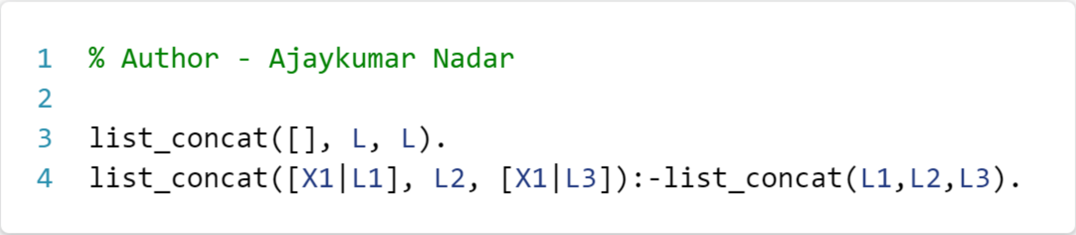
Code:

```
Welcome to SWI-Prolog (threaded, 64 bits, version 9.0.4)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

1 ?- members(x,[x,y,z]).
ERROR: Unknown procedure: members/2 (DWIM could not correct goal)
2 ?- member(x,[x,y,z]).
true ;
false.

3 ?- member(p,[x,y,z]).
false.

4 ?- member(my(x,y,z),[q,r,s,my(x,y,z),w]).
true ;
false.

5 ?- member(x,[x,y,z]).
true ;
false.

6 ?- member(v, []).
false.

7 ?- length([a,b],M).
M = 2.

8 ?- length([1,2,3], M).
M = 3.

9 ?- length([1,2,3],a).
ERROR: Type error: `integer' expected, found `a' (an atom)
ERROR: In:
ERROR:   [11] throw(error(type_error(integer,a),context(...,_162)))
ERROR:    [9] toplevel_call(user:user: ...) at c:/program
files/swipl/boot/toplevel.pl:1173
ERROR:
ERROR: Note: some frames are missing due to last-call optimization.
ERROR: Re-run your program in debug mode (:- debug.) to get more detail.
10 ?- length([1,2,3],X1).
X1 = 3.

11 ?- length([1,2,3],X-1).
ERROR: Type error: `integer' expected, found `_1008-1' (a compound)
ERROR: In:
ERROR:   [11] throw(error(type_error(integer,...),context(...,_1072)))
ERROR:    [9] toplevel_call(user:user: ...) at c:/program
files/swipl/boot/toplevel.pl:1173
ERROR:
```

```
ERROR: Note: some frames are missing due to last-call optimization.
ERROR: Re-run your program in debug mode (:- debug.) to get more detail.
12 ?- length([[a,c],[e,f],[h,i]],N).
N = 3.

13 ?- length([], P).
P = 0.

14 ?- length([a,b,c], 3).
true.

15 ?- reverse([1,2,3], A).
A = [3, 2, 1].

16 ?- length([1,2,3], 3).
true.

17 ?- reverse(B, [1,2,3]).
B = [3, 2, 1].

18 ?- reverse([[dog, cat], [1,2], [bird, mousse]], L).
L = [[bird, mousse], [1, 2], [dog, cat]].

19 ?- reverse([1,2,3,4],[4,5,6,8]).
false.

20 ?- reverse([1,2,3,4],[4,3,2,1]).
true.

21 ?- append([], [1,2,3],L).
L = [1, 2, 3].

22 ?- append([a,b],[1,2,3],L).
L = [a, b, 1, 2, 3].

23 ?- append([a,b,23],[1,2,3],L).
L = [a, b, 23, 1, 2, 3].
```

2. Write a program in Prolog to concatenate two lists.

Code:



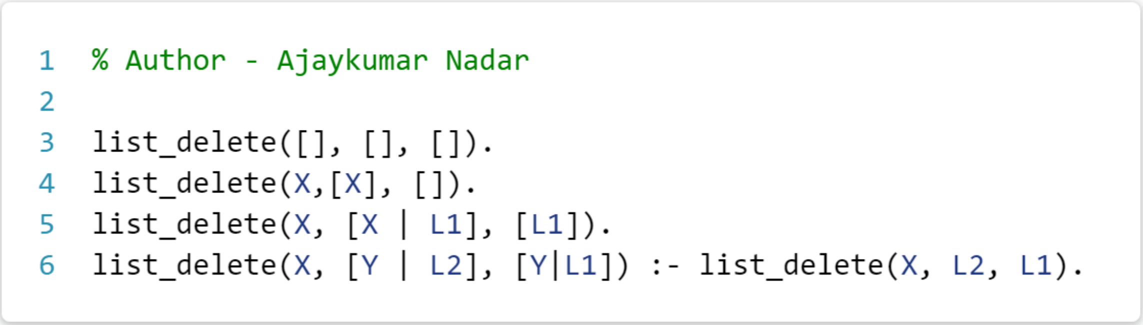
```prolog
% Author - Ajaykumar Nadar

list_concat([], L, L).
list_concat([X1|L1], L2, [X1|L3]):-list_concat(L1,L2,L3).
```

