Exercise 4.60.

By giving the query

```
(lives-near ?person (Hacker Alyssa P))
```

Alyssa P. Hacker is able to find people who live near her, with whom she can ride to work. On the other hand, when she tries to find all pairs of people who live near each other by querying

```
(lives-near ?person-1 ?person-2)
```

she notices that each pair of people who live near each other is listed twice; for example,

```
(lives-near (Hacker Alyssa P) (Fect Cy D))
(lives-near (Fect Cy D) (Hacker Alyssa P))
```

Why does this happen? Is there a way to find a list of people who live near each other, in which each pair appears only once? Explain.

Answer.

Alyssa's trouble arouse from the fact that `lives-near` matches a pair of symmetrical assertions twice while consulting the data base. A plausible strategy to list each pair only once is to stipulate a topological order among the employees. For example, we may assign each individual a unique ID:

```
(id (Warbuks Oliver) 001)
(id (Aull Dewitt) 002)
(id (Bitdiddle Ben) 011)
(id (Hacker Alyssa P) 012)
(id (Fect Cy D) 013)
(id (Tweakit Lem E) 014)
(id (Reasoner Louis) 015)
(id (Scrooge Eben) 021)
(id (Cratchet Robert) 022)
```

and use a query equipped with ID comparison to find a list of people who live near each other without duplicates

```
(and (lives-near ?person-1 ?person-2)
     (id ?person-1 ?id-1)
     (id ?person-2 ?id-2)
     (lisp-value > ?id-1 ?id-2))
```

and result in

```
(and (lives-near (reasoner louis) (aull dewitt))
     (id (reasoner louis) 15)
     (id (aull dewitt) 2)
     (lisp-value > 15 2))

(and (lives-near (reasoner louis) (bitdiddle ben))
     (id (reasoner louis) 15)
     (id (bitdiddle ben) 11)
     (lisp-value > 15 11))
```


Email address: informlarry@gmail.com

```
(and (lives-near (fect cy d) (hacker alyssa p))
     (id (fect cy d) 13)
     (id (hacker alyssa p) 12)
     (lisp-value > 13 12))

(and (lives-near (bitdiddle ben) (aull dewitt))
     (id (bitdiddle ben) 11)
     (id (aull dewitt) 2)
     (lisp-value > 11 2))
```