◊ **Best before:** December 15.

1. Prove that $\{\oplus, \rightarrow\}$ is a complete set of connectives.
   See page 139 of course notes for definition of $\oplus$.

2. (a) Is $\{\neg, \leftrightarrow\}$ a complete set of connectives? Justify your answer.
   (b) Is $\{\oplus, \wedge, \vee\}$ a complete set of connectives? Justify your answer.

3. We have seen one *unary* connective ($\neg$) and several *binary* connectives ($\wedge, \vee, \rightarrow, \leftrightarrow, |, \downarrow, \oplus$) in the course notes. We now introduce the notion of *ternary* connectives and a convention for writing propositional formulas with them. A ternary connective connects three formulas. We use a pair of symbols, placing the first symbol between the first and second formulas, and the second symbol between the second and third formulas. We illustrate with two examples, $(\pm, :)$ and $(\mp, :)$, called *Majority* and *Minority* respectively (see explanation about *Majority* on pages 131-132 of the notes). These are defined by the following truth table.

Q_1	Q_2	Q_3	$(Q_1 \pm Q_2 : Q_3)$	$(Q_1 \mp Q_2 : Q_3)$
0	0	0	0	1
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	0

   (a) There are 4 distinct unary connectives and 16 distinct binary connectives.
   How many distinct ternary connectives are there? Explain your answer.
   (b) Informally explain why $\{(\mp, :), \rightarrow\}$ is a complete set of connectives.
   (c) Informally explain why $\{(\pm, :), \rightarrow\}$ is **not** a complete set of connectives.

4. Let x, y and z be propositional variables and consider the propositional formula

$$(\neg x \rightarrow (y \wedge z)) \wedge (\neg y \rightarrow (x \wedge z)).$$

   (a) Give a truth table for the above formula. Show all columns (as shown in class).
   (b) Using part (a), write a DNF formula that is logically equivalent to the given formula.
   (c) Using part (a), write a CNF formula that is logically equivalent to the given formula.
   (d) Using only substitution of logically equivalent sub-formulas (and in particular **without** using truth tables), derive a CNF formula that is logically equivalent to the given formula. Show the steps you use to derive your formula.
   (e) Make up more propositional formulas and repeat parts (a) through (d) with your formulas.

5. This question concerns the binary connective $\leftrightarrow$.

   (a) Is $((x \leftrightarrow y) \leftrightarrow z)$ logically equivalent to $(x \leftrightarrow (y \leftrightarrow z))$?
   Derive your answer both **with** and **without** using truth tables.

   (b) For any integer $n > 0$, when exactly is $x_1 \leftrightarrow x_2 \leftrightarrow \cdots \leftrightarrow x_n$ satisfied?
   Find a pattern, then use induction to prove it.

6. Do exercise 2 on page 180 of the course notes (about prime conjectures).

7. Do exercise 5 on page 181 of the course notes (about a result of Professor John Friedlander).

8. Do exercise 6 on page 181 of the course notes (about logical implication/equivalence of first-order formulas).

9. Consider a first-order language with binary predicate R and equality predicate $=$.
   We define the following formulas.

   F_1: $\forall x\ \exists y\ \Big(\neg{=}(x,y) \rightarrow R(x,y)\Big)$.

   F_2: $\exists x\ \forall y\ \Big(\ {=}(x,y) \wedge \neg R(y,x)\Big)$.

   F_3: $\exists y\ \forall x\ \neg\Big(\ {=}(x,y) \vee R(y,x)\Big)$.

   F_4: $\forall x\ \forall y\ \neg{=}(x,y) \leftrightarrow \forall x\ \forall y\ \Big(R(x,y) \leftrightarrow \neg R(y,x)\Big)$.

   F_5: $\forall x\ \forall y\ \Big(\neg{=}(x,y) \leftrightarrow (R(x,y) \leftrightarrow \neg R(y,x))\Big)$.

   For each of the following formulas, state whether it is (i) valid, (ii) unsatisfiable, or (iii) both satisfiable and falsifiable. Justify your answers.

   (a) $F_1 \leftrightarrow F_2$.

   (b) $F_1 \rightarrow F_3$.

   (c) F_4.

   (d) F_5.

10. (a) Using the summary of logical equivalences from the additional notes, transform the following formula into a logically equivalent PNF formula in which the quantifier-free portion uses only the connective $\rightarrow$.
   $$\Big(\forall x\, A(x) \wedge \forall x\, B(x) \wedge \forall x\, C(x)\Big) \rightarrow \forall x\, D(x).$$

   (b) Make up more first-order formulas and transform them into logically equivalent PNF formulas. Try to use every equivalence law at least once.

11. *Fun with the CNF satisfiability problem!*

   A *3-CNF* formula is a CNF formula with exactly 3 literals in each clause.
   We want to answer the following question.

   What is the probability that a random 3-CNF formula with n variables and k clauses is satisfiable?

   Consider what should happen if we were to fix n and let k vary. The formula is very likely to be satisfiable when k is small, and very likely to be unsatisfiable when k is large. Do you see why? We would like to find the value of k, as a function of n, when the probability of the formula being satisfiable is exactly one half, or the range of values of k when the probability is in some range, say $[0.25, 0.75]$.

(a) Write a function that takes a 3-CNF formula and returns whether it is satisfiable. You may use any programming language, and choose any way of representing a 3-CNF formula.

(b) Write a function that takes numbers n, k and t, and randomly generates t 3-CNF formulas, each with n variables and k clauses, and returns how many of them are satisfiable. Of course, this function should call your function from part (a).

(c) Experiment with your function from part (b) to get estimates for the values we want to find.