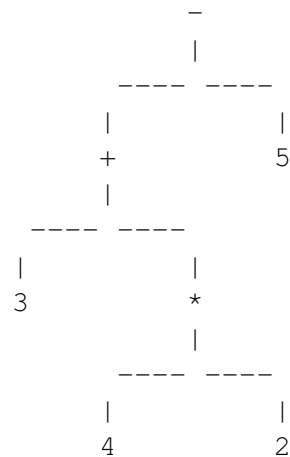


## Nombre d'expressions amb avaluació negativa

X30191\_ca

### INTRODUCCIÓ:

En aquest exercici considerarem arbres que representen expressions sobre els operadors  $+$ ,  $-$ ,  $*$ , i sobre operands naturals. Per exemple, el següent arbre representa l'expressió  $3+4*2-5$ .



### EXERCICI:

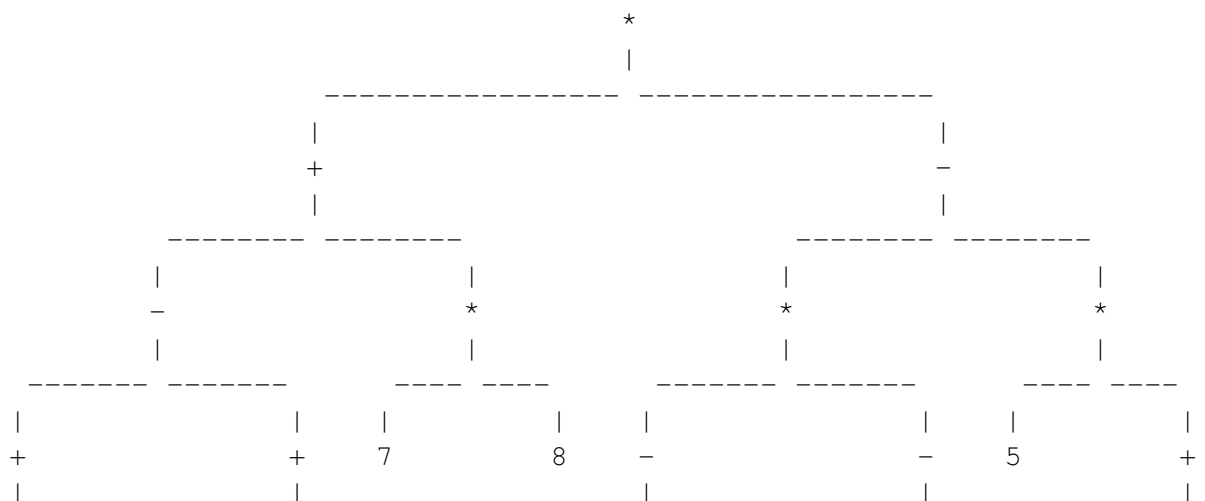
Implementeu una funció que, donat un arbre binari d'strings que representa una expressió correcta sobre naturals i operadors  $+$ ,  $-$ ,  $*$ , retorna el nombre de subexpressions tals que la seva avaluació és estrictament menor que 0. Aquesta és la capcelera:

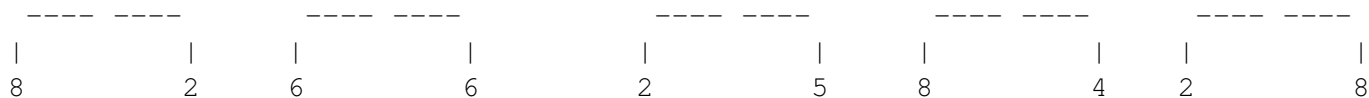
```

// Pre:  t és un arbre no buit que representa una expressió correcta
//        sobre els naturals i els operadors +,-,* .
//        Les operacions no produeixen errors d'overflow.
// Post: Retorna el nombre de subexpressions de l'expressió representada per t
//        amb avaluació estrictament menor que 0.
int numNegative(BinTree<string> t);

```

Aquí tenim un exemple de paràmetre d'entrada de la funció i la corresponent sortida:





=>

5

Fixeu-vos que l'enunciat d'aquest exercici ja ofereix uns fitxers que haureu d'utilitzar per a compilar: `main.cc`, `BinTree.hh`, `numNegative.hh`, `utils.hh`, `utils.cc`. Us falta crear el fitxer `numNegative.cc` amb els corresponents `includes` i implementar-hi la funció anterior. Valdrà la pena que utilitzeu algunes de les funcions oferides a `utils.hpp`. Només cal que pugeu `numNegative.cc` al jutge.

