Heightbook is a social network platform where you can compare the height of your friends!

The **first-order language of Heightbook** has the following constant/function/relation symbols:

- binary relation symbols **Friend** and **Taller** (standing for “is friends with” and “is taller than”),
- a unary function symbol **Crush** (everyone on Heightbook has a unique crush),
- and constant symbols **Alice** and **Bob** (two celebrities whom everyone knows by name).

The syntactic objects of this language — *terms* and *formulas* — are finite strings of symbols from the set

\[
\{\text{Friend, Taller, Crush, Alice, Bob}\} \cup \{=, (,), \neg, \lor, \forall\} \cup \{v_1, v_2, \ldots\}.
\]

(This set of symbols is the “alphabet” of the language.)

**Terms** have the inductive definition:

- Every variable symbol (viewed as a string of length 1) is a term.
- Constant symbols **Alice** and **Bob** (viewed as strings of length 1) are terms.
- If \( t \) is a term, then so is **Crush** \( t \).

Terms are the “nouns” of the language: they refer to elements in the universe of a structure (specific users in an implementation of the Heightbook platform). Note that every term has the form \( \text{Crush} \cdots \text{Crush} \sigma \) where \( \sigma \in \{\text{Alice, Bob, } v_1, v_2, \ldots\} \). (More complicated terms arise in language with function symbols of arity \( \geq 2 \).)

**Formulas** have the inductive definition:

- (**atomic formulas**) If \( t_1 \) and \( t_2 \) are terms, then \( = t_1 t_2 \) and **Friend** \( t_1 t_2 \) and **Taller** \( t_1 t_2 \) are formulas.
• (logical connectives) If $\alpha$ and $\beta$ are formulas, then so are $\neg(\alpha)$ and $(\alpha \lor \beta)$.
• (universal quantification) If $\alpha$ is a formula and $x$ is a variable symbol, then $(\forall x)(\alpha)$ is a formula.

Formulas are the “declarative statements” of the language: they are either true or false with respect to a particular structure and assignment of free variables. An example of a formula is the following string of 18 alphabet symbols:

$$(\neg(=\text{Alice Crush})v_5 \lor (\forall v_8)(\text{Friend Bob } v_8))$$

Of course, not every string of alphabet symbols is a term or formula. For example, $\text{Crush}(\forall \text{Alice } v_4)$ is neither a term nor a formula.

The official syntax of our first-order language is rather unpleasant to use. For readability sake, we employ various “syntactic sugar” when writing examples. We liberally use or omit parentheses (…) and square brackets [...]. We write $t_1 = t_2$ instead of $= t_1 t_2$. We use the following abbreviations:

$$(\exists x)(\alpha) :\equiv \neg[(\forall x)(\neg \alpha)] \text{ and } (\alpha \land \beta) :\equiv \neg(\neg \alpha \lor \neg \beta) \text{ and } (\alpha \rightarrow \beta) :\equiv (\neg \alpha \lor \beta).$$

Thus, we might informally write the above formula as

$$[\text{Alice } = \text{Crush}(v_5)] \rightarrow [\forall v_8 \text{Friend}(\text{Bob }, v_8)].$$

This formula expresses: “If Alice is the crush of $v_5$, then Bob is friends with everyone.” (Note that this formula has one free variable $v_5$.) Finally, we let $x, y, z$ stand for arbitrary distinct variable symbols, since this is often cleaner than writing $v_i, v_j, v_k$ where $i, j, k$ are distinct indices.

The axioms of Heightbook are the following first-order formulas:

• $(\forall x)(\forall y) \text{Friend}(x, y) \rightarrow \text{Friend}(y, x)$
  “Friendship is symmetric.”
• $(\forall x)(\forall y)(\forall z) [\text{Taller}(x, y) \land \text{Taller}(y, z)] \rightarrow \text{Taller}(x, z)$
  “The taller-than relation is transitive.”
• $(\forall x) \neg\text{Friend}(x, x) \land \neg\text{Taller}(x, x)$
  “Nobody is friends with or taller than themselves.”
• $\neg(\text{Alice } = \text{Bob})$
  “Alice and Bob are distinct people.”
Think of these axioms as the specification or design principles of the Heightbook platform, which every implementation must adhere to. Note that these axioms have no free variables, that is, they are *sentences*.

A **structure** $A$ in the language of Heightbook consists of a universe $A$ (a nonempty set of “users”) together with interpretations for the symbols of the language: binary relations $\text{Friend}^A \subseteq A \times A$ and $\text{Taller}^A \subseteq A \times A$, a unary function $\text{Crush}^A : A \to A$, and distinguished elements $\text{Alice}^A \in A$ and $\text{Bob}^A \in A$. A structure $A$ which satisfies the axioms of Heightbook is said to be *model of Heightbook*. Not every structure in the language of Heightbook is a model of Heightbook, for instance, any structure $A$ where $\text{Alice}^A = \text{Bob}^A$.

**Exercise.** Translate the following statements into sentences of our first-order language:

- “Bob is friends with everyone.”
- “All users with the same height are friends.”
- “Bob has exactly one friend.”
- “No one has more than two degrees of separation from Bob (i.e., Bob is either a friend or a friend-of-a-friend of everyone).”
- “Alice is friends with the shortest user (or shortest users in case of a tie).”
- “Everyone’s crush is taller than they are.”

Note that this statement, together with the axioms of Heightbook, logically implies that the number of users is infinite. It also implies the unfortunate formula $(\forall x) \neg[x = \text{Crush}(\text{Crush}(x))]$. What is the English translation of this formula?

Next, translate the following first-order sentences into English:

- $(\forall x)(\forall y)[\text{Crush}(x) = \text{Crush}(y)]$
  Note that this sentence logically implies $(\exists z)(\forall x)[\text{Crush}(x) = z]$.
- $(\forall x)[\text{Friend}(\text{Bob}, x) \rightarrow \text{Friend}(\text{Alice}, \text{Crush}(x))]$
- $\text{Friend}(\text{Bob}, \text{Crush}(\text{Crush}(\text{Bob})))$
- $(\forall x)[(x = \text{Crush}(x)) \rightarrow \neg(\exists y)(\text{Friends}(x, y))]$