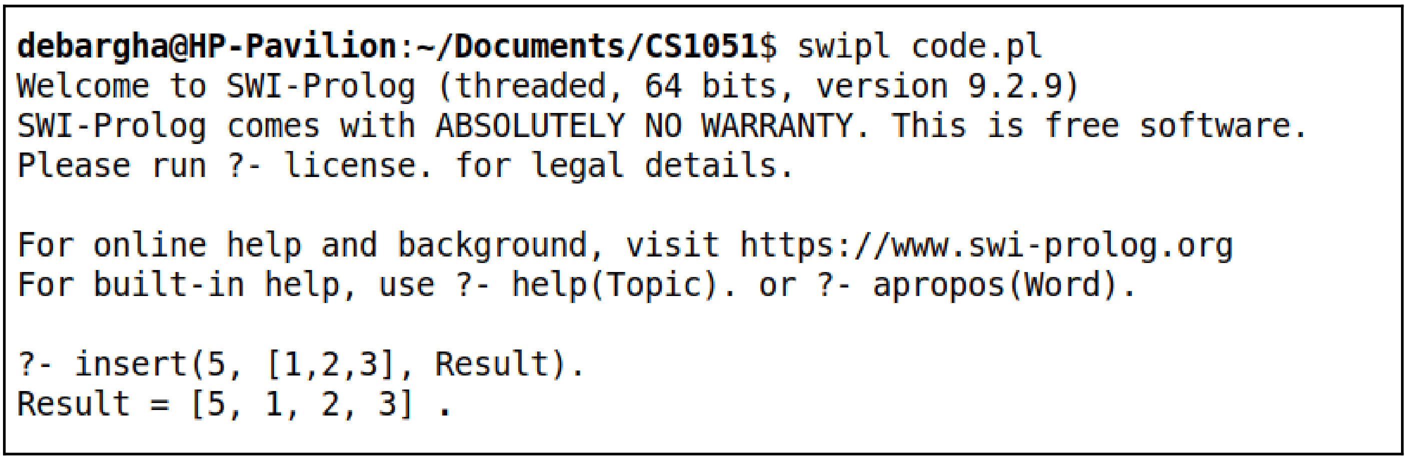
Assignment-2

1. Write a Prolog program to insert an element in a list.

code.pl

```
insert(E, L, [E|L]).
```

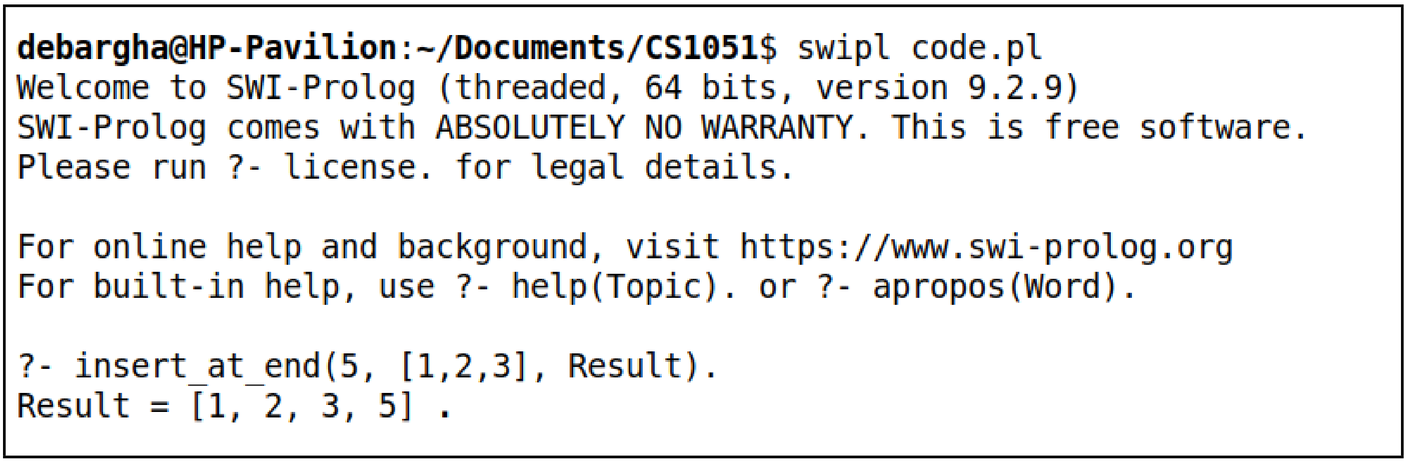


```
debargha@HP-Pavilion:~/Documents/CS1051$ swipl code.pl
Welcome to SWI-Prolog (threaded, 64 bits, version 9.2.9)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

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

?- insert(5, [1,2,3], Result).
Result = [5, 1, 2, 3] .
```

