

Programozás Alapjai Projektfeladatok

Szoftverfejlesztés Tanszék

2013, ősz

Általános információk

A kötelező programot C nyelven kell megírni, és a *Bíró* webes felületén keresztül lehet benyújtani (<https://biro.inf.u-szeged.hu>). Egy C program kiterjesztése `c`. A *Bíró* a fájl nevében található első pont utáni részt tekinti kiterjesztésnek.

Kiértékelés

A kötelező programot a *Bíró* fogja kiértékelni. Feltöltés után a *Bíró* programot a `gcc` fordítóval `-static -O2` paraméterezéssel lefordítja, majd a programot 20 különböző tesztesetre futtatja. Minden helyes teszteset 1 pontot ér. A teszteset akkor helyes, ha a program futása nem tartott tovább 5 másodpercnél és a futása hiba nélkül (0 hibakóddal) fejeződött be és az adott inputhoz tartozó kimenet **minden egyes karaktere** megegyezik az előre eltárolt referencia megoldással. A programot 10 alkalommal lehet benyújtani, a megadott határidőig.

A *Bíró* által a `riport.txt`-ben visszaadott lehetséges hibakódok:

Futási hiba: 6	Memória- vagy időkorlát túllépés.
Futási hiba: 8	Lebegőpontos hiba, például nullával való osztás.
Futási hiba: 11	Memória-hozzáférési probléma, pl. tömb-túlindekselés, null pointer használat.

Minden programra vonatkozó követelmények

A program bemenő adatait a `be.txt` nevű fájlból kell beolvasni, az eredményt pedig a `ki.txt` nevű fájlba kell írni akkor is, ha ez nincs külön megemlítve a feladat leírásában. A `be.txt` állomány csak olvasásra, a `ki.txt` állomány pedig csak írásra nyitható meg, más megnyitási mód esetén a *Bíró* nem engedélyezi a hozzáférést. Más fájl megnyitását a *Bíró* szintén nem engedélyezi.

A program bemenet/kimenet leírásokban a „sor” egy olyan karaktersorozatot jelöl, amelyben pontosan egy sorvége jel (`'\n'`) található, és az az utolsó karakter. Tehát minden sort sorvége jel zár! Elképzelhető olyan output, amelyben nincs sorvége jel, de akkor a feladat kiírásának egyértelműen jelznie kell, hogy a sorvége jel hiányzik!

A hibakód nélküli befejezést a `main` függvény végén végrehajtott `return 0;` utasítás biztosíthatja.

A feladatkiírással kapcsolatos megjegyzések

Igyekeztünk a megoldandó feladat leírását pontossá, a feladatot egyértelművé tenni. Ha ennek ellenére bármilyen pontatlanságot, kétértelműséget vagy hibát észlel a kiírásban, illetve bármilyen egyéb okból kérdése van a feladattal kapcsolatban, azt jelezze a gyakorlatvezetőjénél. A gyakorlatvezető a leadás előtti utolsó héten már nem válaszol a konkrétan a feladattal kapcsolatos kérdésekre.

[3304] PreFix-PostFix converter

A feladat prefix-postfix, és postfix-prefix, illetve infix-prefix, infix-postfix konverter megírása. Egy formula infix alakja: $a + b$; a formula prefix alakja: $+ab$; a formula postfix alakja: $ab+$; ahol a és b további adott alakú formulák lehetnek. A feladat megoldása során nem kell ellenőrizni, hogy a bemenő adat megfelelő formátumú. Azaz ha azt akarom, hogy a prefixből postfix legyen, akkor feltételezhető, hogy a prefix alakja helyes!

Bemenet

A feladat inputja a **be.txt** fájl, melynek formája a következő. A fájl első sora tartalmaz két számot szóközzel elválasztva. Az első szám azt jelenti, hogy miben adom meg a kifejezés formátumát. 0 jelentse az infix alakot, 1 jelentse a prefix alakot, 2 jelentse a postfix alakot. A második szám jelezze, hogy melyik konvertálást akarom. 1 jelenti a prefix, 2 jelenti az postfix konverziót. A fájl második sora tartalmazza a kifejezést. A kifejezésben szereplhetnek 0 és 9 közötti pozitív egész számok és változók (angol ábécé kisbetűi). A kifejezésben a következő műveletek szerepelhetnek: $+$, $-$, $*$, $/$. A bemeneti fájl végét újsor zárja. A kifejezések nem tartalmazznak zárójeleket. A kifejezés maximum 100 karakter hosszúságú lehet. Infix alak esetén a műveletek között nincs precedencia, mindegyik azonos prioritásúnak számít és bal-asszociatívak.

Kimenet

A kimenet a konvertált kifejezés. Ha az első sor nem felel meg a követelménynek, akkor a „HIBA” szó. A kimeneti fájl neve **ki.txt**. A kimeneti fájl újsort zárja.

Példák

1. példa

Input

```
2 1
fgd+*
```

Output

```
*f+gd
```

2. példa

Input

```
0 1
a+b/d-4+5*3-a+b/d-4+5*3-g+4+5-3*6+4-3/4-t/5-5+2
```

Output

```
+ - / - / - + * - + - - + - / + - * - / + abd453abd453g4536434t552
```