

"101"

En OCaml, los valores de tipo *string* (“cadenas de caracteres”) son como “arrays” (secuencias indexadas) de caracteres (valores de tipo *char*). El índice del primer carácter es 0 y el del último es $(n-1)$ si la cadena tiene longitud n (i.e. contiene n caracteres).

Si $\langle s \rangle$ es una expresión de tipo *string*, $\langle s \rangle.[i]$ representa el i -ésimo carácter de la cadena. La función *String.length* sirve para obtener el número de caracteres de un string.

```
# let s = "abcd";;  
val s : string = "abcd"  
# String.length s;;  
- : int = 4  
# s.[1];;  
- : char = 'b'  
# ("aei" ^ "ou").[3];;  
- : char = 'o'
```

La función *String.sub* sirve para obtener una copia de un segmento de un string.

```
# String.sub "abcd" 1 2;;  
- : string = "bc"
```

Consideraremos que un string “representa un número natural en binario” si no es vacío y contiene sólo los caracteres '0' y '1'.

Escriba en un archivo ***binarystr.ml*** las definiciones en OCaml que se piden en el resto de este enunciado. Utilice recursividad y manténgase dentro del paradigma funcional.

Defina una función ***binstr_of_int : int -> string*** tal que, para cualquier $\text{int } n \geq 0$, *binstr_of_int* n sea la representación binaria de n . Así, por ejemplo, se debería tener:

```
# binstr_of_int 0;;  
- : string = "0"  
# binstr_of_int 5;;  
- : string = "101"  
# binstr_of_int 255;;  
- : string = "11111111"  
# binstr_of_int 2147483647;;  
- : string = "11111111111111111111111111111111"
```

Defina también una función ***int_of_binstr : string -> int*** de modo que si s representa un número natural binario *int_of_binstr* s sea el valor de tipo *int* que le corresponde. En OCaml, los valores de tipo *int* se representan con *Sys.int_size* bits (Lo más probable es que en su

equipo `Sys.int_size` sea 63). Puesto que un string puede tener muchos más que estos caracteres, la función que se pide debe devolver su “valor” módulo $2^{\text{Sys.int_size}}$.

[illegible]

Si la evaluación de la última expresión provoca un error de ejecución o tarda en ejecutarse (digamos más de 1 segundo en el compilador interactivo *ocaml*) debería considerar la posibilidad de mejorar la definición de la función *int_of_binstr* (tenga en cuenta que un valor de tipo *string* podría tener teóricamente miles de millones de caracteres).

Defina *int_of_binstr' : string -> int* como una nueva versión de la función anterior que ignore los espacios en blanco dentro del string. (Sugerencia: para esta definición puede ser útil el uso de las funciones *String.contains* y *String.index*)

```
# int_of_binstr' "01011001 11110000";;
- : int = 23024
# int_of_binstr' "0101 1001 1111 0000";;
- : int = 23024
# int_of_binstr' "0 1 0 1 1 0 0 1 1 1 1 0 0 0 0";;
- : int = 23024
# int_of_binstr' "    01 01    10 01    11 11    00 00 ";;
- : int = 23024
```