

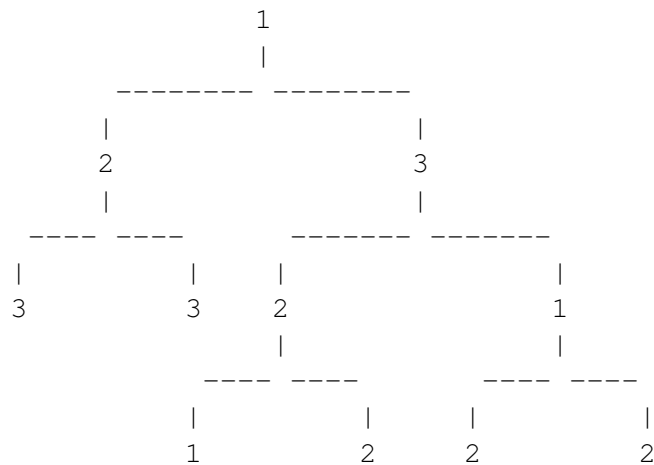
Equilibri parell/senar en arbres binaris

X33722_ca

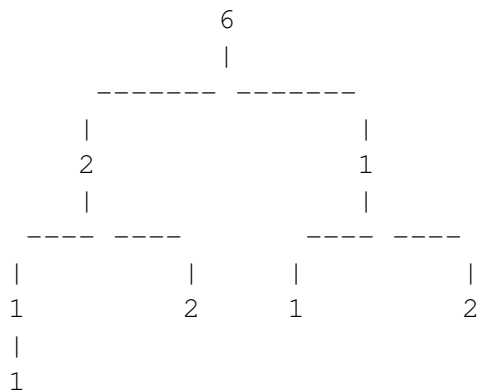
Un arbre està **equilibrat parell/senar** si compleix **dues** condicions:

1. La diferència entre el nombre de números parells i el nombre de números senars que hi ha a l'arbre és, com a màxim, 1. És a dir, que o bé hi ha tants números parells com senars a l'arbre, o bé hi ha com a màxim un número més de senars, o bé un número més de parells. Algebraicament: $\text{abs}(\text{Parells}(A) - \text{Senars}(A)) \leq 1$.
2. La condició anterior s'aplica a tots els subarbres.

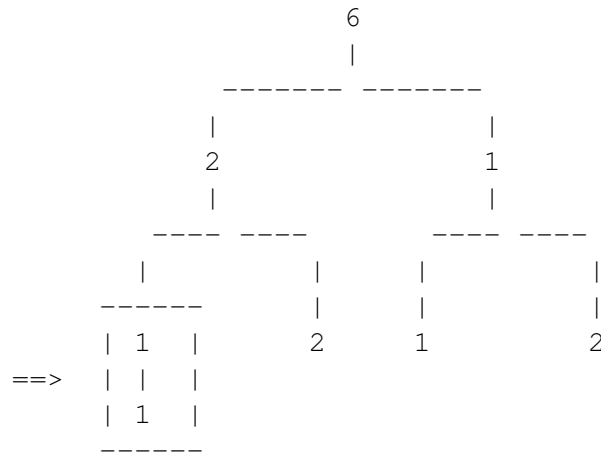
Observeu el següent exemple:



L'arbre té, en total, 6 números senars, i té 5 números parells. Per tant, compleix la primera condició. A més, com que **tots els seus subarbres** compleixen aquesta condició, l'arbre està parell/senar equilibrat. Considerem, per exemple, el següent subarbre de l'arbre anterior:



Aquest subarbre té 4 senars i 4 parells. Per tant, compleix la primera condició. Però hi ha un dels subarbres que no compleix aquesta condició, concretament, el subarbre marcat a dins del quadre, que té dos senars però cap parell:



Aquest arbre **NO** està parell/senar equilibrat.

Observació: Un arbre pot complir la condició 1, però això no implica necessàriament que tots els seus subarbres la compleixin.

Observació: Un arbre buit o que té un sol node està sempre parell/senar equilibrat.

Heu d'implementar una funció que rep un arbre d'enters, i retorna un booleà indicant si l'arbre està parell/senar equilibrat. Aquesta és la capçalera:

PRE A és un arbre binari d'enters.

