

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% CS_125 Logic Programming
% Spring 2002
% Solutions to Exercises 2
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Question 1.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```
pythagoras(A,B,X) :- X is A*A + B*B.
```

```
% ?- pythagoras(3,4,P).
```

```
% P = 25 ;
% No
```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Question 2.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```
nat_init(N,L) :- interval(0,N,L).
```

```
% interval(K,N,L) computes for given numbers
% K <= N, the list L = [K,K+1,...,N-1,N]
```

```
interval(N,N,[N]).
interval(K,N,[K|L]) :-
  K < N, K1 is K+1,
  interval(K1,N,L).
```

```
% ?- nat_init(5,L).
```

```
% L = [0, 1, 2, 3, 4, 5] ;
% No
```

```
% Alternative solution
```

```
nat_init1(N,L) :-
  nat_init_rev(N,L1),
  reverse(L1,L).
```

```
% nat_ini_rev(N,L) computes for a given number
% N, the list L = [N,N-1,...,1,0]
```

```
nat_init_rev(0,[0]).
nat_init_rev(N,[N|L]) :-
  N > 0, N1 is N-1,
  nat_init_rev(N1,L).
```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Question 3.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```
sum_list([],0).
sum_list([H|T],N) :-
  sum_list(T,M), N is H+M.
```

```
% ?- sum_list([3,4,1],N).
```

```
% N = 8 ;
% No
```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Question 4.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```
fib(1,1).
fib(2,1).
```

```
fib(N,F) :-
  N > 2, N1 is N-1, N2 is N-2,
  fib(N1,F1), fib(N2,F2),
  F is F1+F2.
```

```
% ?- fib(20,N).
```

```
% N = 6765 ; (takes a few seconds)
% No
```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Question 5.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```
fib1(N,F) :- fib_pair(N,F,_).
```

```
% fib_pair(N,F,G) :<=> F and G are the Nth, and
% N+1st Fibonacci numbers, respectively.
```

```
fib_pair(1,1,1).
```

```
fib_pair(N,G1,G) :-
  N > 1, N1 is N-1,
  fib_pair(N1,F1,G1),
  G is F1+G1.
```

```
% ?- fib1(40,N).
```

```
% N = 102334155 ; (instantly)
% No
```

p.t.o.

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Question 6.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

sublist(S,L) :-
  append(_,L1,L),
  append(S,_,L1).

```

```

write_sums_of_sublists(L) :-
  sublist(S,L), sum_list(S,N),
  write(N), write(' '), write(S), nl,
  fail.

```

```

write_sums_of_sublists(_).

```

```

% ?- write_sums_of_sublists([1,2,3]).
% 0 []
% 1 [1]
% 3 [1, 2]
% 6 [1, 2, 3]
% 2 [2]
% 5 [2, 3]
% 3 [3]

```

```

% Yes

```

```

% Remark: Using the definition of sublist
% above, every sublist ist found exactly
% once.

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Question 7.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

quicksort([], []).

```

```

quicksort([X|Tail], Sorted) :-
  split(X, Tail, Small, Big),
  quicksort(Small, Sortedsmall),
  quicksort(Big, Sortedbig),
  append(Sortedsmall, [X|Sortedbig], Sorted).

```

```

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

```

```

split(X, [Y|Tail], [Y|Small], Big) :-
  X > Y, !,
  split(X, Tail, Small, Big).

```

```

split(X, [Y|Tail], Small, [Y|Big]) :-
  split(X, Tail, Small, Big).

```

```

% ?- quicksort([4,2,5,1,7,4],Sorted).

```

```

% Sorted = [1, 2, 4, 4, 5, 7] ;
% No

```