



# C1 - Piscine C

---

C-CPE-042

## Fir Tree

---

An ascii art resizable Fir tree



# Fir Tree

---

repository name: pool\_c\_sapin  
repository rights: ramassage-tek  
language: C

---

- Technicals details
- Subject
  - Examples
    - Small tree
    - relatively large tree



## TECHNICALS DETAILS

---



Don't push your main function into your delivery directory, we will be adding our own. Your files will be compiled adding our **main.c** and our **my\_putchar.c** files.



You are only allowed to use the **my\_putchar** function to complete the following tasks, but don't push it into your delivery directory, and don't copy it in any of your delivered files.



The only allowed system call for this project is `write`.



You must deliver a file called "tree.c" in the root of your repository : this is the only source file that will be checked-out. It will be compiled with the command :

```
Terminal  
~/C-CPE-042> cc main.c my_putchar.c tree.c
```



## SUBJECT

Write a function that displays a fir tree, based on its given size.

If the size is 0, don't display anything.

The function must be prototyped as follows:

```
void tree(int size);
```



You can find a binary called, **tree** on the intranet along with the project description.

Don't forget that you need a coherent test policy to ensure your program outputs are correct. To do so:

- split your functions in **as many small functions as possible**, so that each function is responsible for one single thing (according to the Coding Style).
- Test each of your functions **individually** AND try to automate your testing process with some (shell?) scripts.



These are the first steps in what we call computing tests, and more precisely: unit tests.

## + EXAMPLES

### SMALL TREE

```
~/C-CPE-042> ./a.out 1
*
***
*****
*****
|
```



4