2. Write a Prolog program to insert an element in the last position of a list.

code.pl

```
insert_at_end(E, [], [E]).
insert_at_end(E, [H|T], [H|R]) :-
    insert_at_end(E, T, R).
```

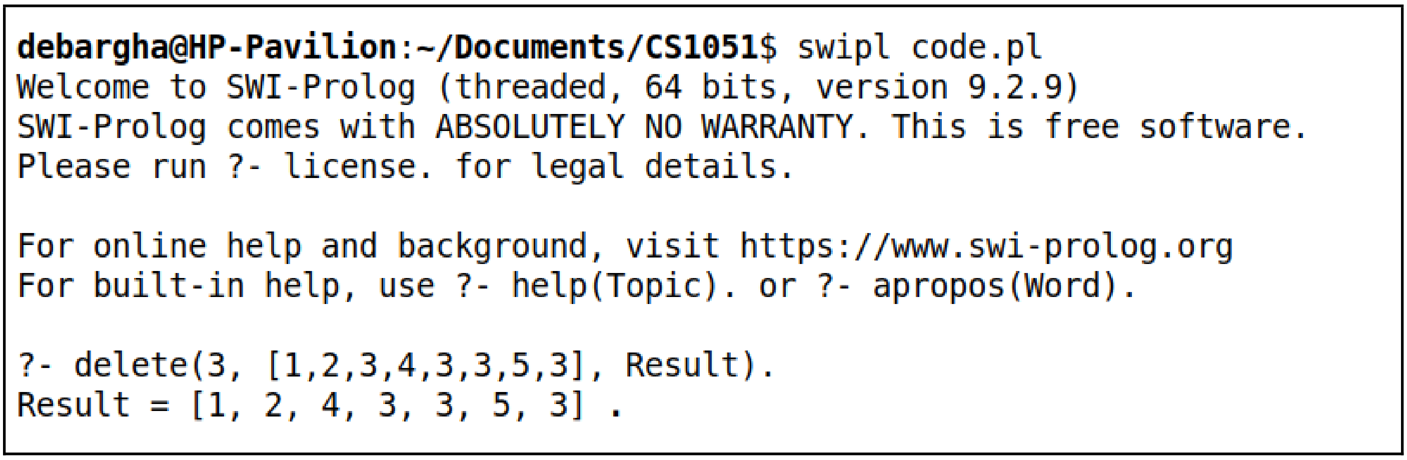


```
debargha@HP-Pavilion:~/Documents/CS1051$ swipl code.pl
Welcome to SWI-Prolog (threaded, 64 bits, version 9.2.9)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

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

?- insert_at_end(5, [1,2,3], Result).
Result = [1, 2, 3, 5] .
```

3. Write a Prolog program to delete an element in a list.

code.pl

```
delete(_, [], []).
delete(E, [E|T], T).
delete(E, [H|T], [H|R]) :-
    E \= H,
    delete(E, T, R).
```

