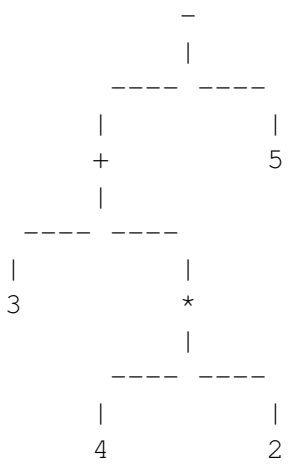
Avaluar expressions sense variables X84774_ca

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 la seva avaluació. 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 l'avaluació de l'expressió representada per t.
int evaluate(const BinaryTree<string> &t);
```

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

```
t=            *
              |
       ------- -------
      |               |
      +               -
      |               |
 ---- ----        ---- ----
|          |     |          |
1          2     5          3

=>

6
```

Fixeu-vos que l'enunciat d'aquest exercici ja ofereix uns fitxers que haureu d'utilitzar per a compilar: `Makefile`, `program.cpp`, `BinaryTree.hpp`, `evaluate.hpp`, `utils.hpp`,

`utils.cpp`. Us falta crear el fitxer `evaluate.cpp` amb els corresponents `includes` i implementar-hi la funció anterior. Valdrà la pena que utilitzeu algunes de les funcions oferides a `utils.hpp`. Quan pugeu la vostra solució al jutge, només cal que pugeu un `tar` construït així:

```
tar cf solution.tar evaluate.cpp
```

Entrada

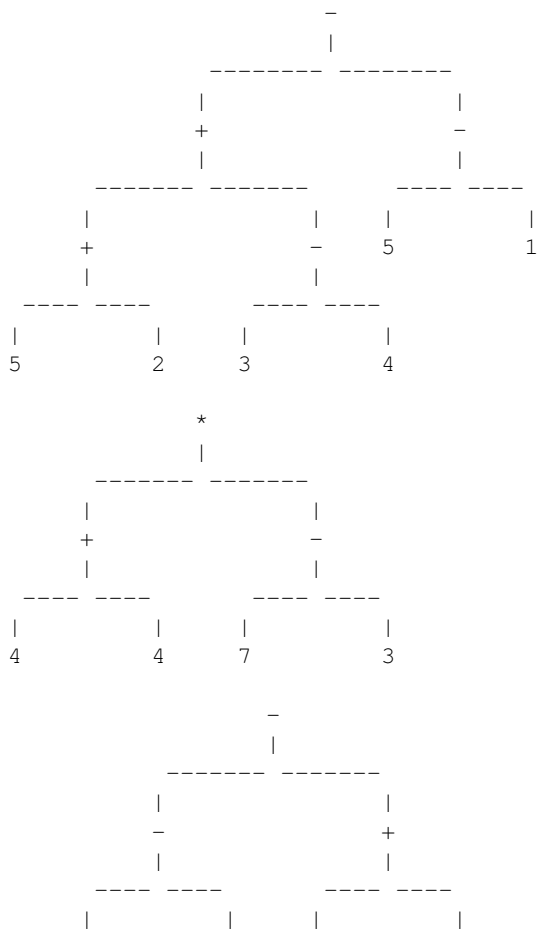
La primera linia de l'entrada descriu el format en el que es descriuen els arbres, o bé INLINEFORMAT o bé VISUALFORMAT. Després venen un nombre arbitrari de casos. Cada cas consisteix en una descripció d'un arbre binari d'strings que representa una expressió correcta amb operadors de suma, resta i multiplicació, i operands naturals. Fixeu-vos en que el programa que us oferim ja s'encarrega de llegir aquestes entrades. Només cal que implementeu la funció abans esmentada.

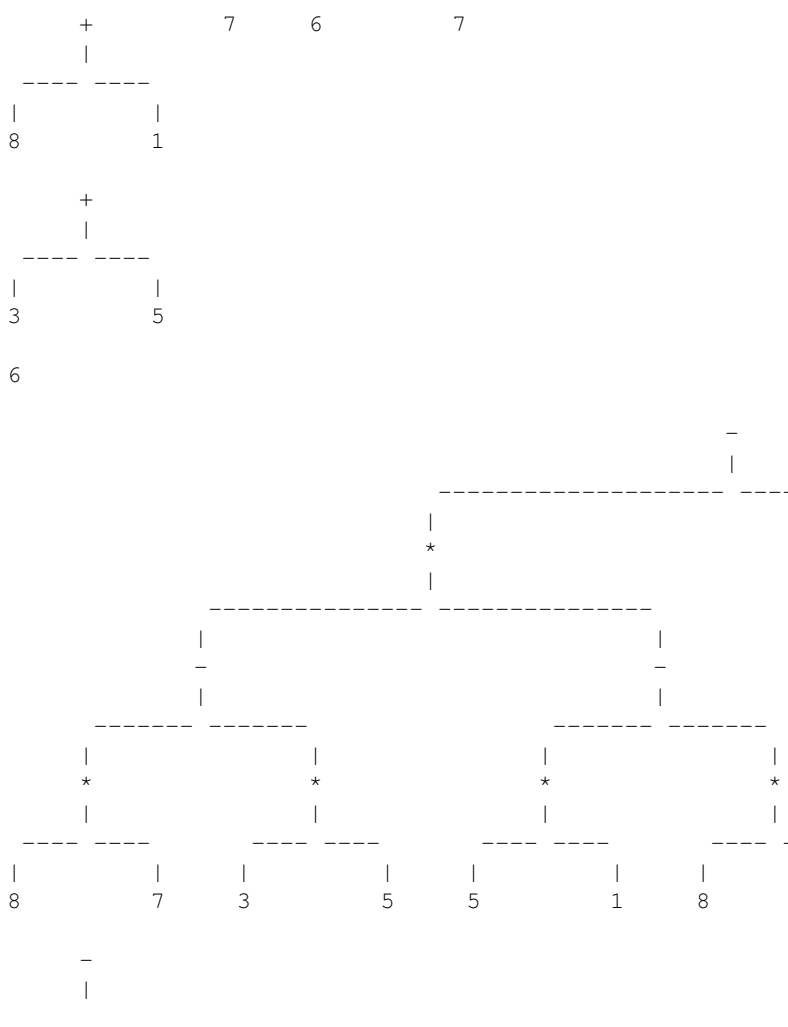
Sortida

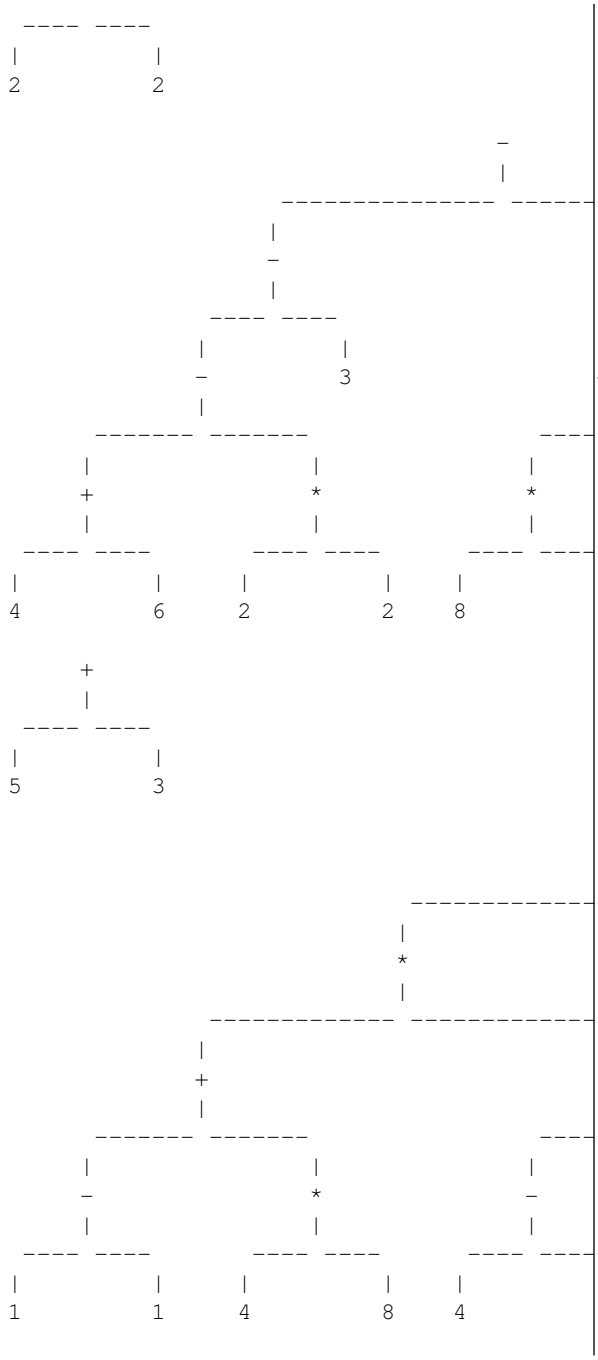
Per a cada cas, la sortida conté la corresponent avaluació de l'arbre. Fixeu-vos en que el programa que us oferim ja s'encarrega d'escriure aquesta avaluació. Només cal que implementeu la funció abans esmentada.

Exemple d'entrada 1

