Javascript Lab Seminar

MELL-JAVASCRIPT-03

Day 03

Merry Christmas

v1.61

Day 03

Merry Christmas

repository name: javascript_lab

branch name: day_03

Your repository must contain the totality of your source files.

You must have one file per Task. my_compute_factorial_it is the task one so you need to have a file named `my_compute_factorial_it.js`.

You are only allow to use **var let const if while** and **for**
Do not use any function of any kind that is not your. If you need to use concact() method then create it.

If one of your files prevents you from compiling and if we are not able to correct your work you will receive a 0.

All of the day's functions must produce an answer in under 2 seconds. Overflows must be handled (as errors).

Here's a complete list of the packages we'll use specifically for developing on the command line:

- chalk — colorizes the output
- clear — clears the terminal screen
- clui — draws command-line tables, gauges and spinners
- figlet — creates ASCII art from text
- inquirer — creates interactive command-line user interface
- minimist — parses argument options
- configstore — easily loads and saves config without you having to think about where and how.

Task 01

Square

The goal of this project is to display a square on the screen. Depending on your assignment, the squares will look differently (see below). You have to write the *square* function, which will be called by our main function, which will look like this:

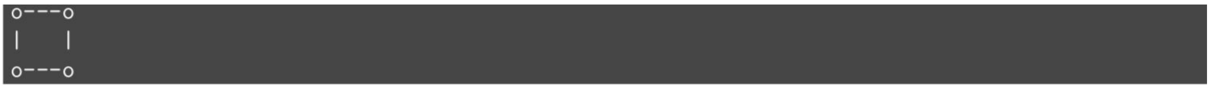
```
void square ( int x, int y, string edge = “o”, string display = “oo-|oo”)
```

<u>Exemple 1</u>

input:

```
$> node square.js 5 3
```

ouput:



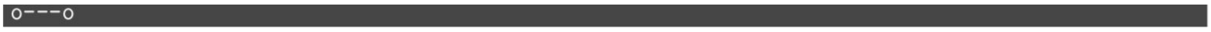
```
o---o
|   |
o---o
```

<u>Exemple 2</u>

input:

```
$> node square.js 5 1
```

ouput:



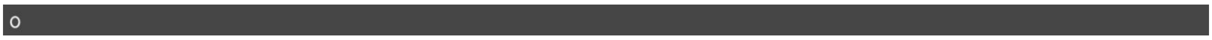
```
o---o
```

Exemple 3

input:

```
$> node square.js 1 1
```

ouput:



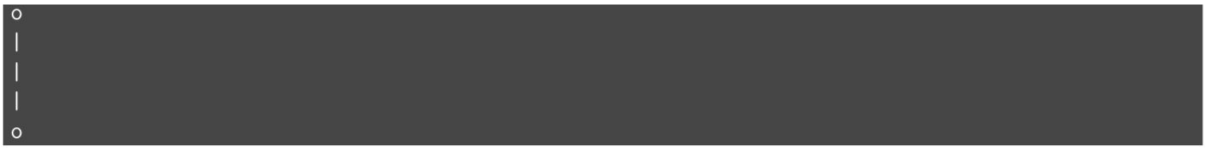
```
o
```

Exemple 4

input:

```
$> node square.js 1 5
```

ouput:



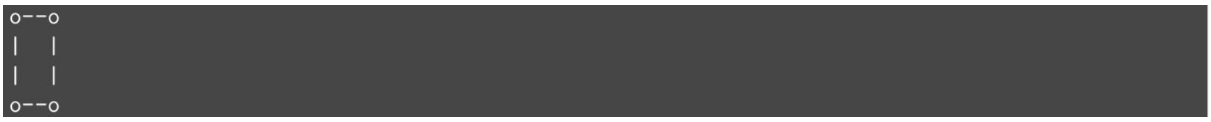
```
o
|
|
|
o
```

Exemple 5

input:

```
$> node square.js 4 4
```

ouput:



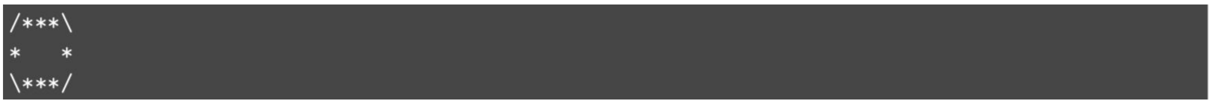
```
o--o
|  |
|  |
o--o
```

Exemple 6

input:

```
$> node square.js 5 3 --d=/\**\/
```

ouput:



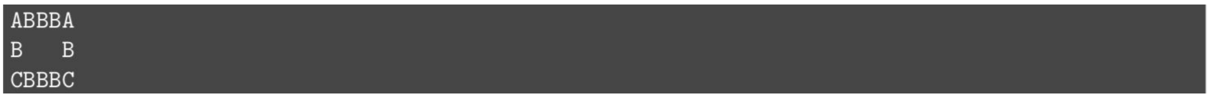
```
/***\
*   *
\***/
```

Exemple 7

input:

```
$> node square.js 5 3 --d=AABBCC
```

ouput:



```
ABBBA
B   B
CBBBC
```

Task 02

Resizable Tree

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:

