SIMPLE FUNCTIONS OVER INTEGERS (20/20 points)

Let's define two functions working with integers:

1. `multiple_of` that takes two integer parameters, `n` and `d`, and determines whether `n` is a multiple of `d`. The function must return a boolean value. This function can be written without recursion. Look at the operators defined on integers in sequence 1.
2. `integer_square_root` that calculates the *integer square root* of a positive integer `n`, that is the largest integer `r` such that `r * r <= n`. **Hint:** you may use floating point arithmetic, but don't forget that you have to convert explicitely between `float` and `int`.

YOUR OCAML ENVIRONMENT

```
let multiple_of n d =
  n mod d = 0;;

let integer_square_root n =
  let intn = float_of_int n in
  int_of_float (sqrt intn);;
```

Evaluate >

Switch >>

Typecheck

Reset Templ

Full-screen |

Check & Sa

Exercise complete (click for details) — 20 pts

∨ Exercise 1: `multiple_of`	Completed, 10 pts
Found `multiple_of` with compatible type.	
Computing `multiple_of 4 2`	
Correct value `true`	1 pt
Computing `multiple_of 8 3`	
Correct value `false`	1 pt
Computing `multiple_of 13 2`	
Correct value `false`	1 pt
Computing `multiple_of 12 4`	
Correct value `true`	1 pt
Computing `multiple_of 2 1`	
Correct value `true`	1 pt
Computing `multiple_of -1 -2`	
Correct value `false`	1 pt
Computing `multiple_of 1 -4`	
Correct value `false`	1 pt
Computing `multiple_of 1 -1`	
Correct value `true`	1 pt
Computing `multiple_of 1 2`	
Correct value `false`	1 pt
Computing `multiple_of -4 4`	
Correct value `true`	1 pt
∨ Exercise 2: `integer_square_root`	Completed, 10 pts
Found `integer_square_root` with compatible type.	

Correct value 4	1 pt
Computing `integer_square_root 144`	
Correct value 12	1 pt
Computing `integer_square_root 89`	
Correct value 9	1 pt
Computing `integer_square_root 9643`	
Correct value 98	1 pt
Computing `integer_square_root 9639`	
Correct value 98	1 pt
Computing `integer_square_root 702`	
Correct value 26	1 pt
Computing `integer_square_root 4945`	
Correct value 70	1 pt
Computing `integer_square_root 4207`	
Correct value 64	1 pt
Computing `integer_square_root 8744`	
Correct value 93	1 pt