a) L1 is off & L2 is on

?- illuminated(l2); not(illuminated(L1)).

not(illuminated(L1)),

illuminated(l2).

light(l2), ok(l2), continuous(l2, outside).

ok(l2), continous(l2, outside).

continuous(l2, outside).

connected_to(l2, w4), continous(w4, outside).

continous(w4, outside).

connected_to(w4, w3), continous(w3, outside).

up(s3), continous(w3, outside).

continous(w3, outside).

connected_to(w3, w5), continous(w5, outside).

ok(cb1), continous(w5, outside).

continous(w5, outside).

connected_to(w5, outside).

true.

b) both L1 & L2 are on?

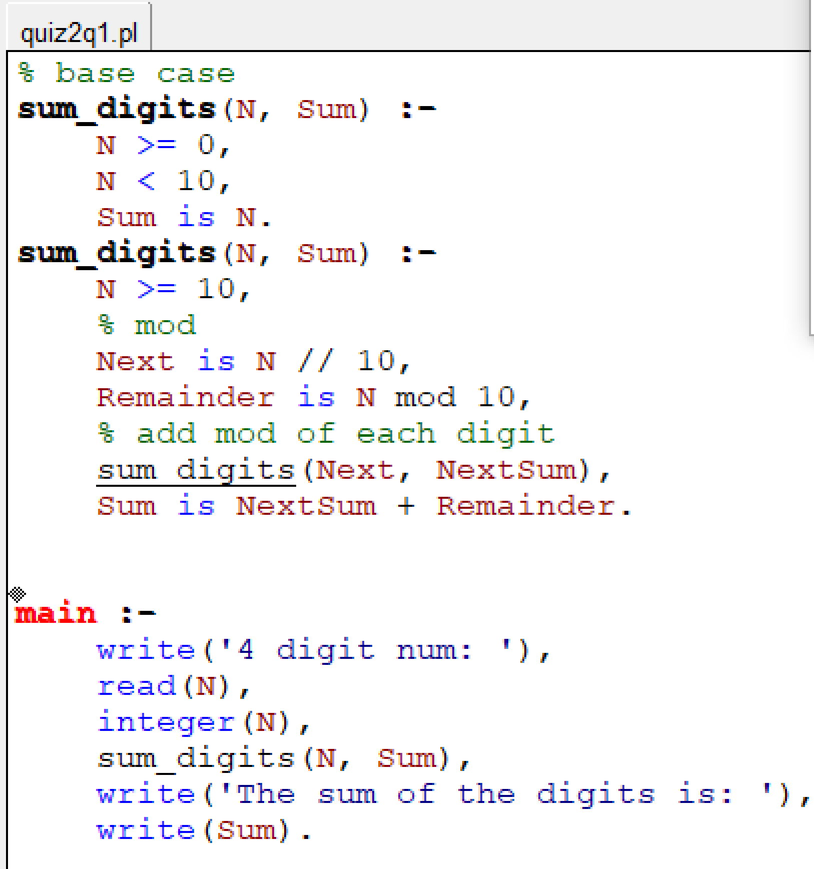
```
?- illuminated(L1); not(illuminated(L2)).
not(illuminated(L2)).
illuminated(L1).
light(L1), OK(L1), continous(L1, outside).
Continous(L1, outside).
Connected_to(L1, W0), continous(W0, outside).
Continous(W0, outside).
Connected_to(W0, W1), continous(W1, outside);
Connected_to(W0, W2), continous(W2, outside).
uP(S1), down(S2), continous(W3, outside);
down(S1), uP(S2), continous(W3, outside).
Continous(W3, outside).
Connected_to(W3, W5), continous(W5, outside).
OK(Cb1), Continous(W5, outside).
Continous(W5, outside).
Connected_to(W5, outside).
true.
```

c) both L1 & L2 off

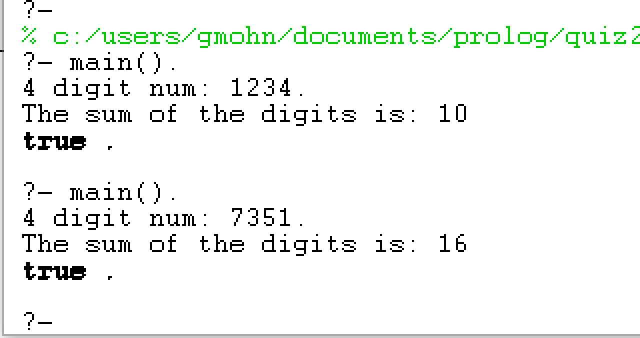
```
?- not(illuminated(L1)); not(illuminated(L2)).
not(illuminated(L1)).
not(illuminated(L2)).
```