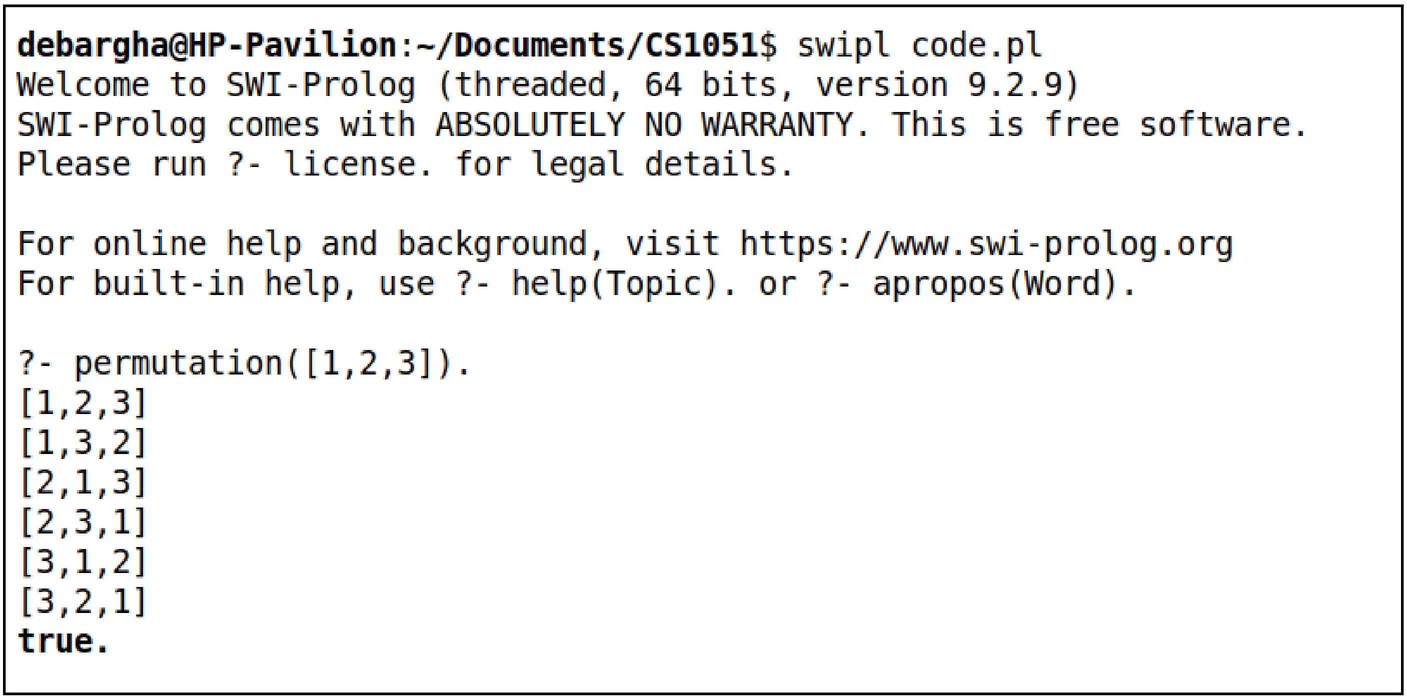


```
debargha@HP-Pavilion:~/Documents/CS1051$ swipl code.pl
Welcome to SWI-Prolog (threaded, 64 bits, version 9.2.9)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

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

?- delete(3, [1,2,3,4,3,3,5,3], Result).
Result = [1, 2, 4, 3, 3, 5, 3] .
```

4. Write a Prolog program to generate all permutations of a list of elements.

code.pl

```
permutation(List) :-
    permutation(List, P),
    write(P), nl,
    fail.
permutation(_).
```

