

Programozás Alapjai 8. ZH

14. feladatsor

Szoftverfejlesztés Tanszék

2023, Ősz

Általános információk

A programot C nyelven kell megírni, és a *Bíró* webes felületén keresztül lehet benyújtani. Egy C program kiterjesztése `c`. A *Bíró* a fájl nevében található első pont utáni részt tekinti kiterjesztésnek.

Kiértékelés

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

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 jeleznie kell, hogy a sorvége jel hiányzik!

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

1. feladat: Zárójelezés (10 pont)

Írj egy programot, amely eldönti, hogy egy matematikai kifejezésben a zárójelezés helyes-e. A kifejezésben a '(', ')', '+', '-', '*', '/' műveleti jelek, kis és nagybetűk, illetve számok szerepelhetnek. A programnak meg kell mondania azt a legkisebb pozíciót, ahol a kifejezésben először előfordul egy zárójelezési hiba. Ha a hiba egy hiányzó bezáró zárójel, akkor a hiba a sztring végén van. Ha a kifejezésben nem fordul elő hiba, akkor az eredmény a -1 érték. A kifejezés esetleges egyéb hibáival (pl. műveleti jel elmaradása) ne foglalkozzunk!

Write a program that decides whether the bracketing in a mathematical expression is correct. The expression can contain the brackets '(', ')', the operation symbols '+', '-', '*', '/', lower and upper case letters, and numbers. The program shall tell the smallest position where a bracketing error occurs for the first time in the expression. If the error is a missing closing parenthesis, the error is at the end of the string. If no error occurs in the expression, the result is -1 . Ignore any other errors in the expression (e.g. missing operation sign).

Bemenet

A be.txt fájl egyetlen sorában az adott matematikai kifejezés szerepel. A kifejezés maximum 100 karakter hosszú lehet.

Kimenet

A ki.txt fájl egyetlen sort kell tartalmazzon, amelyben az eredmény található.

Példák

1. példa

Input

Output

2. példa

Input

Output