[illegible]
[illegible]
[illegible] [illegible] (Prolog)
[illegible]: [illegible] 2014-15
[illegible]: [illegible]

[illegible] 6:
[illegible] (2)

[illegible] append

[illegible] append [illegible]

```
append([], L, L).
append([H|L1], L2, [H|L3]) :-
                    append(L1, L2, L3).
```

[illegible] Prolog. ([illegible] SWI-Prolog [illegible], [illegible] myappend).

[illegible] append() [illegible]

[illegible] (!!!) [illegible] []. [illegible] L [illegible] L.

[illegible] H [illegible] L1 [illegible] L2 [illegible] H [illegible] L1 [illegible] L2 ([illegible] append).

[illegible] Prolog [illegible]:

```
?- append([a, b], [c,d], L).
```

1. [illegible]

```
append([H|L1], L2, [H|L3]) :-
      append(L1, L2, L3).
```

[illegible]: H=a, L1=[b], L2=[c,d].

[illegible]:

```
append(L1, L2, L3)
```

[illegible] ([illegible]) :

```
?- append([b], [c,d], L3).
```

2. [illegible]

```
append([H|L1], L2, [H|L3]) :-
      append(L1, L2, L3).
```

[illegible]): H=b, L1=[], L2=[c,d].

[illegible]:

```
append(L1, L2, L3).
```

[illegible]

```
?- append([], [c,d], L3).
```

3. [illegible]

```
append([], L, L).
```

[illegible] L=[c,d].

4. [illegible] (back-substitution):

To `L3` [illegible] 2 [illegible] `L` [illegible] 3 [illegible]

    L3([illegible]-2) = [c, d]

To `L3` [illegible] 1 [illegible] `[H|L3]` [illegible] 2 [illegible]

    L3([illegible]-1) = [b, c, d]

To `L` [illegible] `[H|L3]` [illegible] 1 [illegible]

    L = [a, b, c, d]

[illegible] append.

[illegible] append. [illegible] append [illegible]. [illegible] Prolog [illegible] (append) [illegible] append) [illegible] append) [illegible].

```
?- append(A,B,[a,b,c]).
A = []
B = [a,b,c];

A = [a]
B = [b,c];

A = [a,b]
B = [c];

A = [a,b,c]
B = [];

No
```

[illegible] A, B [illegible] [a,b,c]?

[illegible]:
[illegible] append ([illegible] [a,b,c]) [illegible] 2 [illegible].

[illegible] reverse()

[illegible] (=[illegible]) [illegible] **(back-substitution)** [illegible].

[illegible] `reverse(L,L1)` [illegible] L [illegible] L1, [illegible]:

```
?- reverse([a,b,c],L).
L = [c,b,a]
```

[illegible]:

```
reverse([],[]).

reverse([H|T],L) :-
    reverse(T,L1),
    append(L1,[H],L).
```

[illegible]: [illegible].

[illegible]: [illegible] L [illegible] reverse [illegible] [H|T]. [illegible]

[illegible] reverse [illegible] T [illegible] L1[illegible]

[illegible] L [illegible] L1 [illegible] H. [illegible] append [illegible].

[illegible]

```
?-reverse([a,b,c],LA).
```

[illegible] (¢=a, ¢=[b,c], LA=L)

```
(1) ?-reverse([b,c],L1),append(L1,[a],LA).
```

[illegible] L1 [illegible] (¢'=b, ¢'=[c], L1=L' [illegible] ¢,¢ [illegible] L). [illegible]

```
(2) ?-reverse([c],L1'), append(L1',[b],L1), append(L1,[a],LA).
```

[illegible] L1' [illegible]

```
(3)?-reverse([],L1''), append(L1'',[c],L1'),append(L1',[b],L1), append(L1,[a],LA).
```

[illegible] **L1''=[].**

[illegible]

- [illegible] reverse [illegible] [] -> [] [illegible] append([],[c],L1') [illegible] L1'=[c].
- [illegible] reverse [illegible] [c] -> [c] [illegible] append([c],[b],L1) το L1=[c,b].
- [illegible] reverse [illegible] [b,c] -> [c,b] [illegible] append([c,b],[a],LA) το LA=[c,b,α].

[illegible]:

[illegible] not/1

[illegible] not() [illegible]

```
?- not(member(2,[1,2,3])).
No

?- not(member(4,[1,2,3])).
Yes
```

[illegible] atomic/1

[illegible] atomic/1 [illegible] Prolog [illegible]

```
?- atomic(a).        ?- atomic(2).
Yes                  Yes

?- atomic([1,2]).    ?- atomic(X).
No                   No
```

[illegible] _ ([illegible] don't care)

[illegible] _ [illegible] X, [illegible] _ [illegible]

```
member(X,[X|_]).
```

[illegible] [H|T] [illegible] SWI-Prolog [illegible] T [illegible]

```
[WARNING: ([illegible])
      Singleton variables: T]
```

[illegible] an;vnymhw [illegible] _ (underscore).

[illegible]

(1) [illegible] **posneg(L,LP,LN)** [illegible]

```
?- posneg([13,-51,-11,29], LP,LN).
LP=[13,29] LN=[-51,-11]
?- posneg([1, 3, 21], LP,LN).
LP=[1,3,21] LN=[]
```

(2) [illegible] **sumlist(L,X)** [illegible]

```
?- sumlist([[2,3],[1,s,d,a],[3]], X).
X=7
?-sumlist([[],[1,3,2],[a,d,s]], X).
X=6
```

(3) [illegible] **enwsh(L1,L2,L)** [illegible] (?). [illegible]

```
?- enwsh([3,5,9,11],[4,3,10,9], L).
L=[5,11,4,3,10,9]
?- enwsh([c,e,f,a,w],[a,b,c,d], X).
X=[e,f,w,a,b,c,d]
```

(4) [illegible] **flat(L1,L2)** [illegible] L2 [illegible] L1: [illegible] L1 [illegible] L2 [illegible] L1 [illegible]

```
?- flat([[a, e], [[[b], c]]], L).
L2 = [a, e, b, c]
?- flat([a, [[b,c], [[d,e], f]], g], L2).
L2 = [a, b, c, d, e, f, g]
```

[illegible]: [illegible] atomic/1.

(5) [illegible] **memberlist(X,L)** [illegible] X [illegible] L. [illegible]

```
?- memberlist([a, [[b,c], [d,e], [f,2,a,3]]).
Yes
?- memberlist([4, [[b,c], [[w]], [f,2,a,3]]).
No
```