# FOCS Homework 20

Edit your answers into this file, or add a separate file in the same directory. If you add a separate file, please include the following at the top:

```

Student Name: Leon Lam [change to your name]

Check one:

[ ] I completed this assignment without assistance or external resources.

[X] I completed this assignment with assistance from \_\_\_

and/or using these external resources: **The SQL join website**

```

## 1. Datalog

a. Consider a dialect of Datalog that includes `not`. [Many implementations include `not`. Racket Catalog does not.] What is value of the query in the following program?

```

likes(amy, susan).

likes(susan, amy).

likes(mark, amy).

unliked(X) :- not likes(Someone, X).

unliked(mark)?

```

**True. Datalog searches for a value of Someone where “not likes(Someone, mark) is true.**

b. How about after extending this program with the following fact?

```

likes(amy, mark).

unliked(mark)?

```

**Still true, I think.**

c. Is Datalog NP-Complete?

**Uh. I feel like it should be, since you’d potentially have to check all your data for an answer. There’s definitely a short-certificate thing in there, where you can point to a single fact and see that it matches.**

## 2. SQL

Read about SQL \*\*joins\*\*: inner join, left (outer) join, right (outer) join, and full (outer) join. Here are some explanations of joins:

\* [What Is a SQL Join](http://www.sql-join.com) and [Basic SQL Join Types](http://www.sql-join.com/sql-join-types) – Stitch Co.

\* [A Visual Explanation of SQL Joins](https://blog.codinghorror.com/a-visual-explanation-of-sql-joins/) – Jeff Atwood

Consider the tables `college` and `town`:

| college\_name | students | town\_name |

| ------------ | -------- | --------- |

| Babson | 2840 | Needham |

| Olin | 350 | Needham |

| Wellesley | 2474 | Wellesley |

| town\_name | population | settled\_date |

| --------- | ---------- | ------------ |

| Needham | 28,888 | 1680 |

| Dedham | 24,729 | 1635 |

What values does each of the following joins select?

All joins are on the column `town\_name` that is present in both tables. (That is, these are \*\*natural joins\*\*.)

a. An inner join of `college` and `town`.

**| Babson | 2840 | Needham |**  **28,888 | 1680 |**

**| Olin | 350 | Needham | 28,888 | 1680 |**

b. A left join of `college` and `town`.

**| Babson | 2840 | Needham |**  **28,888 | 1680 |**

**| Olin | 350 | Needham | 28,888 | 1680 |**

**| Wellesley | 2474 | Wellesley | NULL | NULL |**

c. A right join of `college` and `town`.

**| Babson | 2840 | Needham |**  **28,888 | 1680 |**

**| Olin | 350 | Needham | 28,888 | 1680 |**

**| NULL | NULL | Dedham | 24,729 | 1635 |**

d. A full join of `college` and `town`.

**| Babson | 2840 | Needham |**  **28,888 | 1680 |**

**| Olin | 350 | Needham | 28,888 | 1680 |**

**| Wellesley | 2474 | Wellesley | NULL | NULL |**

**| NULL | NULL | Dedham | 24,729 | 1635 |**

## 3. Prolog

Do chapter 1 of [Learn Prolog Now](http://lpn.swi-prolog.org/lpnpage.php?pageid=online). Note that this is the version with embedded exercises, which is not what a generic search turns up.

You do not have to turn in the exercises in 1.3 (or any others), but those are a good gauge of your understanding.

You are also welcome to download and install [SWI-Prolog](http://www.swi-prolog.org) or [gprolog](http://www.gprolog.org), but these are not required.

## 4. (Optional) Sorting

Using the descriptions of sorting algorithms in the PDF attached to [Day 15](https://sites.google.com/site/focs16fall/in-class-exercises/day-15-sorting-and-friends):

Write out the steps in sorting the following arrays with each of the algorithms. Specifically, show the state of the array after each pass through the outer loop. [NB: you may omit passes that either subdivide or concatenate sub-arrays without changing the order.]

`3 9 4 1 8 3 7 2`

`3 1 2 4 6 5 2 7`

`9 1 8 2 7 6 5 4`

`8 3 4 3 4 8 3 4`