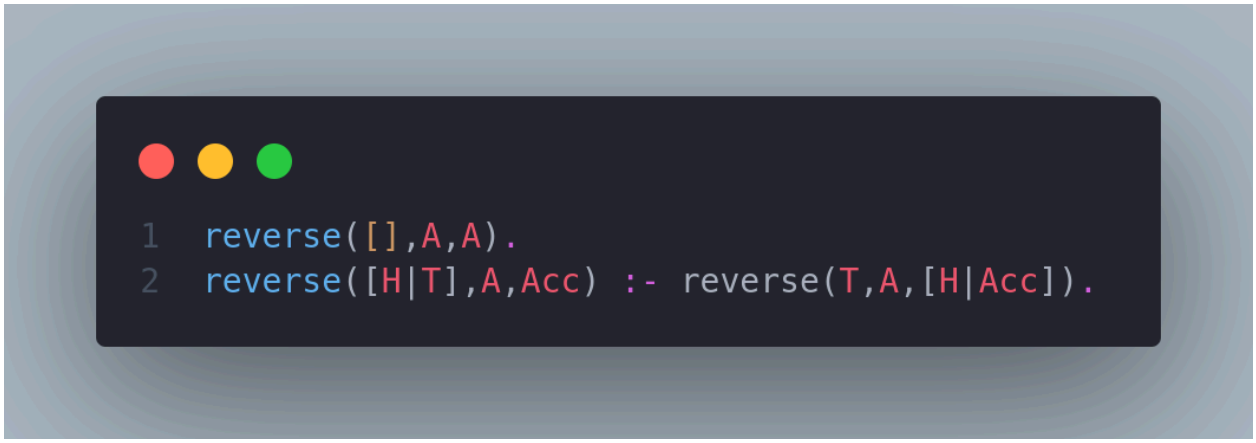
Computer Intelligence Lab - CS354N

Weekly Assignment Report

Question 1:



```
reverse([],A,A).
reverse([H|T],A,Acc) :- reverse(T,A,[H|Acc]).
```

Output:

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

?- reverse([1,2,3,6,5,4], A, []).
A = [4, 5, 6, 3, 2, 1].
```

Question 2:

```
element_at(H,[H|_],1):- !.
element_at(A,[_|T],N) :- N1 is N - 1, element_at(A,T,N1).
```

Output:

```
?- element_at(X, [a, b, c, d, e], 3).
X = c.

?- element_at(X, [a, b, c, d, e], 6).
false.

?- element_at(X, [a, b, c, d, e, f], 6).
X = f.
```

Question 3:

```
reverse([],A,A).
reverse([H|T],A,Acc) :- reverse(T,A,[H|Acc]).
palindrome(L):- reverse(L, L, []).
```

Output:

```
?- palindrome([x, a, m, a, x]).
true.

?- palindrome([x, a, m, a, x, a]).
false.

?- palindrome([r, a, c, e, c, a, r]).
true.

?- palindrome([r, a, c, e, c, a]).
false.
```

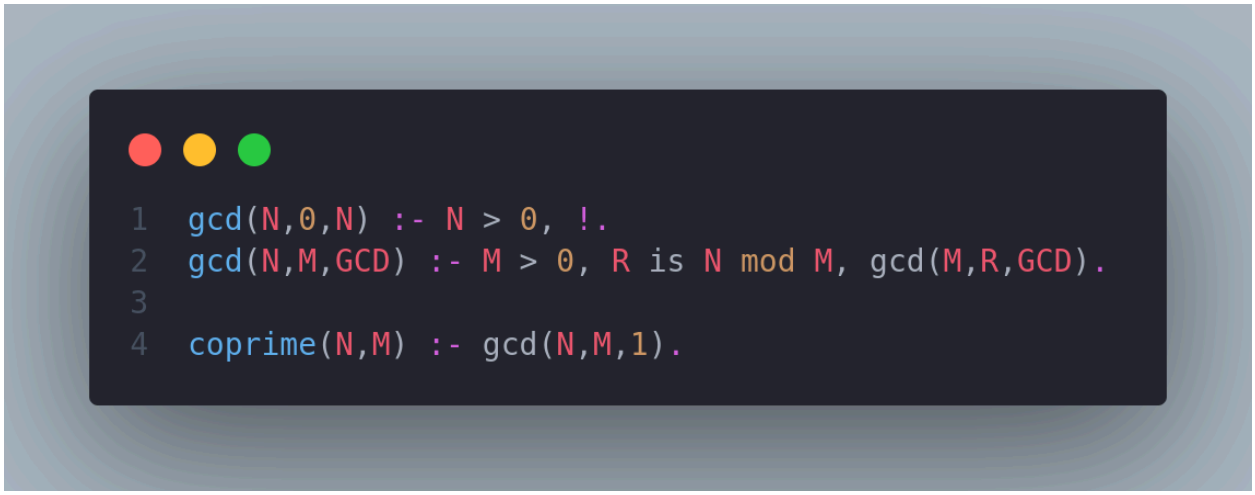
Question 4:

