ENIGMA (30/30 points)

Let us solve the following puzzle:
If you multiply my grand-son age by four, you know how old I am. Now, if you exchange the two digits of our ages then you have to multiply by three my age to get the age of my grand-son!

1. Write a function `exchange` of type `int -> int` that takes an integer `x` between 10 and 99 and returns an integer which is `x` whose digits have been exchanged. For instance, `exchange 73 = 37`.
2. Define `is_valid_answer` of type `int * int -> bool` such that `is_valid_answer (grand_father_age, grand_son_age)` returns `true` if and only if `grand_father_age` and `grand_son_age` verify the constraints of the puzzle.
3. Write a function `find : (int * int) -> (int * int)` that takes a pair `(max_grand_father_age, min_grand_son_age)` and returns a solution `(grand_father_age, grand_son_age)` to the problem, where `min_grand_son_age <= grand_son_age < grand_father_age <= max_grand_father_age` or `(-1, -1)` if there was no valid answer in the given range.

YOUR OCAML ENVIRONMENT

```
let exchange k =
  (k mod 10) * 10 + (k / 10) ;;

let is_valid_answer (grand_father_age, grand_son_age) =
  grand_son_age * 4 = grand_father_age
  &&
  (exchange grand_father_age) * 3 = exchange grand_son_age ;;


let rec find answer =
  let (grand_father_age, grand_son_age) = answer in
  if grand_son_age >= grand_father_age then (-1,-1) else
  if find_down answer <> (-1,-1) then find_down answer else
    find (grand_father_age, grand_son_age + 1)
and
  find_down answer =
  let (grand_father_age, grand_son_age) = answer in
  if grand_son_age >= grand_father_age then (-1,-1) else
  if is_valid_answer answer = true then answer else
    find_down (grand_father_age - 1, grand_son_age);;

```

Evaluate >

Switch >>

Typecheck

Reset Templ

Full-screen

Check & Sa

Exercise complete (click for details) — 30 pts

v Exercise 1: exchange — Completed, 10 pts

Found exchange with compatible type.	
Computing exchange 27	
Correct value 72	1 pt
Computing exchange 98	
Correct value 89	1 pt
Computing exchange 66	
Correct value 66	1 pt
Computing exchange 13	
Correct value 31	1 pt
Computing exchange 54	
Correct value 45	1 pt
Computing exchange 67	
Correct value 76	1 pt
Computing exchange 11	
Correct value 11	1 pt
Computing exchange 78	
Correct value 87	1 pt

Correct value 11 1 pt

v Exercise 2: `is_valid_answer` Completed, 10 pts

Found `is_valid_answer` with compatible type.

Computing `[REDACTED FOR YOUR OWN GOOD]`

Correct value `true` 1 pt

Computing `[REDACTED FOR YOUR OWN GOOD]`

Correct value `false` 1 pt

Computing `[REDACTED FOR YOUR OWN GOOD]`

Correct value `false` 1 pt

Computing `[REDACTED FOR YOUR OWN GOOD]`

Correct value `false` 1 pt

Computing `[REDACTED FOR YOUR OWN GOOD]`

Correct value `false` 1 pt

Computing `[REDACTED FOR YOUR OWN GOOD]`

Correct value `false` 1 pt

Computing `[REDACTED FOR YOUR OWN GOOD]`

Correct value `false` 1 pt

Computing `[REDACTED FOR YOUR OWN GOOD]`

Correct value `false` 1 pt

Computing `[REDACTED FOR YOUR OWN GOOD]`

Correct value `false` 1 pt

Computing `[REDACTED FOR YOUR OWN GOOD]`

Correct value `false` 1 pt

v Exercise 3: `find` Completed, 10 pts

Found `find` with compatible type.

Computing `find (99, 10)`

Correct value `(72, 18)` 1 pt

Computing `find (99, 11)`

Correct value `(72, 18)` 1 pt

Computing `find (87, 10)`

Correct value `(72, 18)` 1 pt

Computing `find (92, 10)`

Correct value `(72, 18)` 1 pt

Computing `find (76, 23)`

Correct value `(-1, -1)` 1 pt

Computing `find (76, 24)`

Correct value `(-1, -1)` 1 pt

Computing `find (80, 23)`

Correct value `(-1, -1)` 1 pt

Computing `find (96, 30)`

Correct value `(-1, -1)` 1 pt

Computing `find (83, 10)`

Correct value `(72, 18)` 1 pt

Computing `find (76, 24)`

Correct value `(-1, -1)` 1 pt