## 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 un arbre binari d'enters. Fixeu-vos en que el programa que us oferim ja s'encarrega de llegir aquestes entrades. Només cal que implementeu la funció abans esmentada.

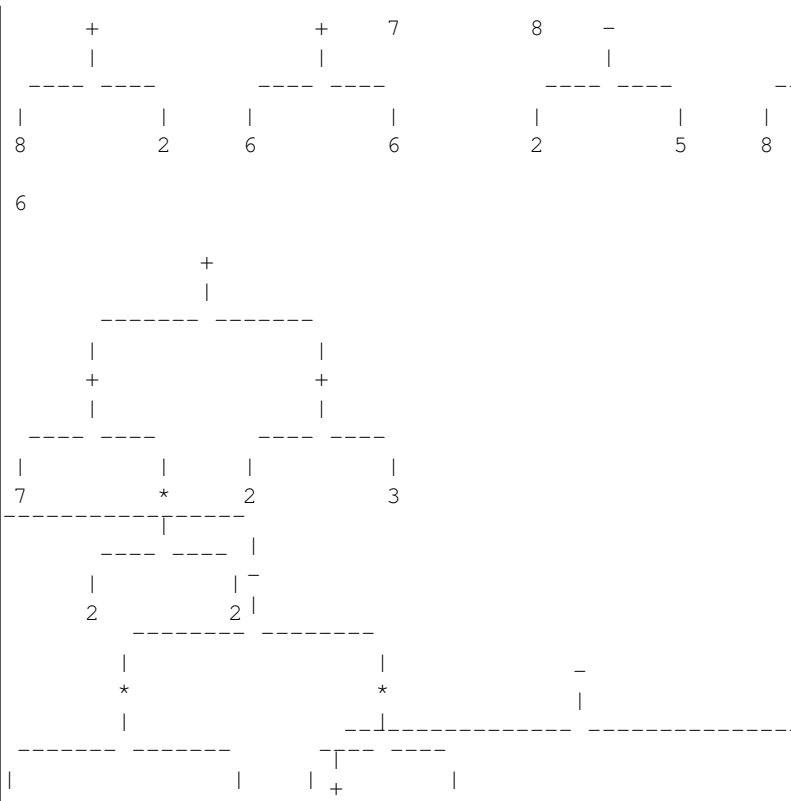
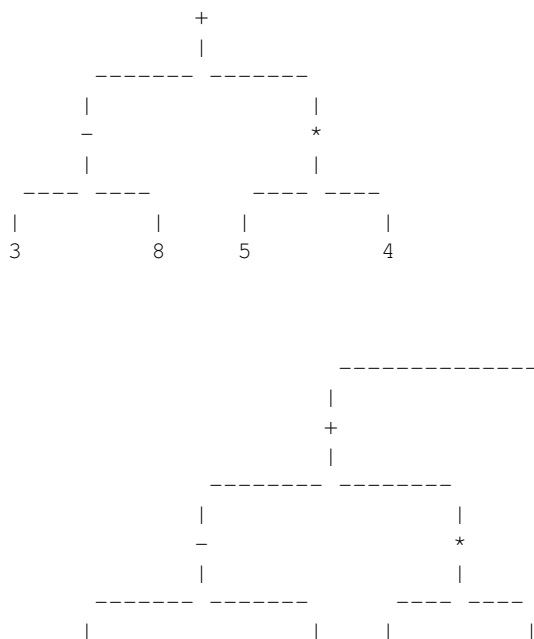
## Sortida

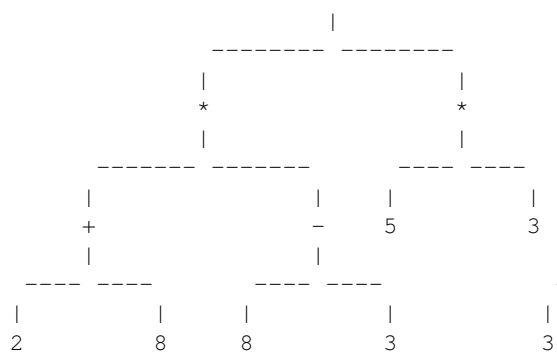
Per a cada cas, la sortida conté el corresponent nombre de subexpressions negatives. Fixeu-vos en que el programa que us oferim ja s'encarrega d'escriure aquesta sortida. Només cal que implementeu la funció abans esmentada.

