

Programozás Alapjai 5. ZH

13. feladatsor

Szoftverfejlesztés Tanszék

2022, 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 tesztessel. Egy tesztet egy bemenet-kimenet pár, amely a megfelelő feladathoz készült. A tesztet 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 tesztet 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 (5 pont)

Írj egy `haromszog` nevű struktúrát, ami az `a`, `b` és `c` nevű `float` típusú mezőiben egy háromszög három oldalának hosszát tárolja. Írj egy `compare` nevű függvényt, ami két ilyen háromszöget kap paraméterül, és a nagyobb területűvel tér vissza. Amennyiben a két háromszög kerülete egyforma, akkor a függvény az első paraméterben kapott háromszöget adja vissza. A háromszög kerülete: $a + b + c$.

`haromszog compare(haromszog h1, haromszog h2);`

2. feladat (5 pont)

Egy irányítótornyhoz tartozó légtérben 30 repülési szint található. Az azonos szinten repülő egynél több gép közül mind potenciális veszélyben van. A feladat megírni azt a függvényt, ami a repülőgépek tömbben tárolt magassági szintjei alapján meghatározza, hogy hány veszélyeztetett gép van a légtérben. A függvény paraméterei a gépek magassági szintjeit $(1, \dots, 30)$ tároló tömb és annak a mérete, visszatérési értéke pedig a veszélyben lévő gépek száma, illetve -1 , ha nincs veszélyben egy gép sem.

`int repulok(int tomb[], int meret);`