7

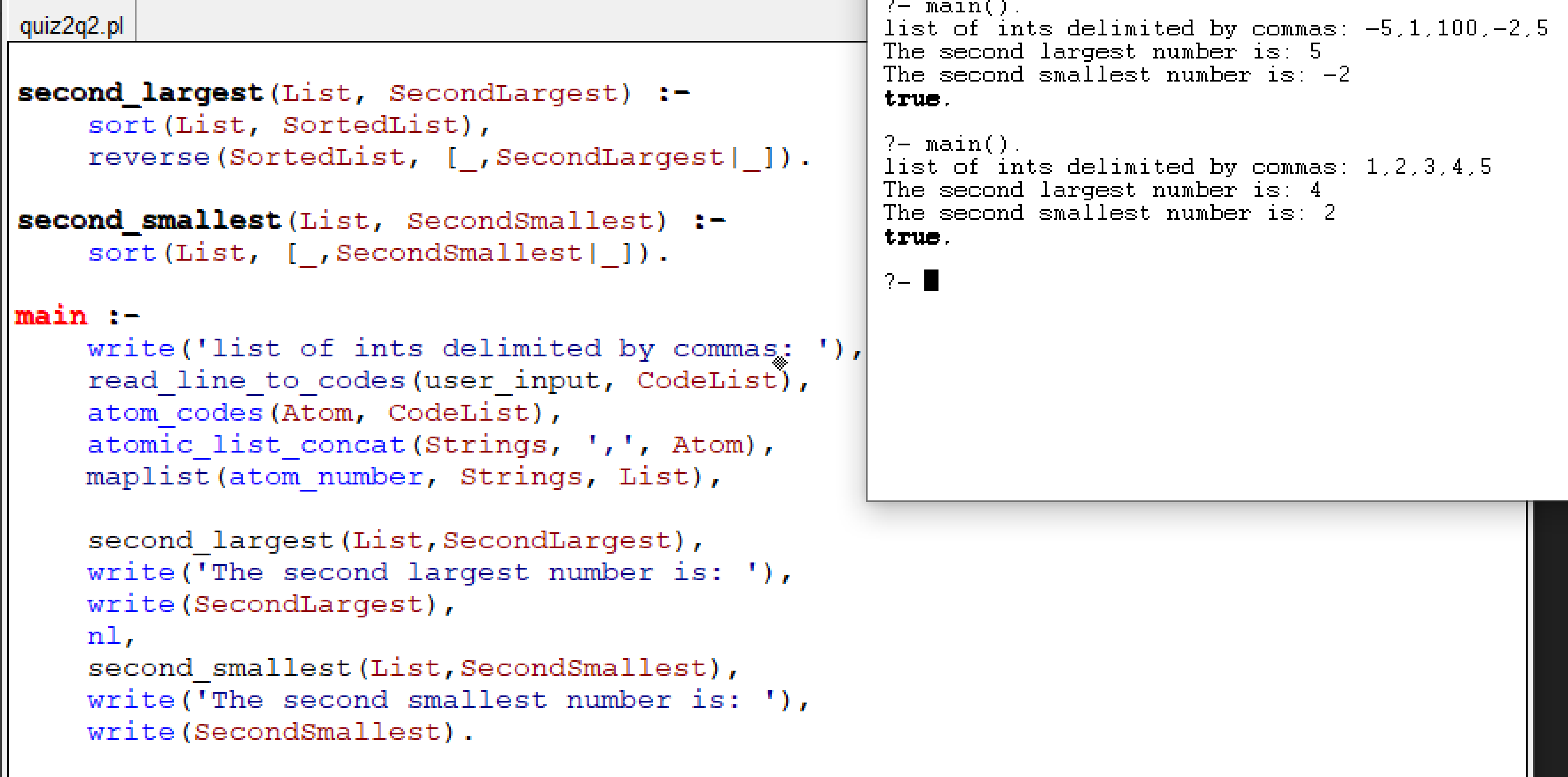
```
minidoku(L) :- [X1, X2, X3, X4] = L,
        worthy([X1, X2, X3, X4]).
worthy(L) :- valid(L), diff(L).
validval(1).
validval(2).
validval(3).
validval(4).
valid([H]) :- validval(H).
valid([H|T]) :- validval(H), valid(T).
diff([H]).
diff([H|T]) :- not(member(H,T)),
        diff(T).
member(H, [H|T]).
member(X, [H|T]) :- member(X,T).
```


