
Avaluar expressions binàries amb variables

Z78925_ca

En aquest exercici considerarem arbres que representen expressions sobre els operadors +, -, *, i sobre operands naturals i variables (una variable serà una seqüència no buida de lletres minúscules).

Per exemple, el següent arbre representa l'expressió $(3 + x * 2) - y$

```
-
|-- +
|   |-- 3
|   '-- *
|       |-- x
|       '-- 2
'-- y
```

Per avaluar una expressió com aquesta, necessitarem un *diccionari* que relaciona cada variable amb el valor que conté (un `map<string, int>`).

Implementeu, doncs, la funció següent:

```
/**
 * @brief Avalua una expressió binària amb variables
 *
 * L'expressió és sobre els naturals i els operadors '+', '-', i '*'.
 * A més, hi ha un diccionari 'env' que conté parelles amb cada
 * variable i el seu valor per a un conjunt de variables que poden
 * aparèixer a l'arbre.
 *
 * @param t Arbre amb l'expressió binària.
 * @param env Diccionari amb parelles (nom de variable, valor). Aquest
 *           diccionari no es pot modificar.
 *
 * @pre 't' és no buit. Totes les variables que apareixen a 't'
 *      estan definides a 'env'. Les operacions expressades per
 *      l'arbre no produeixen errors d'_overflow_
 */
int tree_eval_env(BinTree<string> t, const map<string, int>& env);
```

Observació

Els fitxers públics (icona del gatet) contenen:

main.cc	el programa principal
bintree.hh	la classe BinTree
bintree-io.hh	l'entrada/sortida de BinTree
util.hhiutil.cc	un mòdul d'utilitats: <code>is_number</code> , <code>is_var_name</code> i <code>string_to_int</code>

També hi ha un Makefile i el directori .vscode que té la configuració per compilar i depurar amb VSCode.

Cal implementar `tree_eval_env` en un **fitxer .cc nou**, compilar, i finalment **enviar només el fitxer amb la funció**.

Entrada

L'entrada consisteix en una seqüència de parelles d'arbres i assignacions de variables. Cada grup d'assignacions de variables simplement és una línia amb parelles amb el nom d'una variable i el seu valor, separat per espais. (D'això se n'encarrega el programa principal ja disponible als fitxers públics.)

Sortida

Per a cada cas, la sortida és el resultat d'avaluar l'expressió, cada resultat en una línia separada. (Això també ho fa el programa principal.)

Exemple d'entrada

```
b
b 5 .

+
|-- 6
'-- -
    |-- h
    '-- 4

h 5 .

+
|-- 4
'-- -
    |-- f
    '-- a

f 4 a 8 .

*
|-- +
|   |-- 8
|   '-- b
'-- -
    |-- 6
    '-- g

b 2 g 9 .

+
|-- -
|   |-- 1
|   '-- d
'-- +
    |-- g
    '-- c

d 6 g 6 c 7 .
```

Exemple de sortida

```
5
7
0
-30
8
```

Informació del problema

Autor : PRO2

Generació : 2025-03-22 16:29:03

© *Jutge.org*, 2006–2025.

<https://jutge.org>