hodnota;
21     while (hodnota != 0){
22         if (hodnota % 10 == 0) pocet++;
23         cin >> hodnota;
24     }
25     cout << pocet << endl;
26     return 0;
27 }
```

Příklady

Příklad 1.2 Napište program v jazyce C++ řešící tuto úlohu: Na standardním vstupu se nachází řada desetinných čísel představující měsíční zisky nejmenované firmy v tisících eur. Vypište pořadové číslo měsíce, v němž byl zaznamenán největší nárůst zisku oproti předchozímu měsíci.

Příklad 1.3 Na standardním vstupu jsou zadány rozměry matice (dvě čísla – počet řádků a počet sloupců) a za nimi se nachází řada celočíselných hodnot představujících prvky matice zapsané po řádcích. Počet hodnot může být menší, než je potřebný počet prvků takto zadané matice, v tom případě se zbylé prvky budou považovat za nulové. Pokud bude hodnot naopak více, zpracuje se jen tolik, kolik je pro zadané rozměry potřeba.

Načtete takto zadanou matici, vypište ji přehledným způsobem na výstup, provedte její transpozici a výsledek opět vypište na výstup.

Příklad 1.4 Na každém řádku standardního vstupu je zadán název zboží. Vypište tento seznam názvů na standardní výstup v obráceném pořadí.

Co máme po cvičení umět

- Skládání elementárních algoritmů při řešení úlohy.
- Práce se standardními soubory, přesměrování vstupu a výstupu.
- Definice podprogramu.
- Definice konstanty, datového typu.
- Využití pole, podprogramu, rekurze.

Kontrolní otázky

- 1.9 Jakým způsobem je možné ošetřit vstupní hodnoty?
- 1.10 Jak se definuje procedura a jak funkce?
- 1.11 Je obrácení pořadí vstupních hodnot vhodným případem rekurzivního algoritmu? Proč?
- 1.12 Kdy je vhodné použít statické pole a jaké nebezpečí hrozí?