Output:

```
Welcome to SWI-Prolog (threaded, 64 bits, version 9.0.4)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

1 ?- [prolog2].
true.

2 ?- list_concat([1,2,3],[a,b,c],X).
X = [1, 2, 3, a, b, c].

3 ?- list_concat([1,2,3],[],X).
X = [1, 2, 3].

4 ?- list_concat([],[a,b,c],X).
X = [a, b, c].
```

3. Write a program in Prolog to delete an item from a given list.

Code:



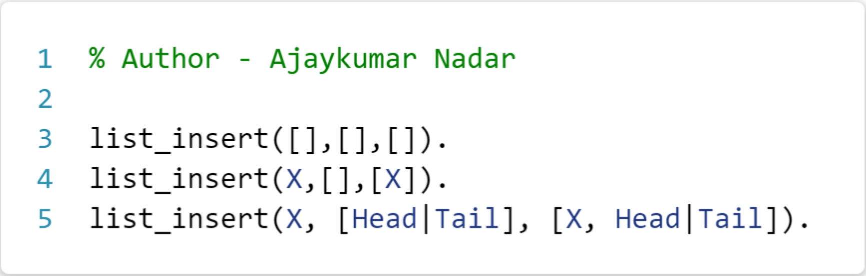
```
% Author - Ajaykumar Nadar

list_delete([], [], []).
list_delete(X,[X], []).
list_delete(X, [X | L1], [L1]).
list_delete(X, [Y | L2], [Y|L1]) :- list_delete(X, L2, L1).
```

Output:

```
Welcome to SWI-Prolog (threaded, 64 bits, version 9.0.4)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

1 ?- [prolog3].
true.

2 ?- list_delete(b, [b,c,d], NewList).
NewList = [[c, d]] ;
false.

3 ?- list_delete([],[], NewList).
NewList = [] ;
false.

4 ?- list_delete(a,[a], NewList).
NewList = [] ;
NewList = [[]] ;
false.

5 ?- list_delete(b, [a,b,c,d], NewList).
NewList = [a, [c, d]] ;
false.
```

4. Write a program in Prolog to insert an item in a given list.

Code:



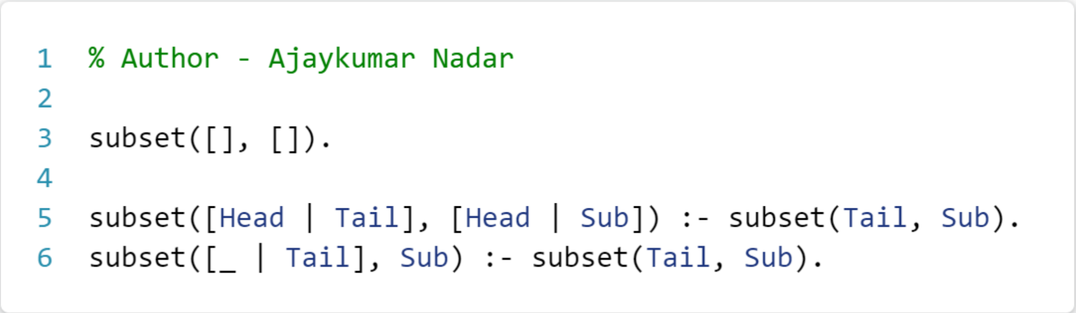
```
% Author - Ajaykumar Nadar

list_insert([],[],[]).
list_insert(X,[],[X]).
list_insert(X, [Head|Tail], [X, Head|Tail]).
```

Output:

```
Welcome to SWI-Prolog (threaded, 64 bits, version 9.0.4)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

1 ?- [prolog4].
true.

2 ?- list_insert(5, [], X).
X = [5] .

3 ?- list_insert(5, [4,6,7], X).
X = [5, 4, 6, 7].
```

Write a program in Prolog to find all possible subsets of a given list.

Code:



```prolog
% Author - Ajaykumar Nadar

subset([], []).

subset([Head | Tail], [Head | Sub]) :- subset(Tail, Sub).
subset([_ | Tail], Sub) :- subset(Tail, Sub).
```

Output:

```
Welcome to SWI-Prolog (threaded, 64 bits, version 9.0.4)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

1 ?- [subset].
true.

2 ?- subset([1,2,3],X).
X = [1, 2] ;
X = [1, 3] ;
X = [1] ;
X = [2, 3] ;
X = [2] ;
X = [3] ;
X = [].
```