### Exemple d'entrada 1

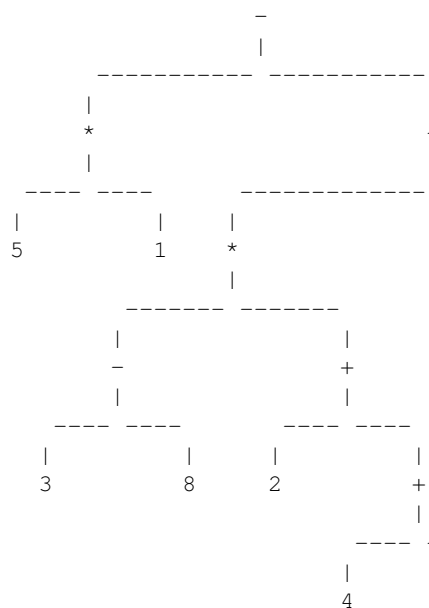
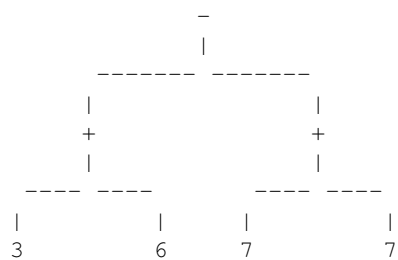
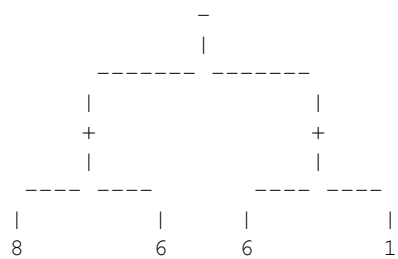
`VISUALFORMAT`

