

Úloha 1.1: Napíšte program, ktorý načíta celé číslo N nasledované znakom konca riadku. Potom načíta postupnosť N celých čísel, každé nasledované znakom konca riadku. Program určí, či načítaná postupnosť čísel je správna. Postupnosť je správna, ak:

- prvé číslo je z rozsahu $\langle 0, 10 \rangle$
- pre každé i -te číslo ($i > 1$) platí, že nie je väčšie ako dvojnásobok predchádzajúceho $(i-1)$ -ho čísla, ani menšie ako polovica predchádzajúceho $(i-1)$ -ho čísla.

Ak je postupnosť správna, vypíše program správu `Postupnost je spravna` a odriadkuje, inak vypíše `Postupnost nie je spravna` a odriadkuje.

Ukážka vstupu:

```
3
5
7
9
```

Výstup pre ukážkový vstup:

```
Postupnost je spravna
```

Úloha 1.2: Napíšte program, ktorý načíta číslo N . Ak je $N < 1$ alebo $N > 15$, program vypíše chybovú hlášku `Cislo nie je z daneho intervalu` a skončí. V opačnom prípade program vypíše N riadkov, kde každý bude obsahovať číslo riadku, dvojbodku, medzeru a čísla oddelené medzerou. V prvom riadku budú za dvojbodkou vypísané čísla od N po 1. V každom nasledovnom riadku bude vždy o jedno číslo menej. Všetky čísla vypisujte na 2 miesta (viď dokumentácia funkcie `printf` <http://www.cplusplus.com/reference/cstdio/printf/>). Výsledok zapíšte aj do textového súboru s názvom „cvicenie1p2.txt“.

Ukážka vstupu:

```
4
```

Výstup pre ukážkový vstup:

```
1:  4  3  2  1
2:  3  2  1
3:  2  1
4:  1
```

Úloha 1.3: Napíšte program, ktorý bude vypisovať tabuľku mocnín čísla x od 1 do n , pričom výpočet aj výpis bude realizovať funkcia `void mocnina(double x, int n)`. Program načíta zo vstupu 2 čísla oddelené jednou medzerou a ukončené znakom konca riadku. Prvé z čísel je reálne číslo a predstavuje základ mocniny (mocnenec) a druhé číslo je celé, pričom predstavuje maximálny exponent (mocniteľ) n . Výstupom je n riadkov, pričom i -ty riadok má formát $x^i = y$, kde x je mocnenec, i je mocniteľ a y je výsledok umocnenia. Mocnenec aj výsledok vypisujte na 2 desatinné miesta. Každý riadok je ukončený znakom konca riadku.

Ukážka vstupu:

3.0 4

Výstup pre ukážkový vstup:

3.00¹ = 3.00

3.00² = 9.00

3.00³ = 27.00

3.00⁴ = 81.00

Úloha 1.4: Napíšte program, na zisťovanie reverzného čísla. Program načíta zo vstupu číslo x ukončené znakom konca riadku. V programe použite funkciu `long reverzne_cislo(long x)` ktorá vráti reverzné číslo k číslu x . Výstupom programu bude vrátené reverzné číslo.

Program rozšírte tak, že bude načítavať všetky čísla zo vstupu. Počet čísel na vstupe nie je známy pred spustením programu. Využite návratovú hodnotu funkcie `scanf`. Ku každému načítanému číslu vytvorí reverzné číslo a navyše zistí, či načítané číslo je palindróm a vypíše správu: Cislo X je palindrom, alebo Cislo X nie je palindrom, kde X je zisťované číslo. Správa je nasledovaná znakom konca riadku.

Ukážka vstupu:

12345

12321

Výstup pre ukážkový vstup:

54321

Cislo 12345 nie je palindrom

12321

Cislo 12321 je palindrom

Úloha 1.5: Napíšte program, ktorý bude čítať znaky zo súboru text.txt pokiaľ nenačíta znak '*'. Ak načíta znak 'x' alebo 'X' vypíše Precital som X, ak znak 'y' alebo 'Y' vypíše Precital som Y, ak načíta znaky '#', '\$' alebo '&' vypíše Precital som riadiaci znak a ak načíta znak '*' vypíše Koniec a skončí čítanie súboru. Po prečítaní súboru vypíše správu Pocet precitanych medzier: nasledovanú medzerou a počtom prečítaných medzier. Každá správa je nasledovaná koncom riadku.

Ukážka súboru text.txt:

\$ abc 5 xyz #

& Q *# abf

Ukážkový výstup pre súbor text.txt:

Precital som riadiaci znak

Precital som X

Precital som Y

Precital som riadiaci znak

Precital som riadiaci znak

Koniec

Pocet precitanych medzier: 6