
Mostra carpetes indentades**Z19994_ca**

Un `'Tree<string>'` ens permet representar una jerarquia de carpetes. Cada node és una carpeta, i les fulles són o bé fitxers, o carpetes buides. Per exemple, l'estructura de carpetes del codi públic d'aquest mateix problema és aquesta:

```
public
|-- .vscode
|   |-- launch.json
|   '-- tasks.json
|-- main.cc
|-- Makefile
|-- tree.hh
'-- tree-io.hh
```

Implementa la funció amb la següent capçalera:

```
/**
 * @brief Mostra un arbre de carpetes i fitxers a la sortida en format
 * indentat
 *
 * Per exemple:
 * ```
 * problema
 *     codi_font
 *         main.cc
 *         Makefile
 *     jocs_de_prova
 *         publics
 *             sample-1.inp
 *         privats
 * ```
 *
 * Cal notar que la última línia és buida per poder veure la separació
 * amb altres arbres de carpetes
 *
 * @param t Un arbre que representa una jerarquia de carpetes i fitxers.
 * @param indent_size Número d'espais d'indentació per a cada nivell.
 *
 * @pre `indent_size` > 0.
 * @post S'ha mostrat per la sortida estàndard l'arbre `t` amb un nivell
 * d'indentació d' `indent_size` espais per a cada nivell de profunditat.
 */
void print_folders(Tree<string> t, int indent_size);
```

Observació

Els fitxers públics (icona del gatet) contenen:

tree.hh	la classe Tree
tree-io.hh	l'entrada/sortida de Tree
main.cc	el programa principal

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

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

Entrada

Cada cas consisteix en un enter que és el nivell d'intendació desitjat, seguit d'una representació textual d'un arbre de carpetes. (Aquesta lectura ja la fa el programa principal.)

Sortida

Per a cada cas, la sortida és el resultat de cridar l'acció `print_folders` per a cada carpeta. (Això també ho fa el programa principal.)

Exemple d'entrada

```
1
problem.yml

2
sample
|-- sample-2.cor
|-- public
|   |-- sample-2.cor
|   |-- tree-io.hh
|   |-- sample-1.cor
|   |-- program.cc
|   '-- sample-1.cor
'-- vendor
    |-- sample-1.inp
    '-- tree-io.hh

3
private-1.inp

4
sample-2.cor

5
private
|-- P68688.zip
|-- program.cc
|-- .vscode
|   |-- sample-2.inp
|   |-- test
|   |   |-- tree.hh
|   |   |-- private-2.inp
|   |   |-- private-2.inp
|   |   '-- program.exe
```

```
|   |-- sample-2.inp
|   |-- test
|   |   |-- problem.yml
|   |   |-- P68688.pdf
|   |   '-- tree-io.hh
|   '-- handler.yml
'-- test
    |-- test
    |   |-- program.exe
    |   '-- main.cc
    '-- .vscode
        |-- sample-2.cor
        '-- program.exe
```

Exemple de sortida

```
problem.yml
```

```
sample
  sample-2.cor
  public
    sample-2.cor
    tree-io.hh
    sample-1.cor
    program.cc
    sample-1.cor
  vendor
    sample-1.inp
    tree-io.hh
```

```
private-1.inp
```

```
sample-2.cor
```

```
private
  P68688.zip
```

```
program.cc
.vscode
  sample-2.inp
  test
    tree.hh
    private-2.inp
    private-2.inp
    program.exe
  sample-2.inp
  test
    problem.yml
    P68688.pdf
    tree-io.hh
  handler.yml
test
  test
    program.exe
    main.cc
  .vscode
    sample-2.cor
    program.exe
```

Informació del problema

Autor : Pau Fernández

Generació : 2025-03-22 13:56:36

© *Jutge.org*, 2006–2025.

<https://jutge.org>