```
div(N, F, M, R, K) :- S is N // F, N =:= S * F, !,
   K1 is K + 1, div(S, F, M, R, K1).
div(N, _, M, N, M).

division(N, F, M, R) :- div(N, F, M, R, 0), M > 0.

next_factor(_, 2, 3) :- !.
next_factor(N, F, NF) :- F * F < N, !, NF is F + 2.
next_factor(N, _, N).

prime_factors(N, L) :- N > 0, prime_factors(N, L, 2).

prime_factors(1, [], _) :- !.
prime_factors(N, [[F, M]|L], F) :- division(N, F, M, R), !,
   next_factor(R, F, NF), prime_factors(R, L, NF).
prime_factors(N, L, F) :- !,
   next_factor(N, F, NF), prime_factors(N, L, NF).

prime_factors_mult(N, L) :- prime_factors(N, L).
```

Output:

```
?- prime_factors(315, L).
L = [[3, 2], [5, 1], [7, 1]].

?- prime_factors(35, L).
L = [[5, 1], [7, 1]].

?- prime_factors(5, L).
L = [[5, 1]].

?- prime_factors(1, L).
L = [].

?- prime_factors(9999, L).
L = [[3, 2], [11, 1], [101, 1]].
```

Question 5:



```
gcd(N,0,N) :- N > 0, !.
gcd(N,M,GCD) :- M > 0, R is N mod M, gcd(M,R,GCD).

coprime(N,M) :- gcd(N,M,1).
```

Output:

```
?- coprime(35, 64).
true.

?- coprime(35, 77).
false.

?- coprime(1, 3).
true.

?- coprime(2, 3).
true.

?- coprime(20, 30).
false.
```

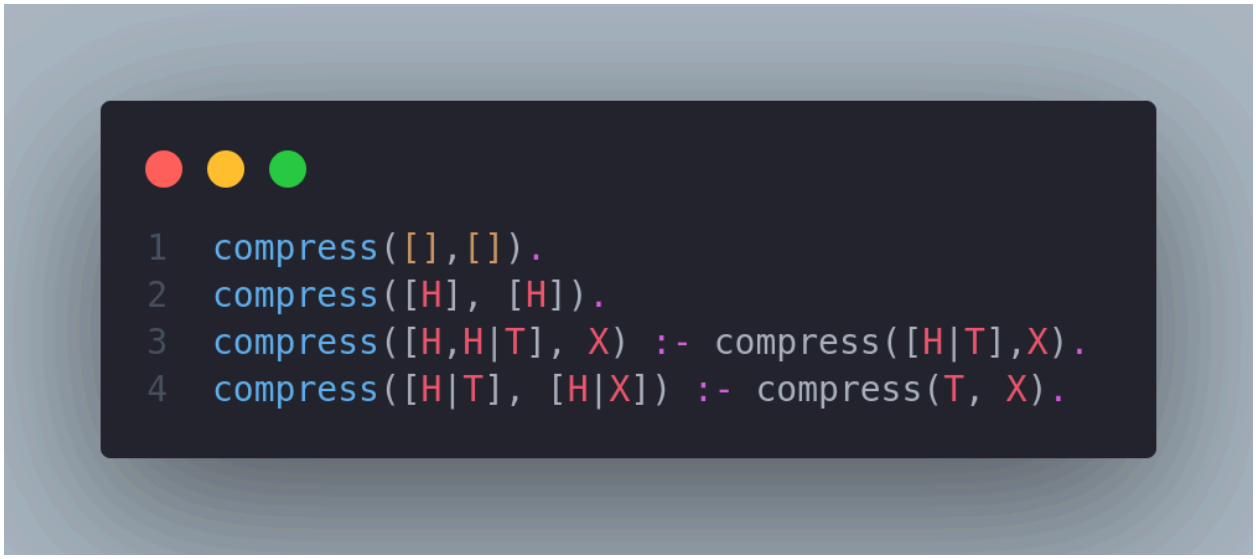
Question 6:

```
my_flatten([],[]):- !.
my_flatten(A,[A]) :- \+ is_list(A).
my_flatten([A|A1],C1) :- my_flatten(A,B), my_flatten(A1,B1), append(B,B1,C1).
```

Output:

```
?- my_flatten([a, [b, [c, d], e]], X).
X = [a, b, c, d, e].

