

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.

A feladat bináris számok (kettédes törtek) átváltása decimális számokká (tizedes törttekké), és viszont. Tizedes számrendszerből tizedes törtet 2-es számrendszerbe átváltva szintén törtet kapunk (kettédes), és fordítva is. Bemenetként adott a számrendszer alapja, továbbá a szám, amit át kell váltani. A negatív számokkal nem kell foglalkozni.

A bemenet egyetlen sorában két érték szerepel. Az első érték egy egész szám, mely az átváltandó szám számrendszerének az alapja (2 vagy 10). A második érték maga a szám, amit át kell váltani. A számérték lehet egész, ebben az esetben csak számjegyek szerepelnek benne. Ha törtről van szó, akkor mindenképpen van legalább egy számjegy a tizedespont előtt és után is. A két érték egymástól szóközzel van elválasztva, és a sor végén egy sortörés található.

Kimenet

A kimenet az eredményt tartalmazza a bemenethez hasonló formátumban. Vagyis elől a kimenet számrendszere, utána egy szóköz, majd az átváltott szám, és a végén egy sortörés. A tizedestörtet pontosan 15 tizedesjegy, a kettedestörtet 30 kettedesjegy pontossággal kell kiíratni tizedespontot (kettedespontot) használva. Ha a kimeneti számrendszerben a szám pontosan nem ábrázolható akkor lefelé kerekítést (csonkolást) kell alkalmazni. Egynél kisebb szám esetén a tizedespont előtt kötelező a 0.

1. példa

10 0.4

2 0.000011001100110011001100110011

2 101.01

10 5.33333333333269

Segédanyag

Andrew S. Tannenbaum: *Számítógépes architektúrák*
 Fried Ervin: *Oszthatóság és számrendszerek*