quiz2q1.pl

```prolog
% base case
sum_digits(N, Sum) :-
    N >= 0,
    N < 10,
    Sum is N.
sum_digits(N, Sum) :-
    N >= 10,
    % mod
    Next is N // 10,
    Remainder is N mod 10,
    % add mod of each digit
    sum_digits(Next, NextSum),
    Sum is NextSum + Remainder.


main :-
    write('4 digit num: '),
    read(N),
    integer(N),
    sum_digits(N, Sum),
    write('The sum of the digits is: '),
    write(Sum).
```

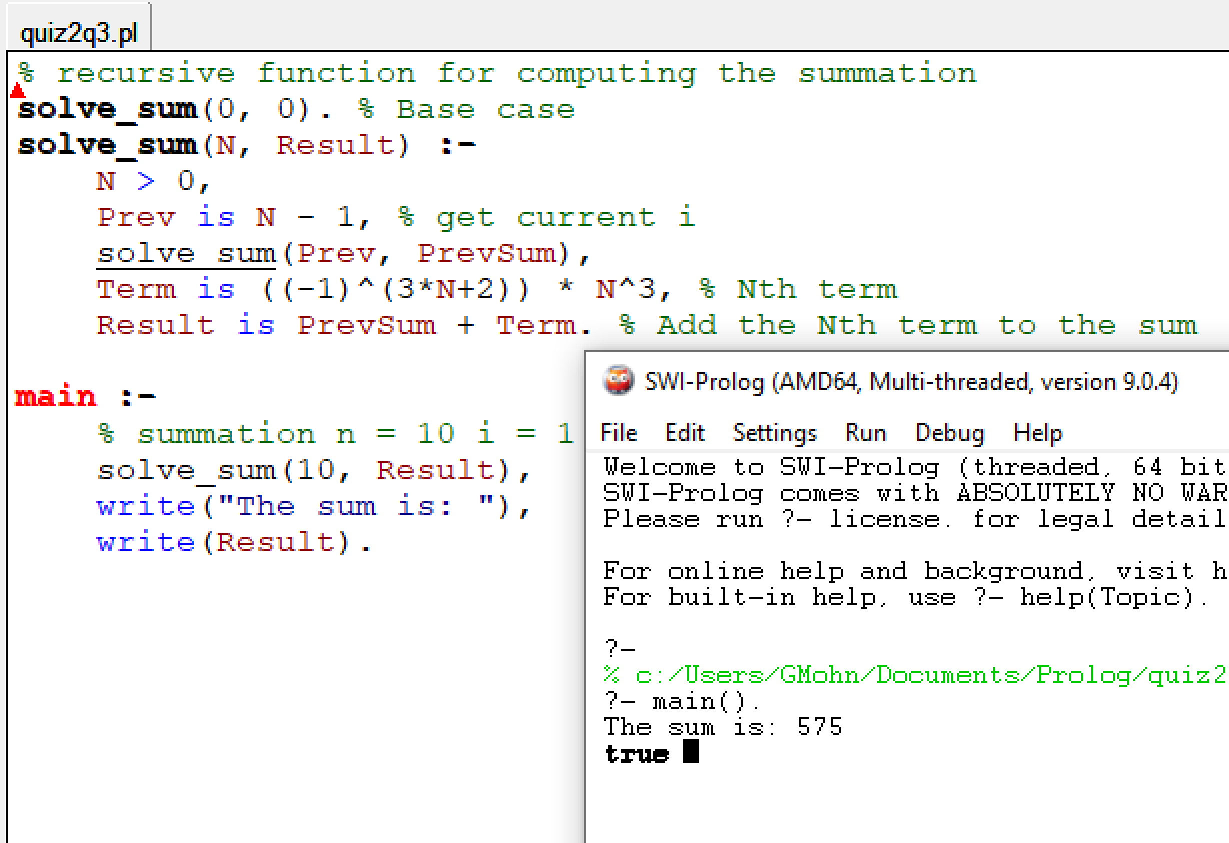


```
?-
% c:/users/gmohn/documents/prolog/quiz2
?- main().
4 digit num: 1234.
The sum of the digits is: 10
true .

?- main().
4 digit num: 7351.
The sum of the digits is: 16
true .

?-
```


