

Tanácsok a prog1 beadandó megoldásához

1. Figyeljete arra, hogy az fgets függvény a sor végén levő sortörést is beolvassa a stringbe!
2. Memóriafoglalási problémák miatt ennél a feladatnál nem kivitelező a „minimális mallocolás, majd minden új adat olvasása előtt reallocolás” módszer. Emiatt segítségként megadom, hogy legfeljebb 300 ellenőrző pont és legfeljebb 100.000 megállapított szabálytalanság található tesztetenként. Továbbra is elvárom viszont, hogy használjatok dinamikus tárkezelő függvényeket, még ha ismert is az adatok száma.
3. Vegyétek figyelembe, hogy a struktúratömbök mindig cím szerint kerülnek átadásra az alprogramoknak, így a rajtuk ott végzett módosítások megmaradnak (tehát az adatok beolvashatók és módosíthatók alprogramokon keresztül).
4. Ha százalék jelet szeretnének a printf/fprintf függvények valamelyikével, akkor ezt dupla százalékjellel kell jelölni (%%), mivel az egyszeres százalékjel I/O formátumot vár.
5. Úgy vettük észre, hogy Windows alatt a qsort függvény nem mindig működik megfelelően (Linux alatt viszont minden esetben jól működik). Ha ilyen hibák előjönnek, javaslom az alábbi módon a hasonlítófüggvény megírását:

```
/* if(a->sebessegtullepes < b->sebessegtullepes)
   return 1;
   if(a->sebessegtullepes == b->sebessegtullepes && strcmp(a->rendszam, b->rendszam) > 0)
   return 1;*/
```

helyett

```
if(a->sebessegtullepes != b->sebessegtullepes)
    return b->sebessegtullepes - a->sebessegtullepes;
else
    return strcmp(a->rendszam, b->rendszam);
```

A lényeg, 0 vagy 1 érték visszaadása helyett, különbséget vagy stringek esetén magának az strcmp visszatérési értékét adjátok vissza. Ilyenkor persze figyeljetek oda, melyikből vonjátok ki melyiket, illetve melyiket írjátok előre, annak függvényében, hogy növekvőbe vagy csökkenőbe kell rendezni!

6. Vegyétek figyelembe, hogy a kiszabott bírságok kumulált összege nagy szám is lehet, ezért számításához *long long int* típust használjatok *%lld* formázással!

Kovács Tamás, s.k.

Debrecen, 2019.04.05.