```
VISUALFORMAT
```







Exemple de sortida 1

```
2
32
-11
8
6
-1505
0
-25
8
317
1
4
4
3
2
4
```

Exemple d'entrada 2

```
INLINEFORMAT
+(12,52)
+(-(+(19,51),-(89,*(58,20))),30)
*(18,43)
-(*(+(-(-(93,87),+(88,89)),+(38,36)),46),*(+(15,58),72))
69
-(*(92,31),*(37,86))
+(+(8,22),*(94,57))
-(*(16,24),7)
-(81,39)
+(-(-(+(43,20),*(78,98)),32),-(+(*(75,62),13),+(+(100,0),-(0,1))))
+(63,18)
```

```
44
-(-(+(+(+(97,*(97,39)),*(46,25)),-(+(+(38,31),+(21,84)),
-(-(-(*(-(67,51),7),+(-(10,57),*(60,9))),*(84,+(25,11))
+(-(-(+(100,32),60),*(80,+(14,94))),-(*(71,35),+(*(11,3
*(63,65)
+(41,-(29,+(-(-(47,76),+(48,46)),*(-(83,20),13))))
+(+(10,4),74)
-(-(+(*(39,75),-(22,*(31,65))),*(79,45)),-(*(9,97),+(+
-(-(*(15,98),*(80,84)),+(-(*(89,-(22,5)),-(-(81,28),*(9
```

Exemple de sortida 2

```
64
1171
774
-9718
69
-330
5388
377
42
-3049
81
44
4491
-6864
-6454
4095
-626
88
-3421
-9122
```

Informació del problema

Autoria: PRO1

Generació: 2026-01-25T21:30:42.666Z