

Úloha 1: Počítání slov

Napište program, který dostane jako argument příkazové řádky jeden parametr: jméno vstupního souboru. Spočítejte kolik slov se nachází ve vstupním souboru, a celkový počet vypíše na standardní výstup.

Pokud program obdrží špatné množství argumentů, program vypíše na standardní výstup řetězec "Argument Error". V případě, že vstupní soubor nejde otevřít (vstupní soubor neexistuje, špatné znaky v názvu, nedostatečná práva, apod.), nebo nastane chyba při čtení, vypíše program řetězec "File Error".

Při zpracování vstupu a výpisu četností je potřeba dodržet následující pravidla:

- Zalomení řádku, tabulátor a mezera (tzn. '\n', '\t' a ' ') považujeme za bílé znaky, vše ostatní za znaky tisknutelné. Pro jednoduchost máte zaručeno, že znak '\r' v textu není.
- Za slovo považujeme libovolnou sekvenci nebílých znaků, která je z obou stran oddělena alespoň jedním bílým znakem nebo začátkem/koncem souboru.
- Standardní výstup se bude skládat z jediného řádku, na kterém bude napsáno jedno celé číslo určující počet slov ve vstupním souboru. Můžete předpokládat, že slov ve vstupním souboru je méně než 2 miliardy [2 000 000 000]).

Příklad 1:

```
$>program.exe plain.txt
```

Vstupní soubor [plain.txt](#)

If a train station is where the train stops, what is a work station?

std. výstup

```
14
```

Příklad 2:

```
$>program.exe
```

std. výstup

```
Argument Error
```

Úloha 2: Četnosti slov

Napište program, který dostane jako argument příkazové řádky jeden parametr: jméno vstupního souboru. Pro každé unikátní slovo, které se vyskytuje v textu vstupního souboru spočítejte a na standardní výstup vypíše jeho četnost.

Pokud program obdrží špatné množství argumentů, program vypíše na standardní výstup řetězec "Argument Error". V případě, že vstupní soubor nejde otevřít (vstupní soubor neexistuje, špatné znaky v názvu, nedostatečná práva, apod.), nebo nastane chyba při čtení, vypíše program řetězec "File Error".

Při zpracování vstupu a výpisu četností je potřeba dodržet následující pravidla:

- Zalomení řádku, tabulátor a mezera (tzn. '\n', '\t' a ' ') považujeme za bílé znaky, vše ostatní za znaky tisknutelné. Pro jednoduchost máte zaručeno, že znak '\r' v textu není.
- Za slovo považujeme libovolnou sekvenci nebílých znaků, která je z obou stran oddělena alespoň jedním bílým znakem nebo začátkem/koncem souboru. Měnit smíte pouze bílé znaky, všechna slova musí zůstat přesně tak, jak jsou ve vstupním souboru.
- Pro každé unikátní slovo, které se ve vstupním souboru vyskytlo, bude na standardním výstupu uveden jeden řádek, který uvádí jeho četnost na vstupu (záleží na velikosti písmen, tedy např. slova *world* a *World* jsou každé unikátním slovem, a každé bude mít tedy spočítanu svoji vlastní četnost).
- Každý výstupní řádek s četností vypadá následujícím způsobem:
slovo: četnost
Kde `slovo` je slovo, kterého se řádek týká; `četnost` je celé kladné číslo udávající jeho četnost (můžete předpokládat, že žádné slovo se ve vstupním souboru nevyskytuje více než miliardkrát [1 000 000 000]).
- Výstupní seznam četností je seříděn podle slova, kterému četnost náleží. Běžné metody .NETu, které slouží k porovnání dvou řetězců (např. `.CompareTo` na instancích řetězců), resp. je využívají, produkují správné uspořádání potřebné pro tuto úlohu.

Výše uvedená pravidla je třeba dodržet velmi důsledně, protože výstup vašeho programu bude porovnán se vzorovým výstupem znak po znaku. Pokud je n počet znaků na nejdelším řádku ve vstupním souboru, a m je součet počtu všech znaků všech unikátních slov ve vstupním souboru, tak můžete předpokládat, že máte k dispozici $O(n + m)$ paměti.

Příklad 1:

```
$>program.exe plain.txt
```

Vstupní soubor [plain.txt](#)

```
If a train station is where the train stops, what is a work station?
```

std. výstup (viz [counts.txt](#))

```
a: 2
If: 1
is: 2
station: 1
station?: 1
stops, : 1
the: 1
train: 2
what: 1
where: 1
work: 1
```

Příklad 2:

```
$>program.exe
```

std. výstup

```
Argument Error
```