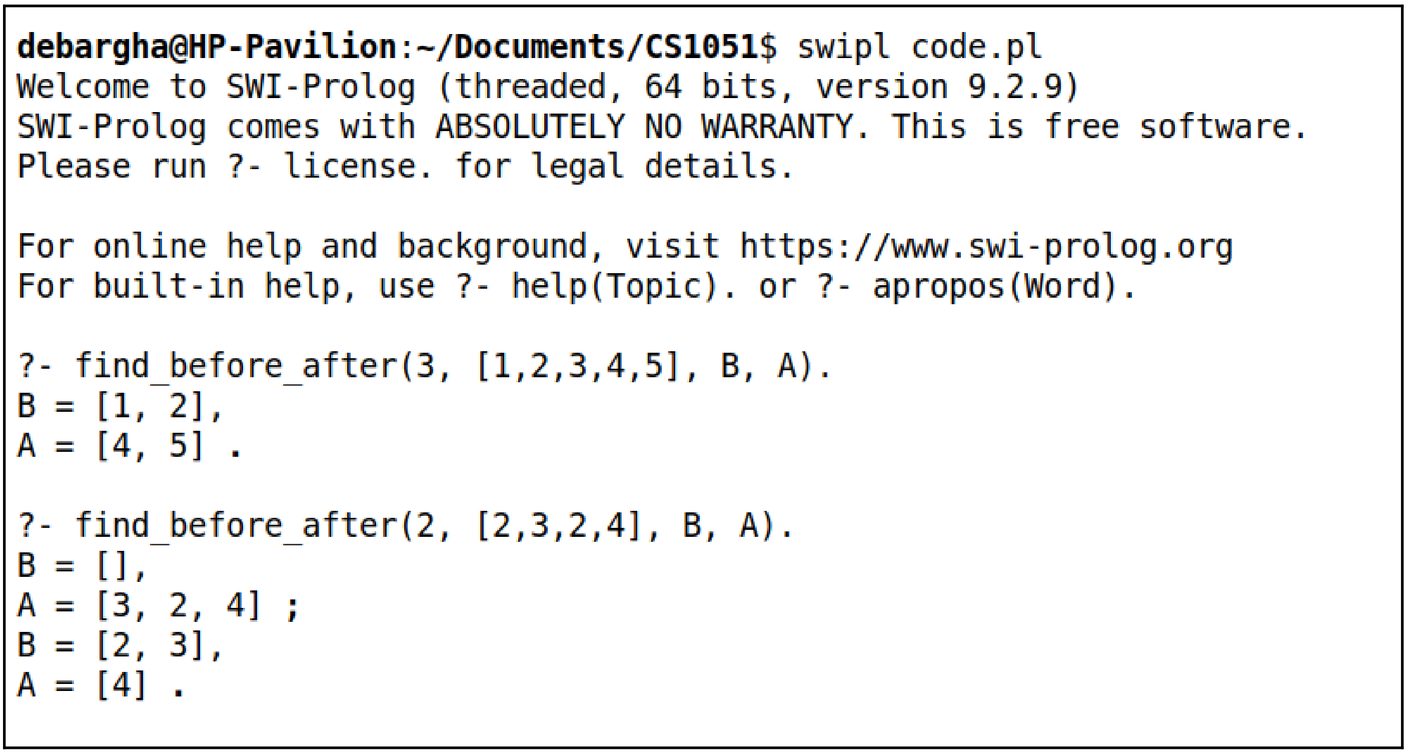


```
debargha@HP-Pavilion:~/Documents/CS1051$ swipl code.pl
Welcome to SWI-Prolog (threaded, 64 bits, version 9.2.9)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

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

?- permutation([1,2,3]).
[1,2,3]
[1,3,2]
[2,1,3]
[2,3,1]
[3,1,2]
[3,2,1]
true.
```

5. Write a Prolog program to find before and after list values of particular elements in the list.

code.pl

```
find_before_after(E, List, Before, After) :-
    append(Before, [E|After], List).
```