quiz2q2.pl

```prolog
second_largest(List, SecondLargest) :-
    sort(List, SortedList),
    reverse(SortedList, [_,SecondLargest|_]).

second_smallest(List, SecondSmallest) :-
    sort(List, [_,SecondSmallest|_]).

main :-
    write('list of ints delimited by commas: '),
    read_line_to_codes(user_input, CodeList),
    atom_codes(Atom, CodeList),
    atomic_list_concat(Strings, ',', Atom),
    maplist(atom_number, Strings, List),

    second_largest(List,SecondLargest),
    write('The second largest number is: '),
    write(SecondLargest),
    nl,
    second_smallest(List,SecondSmallest),
    write('The second smallest number is: '),
    write(SecondSmallest).
```

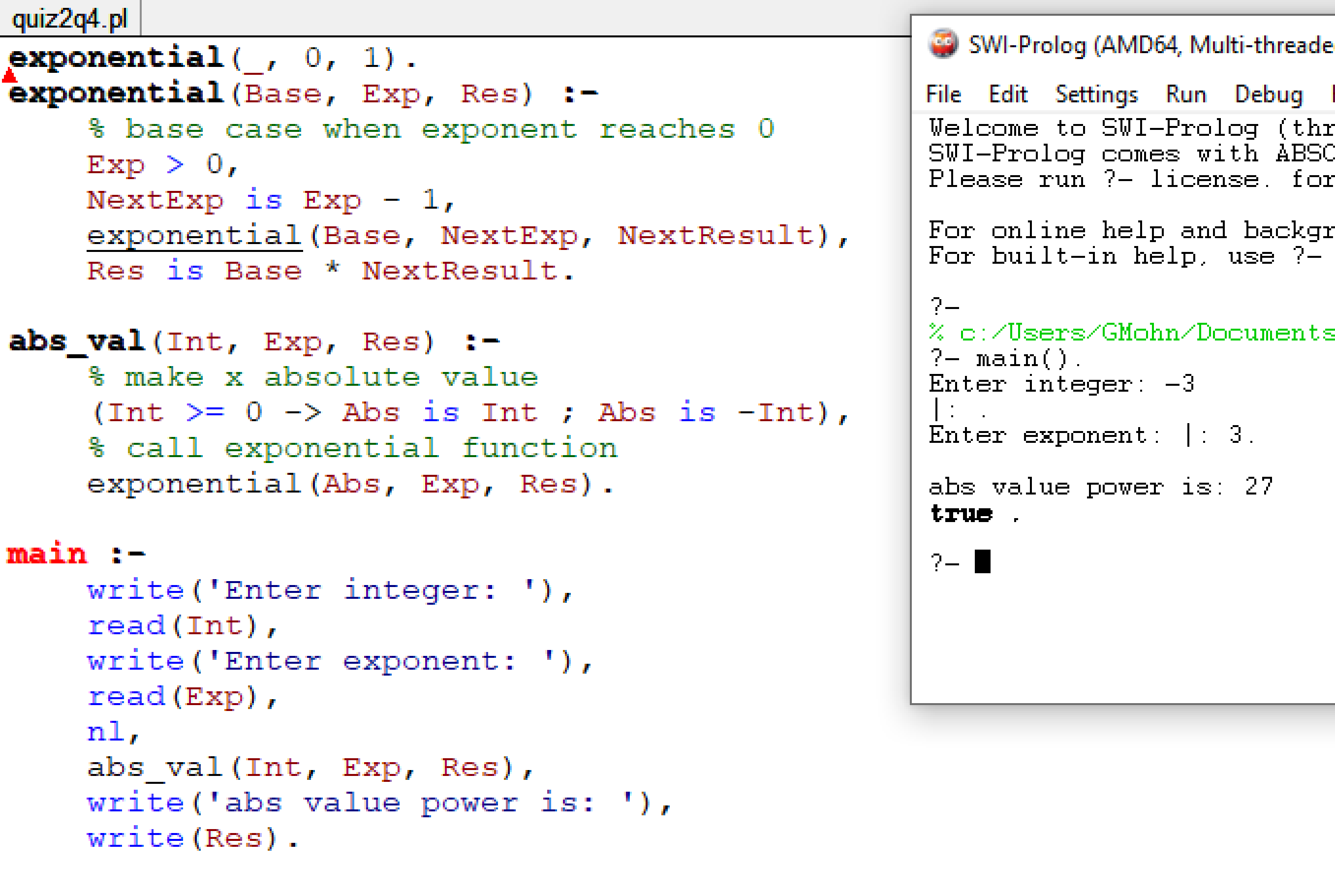
```
?- main().
list of ints delimited by commas: -5,1,100,-2,5
The second largest number is: 5
The second smallest number is: -2
true.

?- main().
list of ints delimited by commas: 1,2,3,4,5
The second largest number is: 4
The second smallest number is: 2
true.

?- 
```



```
% recursive function for computing the summation
solve_sum(0, 0). % Base case
solve_sum(N, Result) :-
    N > 0,
    Prev is N - 1, % get current i
    solve_sum(Prev, PrevSum),
    Term is ((-1)^(3*N+2)) * N^3, % Nth term
    Result is PrevSum + Term. % Add the Nth term to the sum

main :-
    % summation n = 10 i = 1
    solve_sum(10, Result),
    write("The sum is: "),
    write(Result).
```

