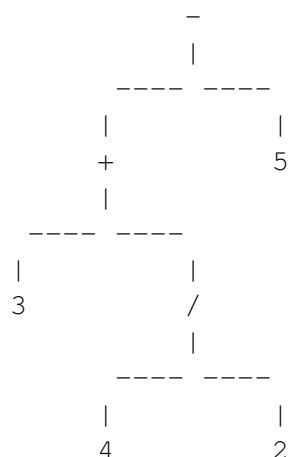


**Avaluar expressions amb divisió (PRO2)****X96833\_ca****INTRODUCCIÓ:**

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



Alhora d'avaluar una divisió, interpretem la divisió entera que ens ofereix C++. Noteu que, en particular,  $(-5)/2 = -2$ , contradient la definició que trobem habitualment en llibres de matemàtiques.

Noteu també que la divisió per 0 no està definida, i això ho haurem de tenir en compte en resoldre l'exercici.

**EXERCICI:**

Implementeu una funció que, donat un arbre binari d'strings que representa una expressió correcta sobre naturals i operadors binaris  $+$ ,  $-$ ,  $*$ ,  $/$ , retorna la seva avaluació i un indicador de si s'ha produït un error de divisió per 0, tot mitjançant paràmetres per referència. Aquesta és la capçalera:

```
// Pre:  t és un arbre no buit que representa una expressió correcta
//        sobre els naturals i els operadors binaris +, -, *, /.
//        Les operacions no produeixen errors d'overflow,
//        però poden produir error de divisió per 0.
// Post:  Si l'avaluació de l'expressió representada per t no produeix errors de
//        llavors 'result' val l'avaluació d'aquesta expressió i 'error' val 'fa
//        En cas contrari, 'error' val 'true', i el valor de 'result' és irrelle
void evaluate(BinTree<string> t, int &result, bool &error);
```

Aquí tenim un exemple de paràmetre d'entrada de la funció i els corresponents valors de sortida:

```
evaluate(/(+ (1,2), -(5,2)), result, error) produeix result=1, error=false
```

Fixeu-vos que l'enunciat d'aquest exercici ja ofereix uns fitxers que haureu d'utilitzar per a compilar: `main.cc`, `BinTree.hh`, `evaluate.hh`, `utils.hh`, `utils.cc`. Us falta

crear el fitxer `evaluate.cc` amb els corresponents `includes` i implementar-hi la funció anterior. Valdrà la pena que utilitzeu algunes de les funcions oferides a `utils.hh`. Haureu de compilar `main.cc`, `utils.cc` i `evaluate.cc` combinadament. Només cal que pugueu `evaluate.cc` al jutge.

