INTERACTIVE THEOREM PROVING, ASSIGNMENT 1

Instructions:

- Submit your work as a zip file, containing files `u5228111/Q1.lean`, ..., `u5228111/Q5.lean`.
- Each file must compile without errors (but `sorry` is okay if you get stuck and want to keep going).
- Please include comments explaining clearly what you are doing, wherever there is the slightest doubt. Don't expect sympathetic marking if you mess something up, and haven't written a comment explaining what you intended!
- If a question asks for, or deserves, extended written comments, feel free to include a pdf as well.
- You may freely import anything from `mathlib`, except where noted.

---

(1) Define the natural numbers as an inductive type, and prove at least one of
  (a) `a * (b + c) = a * b + a * c`
  (b) `a * b = b * a`
  (c) `(a * b) * c = a * (b * c)`

(2) Let's formalise the statements (but not the proofs!) of some famous theorems or conjectures. Choose one of the following, and define a `Prop` corresponding to the statement of the theorem. For example, if you were formalising the statement of Fermat's last theorem, you would write

```
theorem statement_of_fermats_last_theorem : Prop :=
∀ n ≥ 3, ∀ a b c ≥ 1, a ^ n + b ^ n ≠ c ^ n
```

  (a) The Green-Tao theorem, that there are arbitrarily long arithmetic progressions of primes.
  (b) Apéry's theorem, that $\zeta(3)$ is irrational.
  (c) The *abc* conjecture.
    - You'll need to define the *radical*, and may prefer to discover and explain informally an equivalence with a version that does not mention real numbers.
  (d) The Riemann hypothesis.
    - This one would be seriously difficult, if you start to define the Riemann ζ function; I'd be happy if you formalised an elementary statement *equivalent* to RH, and gave a citation to the literature for the equivalence.

(3) (a) (Optional) Define your own list type (otherwise, below, use the built-in one).
  (b) Give sensible definitions of

```
def concat {α : Type} : list (list α) → list α := ...
def len {α : Type} : list α → ℕ := ...
```

  (c) Show that the concatenation of a nonempty list of nonempty lists is nonempty. (Hint: Using the `len` function you just defined may not be the best way to formalise 'non-empty'; can you define an inductive predicate?)

(4) Define the binomial coefficients $\binom{n}{m}$, and prove $\sum_{m=0}^{n} \binom{n}{m} = 2^n$.
  - You'll want to use `finset.sum` from `algebra/big_operators.lean`. (This one may be awkward at first: I'll provide some hints later.)