

Programozás Alapjai 6. 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)

Számold ki egy s_1, s_2, s_3, \dots mértani sorozat elemeinek összegét. A függvény megkapja a sorozat első elemét (`elso`) és a kvóciensét (`q`), valamint egy egész intervallum alsó (`a`) és felső (`b`) végpontját. A függvény visszatérési értéke az $s_a + \dots + s_b$ összeg. A mértani sorozat n -edik eleme: $s_n = s_1 \cdot q^{n-1}$

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

2. feladat (5 pont)

A feladat egy függvény megírása, amely paraméterül vár egy autókából álló tömböt, annak hosszát, illetve egy utas számot. A függvény adja vissza azon autók átlagsebességét, amelyben legalább annyian utaznak, mint a harmadik paraméterként kapott utasszám. Amennyiben nincs az autók között ilyen, akkor az eredmény legyen -1 .

Kódold le C nyelven a függvényt! A függvény fejlécén és a struktúráján ne változtass! A függvény inputjai a paraméterek, outputja a visszatérési érték. A függvény nem végez IO műveleteket!

```
float atlagsebesség(auto_t autok[], int db, int utasok);
```