## Entrada

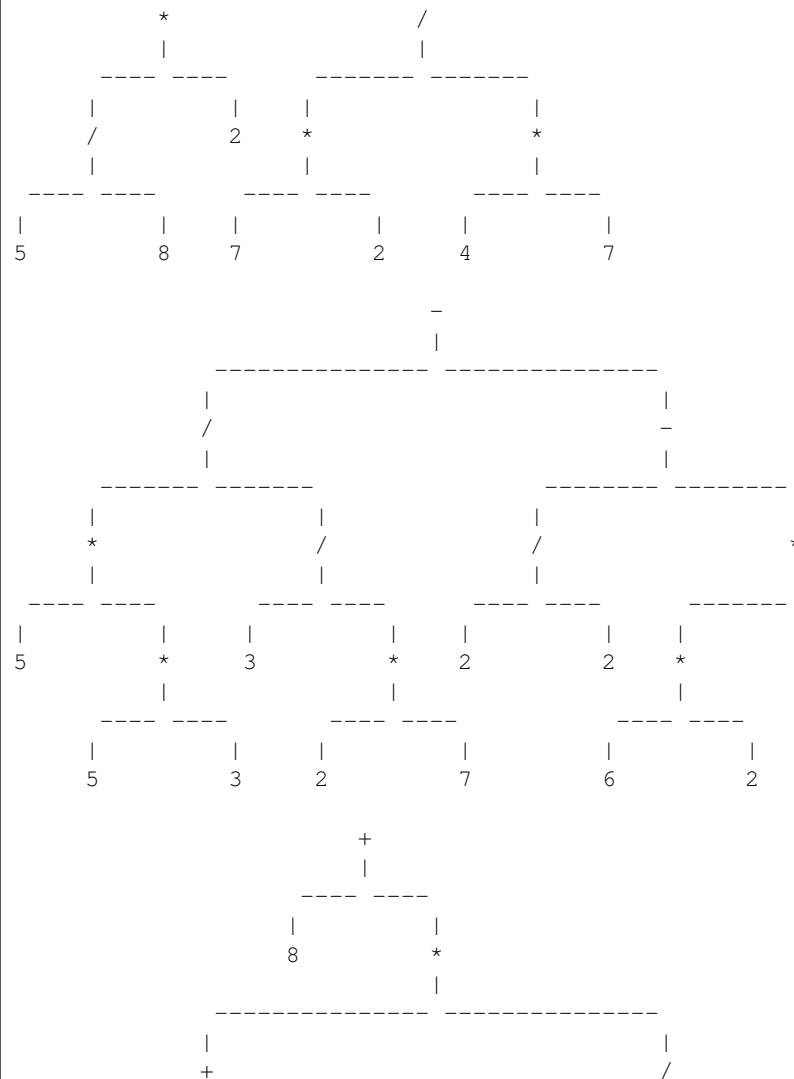
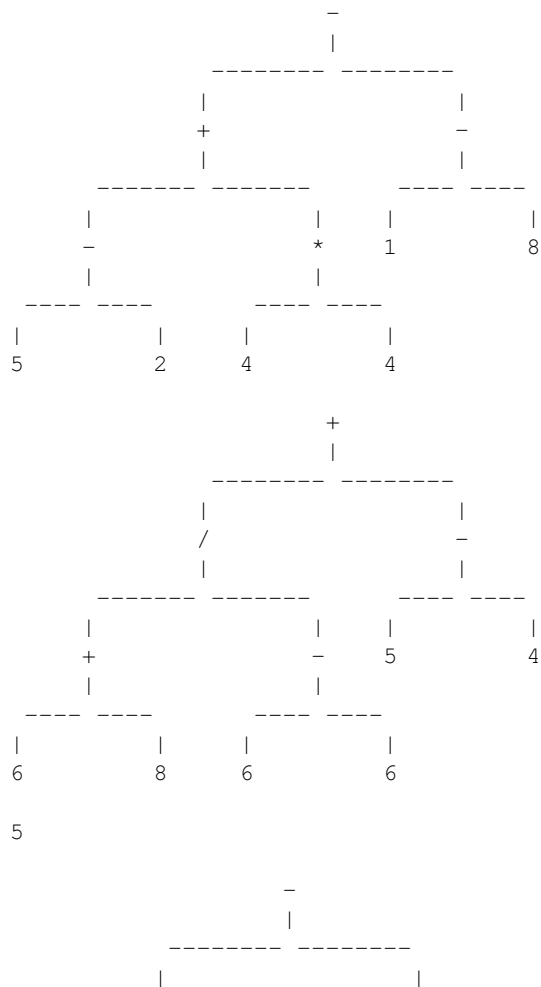
La primera línia de l'entrada descriu el format en el que es descriuen els arbres, o bé `INLINE-FORMAT` o bé `VISUALFORMAT`. Després venen un nombre arbitrari de casos. Cada cas consisteix en una descripció d'un arbre binari que representa una expressió. Fixeu-vos en que el programa que us oferim ja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

