

# Homework # 10

## due Tuesday, April 12, 3:30 PM

### 1 Parameters

Do exercise 6 on page 384 by completing the Java program `ParameterTest.java` from a skeleton in `$CLASSHOME/src/homework10/`: but skip 6(d) (macro expansion). Don't hard-code the values expected into the parameter classes!

Answer the following questions in a file `README`:

- There is more than one possibility for value-result parameters. Why?
- Why does it not make sense to implement macro-expansion in the test?

### 2 Prolog Programs

- Write a Prolog predicate `insert` that places an element somewhere in a list:

```
?- insert(A,[B,C,D],L).
L = [A, B, C, D] ;
L = [B, A, C, D] ;
L = [B, C, A, D] ;
L = [B, C, D, A] ;
No
```

- Write a Prolog predicate `permute` that returns true if the first list is a permutation of the second:

```
?- permute([a,b,c],[a,b,c]).
Yes
?- permute([a,b,c],[b,c,a]).
Yes
?- permute([a,b,c],[a,b,a,c]).
No
?- permute([a,b,c],[c,b]).
No
```

The `permute` predicate should use `insert` as well as being recursive.

- Write a 2x2 Sudoku solver. (If you don't know Sudoku, please look it up.)

```
?- X=[[1,_,_,_],
|     [_,2,1,_,_],
|     [_,_,3,_,_],
|     [_,_,_,4]], sudoku2(X).

X = [[1, 3, 4, 2], [4, 2, 1, 3], [2, 4, 3, 1], [3, 1, 2, 4]] ;

No
```

(Here the `No` shows that the solution given is unique.)

The sudoku predicate should be defined entirely using the `permute` predicate.

The code for all predicates should go into `homework10.pl` in your `homework10` directory of your AFS volume.