SWI-Prolog (AMD64, Multi-threaded, version 9.0.4)
File Edit Settings Run Debug Help

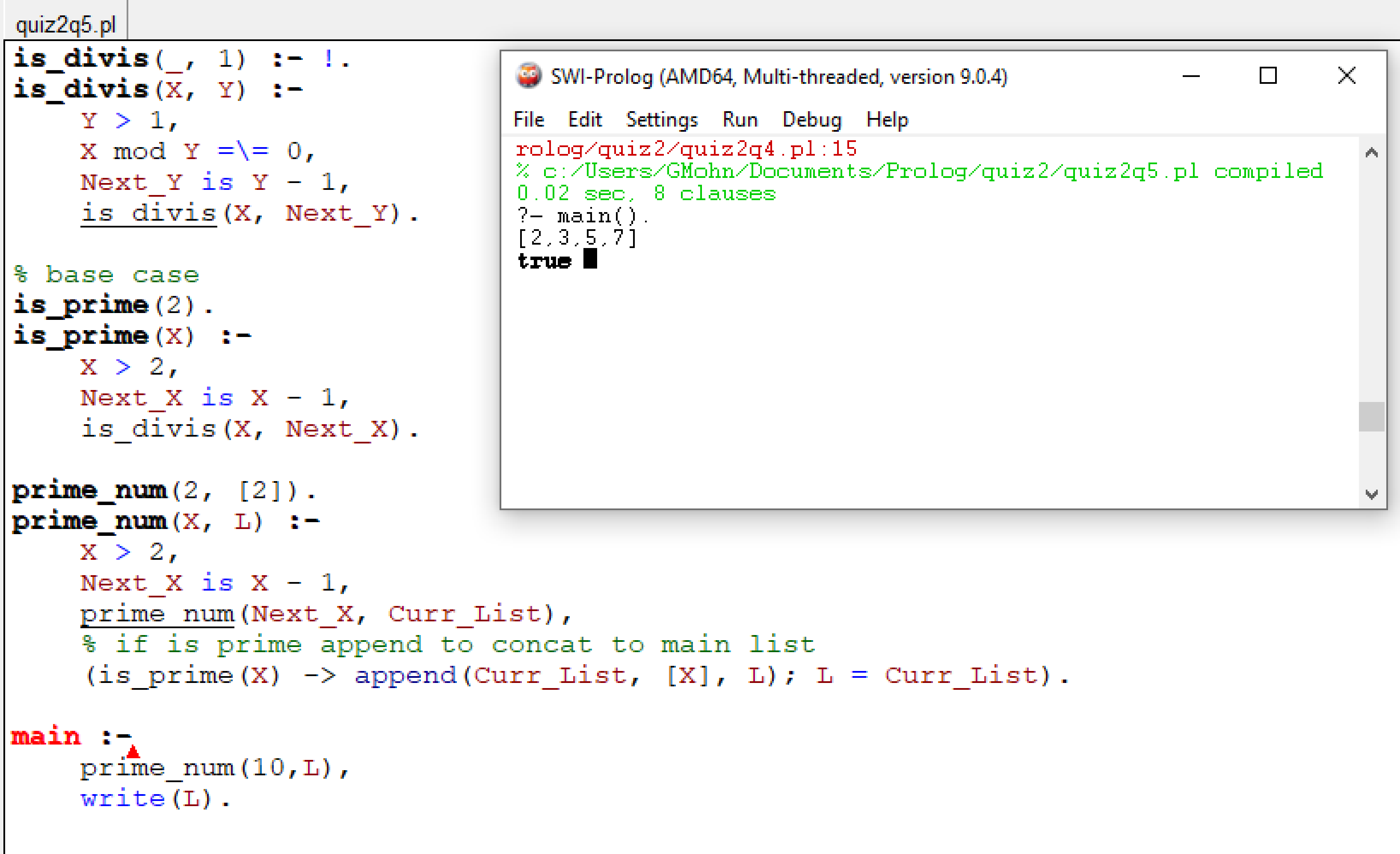
```
Welcome to SWI-Prolog (threaded, 64 bit
SWI-Prolog comes with ABSOLUTELY NO WAR
Please run ?- license. for legal detail

For online help and background, visit h
For built-in help, use ?- help(Topic).

?-
% c:/Users/GMohn/Documents/Prolog/quiz2
?- main().
The sum is: 575
true
```



```
exponential(_, 0, 1).
exponential(Base, Exp, Res) :-
    % base case when exponent reaches 0
    Exp > 0,
    NextExp is Exp - 1,
    exponential(Base, NextExp, NextResult),
    Res is Base * NextResult.

abs_val(Int, Exp, Res) :-
    % make x absolute value
    (Int >= 0 -> Abs is Int ; Abs is -Int),
    % call exponential function
    exponential(Abs, Exp, Res).

main :-
    write('Enter integer: '),
    read(Int),
    write('Enter exponent: '),
    read(Exp),
    nl,
    abs_val(Int, Exp, Res),
    write('abs value power is: '),
    write(Res).
