

Programozás Alapjai 4. próba feladat

1. 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)

Az emberiség egy önálló működésre képes kolóniát létesített a Marson. Az építkezés és bővítés még javában folyik. Azonban néhány naponta homokviharok söpörnek végig a kolónia területén, amelyek miatt sajnos már több kint rekedt telepes életét veszlette.

Dr. Zhang, a kolónia meteorológiai szakértője feljegyezte a homokviharok előfordulásait, és azt állapította meg, hogy a helyi speciális mikroklíma következtében pontos rendszer van a viharok érkezésében: a köztük eltelt napok egy ismétlődő mintát követnek. Arról nincs adatunk, hogy a minta milyen hosszú, de tudjuk azt, hogy mostanra már legalább egyszer teljesen feljegyzésre került, valamint az is ismert, hogy az összes feljegyzett nap illeszkedik ehhez a mintához.

Sajnos elődöd és mentorod, Dr. Halliburton szintén odaveszett az előző viharban, így máától te látod el az informatikai feladatokat a kolónián. Mint a szuperszámítógép új kezelője, minden felelősség rád hárul. A feladatod elérni, hogy a szuperszámítógép meg tudja jósolni a következő vihar jöveteléig várható napok számát. Ennek sikerén emberi életek és a Mars kolonizációjának teljes jövője múlhat.

Írd meg a szuperszámítógép vihar-előrejelző függvényét, amely egy egész számokból álló tömbben kapja paraméterül a viharok között eltelt napok számát, majd az ismétlődő minta meghatározása után kiszámítja, hogy mely értékkel folytatódna a tömb. A függvény ezzel az értékkel, azaz a következő viharig várható napok számával térjen vissza. Az input tömböt a `-1` érték zárja (ez az érték már nem része a feljegyzések sorozatának). A minta mindig a tömb elején kezdődik.

Az alábbi példában 17 vihar alapján készült feljegyzéseket látjuk és a 17. után a 18. viharig hátralévő napok számát keressük. Mint látszik az első 7 vihar rendre 1, 2, 3, 1, 2, 3, 4 naponta követi egymást, majd a mintázat a 8. viharnál újratekődik. Amennyiben ezt a 7 elemű mintát megfelelően hozzáillesztjük a feljegyzéseink végéhez, akkor azt kapjuk, hogy a következő vihar 1 nap múlva várható.

1	2	3	1	2	3	4	1	2	3	1	2	3	4	1	2	3	?
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18

1	2	3	1	2	3	4	1	2	3	1	2	3	4	1	2	3	1	2	3	4
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21

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

```
int előrejelez(int feljegyzesek[]);
```