- Introduction and overview
- Basic types, definitions and functions
- Basic data structures

Table of Contents

Greetings

User-defined types
Week 2 Echéance le déc 12, 2016 at 23:30 UTC

Tuples
Week 2 Echéance le déc 12, 2016 at 23:30 UTC

Records
Week 2 Echéance le déc 12, 2016 at 23:30 UTC

Arrays
Week 2 Echéance le déc 12, 2016 at 23:30 UTC

Case study: A small typed database
Week 2 Echéance le déc 12, 2016 at 23:30 UTC

- More advanced data structures
- Higher order functions
- Exceptions, input/output and imperative constructs
- Modules and data abstraction

SEARCHING FOR STRINGS IN ARRAYS (30/30 points)

1. Write a function `is_sorted : string array -> bool` which checks if the values of the input array are sorted in strictly increasing order, implying that its elements are unique (use `String.compare`).
2. Using the binary search algorithm, an element can be found very quickly in a sorted array. Write a function `find : string array -> string -> int` such that `find arr word` is the index of the `word` in the sorted array `arr` if it occurs in `arr` or -1 if `word` does not occur in `arr`.
The number or array accesses will be counted, to check that you obtain the expected algorithmic complexity. Beware that you really perform the minimal number of accesses. For instance, if your function has to test the contents of a cell twice, be sure to put the result of the access in a variable, and then perform the tests on that variable.

YOUR OCAML ENVIRONMENT

```
let is_sorted a =
  let rec is_sorted_rec array inf sup=
    if sup <= inf then true else
    if array.(inf) >= array.(inf + 1) then false else
      true && is_sorted_rec array (inf + 1) sup
  in
  is_sorted_rec a 0 (Array.length a - 1)
;;


let find dict word =
  let rec bin_search dict word min max =
    if min > max then -1 else
    if min = max then
      if dict.(min)= word then min else -1
    else
      let med = (min + max) / 2 in
      if dict.(med) >= word then bin_search dict word min med else
        bin_search dict word (med + 1) max
  in
  bin_search dict word 0 (Array.length dict - 1)
;;

```

Evaluate >

Switch >>

Typechecl

Reset Templ

Full-screen |

Check & Sa

Exercise complete (click for details) 30 pts

v Exercise 1: is_sorted Completed, 10 pts

Found `is_sorted` with compatible type.

```
Computing is_sorted [||]
Correct value true                                                        1 pt
Computing is_sorted [|"single"|]
Correct value true                                                        1 pt
Computing is_sorted [|"clown"; "elephant"; "alpha"; "clown"|]
Correct value false                                                       1 pt
Computing is_sorted [|"alpha"; "bowl"; "clown"; "diddy"; "elephant"|]
Correct value true                                                        1 pt
Computing is_sorted [|"alpha"; "bowl"; "diddy"|]
Correct value true                                                        1 pt
Computing is_sorted [|"bowl"; "bowl"; "clown"; "elephant"|]
Correct value false                                                       1 pt
Computing is_sorted [|"bowl"; "clown"; "diddy"; "elephant"|]
Correct value true                                                        1 pt
Computing is_sorted [|"diddy"; "bowl"; "bowl"; "alpha"|]
Correct value false                                                       1 pt
Computing is_sorted [|"alpha"; "elephant"|]
Correct value true                                                        1 pt
Computing is_sorted [|"clown"; "alpha"; "clown"; "diddy"; "elephant"; "elephant"|]
Correct value false                                                       1 pt
```

```
Computing find [|"a"; "b"; "c"|] "a"
Correct value 0                                                          1 pt
Search in 3 operations.                                                  1 pt
Computing find [|"a"; "b"; "c"|] "b"
Correct value 1                                                          1 pt
Search in 3 operations.                                                  1 pt
Computing find [|"a"; "b"; "c"|] "c"
Correct value 2                                                          1 pt
Search in 2 operations.                                                  1 pt
Computing find [|"a"; "b"; "c"|] "d"
Correct value -1                                                         1 pt
Search in 2 operations.                                                  1 pt
Computing find [||] ""
Correct value -1                                                         1 pt
Search in 0 operations.                                                  1 pt
Computing find [|"a"; "b"; "c"; "d"; "e"; "f"; "g"; "h"; "i"; "j"; "k"|] "l"
Correct value -1                                                         1 pt
Search in 4 operations.                                                  1 pt
Computing find [|"b"; "c"|] "h"
Correct value -1                                                         1 pt
Search in 2 operations.                                                  1 pt
Computing find [|"c"; "d"; "e"; "f"; "g"; "h"; "i"; "j"; "k"|] "i"
Correct value 6                                                          1 pt
Search in 4 operations.                                                  1 pt
Computing find [|"e"; "f"; "g"|] "j"
Correct value -1                                                         1 pt
Search in 2 operations.                                                  1 pt
Computing find [|"d"; "e"; "f"; "g"; "h"; "i"|] "c"
Correct value -1                                                         1 pt
Search in 4 operations.                                                  1 pt
```