PRIME NUMBERS (30/30 points)

Let's define some usual arithmetical functions.

1. `gcd` that takes two non-negative integers `n` and `m`, and that returns the greatest common divisor of `n` and `m`, following Euclid's algorithm.
2. `multiple_upto : int -> int -> bool` that takes two non-negative integers `n` and `r`, and that tells whether `n` admits at least one divisor between `2` and `r`, inclusive. In other words that there exists a number `d >= 2` and `<= r`, such that the remainder of the division of `n` by `d` is zero.
3. `is_prime` a takes a non-negative integer `n` and checks whether it is a prime number.

Important note: You can assume that both `integer_square_root` and `multiple_of` exist, and that they are correct answers to the *Simple functions over integers* exercise from the previous sequence.

Once `is_prime` works, you can try writing a new version of it which is self-contained (that contains all definitions of auxiliary functions as locally defined functions).

YOUR OCAML ENVIRONMENT

```
let rec gcd n m =
  if m = 0 then n else gcd m (n mod m);;

let rec multiple_upto n r =
  if r < 2 then false else
  if r = 2 || multiple_of n r = true then multiple_of n r else multiple_upto n (r - 1);;

let is_prime n =
  if n = 1 then false else
    not (multiple_upto n (integer_square_root n));;

```

Evaluate >

Switch >>

Typechecl

Reset Templ

Full-screen |

Check & Sa

Exercise complete (click for details) 30 pts

v Exercise 1: gcd Completed, 10 pts

```
Found gcd with compatible type.
Computing gcd 10 12
Correct value 2                                              1 pt
Computing gcd 3 19
Correct value 1                                              1 pt
Computing gcd 16 24
Correct value 8                                              1 pt
Computing gcd 33 77
Correct value 11                                             1 pt
Computing gcd 1 7
Correct value 1                                              1 pt
Computing gcd 10 1
Correct value 1                                              1 pt
Computing gcd 9 9
Correct value 9                                              1 pt
Computing gcd 7 9
Computing gcd 3 6
Correct value 3                                                   1 pt
v Exercise 2: multiple_upto                          Completed, 10 pts
Found multiple_upto with compatible type.
Computing multiple_upto 10 3
Correct value true                                                1 pt
Computing multiple_upto 30 2
Correct value true                                                1 pt
Computing multiple_upto 25 6
Correct value true                                                1 pt
Computing multiple_upto 11 10
Correct value false                                               1 pt
Computing multiple_upto 6 6
Correct value true                                                1 pt
Computing multiple_upto 8 5
Correct value true                                                1 pt
Computing multiple_upto 3 5
Correct value true                                                1 pt
Computing multiple_upto 3 6
Correct value true                                                1 pt
Computing multiple_upto 8 1
Correct value false                                               1 pt
Computing multiple_upto 9 2
Correct value false                                               1 pt
v Exercise 3: is_prime                               Completed, 10 pts
Found is_prime with compatible type.
Computing is_prime 1
Correct value false                                               1 pt
Computing is_prime 2
Correct value true                                                1 pt
Computing is_prime 19
Correct value true                                                1 pt
Computing is_prime 41
Correct value true                                                1 pt
Computing is_prime 67
Correct value true                                                1 pt
Computing is_prime 3
Correct value true                                                1 pt
Computing is_prime 3
Correct value true                                                1 pt
Computing is_prime 2
Correct value true                                                1 pt
Computing is_prime 3
Correct value true                                                1 pt
Computing is_prime 3
Correct value true                                                1 pt
```