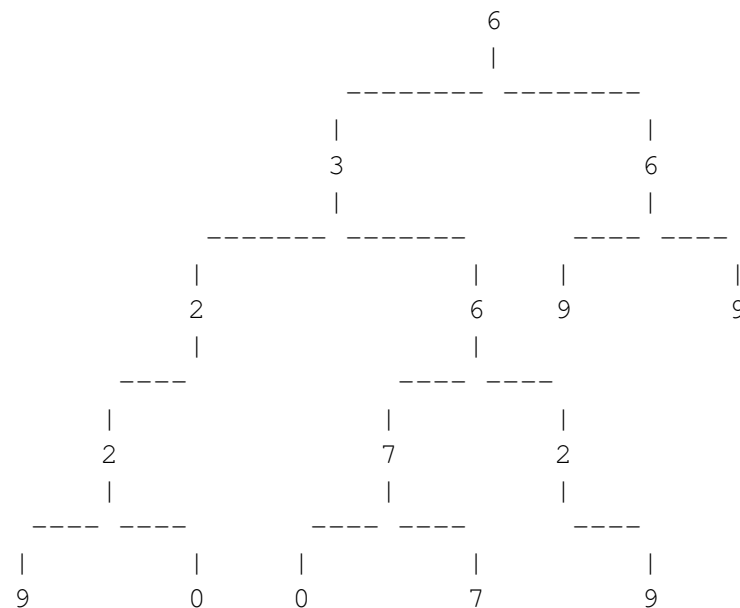

Mètode de la classe `Arbre` per a comptar el nombre de nodes amb valor estrictament major que el valor del seu node pare `X61999_ca`

En aquest exercici afegirem un nou mètode `numIncreasing` a la classe `Arbre` per a calcular el nombre de nodes que guarden un valor estrictament major que el valor guardat en el seu node pare. Fixeu-vos que l'arrel de l'arbre no compta perquè no té node pare.

Per exemple, suposeu que aquest és l'arbre representat per la variable `a` de tipus `Arbre`:



Llavors, la crida `a.numIncreasing()` ha de retornar 6.

D'entre els fitxers que s'adjunten en aquest exercici, trobareu `Arbre.hh`, a on hi ha una implementació de la classe genèrica `Arbre`. Haureu de buscar dins `Arbre.hh` les següents línies i implementar els mètodes que s'hi indiquen:

```

// Pre:
// Post: Retorna el nombre de parelles de nodes de l'arbre representat pel pa
// que no son l'arrel i que guarden un valor
// estrictament major que el valor guardat al seu node pare.
// Descomenteu les següents dues línies i implementeu el mètode:
// int numIncreasing() const{
// }
private:

// Pre:
// Post: Retorna el nombre de parelles de nodes de l'arbre representat per n
// que no son l'arrel i que guarden un valor
// estrictament major que el valor guardat al seu node pare.
// Descomenteu les següents dues línies i implementeu el mètode:
// static int numIncreasingAux(node_arbre *n ){
// }

```

Podeu suposar que el tipus genèric `T` de la classe té predefinida la operació de comparació `<`. Fins i tot, si voleu, poseu suposar que també teniu `>`, `<=`, `>=`, tot i que realment no cal. De fet, es testearà la vostra implementació amb el tipus `T=int`. Ara bé, una solució que no sigui genèrica es considerarà incorrecta i serà invalidada a posteriori, encara que superi els jocs de proves.

D'entre els fitxers que s'adjunten a l'exercici també hi ha `main.cc` (programa principal), i el podeu compilar directament, doncs fa `include` de `Arbre.hh`. Només cal que pugueu `Arbre.hh` al jutge.

Entrada

L'entrada conté un nombre arbitrari d'arbres. Cada cas consisteix en una descripció d'un arbre binari d'enters. La descripció consisteix en un recorregut en preordre dels nodes de l'arbre, amb marques on hi anirien els arbres buits. Fixeu-vos en que el programa que us oferim ja s'encarrega de llegir aquestes entrades. Només cal que implementeu els mètodes abans esmentats.

Sortida

Per a cada cas, la sortida conté la corresponent avaluació de l'arbre. Fixeu-vos en que el programa que us oferim ja s'encarrega d'escriure aquesta avaluació. Només cal que implementeu els mètodes abans esmentats.