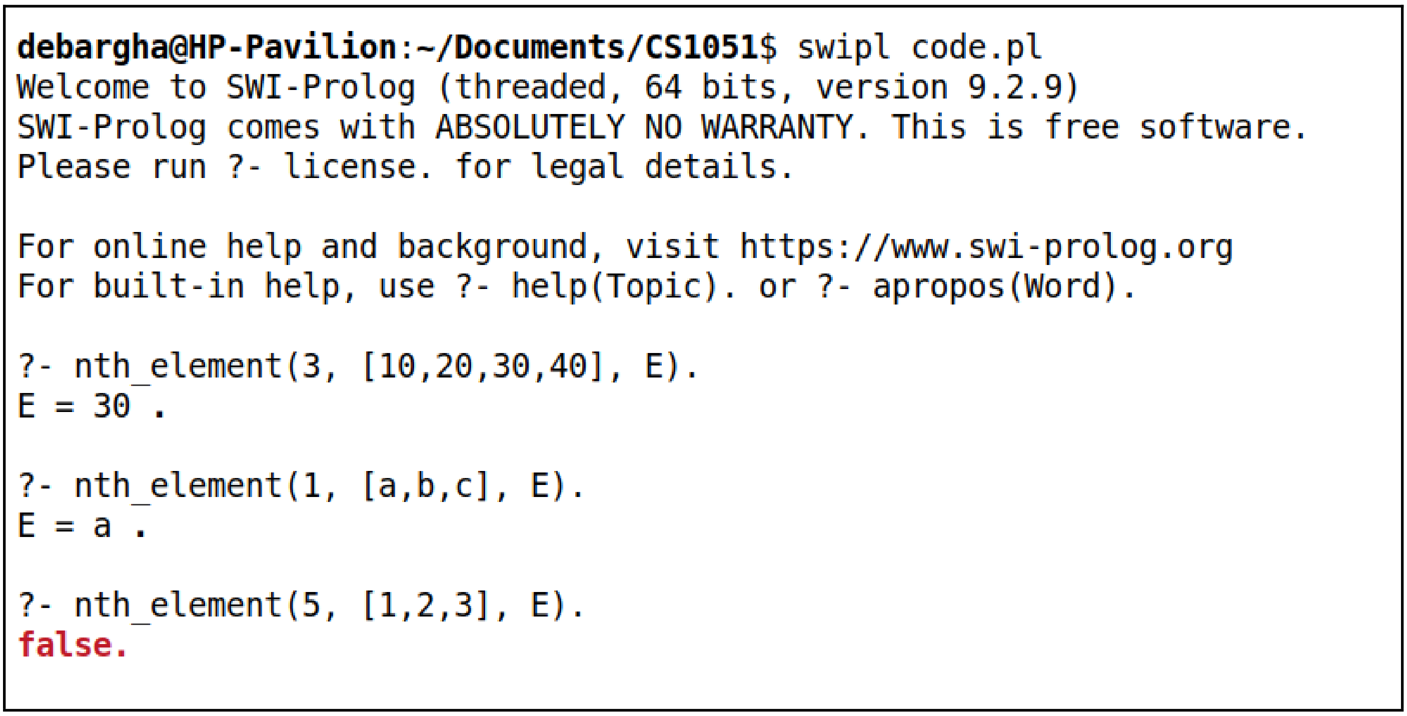


```
debargha@HP-Pavilion:~/Documents/CS1051$ swipl code.pl
Welcome to SWI-Prolog (threaded, 64 bits, version 9.2.9)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

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

?- find_before_after(3, [1,2,3,4,5], B, A).
B = [1, 2],
A = [4, 5] .

?- find_before_after(2, [2,3,2,4], B, A).
B = [],
A = [3, 2, 4] ;
B = [2, 3],
A = [4] .
```

6. Write a Prolog program to find the nth element of a list.

code.pl

```
nth_element(1, [H|_], H).
nth_element(N, [_|T], E) :-
    N > 1,
    N1 is N - 1,
    nth_element(N1, T, E).
```