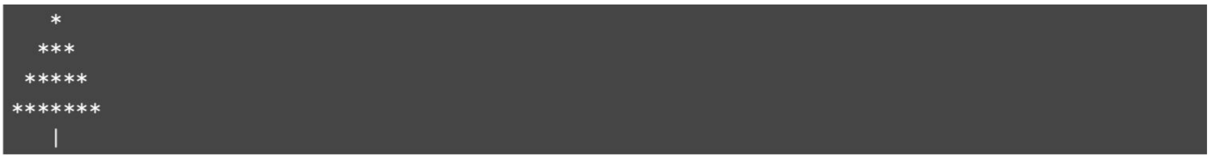
```
void tree ( int size ) ;
```

Exemple 1

input:

```
$> node my_tree.js 1
```

ouput:



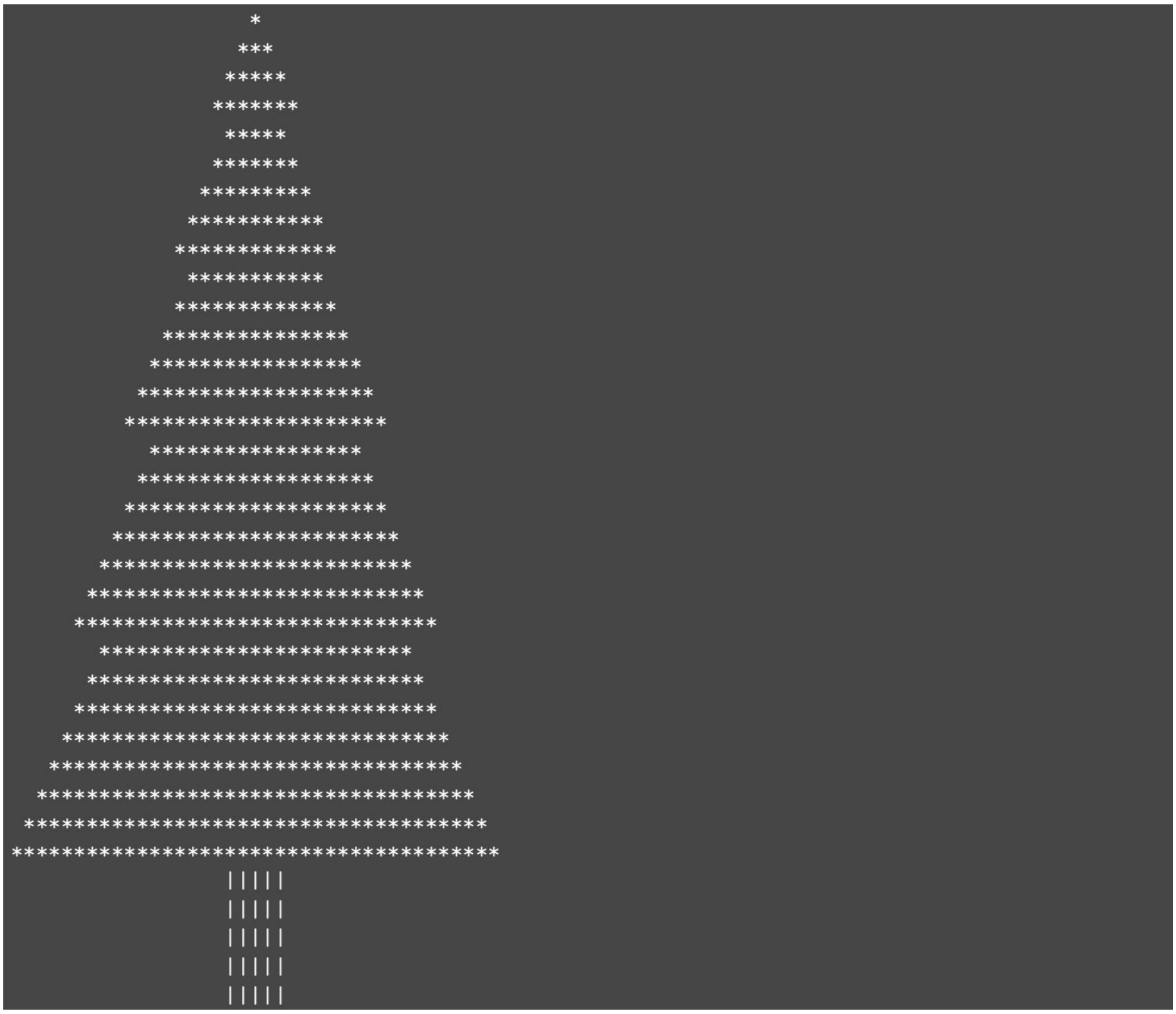
```
   *
  ***
 *****
*******
   |
```

Exemple 2

input:

```
$> node my_tree.js 5
```

ouput:



```
                   *
                  ***
                 *****
                *******
                 *****
                *******
               *********
              ***********
             *************
              ***********
             *************
            ***************
           *****************
          *******************
         *********************
           *****************
          *******************
         *********************
        ***********************
       *************************
      ***************************
     *****************************
       *************************
      ***************************
     *****************************
    *******************************
   *********************************
  ***********************************
 *************************************
***************************************
                 |||||
                 |||||
                 |||||
                 |||||
                 |||||
```