Homework 6  
CSE 4102 Homework 6, Spring 2016  
William Dickson  
05/01/2016  
Section: 001  
Instructor: Jeffrey A. Meunier

# Introduction:

# In this assignment I will wrote a series of Prolog predicates (theorems). These are short predicates to get accustomed to Prolog.

# Output:

liam@Liams-MacBook-Air:[{...}/CSE4102/hw6]$ gprolog

GNU Prolog 1.4.4 (64 bits)

Compiled Apr 9 2016, 13:47:50 with clang

By Daniel Diaz

Copyright (C) 1999-2013 Daniel Diaz

| ?- [hw6].

compiling {...}/CSE4102/hw6/hw6.pl for byte code...

{...}/CSE4102/hw6/hw6.pl compiled, 22 lines read - 4421 bytes written, 6 ms

(2 ms) yes

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

X = [a,b,c,a,d,e]

yes

| ?- my\_flatten([a, [b, [c, d], e]], X).

X = [a,b,c,d,e]

yes

| ?- pack([a,a,a,a,b,c,c,a,a,d,e,e,e], X).

X = [[a,a,a,a],[b],[c,c],[a,a],[d],[e,e,e]]

yes

| ?- rlencode([a,a,a,a,b,c,c,a,a,d,e,e,e], X).

X = [[a,4],[b,1],[c,2],[a,2],[d,1],[e,3]]

yes

| ?- range(2,10,L).

L = [2,3,4,5,6,7,8,9,10]

(1 ms) yes

| ?-

# Source Code:

/\* collapses sequences of consecutive identical ground terms \*/

compress([], []).

compress([X|[X|Zs]], Y) :- !, compress([X|Zs], Y).

compress([X|Xs], [X|Ys]) :- compress(Xs, Ys).

/\* collapses lists into a single \*/

my\_flatten([], []) :- !.

my\_flatten([X|Xs], Y) :- !, my\_flatten(X, T1), my\_flatten(Xs, T2), append(T1, T2, Y).

my\_flatten(X, [X]).

/\* packs the list based on repeating values \*/

pack([X], [[X]]) :- !.

pack([X|[X|Xs]], [[X|Zs]|Y]) :- !, pack([X|Xs], [Zs|Y]).

pack([X|[Z|Zs]], [[X]|Ys]) :- pack([Z|Zs], Ys).

/\* count the packed terms \*/

rlencode(X,Y) :- pack(X,T), rlencode2(T,Y).

rlencode2([], []).

rlencode2([[X|Xs]|Ys], [[X, Z]|Zs]) :- length([X|Xs], Z), rlencode2(Ys, Zs).

/\* create terms in given range and return list \*/

range(X, X, [X]) :- !.

range(X, Y, [X|Ls]) :- !, Z is X + 1, range(Z, Y, Ls).