3

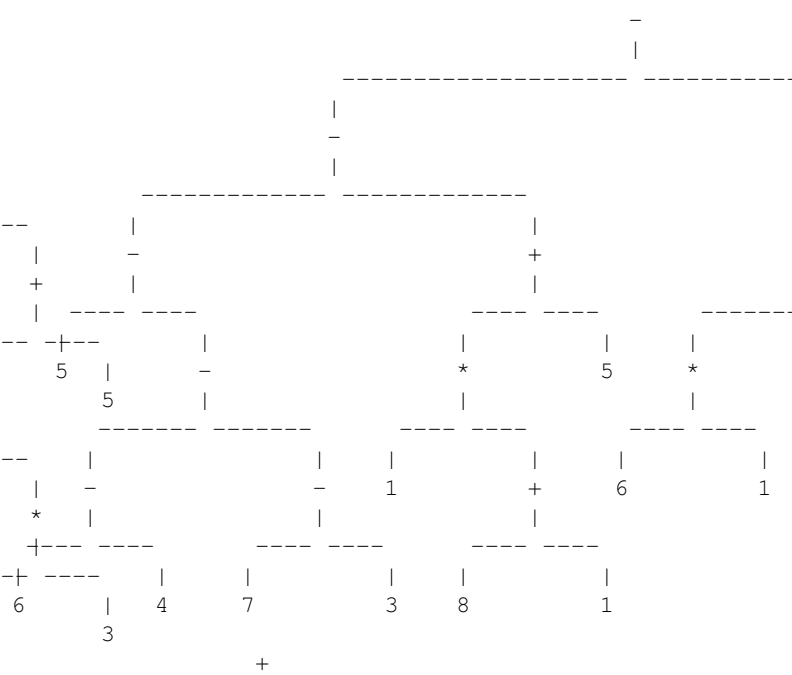
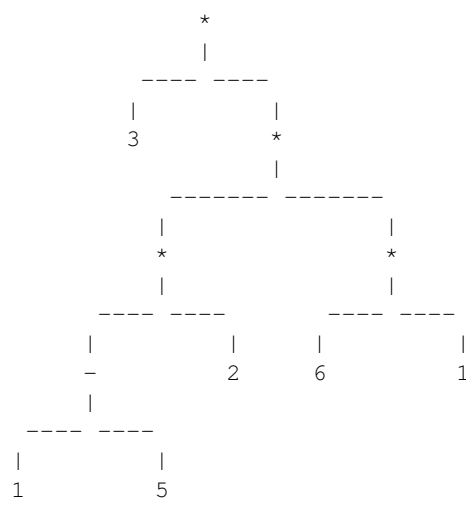
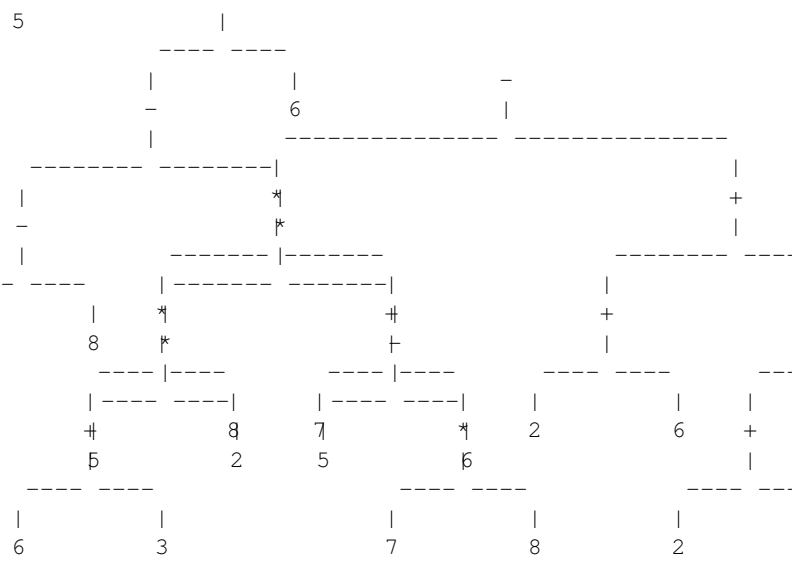


