

Programozás Alapjai 8. ZH

1. feladatsor

Szoftverfejlesztés Tanszék

2021, Ősz

Általános információk

A programot C nyelven kell megírni, és a *Bíró* webes felületén keresztül lehet benyújtani. Egy C program kiterjesztése `c`. A *Bíró* a fájl nevében található első pont utáni részt tekinti kiterjesztésnek.

Kiértékelés

A programot a *Bíró* fogja kiértékelni. Feltöltés után a *Bíró* a programot a `gcc` fordítóval és a `-O2 -static -o feladat feladat.c` paraméterezéssel lefordítja, majd a programot különböző tesztesetekre futtatja. Minden helyes teszteset 1 pontot ér. A teszteset akkor helyes, ha a program futása nem tartott tovább 5 másodpercnél, a futása hiba nélkül (0 hibakóddal) fejeződött be és az adott inputhoz tartozó kimenet **minden egyes karaktere** megegyezik az előre eltárolt referencia megoldással.

A *Bíró* által a `riport.txt`-ben visszaadott lehetséges hibakódok:

Futási hiba: 6	Memória- vagy időkorlát túllépés.
Futási hiba: 8	Lebegőpontos hiba, például nullával való osztás.
Futási hiba: 11	Memória-hozzáférési probléma, pl. tömb-túlinde克斯, null pointer használat.

Minden programra vonatkozó követelmények

A program bemenő adatait a `be.txt` nevű fájlból kell beolvasni, az eredményt pedig a `ki.txt` nevű fájlba kell írni akkor is, ha ez nincs külön megemlítve a feladat leírásában. A `be.txt` állomány csak olvasásra, a `ki.txt` állomány pedig csak írásra nyitható meg, más megnyitási mód esetén a *Bíró* nem engedélyezi a hozzáférést. Más fájl megnyitását a *Bíró* szintén nem engedélyezi.

A program bemenet/kimenet leírásokban a „sor” egy olyan karaktersorozatot jelöl, amelyben pontosan egy sorvége jel (`'\n'`) található, és az az utolsó karakter. Tehát minden sort sorvége jel zár! Elképzelhető olyan output, amelyben nincs sorvége jel, de akkor a feladat kiírásának egyértelműen jeleznie kell, hogy a sorvége jel hiányzik!

A hibakód nélküli befejezést a `main` függvény végén végrehajtott `return 0;` utasítás biztosíthatja.

1. feladat: Tűzijáték (10 pont)

Egy egyenes országút mentén a városok a megadott távolságokban követik egymást. Az egyik városban tűzijátékot szeretnénk rendezni. A feladat kiszámolni, hogy melyik városban célszerű megrendezni az eseményt, ha arra törekszünk, hogy az minél több városból látható legyen. Feltesszük, hogy a látványosság csak a rendező város megadott sugarú környezetében látható. Több optimális megoldás esetén az a város nyer, amelyiket az országúton balról-jobbra haladva (az indexeket $0 \dots N$ sorrendben véve) először érjük el.

Bemenet

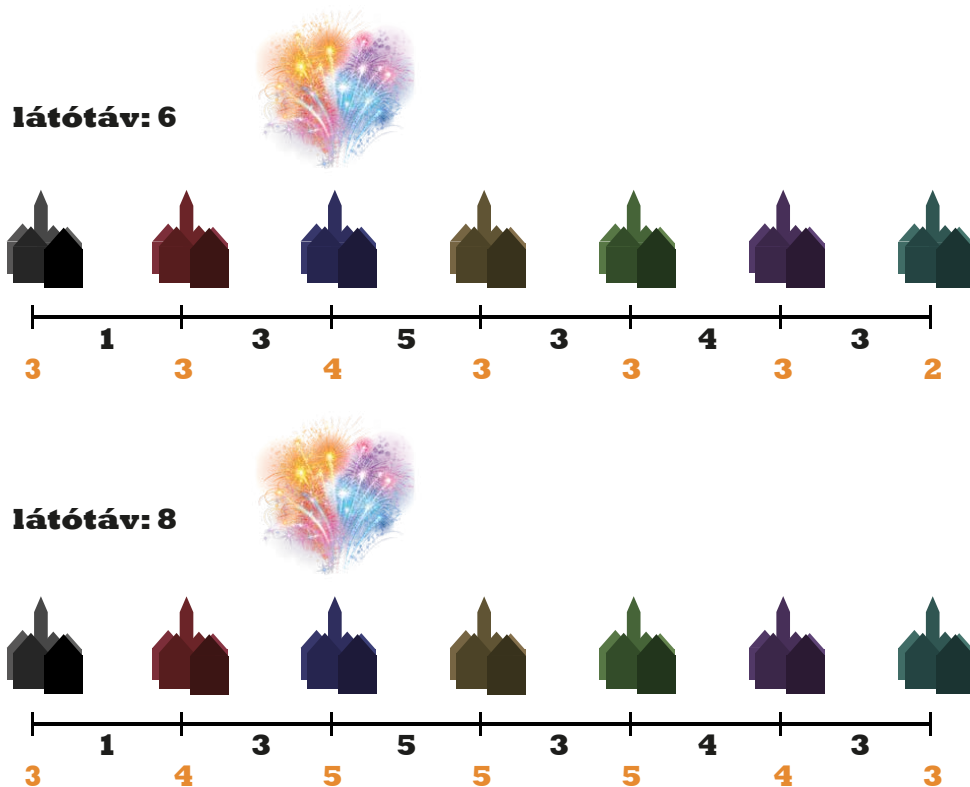
A be.txt fájl első két sorában egy-egy egész szám található. Az első a látótávolság, a második a városok száma. A látótávolság azt a távolságot jelenti, ahonnan még látható a tűzijáték. A további sorokban a városok számánál eggyel kevesebb egész szám található, amely számok a városok közötti távolságok úgy, hogy az i . szám az i . és $i + 1$. város távolságát jelenti.

Kimenet

A ki.txt fájl egyetlen sort kell tartalmazzon, amelyben a megoldásnak tekinthető város sorszáma (0-tól indexelve) található.

Példák

A következő két példában a fekete számok a városok közötti távolságot, a narancssárgák pedig azt mutatják, hogy az adott városban fellőtt tűzijátékot hány városból lehet látni (beleértve az aktuális várost is). A példákhoz tartozó be- és kimenetek alább megtekinthetők.



1. példa

Input

6
7
1
3
5
3
4
3

Output

2

2. példa

Input

8
7
1
3
5
3
4
3

Output

2