?- my_flatten([a, [b, [c, d]]], X).
X = [a, b, c, d].

?- my_flatten([a, e], X).
X = [a, e].

?- my_flatten([a, [b, [c, [f, g], d], e]], X).
X = [a, b, c, f, g, d, e].
```

Question 7:



```
compress([],[]).
compress([H], [H]).
compress([H,H|T], X) :- compress([H|T],X).
compress([H|T], [H|X]) :- compress(T, X).
```

Output:

```
?- compress([a, a, a, a, b, c, c, a, a, d, e, e, e, e], X).
X = [a, b, c, a, d, e] .

?- compress([a, a, a, a, b, c, c, a, a, e, e, e, e], X).
X = [a, b, c, a, e] .

?- compress([a, b, c, d, e], X).
X = [a, b, c, d, e] .
```

Question 8:

```prolog
com(A, [A|L], L).
com(A, [_|L], R) :- com(A, L, R).

combination(0, _, []):- !.
combination(N, L, [H|T]) :- com(H, L, R), N1 is N-1,  combination(N1, R, T).
```

Output:

```
?- combination(3, [a, b, c, d, e, f], L).
L = [a, b, c] ;
L = [a, b, d] ;
L = [a, b, e] ;
L = [a, b, f] ;
L = [a, c, d] ;
L = [a, c, e] ;
L = [a, c, f] ;
L = [a, d, e] ;
L = [a, d, f] ;
L = [a, e, f] ;
L = [b, c, d] ;
L = [b, c, e] ;
L = [b, c, f] ;
L = [b, d, e] ;
L = [b, d, f] ;
L = [b, e, f] ;
L = [c, d, e] ;
L = [c, d, f] ;
L = [c, e, f] ;
L = [d, e, f] ;
false.
```

Question 9:

```prolog
add_key([],[]).
add_key([A|A1],[L-p(A)|B1]) :-!, length(A,L), add_key(A1,B1).

rem_key([],[]).
rem_key([_-p(A)|A1],[A|B1]) :- rem_key(A1,B1).

lsort(In,Out) :-
    add_key(In,K),
    keysort(K,SK),
    rem_key(SK,Out).
```

Output:

```
?- lsort([[a, b, c], [d, e], [f, g, h], [d, e], [i, j, k, l], [m, n], [o]], L).
L = [[o], [d, e], [d, e], [m, n], [a, b, c], [f, g, h], [i, j|...]].

?- lsort([[a, b, c], [d, e], [f, g, h], [d, e], [i, j, k, l]], L).
L = [[d, e], [d, e], [a, b, c], [f, g, h], [i, j, k, l]].

?- lsort([[a, b, c], [d, e], [i, j, k, l], [m, n], [o]], L).
L = [[o], [d, e], [m, n], [a, b, c], [i, j, k, l]].
```

Question 10:

```
add_key([],[]).
add_key([A|A1],[L-p(A)|B1]) :-!, length(A,L), add_key(A1,B1).

rem_key([],[]).
rem_key([_-p(A)|A1],[A|B1]) :- rem_key(A1,B1).

lsort(In,Out) :-
    add_key(In,K),
    keysort(K,SK),
    rem_key(SK,Out).

pack([],[]).
pack([L-A|A1],[[L-A|C]|C1]) :- transf(L-A,A1,B1,C), pack(B1,C1).

transf(_,[],[],[]).
transf(L-_,[K-B|B1],[K-B|B1],[]) :- L \= K.
transf(L-_,[L-A|A1],B1,[L-A|C1]) :- transf(L-A,A1,B1,C1).

lfsort(In,Out) :-
    add_key(In,K),
    keysort(K,SK),
    pack(SK,PK),
    lsort(PK,SPK),
    flatten(SPK,FK),
    rem_key(FK,Out).
```

Output:

```
?- lfsort([[a, b, c], [d, e], [f, g, h], [d, e], [i, j, k, l], [m, n], [o]], L).
L = [[o], [i, j, k, l], [a, b, c], [f, g, h], [d, e], [d, e], [m, n]] .

?- lfsort([[a, b, c], [d, e], [f, g, h], [d, e], [i, j, k, l], [m, n]], L).
L = [[i, j, k, l], [a, b, c], [f, g, h], [d, e], [d, e], [m, n]] .

?- lfsort([[a, b, c], [d, e], [f, g, h], [m, n], [o]], L).
L = [[o], [d, e], [m, n], [a, b, c], [f, g, h]] .
```