POST Torna true si i només si A està parell/senar equilibrat.

```
bool equilibriSenarParell(const BinaryTree<int> &A);
```

Fixeu-vos que l'enunciat d'aquest exercici ja ofereix uns fitxers que haureu d'utilitzar per a compilar: `Makefile`, `program.cpp`, `BinaryTree.hpp`, `equilibriSenarParell.hpp`. Us falta crear el fitxer `equilibriSenarParell.cpp` amb els corresponents `includes` i implementar-hi la funció anterior. Quan pugeu la vostra solució al jutge, només cal que pugeu un `tar` construït així:

```
tar cf solution.tar equilibriSenarParell.cpp
```

Entrada

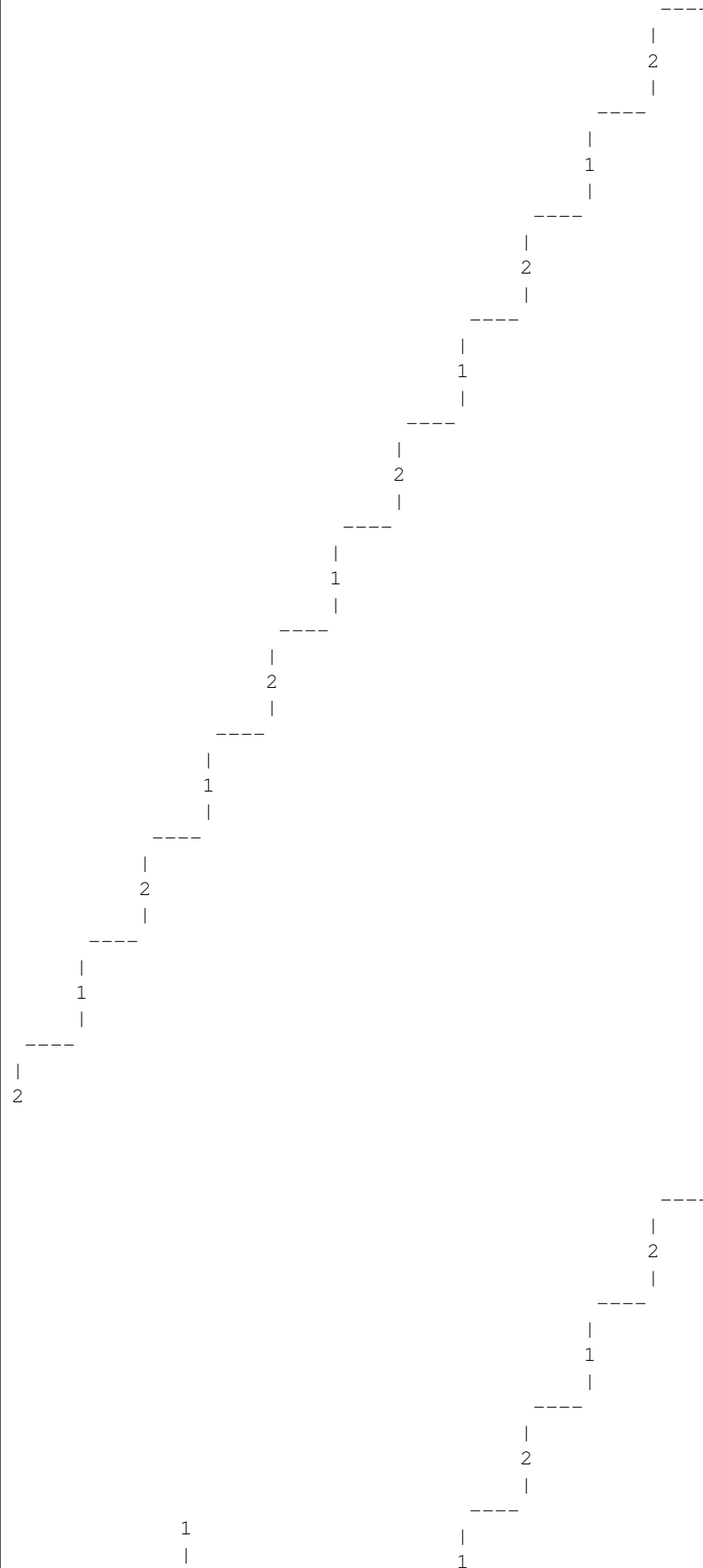
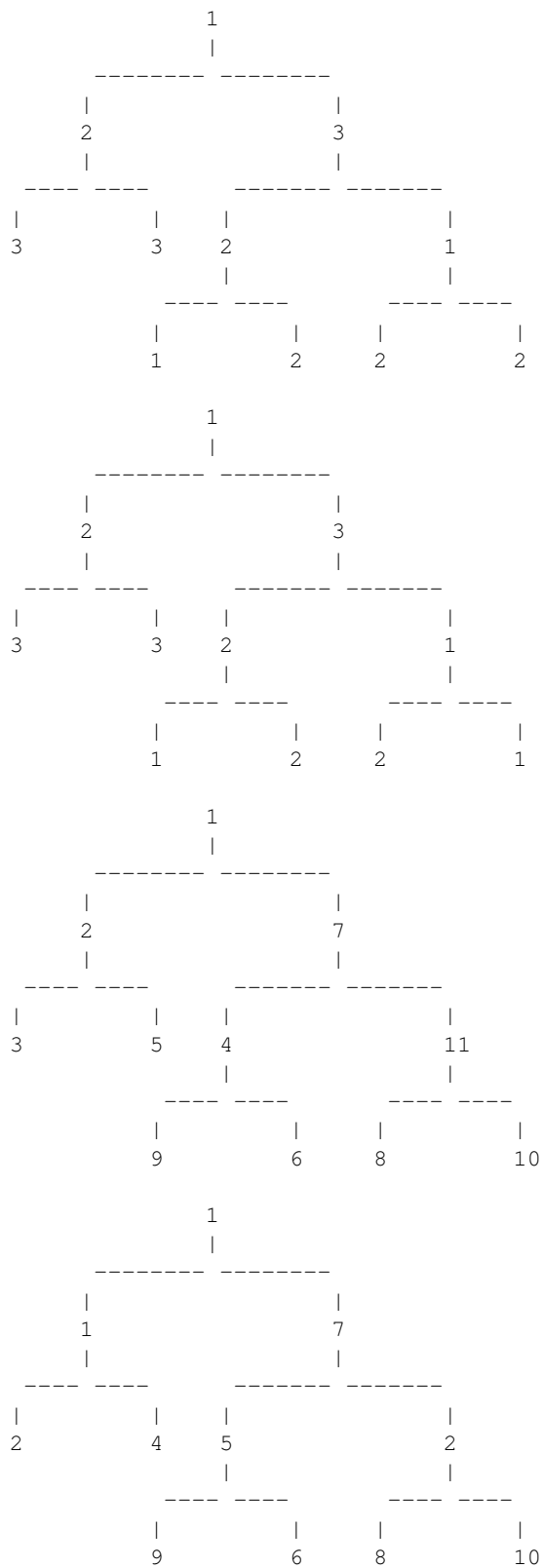
La primera línia de l'entrada descriu el format en el que es descriuen els arbres, o bé IN-LINEFORMAT o bé VISUALFORMAT. Després venen un nombre arbitrari de casos. Cada cas consisteix en una descripció d'un arbre binari d'strings que representa un arbre de directoris i fitxers. Fixeu-vos en que el programa que us oferim ja s'encarrega de llegir aquestes entrades. Només cal que implementeu la funció abans esmentada.

Sortida

Per a cada cas, la sortida conté `cert` si l'arbre està parell/senar equilibrat, `fals` altrament. Fixeu-vos en que el programa que us oferim ja s'encarrega d'escriure el resultat. Només cal que implementeu la funció abans esmentada.

Exemple d'entrada 1

VISUALFORMAT



- solució ràpida + justificació: 10 punts.

Entenem com a solució lenta una que és correcta i capaç de superar els jocs de proves públics. Entenem com a solució ràpida una que és correcta i capaç de superar els jocs de proves públics i privats. La justificació val 2 punts i consisteix en definir correctament les PRE/POST de les funcions auxiliars que afegiu i en definir correctament les hipòtesis d'inducció i funcions de fita.

Informació del problema

Autoria: PRO1

Generació: 2026-01-25T21:08:57.379Z

© *Jutge.org*, 2006–2026.

<https://jutge.org>