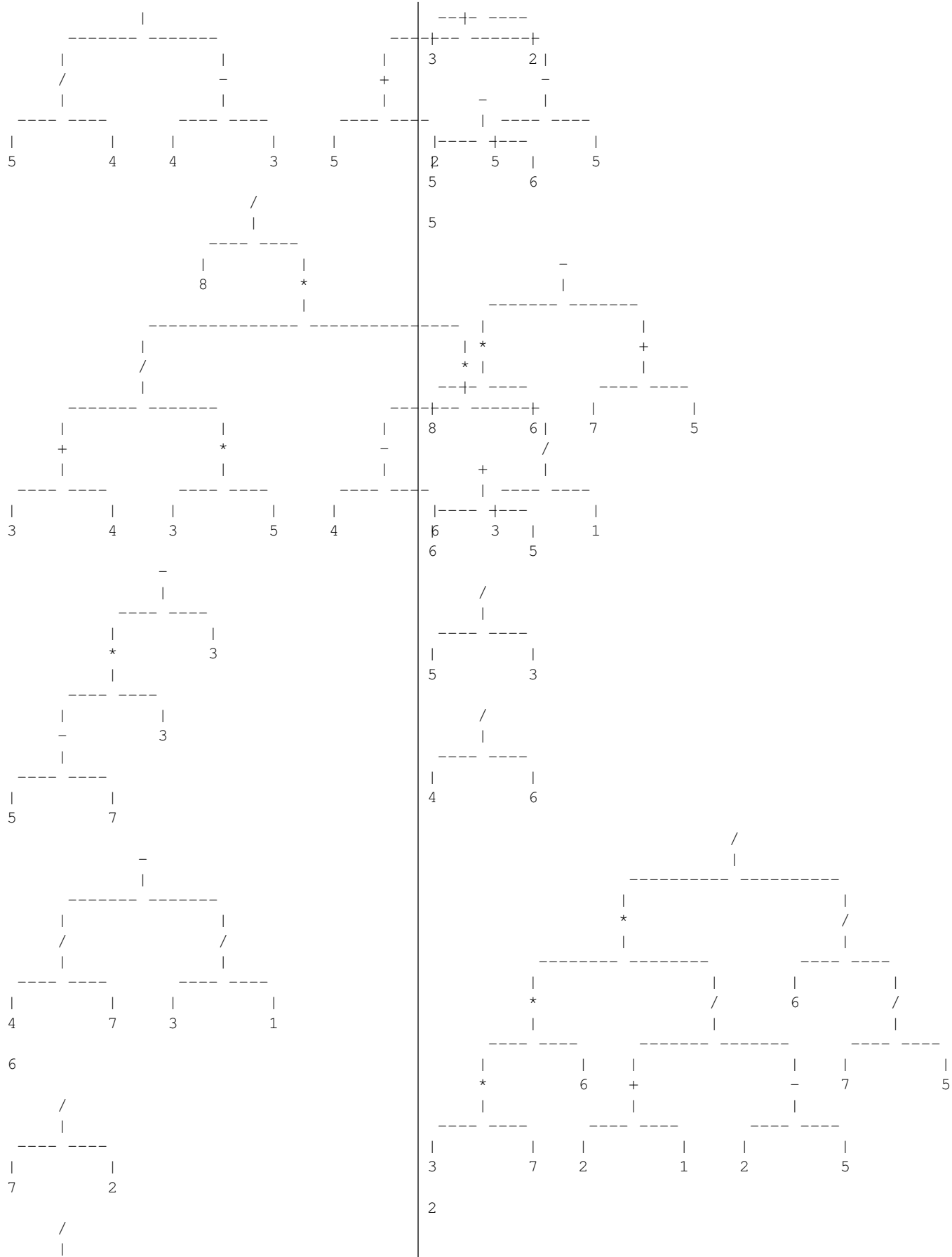
## Sortida

Per a cada cas, la sortida conté la corresponent avaluació de l'arbre o bé una indicació de que s'ha produït un error de divisió per 0 durant el procés d'avaluar l'arbre. 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`





### Exemple de sortida 1

```
26
Division by 0
5
0
Division by 0
Division by 0
Division by 0
-9
-3
```

```
6
3
1
-1
5
36
11
1
0
-21
2
```

### Exemple d'entrada 2

```
INLINEFORMAT
+(12,52)
44
+(65,19)
5
- (/ (/ (- (7,20), +(71,97)), + (/ (75,29), -(87,640)), 40)
* (/ (43,89), 5)
-(8,38)
- (/ (/ (80,30), *(76,21)), -(22, *(38,94)))
/ (*( +(78,53), / (22,60)), -( *(43,20), +(98,42)) )
/ (-( -( *(40,40), +(57,82)), / (100, +(66,63))), 61)
+(37, -( +(73, / (60,90)), 49))
- (*( -(67,51), 36), +(10, +(54,23)))
*(44, -(83,8))
-(35, *(96,39))
- (+ (55,87), *( *( / (60,81), *(16,53)), 99))
- (/ (25, *( +(27,94), 64)), 44)
/ (59, -( / (57, -(92,33)), / (55, +(-(27,12), *(83,242))))
- (*(64, / (+(18,91), -( / (75,35), 23))), -(7, +(-(54,83), -( *(42,93), / (83,10)))))
+(-( / (*(8,9), -(43,66)), -(78, / (5,14))), *( / (77,92), 41))
7
```

### Exemple de sortida 2

```
64
44
84
5
-40
-30
3550
0
Division by 0
61
489
3300
-3709
142
-44
Division by 0
33242
-54
```

### Observació

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

### Informació del problema

Autoria: PRO2

Generació: 2026-01-25T17:33:33.069Z

© Jutge.org, 2006–2026.

<https://jutge.org>