

### Exercise 10.1

Suppose we have the following database:

`p(1) .`

`p(2) :- ! .`

`p(3) .`

Write all of Prolog's answers to the following queries:

`?- p(X) .`

`X = 1`

`X = 2`

`?- p(X), p(Y) .`

`X = 1`

`Y = 1`

`X = 1`

`Y = 2`

`X = 2`

`Y = 1`

`X = 2`

`X = 2`

`?- p(X), !, p(Y) .`

`X = 1`

`Y = 1`

`X = 1`

`Y = 2`

### Exercise 10.2

First, explain what the following program does:

```

class(Number,positive) :- Number > 0.

class(0,zero).

class(Number,negative) :- Number < 0.

```

The program determines the class of a number (positive if greater than 0, zero if equal to 0, or negative if less than 0).

Second, improve it by adding green cuts.

```

class(Number,positive) :- Number > 0, !.

class(0,zero) :- !.

class(Number,negative) :- Number < 0, !.

```

### Exercise 10.3

Without using cut, write a predicate `split/3` that splits a list of integers into two lists: one containing the positive ones (and zero), the other containing the negative ones. For example:

```
split([3,4,-5,-1,0,4,-9],P,N)
```

should return:

```

P = [3,4,0,4]
N = [-5,-1,-9].

```

```

split([],[],[]).

split([X|Xs],[X|P],N) :- X >= 0, split(Xs,P,N).

split([X|Xs],P,[X|N]) :- X < 0, split(Xs,P,N).

```

Then improve this program, without changing its meaning, with the help of cut.

```

split([],[],[]) :- !.

split([X|Xs],[X|P],N) :- X >= 0, !, split(Xs,P,N).

split([X|Xs],P,[X|N]) :- X < 0, !, split(Xs,P,N).

```