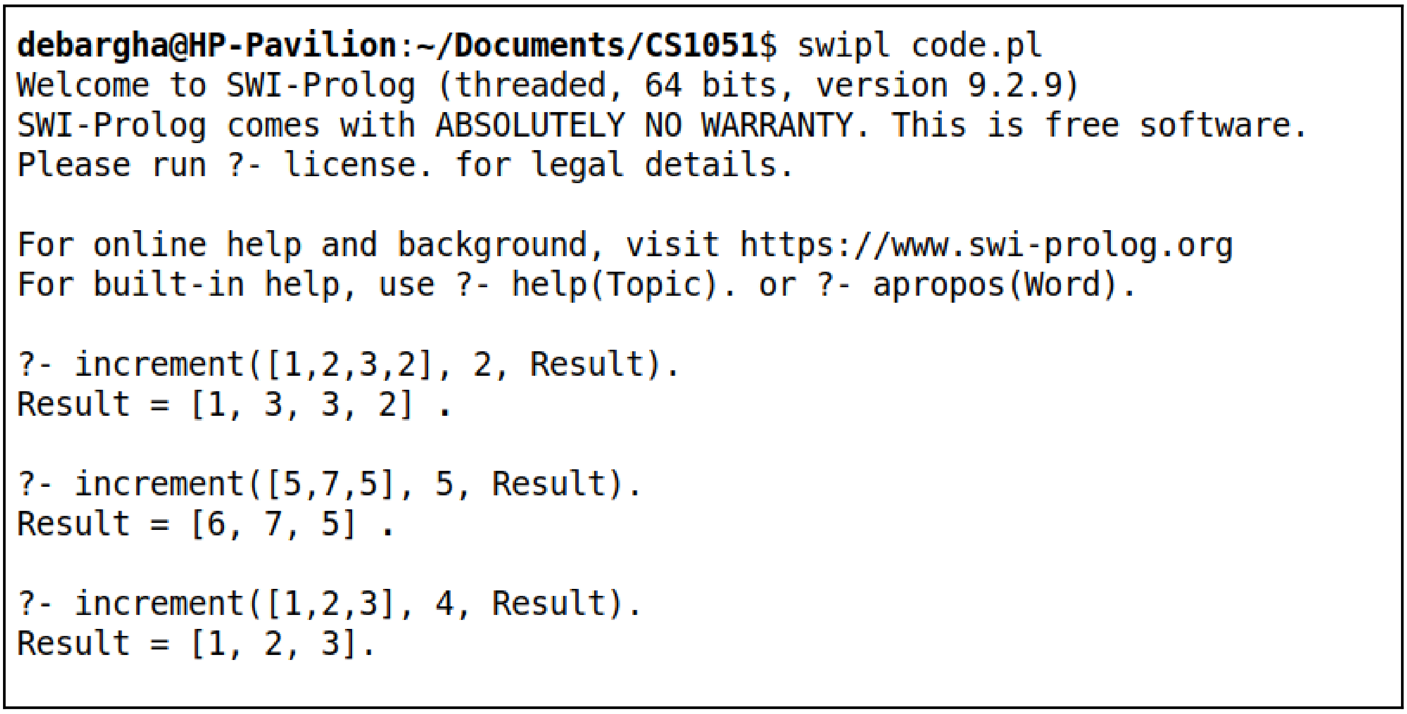


```
debargha@HP-Pavilion:~/Documents/CS1051$ swipl code.pl
Welcome to SWI-Prolog (threaded, 64 bits, version 9.2.9)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

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

?- nth_element(3, [10,20,30,40], E).
E = 30 .

?- nth_element(1, [a,b,c], E).
E = a .

?- nth_element(5, [1,2,3], E).
false.
```

7. Write a Prolog program to increment one value of a list element.

code.pl

```
increment([], _, []).
increment([H|T], H, [H1|T]) :-
    H1 is H + 1.
increment([H|T], E, [H|R]) :-
    H \= E,
    increment(T, E, R).
```