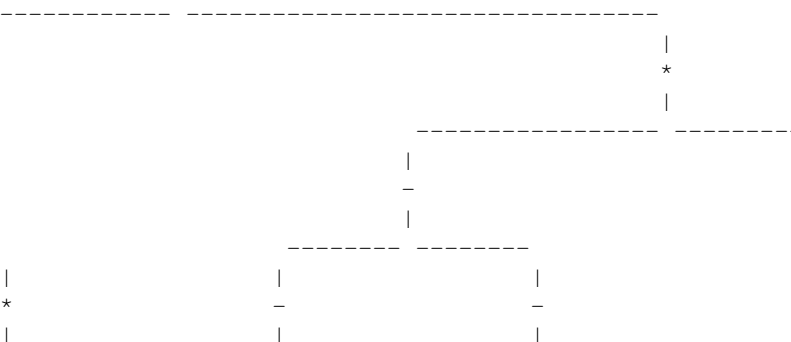
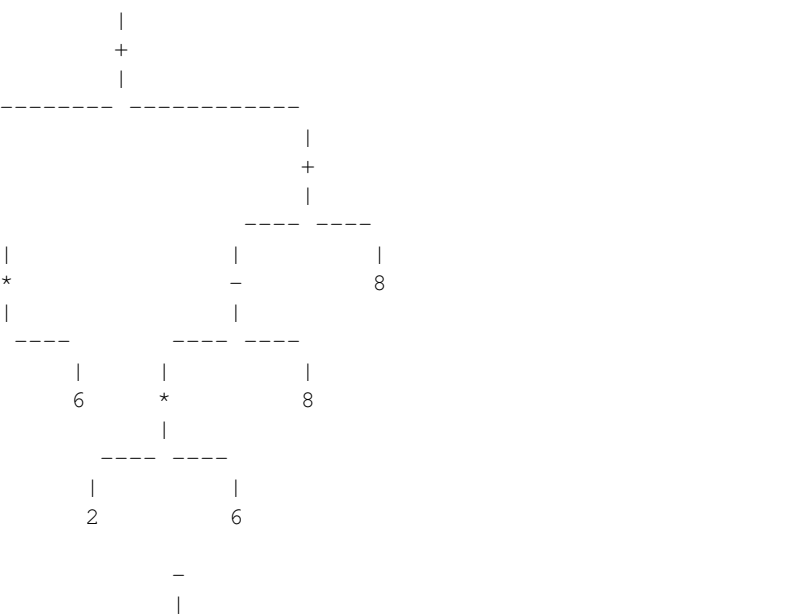
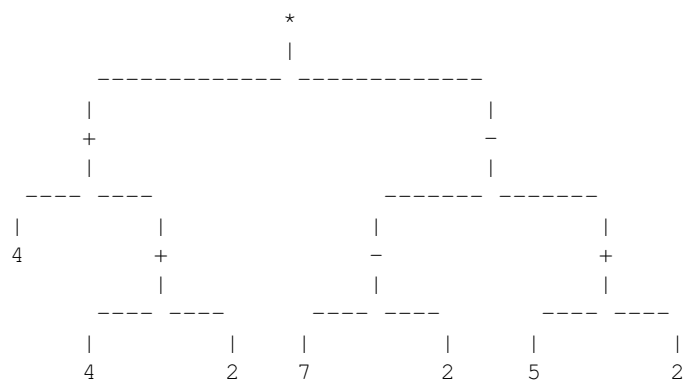
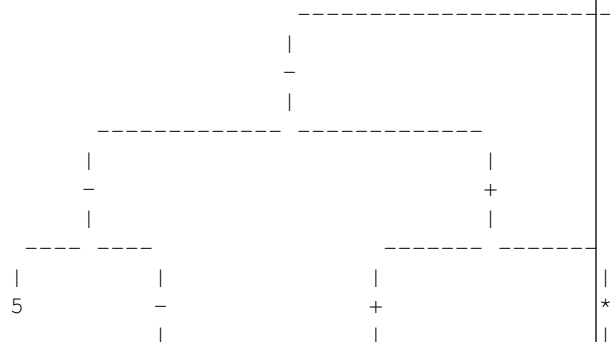
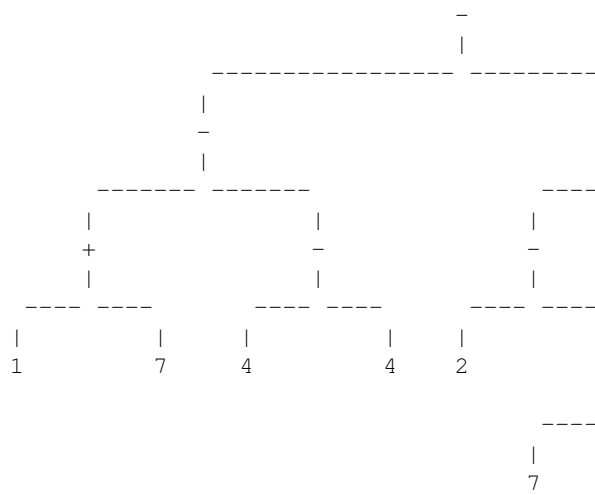
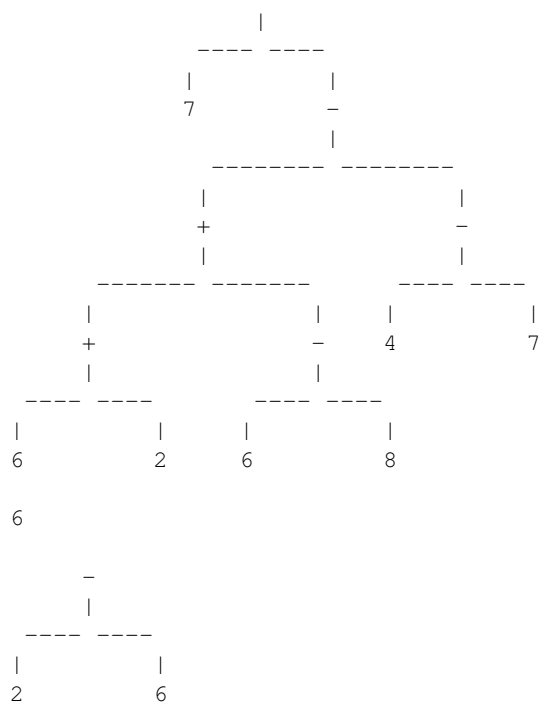


