Programozás Alapjai 4. ZH

30. feladatsor

Szoftverfejlesztés Tanszék

2023, Ősz

Feladat Töltsd le a bíróról a `minta.zip` állományt, majd tömörítsd ki! A `feladat.c` fájlban megtalálod a feladatok megoldás-kezdeményeit. Bővítsd ezt az alább olvasható feladatok alapján! Lehetőség szerint ellenőrizd megoldásod, majd töltsd fel a `feladat.c` fájlt a bíróra!

Kiértékelés A bíró lefordítja a programot, majd lefuttatja azt a feladat pontszámának megfelelő számú tesztesettel. Egy teszteset egy bemenet-kimenet pár, amely a megfelelő feladathoz készült. A teszteset akkor helyes, ha az adott bemenethez tartozó kimenet **minden egyes karaktere** megegyezik az előre eltárolt referencia kimenettel. *További feltételek: a program futása nem tarthat tovább 5 másodpercnél, egyszerre nem fogyaszthat többet 16 MiB memóriánál és nem történhet futási hiba (pl. illetéktelen memória hozzáférés).*

Ellenőrzés Feltöltés előtt érdemes ellenőrizni a megoldásod.

1. **Fordítás** Ellenőrizd, hogy a programod lefordul-e! A bíró a `gcc -O2 -static -o feladat feladat.c` paranccsal fordít, érdemes ezt használni. A `-Wall` kapcsoló is hasznos lehet.
2. **Példa tesztesetek** Ellenőrizd, hogy a programod helyesen működik-e! A `minta.zip` tartalmaz a bíró által futtatott tesztesetek közül feladatonként egyet-egyet. Az első feladat teszteléséhez másold a programod mellé az `ex1.be` fájlt `be.txt` néven, futtasd le a programod, majd az így kapott `ki.txt` tartalmát hasonlítsd össze az `ex1.ki` fájlban található referencia kimenettel.
3. **Extra tesztesetek** Ellenőrizd a programod működését további példák segítségével! Néhány további teszteset is elérhető, de ezek csupán ellenőrzésre használhatóak, a bíró nem futtatja őket. Ezek használatához futtasd a programod a `-t` vagy `-test` kapcsolóval, például a `./feladat -test` paranccsal. Csak az első feladat teszteléséhez futtasd a programod a `./feladat -t 1` paranccsal.

1. feladat (1 pont)
Javítsd ki a `fgv_1` függvény hibáit a fordító visszajelzései alapján!
```
int fgv_1(char sz[]);
```
2. feladat (1 pont)
Javítsd ki a `fgv_2` függvény hibáit a fordító visszajelzései alapján!
```
int fgv_2(char sz[]);
```
3. feladat (1 pont)
Javítsd ki a `fgv_3` függvény hibáit a fordító visszajelzései alapján!
```
int fgv_3(char sz[]);
```
4. feladat (1 pont)
Az alábbi függvények megvalósításai hibákat tartalmaznak. Javítsd ki ezeket a hibákat.
```
int f4_read(struct f4_data*);
void f4_write(struct f4_data);
struct f4_data f4_diff(struct f4_data, struct f4_data);
int f4_main();
```
5. feladat (1 pont)
Az alábbi függvények megvalósításai hibákat tartalmaznak. Javítsd ki ezeket a hibákat.
```
int f5_main();
double f5_m(double);
double f5_r(double);
f5_t f5_g();
double f5_d(f5_t, f5_t);
double f5_a(f5_t, f5_t, f5_t);
```