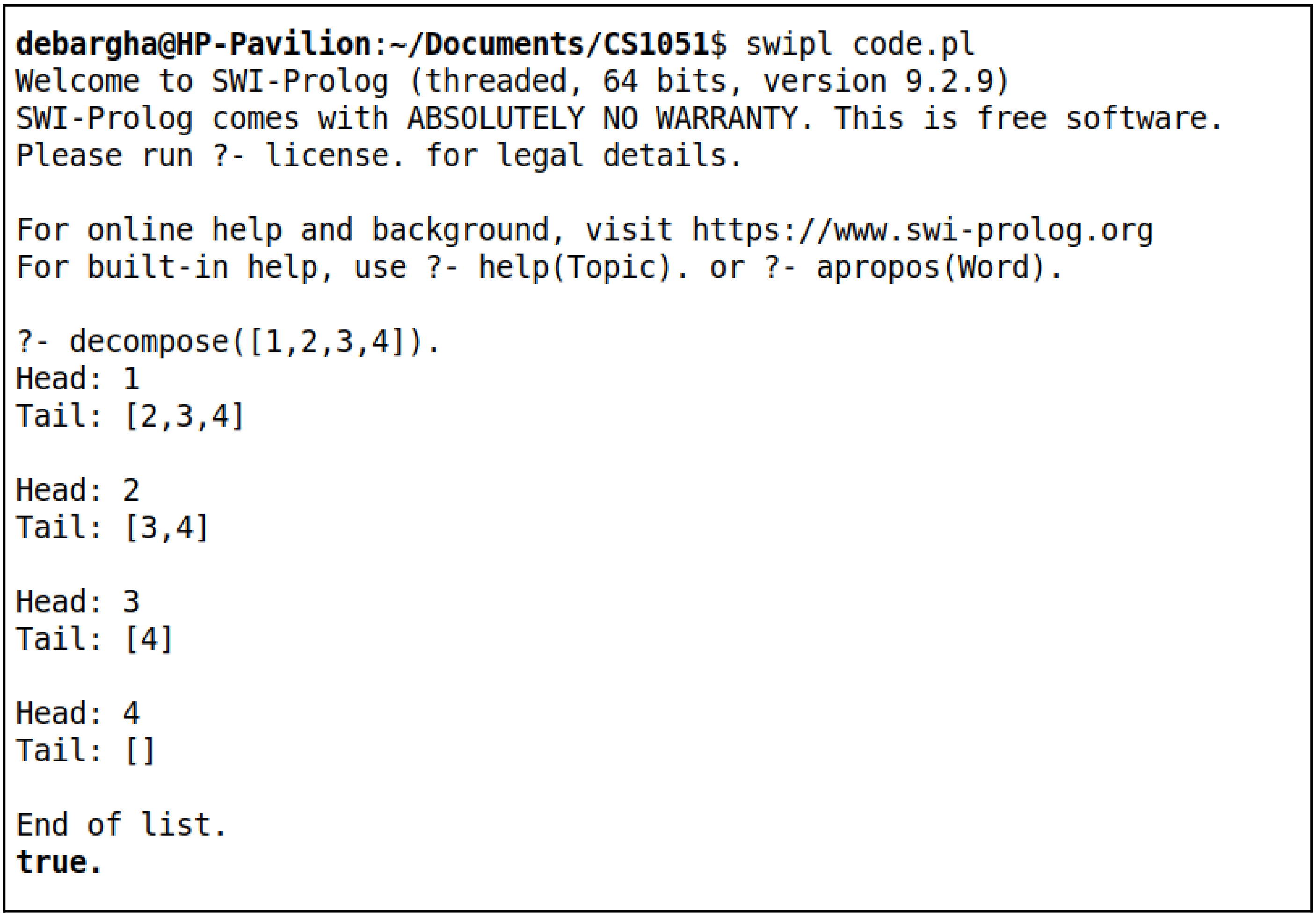


```
debargha@HP-Pavilion:~/Documents/CS1051$ swipl code.pl
Welcome to SWI-Prolog (threaded, 64 bits, version 9.2.9)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

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

?- increment([1,2,3,2], 2, Result).
Result = [1, 3, 3, 2] .

?- increment([5,7,5], 5, Result).
Result = [6, 7, 5] .

?- increment([1,2,3], 4, Result).
Result = [1, 2, 3].
```

8. Write a Prolog program to decompose a list.

code.pl

```
decompose([]) :-
    write('End of list.'), nl.

decompose([H|T]) :-
    write('Head: '), write(H), nl,
    write('Tail: '), write(T), nl, nl,
    decompose(T).
```