Exemple d'entrada 1

6 # #	4 4 2 3 0 # # 9 # # 4 2 # 3 # # # # 0 2 1 4 # # # # 4
5 # #	3 7 6 # # 4 # # 8 # #
2 9 # # 0 # 6 # #	5 9 # # #
9 2 5 # # 2 # # 7 1 # # 3 4 4 # # 3 # # #	6 9 # 9 4 # # 6 # # #
2 # 1 # #	1 # 1 # 9 # 0 # #
7 6 7 2 # # # #	1 0 1 6 # # 2 # # 7 # 1 # # 7 # #
4 0 # # #	2 6 9 2 # 3 # 1 # # 5 3 # # 6 # 5 # # 5 0 7 # # # 1 8 0
8 3 9 # 6 # # 2 # 4 # # 0 7 2 # # 2 # # 0	6 1 # 8 8 # # 7 # # 9 # #
#	7 6 1 8 # # 6 8 # # 8 # # # 7 4 5 # # 2 1 # # 9 # # 9 4
4 2 9 # 8 # # 9 8 # # #	5 9 # 1 6 # # 1 9 # # 6 2 # # 8 # 9 # 4 # # 1 # 9 # #
1 8 # 1 0 # 4 # # 6 7 # # 6 # # #	4 4 6 # # 9 # # #
#	1 9 # # 3 8 3 9 3 # # 0 # # 6 7 # # 1 # # 3 4 # # 2 9 #
5 1 9 3 3 9 # # 3 # # 8 # 4 # # 8 # # 7 #	6 7 # # 7 # #
0 0 1 9 # # 8 # # 6 0 # 6 # # 7 # 9 2 #	6 3 0 8 # # 9 # # 0 9 2 # # # 6 0 # # 8 # # 1 # 5 1 # #
6 # 7 3 5 1 # # 1 # # 5 0 # # #	3 0 9 # # 5 5 6 2 2 7 # # # 8 8 6 # # 2 # # 5 0 # # 0
4 6 4 # # 2 5 0 # # 3 # 4 # # 1 2 # # 4	0 7 9 # # 9 # # 0 9 # # #
0 4 # # #	1 7 3 2 # 4 # # 3 2 # # # 4 4 # # 5 # # 9 2 9 # # 0 7 #
7 1 # 9 # # 2 # 1 # #	8 0 # # #
2 # 8 # #	9 6 8 # # 6 7 # # 0 # 4 # # 7 # #
8 3 # 4 # #	2 9 # # 8 # #
9 # 8 # #	2 9 6 0 2 7 # # 2 # # 4 # # # 6 4 1 7 # # # 0 # # 3 7 #
3 7 6 6 # # 7 # # 0 0 # # # 7 # #	5 1 6 # # 1 # # 2 # #
2 9 2 0 8 # # 5 # # 2 4 4 # # 3 # # 0	9 4 4 9 # # 9 # # 6 9 # # # 7 1 3 # # 4 # # 1 7 # # 8 #
1 7 1 # # 5 # # 7 6 4 # # 8 0 # 4 # # 8	0 8 8 5 6 # # 2 # # 6 7 # # 6 # # # 7 7 5 8 1 # # 3 # #
7 6 # # 2 # #	9 9 3 4 # # 5 0 6 1 # #
9 5 3 # # 8 4 # # 6 7 # # # 7 3 3 # # 3 7	9 6 3 1 # # 0 # #
1 0 9 2 # # 7 # # 2 # # 1 1 1 1 # # 0 # #	3 7 2 0 # # 5 4 # # # 3 6 4 4 # # 5 # # 4 1 3 # # #
1 2 5 # # 6 # # 9 # 9 0 # # 8 # #	7 8 9 # 2 8 8 # # #
1 1 4 # # 3 # # 2 # 7 # #	8 6 # 0 1 # 9 4 # #
3 3 3 2 9 # 4 # # 5 3 # # 8 # # 0 6 0 # #	3 4 8 # # 3 # # 8 3 5 # # 9 # # 2 # #
7 7 5 # 5 # # 1 # 6 # #	3 6 1 # 1 6 4 # # 0 5 # # 7 # # 9 2 # # 9 0 # # 5 # #
	2 4 # # 3 3 5 # 3 # # 5 # # #
	6 7 # # 9 # #

[illegible]

Exemple de sortida 1

#	#	#	1	#	#
0					
8	4	#	#	0	9
2			6	#	#
0			9	#	#
1			#	2	#
0	0	1	#	9	2
8			#	7	#
0			#	7	9
6			#	#	4
2	2	#	0	#	#
4			4	#	#
0			2	#	#
1	#	#	#	1	7
7			4	5	#
2			#	#	#
6			2	7	#
1			#	#	1
1			#	#	3
1			#	#	7
0			3	4	9
3			#	#	#
6					
4					
0					
5					
4					
4					
4					
6					
1					
8					
2					
1					
1					
1					
5					
6					
2					
9					
6					
2					
11					
2					
6					
1					
4					
6					
0					
3					
2					
12					
1					
8					
6					
1					
1					
5					
3					

2	1
5	9
6	2
1	1
2	5
1	1
2	2
5	1
0	16
6	3
2	

Observació

Avaluació sobre 10 punts: (Afegiu comentaris si el vostre codi no és prou clar)

- Solució lenta: 5 punts.
- solució ràpida: 10 punts.

Entenem com a solució ràpida una que és correcta, de cost lineal i capaç de superar els jocs de proves públics i privats. Entenem com a solució lenta una que no és ràpida, però és correcta i capaç de superar els jocs de proves públics.

Una solució que no sigui genèrica (per a qualsevol tipus T amb $<$, $>$, $<=$, $>=$ predefinits) serà invalidada i rebrà nota 0, encara que superi els jocs de proves.

Informació del problema

Autoria: PRO2

Generació: 2026-01-27T18:53:53.884Z

© *Jutge.org*, 2006–2026.

<https://jutge.org>