

# Factores

Según el artículo<sup>1</sup> correspondiente de la wikipedia, la “división por tentativa” es el método más sencillo (y más costoso) para descomponer un entero en factores primos. Este algoritmo fue presentado por primera vez por Fibonacci en su libro *Liber Abaci*.

1. Utilice este método para definir en OCaml (dentro de un archivo con nombre **factores.ml**) una función **factoriza: int -> string** de modo que *factoriza n* devuelva un string con la descomposición en factores de *n*, siguiendo exactamente el formato de los siguientes ejemplos. El resultado de su aplicación debe coincidir exactamente con el dado en cada uno de estos ejemplos (que se suponen ejecutados en un sistema con `Sys.int_size = 63`)

[illegible]

<sup>1</sup> [https://en.wikipedia.org/wiki/Trial\\_division](https://en.wikipedia.org/wiki/Trial_division)

Sería normal que alguno de estos cálculos llevase un tiempo apreciable. En ese sentido, probablemente el último ejemplo sea el que más tiempo podría llevar. Como referencia piense que ese cálculo podría consumir un tiempo similar al de calcular el fibonacci de 46 utilizando la siguiente definición:

```
let rec fib n = if n <= 2 then 1 else fib (n-1) + fib (n-2)
```

2. (Opcional) Defina (en un archivo con nombre *factores\_plus.ml*) una variante de la función del ejercicio anterior, *factoriza: int -> string* de modo que el string con la factorización indique la repetición de factores utilizando exponentes. El formato del string de salida debe ajustarse estrictamente al que se observa en los siguientes ejemplos (que se suponen ejecutados en sistema con *Sys.int\_size = 63*)

```
# factoriza 0;;
- : string = "0"
# factoriza 1;;
- : string = "1"
# factoriza (-1);;
- : string = "(-1)"
# factoriza (-6);;
- : string = "(-1) * 2 * 3"
# factoriza 8;;
- : string = "2^3"
# factoriza 18;;
- : string = "2 * 3^2"
# factoriza 72;;
- : string = "2^3 * 3^2"
# factoriza 128;;
- : string = "2^7"
# factoriza 1183350;;
- : string = "2 * 3 * 5^2 * 7^3 * 23"
# factoriza max_int;;
- : string = "3 * 715827883 * 2147483647"
# factoriza (max_int / 2 + 1);;
- : string = "2^61"
# factoriza min_int;;
- : string = "(-1) * 2^62"
# factoriza (min_int + 1);;
- : string = "(-1) * 3 * 715827883 * 2147483647"
# factoriza 4611686018427387847;;
- : string = "4611686018427387847"
```