Programozás Alapjai 6. ZH

14. 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 (5 pont)
Számold ki egy $s_1, s_2, s_3, \ldots$ számtani sorozat elemeinek szorzatát. A függvény megkapja a sorozat első két elemét (`elso` és `masodik`), valamint egy egész intervallum alsó (`a`) és felső (`b`) végpontját. A függvény visszatérési értéke az $s_a \cdot \ldots \cdot s_b$ szorzat. A számtani sorozat n-edik eleme: $s_n = s_1 + (n-1) \cdot d$

```
double sorozat(double elso, double masodik, int a, int b);
```

2. feladat (5 pont)
A feladat egy függvény megírása, amely paraméterül vár egy állatkertbeli ketrecekből álló nem üres tömböt, illetve annak hosszát. A függvény adja vissza a legzsúfoltabb ketrec azonosítóját. Az a ketrec a legzsúfoltabb, amelyben átlagosan egy állatra a legkisebb terület jut.
Kódold le C nyelven a függvényt! A függvény fejlécén és a struktúrán ne változtass! A függvény inputjai a paraméterek, outputja a visszatérési érték. A függvény nem végez IO műveleteket!

```
long legzsufoltabb(Ketrec ketrecek[], int db)
```