Write a query evaluation engine for conjunctive queries with your favorite programming language. The command line syntax for invoking the query evaluation engine should be similar to the following (where we assume that the program is written in java).

```
java cq-evaluator <queryfile> <dbfile>
```

The above command returns the result of evaluating the conjunctive query in `<queryfile>` against the database in `<dbfile>`.

Example of the contents of a `<queryfile>`:

```
H(x,y) :- R(x,y,z), S(w,y,u).
```

Example of the contents of a `<dbfile>`:

```
R = { (1,2,3) (4,5,6) },
S = { (2,3,1) (1,3,3) (1,5,1) },
T = { },
U = { () },
V = { (1,2,8) (4,5,7) (1,1,1) (9,9,1) }.
```

Running `java cq-evaluator <queryfile> <dbfile>` produces `H(4,5)`. 
- **Hint on how to approach the problem:**
  1. Write a parser for conjunctive queries.
  2. Write a parser for the database format.
  3. Write a query evaluation module for conjunctive queries.
  
  For example, you may choose to use javacc grammar to write the grammars for (1) and (2). These grammars can then be compiled to generate the java programs that would parse a query and database in the `<queryfile>` and `<dbfile>` files respectively.

  For (3), the java code for cq-evaluator that you write should contain calls to the parsers to validate the format of the query and database in `<queryfile>` and `<dbfile>` respectively, before evaluating the query against the database.

- **Javacc reference page:**
  - [https://javacc.dev.java.net/](https://javacc.dev.java.net/)