

P B 0 7 1 Z Á P O Č T O V Ý P Ř Í K L A D

W O R D C O U N T

Napište program `wc` (*word count*, nemá nic společného se zařízením toalety), který bude pracovat jako zjednodušený UNIXový nástroj `wc(1)`.

```
./wc [OPTION] [FILE...]
```

Program bude očekávat libovolný počet parametrů, které reprezentují názvy souborů. Pokud není zadán žádný soubor, program bude číst ze standardního vstupu. Taky z něj bude číst místo souboru se jménem `-`. Navíc může být prvním parametrem jeden z přepínačů `-s`, `-l`, `-w` nebo `-t`.

Formát vstupních souborů

Každý vstupní soubor je **textový**. Počet řádků ani jejich délky nejsou omezeny.

Formát výstupu

Program pro každý vstupní soubor zjistí počet bytů, počet řádků a počet slov. Pro zjednodušení považujte za slovo každou neprázdnou sekvenci znaků, která neobsahuje žádné bílé znaky (*whitespaces*).

Pro každý soubor program vypíše řádek tvaru

```
VELIKOST ŘÁDKY SLOVA NÁZEV
```

Soubory budou seřazené podle přepínače:

žádný	vzestupně podle názvu
<code>-s</code>	sestupně podle velikosti
<code>-l</code>	sestupně podle počtu řádků
<code>-w</code>	sestupně podle počtu slov

Pokud je pro některé soubory hodnota řadícího kritéria stejná, seřaďte tyto soubory vzestupně podle názvu. Pokud je zadán přepínač `-t`, místo seřazeného výstupu vypíšte součty odpovídajících hodnot pro všechny soubory.

Ukázka

Pokud program spustíme nad soubory s těmito vlastnostmi (maximální hodnoty jsou zvýrazněné):

název souboru	velikost	řádky	slova
<code>words.txt</code>	338	64	98
<code>lines.txt</code>	217	79	50
<code>bytes.txt</code>	579	14	40

pak na standardní výstup vypíše

bez přepínače:	s přepínačem <code>-w</code> :	s přepínačem <code>-t</code> :
579 14 40 <code>bytes.txt</code>	338 64 98 <code>words.txt</code>	1134 157 188 <code>total</code>
217 79 50 <code>lines.txt</code>	217 79 50 <code>lines.txt</code>	
338 64 98 <code>words.txt</code>	579 14 40 <code>bytes.txt</code>	

Požadavky a poznámky

- Pro reprezentaci informací o souboru si definujte vlastní **strukturu**.
- Na uložení záznamů použijte nějakou **dynamickou** datovou strukturu.
- Pokud selže otevření některého vstupního souboru, program vypíše na **standardní chybový výstup** nějakou smysluplnou zprávu a bude pokračovat zpracováním dalších souborů.
- Selhání ostatních standardních funkcí (např. pro alokaci paměti) ošetřete způsobem známým z domácích úkolů a cvičení.
- Program musí před ukončením uvolnit všechny alokované zdroje.