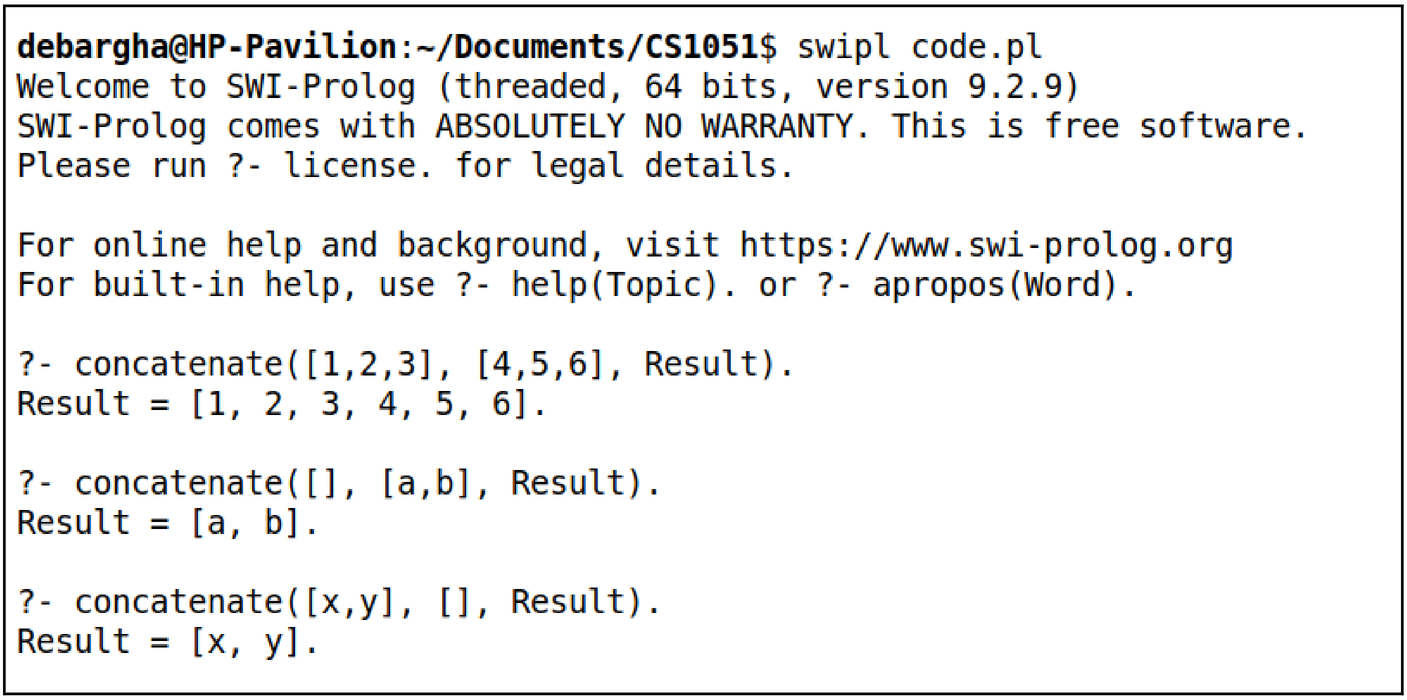
```
debargha@HP-Pavilion:~/Documents/CS1051$ swipl code.pl
Welcome to SWI-Prolog (threaded, 64 bits, version 9.2.9)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

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

?- decompose([1,2,3,4]).
Head: 1
Tail: [2,3,4]

Head: 2
Tail: [3,4]

Head: 3
Tail: [4]

Head: 4
Tail: []

End of list.
true.
```

---

9. Write a Prolog program to concatenate two lists.

code.pl

```
concatenate([], L2, L2).
concatenate([H|T], L2, [H|R]) :-
    concatenate(T, L2, R).
```



```
debargha@HP-Pavilion:~/Documents/CS1051$ swipl code.pl
Welcome to SWI-Prolog (threaded, 64 bits, version 9.2.9)
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).

?- concatenate([1,2,3], [4,5,6], Result).
Result = [1, 2, 3, 4, 5, 6].

?- concatenate([], [a,b], Result).
Result = [a, b].

?- concatenate([x,y], [], Result).
Result = [x, y].
```