```

SWI-Prolog (AMD64, Multi-threade
File Edit Settings Run Debug

```
Welcome to SWI-Prolog (thr
SWI-Prolog comes with ABSC
Please run ?- license. for

For online help and backgr
For built-in help, use ?-

?-
% c:/Users/GMohn/Documents
?- main().
Enter integer: -3
|: .
Enter exponent: |: 3.

abs value power is: 27
true .

?- 
```



```prolog
is_divis(_, 1) :- !.
is_divis(X, Y) :-
    Y > 1,
    X mod Y =\= 0,
    Next_Y is Y - 1,
    is_divis(X, Next_Y).

% base case
is_prime(2).
is_prime(X) :-
    X > 2,
    Next_X is X - 1,
    is_divis(X, Next_X).

prime_num(2, [2]).
prime_num(X, L) :-
    X > 2,
    Next_X is X - 1,
    prime_num(Next_X, Curr_List),
    % if is prime append to concat to main list
    (is_prime(X) -> append(Curr_List, [X], L); L = Curr_List).

main :-
    prime_num(10,L),
    write(L).
```

```
rolog/quiz2/quiz2q4.pl:15
% c:/Users/GMohn/Documents/Prolog/quiz2/quiz2q5.pl compiled
0.02 sec, 8 clauses
?- main().
[2,3,5,7]
true
```