- 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

TIME ON PLANET SHADOKUS (30/30 points)

On planet Shadokus, a year has 5 months, each month has 4 days, each day has 3 hours and each hour has 2 minutes. A calendar date is therefore defined as the record type `date` of the given prelude.

1. A date is well-formed if its year index is `>= 1`, its month index is `>= 1` and `<= 5`, its day index is `>= 1` and `<= 4`, its hour index is `>= 0` and `<= 2`, and its minute index is `>= 0` and `<= 1`.
   The start of year `12` would be:

   ```
   { year = 12; month = 1; day = 1; hour = 0; minute = 0 }
   ```

   The end of year `12` would be:

   ```
   { year = 12; month = 5; day = 4; hour = 2; minute = 1 }
   ```

   Write a function `wellformed : date -> bool` which checks that the input date is well formed.
2. On planet Shadokus, the origin of time is the discovery of the Big-Lambda-Machine, a magical computer that evaluates the infinite lambda-term of time. It is defined by value `the_origin_of_time` of the given prelude.
   Write a function `next : date -> date` which computes the date which comes one minute after the input date.
3. In this computer, the time is represented by an integer that counts the number of minutes since `1/1/1 0:0` (the origin of time).
   Write a function `of_int : int -> date` that converts such an integer into a `date`.

THE GIVEN PRELUDE

```
type date =
  { year : int; month : int; day : int;
    hour : int; minute : int }

let the_origin_of_time =
  { year = 1; month = 1; day = 1;
    hour = 0; minute = 0 }
```

YOUR OCAML ENVIRONMENT

```
let wellformed date =
  if date.year >= 1 &&
     date.month >= 1 && date.month <= 5 &&
     date.day >= 1 && date.day <= 4 &&
     date.hour >= 0 && date.hour <= 2 &&
     date.minute >=0 && date.minute <= 1
  then true else false;;

let next date =
  let rest_min = (date.minute + 1) / 2 in
  let rest_hour = (date.hour + rest_min) / 3 in
  let rest_day = (date.day + rest_hour) / 5 in
  let rest_month = (date.month + rest_day) / 6 in
  { year = (date.year + rest_month);
    month = (date.month + rest_day - 1) mod 5 + 1;
    day = (date.day + rest_hour - 1) mod 4 + 1;
    hour = (date.hour + rest_min) mod 3;
    minute = (date.minute + 1) mod 2
  };;

let rec of_int minutes =
  if minutes = 0 then the_origin_of_time else
    next (of_int (minutes - 1))
;;

```

Evaluate >

Switch >>

Typechecl

Reset Templ

Full-screen |

Check & Sa

```
v Exercise 1: wellformed                                                          Completed, 10 pts
 Found wellformed with compatible type.
 Computing wellformed {year = 1; month = 1; day = 1; hour = 0; minute = 0}
 Correct value true                                                                           1 pt
 Computing wellformed {year = 1000; month = 1; day = 1; hour = 1; minute = 0}
 Correct value true                                                                           1 pt
 Computing wellformed {year = 1; month = 1; day = 1; hour = 1; minute = -1}
 Correct value false                                                                          1 pt
 Computing wellformed {year = -1; month = 1; day = 1; hour = 1; minute = 1}
 Correct value false                                                                          1 pt
 Computing wellformed {year = 1; month = 1; day = 3; hour = -1; minute = 1}
 Correct value false                                                                          1 pt
 Computing wellformed {year = -1; month = 2; day = 2; hour = 0; minute = 2}
 Correct value false                                                                          1 pt
 Computing wellformed {year = 0; month = 5; day = 2; hour = 3; minute = 1}
 Correct value false                                                                          1 pt
 Computing wellformed {year = 5; month = 2; day = 0; hour = -1; minute = 1}
 Correct value false                                                                          1 pt
 Computing wellformed {year = -1; month = 4; day = 2; hour = 3; minute = 2}
 Correct value false                                                                          1 pt
 Computing wellformed {year = 1; month = 1; day = 5; hour = 0; minute = -1}
 Correct value false                                                                          1 pt
v Exercise 2: next                                                                Completed, 10 pts
 Found next with compatible type.
 Computing next {year = 5; month = 3; day = 4; hour = 1; minute = 0}
 Correct value {year = 5; month = 3; day = 4; hour = 1; minute = 1}                           1 pt
 Computing next {year = 7; month = 1; day = 3; hour = 1; minute = 0}
 Correct value {year = 7; month = 1; day = 3; hour = 1; minute = 1}                           1 pt
 Computing next {year = 2; month = 3; day = 3; hour = 0; minute = 0}
 Correct value {year = 2; month = 3; day = 3; hour = 0; minute = 1}                           1 pt
 Computing next {year = 1; month = 2; day = 4; hour = 0; minute = 1}
 Correct value {year = 1; month = 2; day = 4; hour = 1; minute = 0}                           1 pt
 Computing next {year = 2; month = 3; day = 3; hour = 2; minute = 0}
 Correct value {year = 2; month = 3; day = 3; hour = 2; minute = 1}                           1 pt
 Computing next {year = 3; month = 4; day = 3; hour = 0; minute = 0}
 Correct value {year = 3; month = 4; day = 3; hour = 0; minute = 1}                           1 pt
 Computing next {year = 8; month = 3; day = 1; hour = 2; minute = 0}
 Correct value {year = 8; month = 3; day = 1; hour = 2; minute = 1}                           1 pt
 Computing next {year = 8; month = 4; day = 1; hour = 2; minute = 1}
 Correct value {year = 8; month = 4; day = 2; hour = 0; minute = 0}                           1 pt
 Computing next {year = 5; month = 5; day = 4; hour = 1; minute = 1}
 Correct value {year = 5; month = 5; day = 4; hour = 2; minute = 0}                           1 pt
 Computing next {year = 9; month = 2; day = 2; hour = 1; minute = 0}
 Correct value {year = 9; month = 2; day = 2; hour = 1; minute = 1}                           1 pt
v Exercise 3: of_int                                                              Completed, 10 pts
 Found of_int with compatible type.
 Computing of_int 2
 Correct value {year = 1; month = 1; day = 1; hour = 1; minute = 0}                           1 pt
 Computing of_int 78
 Correct value {year = 1; month = 4; day = 2; hour = 0; minute = 0}                           1 pt
 Computing of_int 161
 Correct value {year = 2; month = 2; day = 3; hour = 2; minute = 1}                           1 pt
 Computing of_int 47
 Correct value {year = 1; month = 2; day = 4; hour = 2; minute = 1}                           1 pt
 Computing of_int 13
 Correct value {year = 1; month = 1; day = 3; hour = 0; minute = 1}                           1 pt
 Computing of_int 97
 Correct value {year = 1; month = 5; day = 1; hour = 0; minute = 1}                           1 pt
 Computing of_int 85
 Correct value {year = 1; month = 4; day = 3; hour = 0; minute = 1}                           1 pt
 Computing of_int 31
 Correct value {year = 1; month = 2; day = 2; hour = 0; minute = 1}                           1 pt
```

Correct value `{year = 1; month = 5; day = 4; hour = 1; minute = 0}` 1 pt