Programming Paradigms

Lab 11. Lists and arithmetic in Prolog

Understanding `member`

Exercise 11.1(a)

Draw the search tree for the query `member(3, [X, 3, Y])`.

```
member(X, [X|_]).
member(X, [_|T]) :- member(X, T).
```

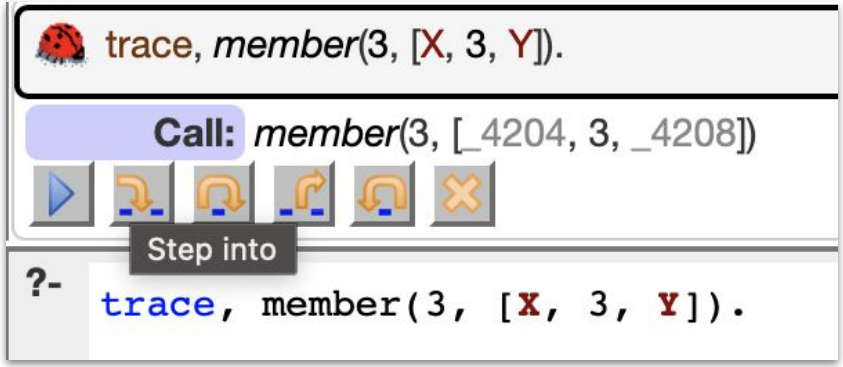
Understanding `member`

Exercise 11.1(b)

Verify your solution to Exercise 11.1(a):

Run the following query in SWISH and hit “Step into” button repeatedly.

```
?- trace, member(X, [X|_]).
```



List prefix

Exercise 11.2
Implement a predicate `prefix/2` that checks if a given list is a prefix of another list.

```
?- prefix([1, 2], [1, 2, 3, 4])
true

?- prefix(X, [1, 2, 3])
X = []
X = [1]
X = [1, 2]
X = [1, 2, 3]
```

Interleaving lists

Exercise 11.3
Implement a predicate `interleave/3` that checks if first two lists give the third when interleaved.

```
?- interleave([1, 2], [a, b], X)
X = [1, a, 2, b]

?- interleave([1, 2], [a, b, c], X)
X = [1, a, 2, b, c]

?- interleave(X, Y, [1, 2, 3, 4])
X = [1, 3], Y = [2, 4]
```

Squares

Exercise 11.4

Implement a predicate `squares/2` that checks
if the second list is a list of squares of numbers from the first list (in the same order).

```
?- squares([1, 2, 3, 4], X)
X = [1, 4, 9, 16]
```

Selecting elements from lists

Exercise 11.5

Implement a predicate `select/3` that splits a list into its element a list of remaining elements.

```
?- select([6, 3, 8, 5], X, R)
H = 6, R = [3, 8, 5]
H = 3, R = [6, 8, 5]
H = 8, R = [6, 3, 5]
H = 5, R = [6, 3, 8]
```

Permutations

Exercise 11.6

Implement a predicate `anagram/2` that checks whether a list is a permutation of another list.

```
?- anagram([d,o,r,m,i,t,o,r,y], [d,i,r,t,y, r,o,o,m])
true
```

```
?- anagram([1, 2], X)
X = [1, 2]
X = [2, 1]
```

A logic puzzle

Exercise 11.7
There is a street with three neighbouring houses that all have a different colour, namely **red**, **blue**, and **green**. People of different nationalities live in the different houses and they all have a different pet. Here are some more facts about them:

- The Englishman lives in the **red** house.
- The jaguar is the pet of the Spanish family.
- The Japanese lives to the right of the snail keeper.
- The snail keeper lives to the left of the **blue** house.

Who keeps the zebra? Don't work it out for yourself:
define a predicate `zebra/1` that tells you the nationality of the owner of the zebra!

Hint: Think of a representation for the houses and the street.
Code the four constraints in Prolog. You may find `member/2` and `sublist/2` useful.

Sorted lists

Exercise 11.8

Implement a predicate `sorted/1` that checks if a list is sorted.

```
?- sorted([6, 3, 8, 5])
false
```

```
?- sorted([3, 5, 6, 8])
true
```

Sorted lists

Exercise 11.9

Implement a predicate `sort/2` that checks
if the second list is a sorted version of the first one.

```
?- sort([6, 3, 8, 5], X)
X = [3, 5, 6, 8]
```

Shuffling both ways

Exercise 11.10 (***)

Examine current implementation of `shuffled/2`. Explain why it's not working in one direction. Update the implementation of `shuffled/2` to work in both directions:

```
?- shuffled([1, 2], X)
X = [1, 2]
X = [2, 1]

?- shuffled(X, [1, 2])
X = [1, 2]
X = [2, 1]
```