7



5





### Exemple de sortida 1

0  
1  
5  
0  
0  
3  
0  
0  
1

3  
0  
1  
4  
4  
2  
0  
1  
2  
4  
2

### Exemple d'entrada 2

INLINEFORMAT

3

+ ( - ( 3 , 8 ) , \* ( 5 , 4 ) )

\* ( + ( - ( + ( 8 , 2 ) , + ( 6 , 6 ) ) , \* ( 7 , 8 ) ) , - ( \* ( - ( 2 , 5 ) , - ( 8 , 4 ) ) , \* ( 5 , + ( 2 , 8 ) ) ) )

6

+ ( + ( 7 , \* ( 2 , 2 ) ) , + ( 2 , 3 ) )

- ( + ( \* ( + ( 2 , 8 ) , - ( 8 , 3 ) ) , \* ( 5 , 3 ) ) , \* ( - ( - ( 3 , 8 ) , \* ( 5 , 2 ) , - ( 5 , 6 ) ) ) , 6 ) )

7

- ( + ( 8 , 6 ) , + ( 6 , 1 ) )

- ( + ( 3 , 6 ) , + ( 7 , 7 ) )

- ( \* ( 5 , 1 ) , - ( \* ( - ( 3 , 8 ) , + ( 2 , + ( 4 , 2 ) ) ) , + ( \* ( 1 , \* ( 2 , 3 ) ) , 5 ) ) )

5

- ( \* ( \* ( + ( 6 , 3 ) , 8 ) , + ( 7 , \* ( 7 , 8 ) ) ) , + ( + ( 2 , 6 ) , + ( 4 , 7 ) , - ( 3 , 5 ) ) ) )

\* ( 3 , \* ( \* ( - ( 1 , 5 ) , 2 ) , \* ( 6 , 1 ) ) )

- ( - ( - ( 5 , - ( - ( 6 , 4 ) , - ( 7 , 3 ) ) ) , + ( \* ( 1 , + ( 8 , 1 ) ) , 5 ) ) , - ( + ( \* ( 6 , 1 ) , - ( 5 , - ( 4 , 2 ) ) ) , - ( 6 , 7 ) ) )

+ ( 7 , - ( + ( + ( 6 , 2 ) , - ( 6 , 8 ) ) , - ( 4 , 7 ) ) )

6

- ( 2 , 6 )

- ( - ( + ( 1 , 7 ) , - ( 4 , 4 ) ) , + ( - ( 2 , \* ( 7 , 6 ) ) , + ( - ( \* ( 2 , 6 , 8 ) , 8 ) ) )

- ( - ( - ( 5 , - ( 5 , 4 ) ) , + ( + ( 4 , 2 ) , \* ( 5 , 7 ) ) ) , \* ( - ( - ( \* ( 2 , 4 ) , \* ( 7 , 8 ) ) , - ( 2 , 2 ) ) , - ( \* ( + ( 1 , 5 ) , - ( 5 , 3 ) ) , + ( 3 , 3 ) ) ) )

\* ( + ( 4 , + ( 4 , 2 ) ) , - ( - ( 7 , 2 ) , + ( 5 , 2 ) ) )

### Exemple de sortida 2

0

1

5

0

3

0

1

3

0

1

3

1

4

2

0

1

2

### Observació

La vostra funció i subfuncions que creeu han de treballar només amb arbres. Heu de trobar una solució **RECURSIVA** del problema. Avaluació sobre 10 punts:

- 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.

### Informació del problema

Autoria: PRO2

Generació: 2026-01-25T15:00:53.856Z

© Jutge.org, 2006–2026.

<https://jutge.org>