CSE3401 Summer 2009 Assignment #3, Due by August 7th 2009 11:59PM

Exercise 1

Write the predicate `sumOf(Integers1, Integers2, Integers3)` that asserts that `Integers3` contains the sum of the integers in the corresponding position in `Integers1` and `Integers2`. Assume the shorter list is extended with zeros to the length of the longer list.

Example:

```
sumOf([0, 11, 22, 33, 44], [-1, -2, -3, -4], Sum).
Sum = [-1, 9, 19, 29, 44]
```

Exercise 2

Write the predicate `everyNth(N, List1, List2)` that asserts that `List2` contains the first item followed by every N'th item after the first item of `List1`.

Examples:

```
everyNth( 3 , [1, 2], List2).
List2 = [1] ;
no
everyNth( 3 , [1, 2, [3, 4], [[5]], 6, [7, 8, [[9]]], 10]
, List2).
List2 = [1, [[5]], 10] ;
no
everyNth( 3 , [1, 2, [3, 4], [[5]], 6, [7, 8, [[9]]], 10, 11]
, List2).
List2 = [1, [[5]], 10] ;
no
```

Exercise 3

Define a Prolog predicate `removeAll(Item,Alist,Blist)` that is true if `Blist` is the same as `Alist` with every `Item` removed from all levels. Use the predicates `=` {e.g. `A=B`} and `not` {e.g. `not(A=B)`} to distinguish cases.

Examples:

```
?- removeAll(a, [a], TheList).
TheList = [] ;
no
?- removeAll(a, [a, b], TheList).
TheList = [b] ;
no
?- removeAll([a, [a, a, [a]]], TheList).
TheList = [[[]]] ;
no
```

DO NOT HAND IN the following: As an exercise try variations on the definition by:
(1) removing both = and not;
(2) removing only not;
(3) removing only =.
Think about what happens in those cases.

Exercise 4

Write the predicate `listCount_na(InList,Count)` without using an accumulator that asserts that a `Count` number of lists occurring at all levels of the list structure `InList`. Use cut,`!`, and not `not`, `\+`, to eliminate multiple answers.

Examples:

```
listCount_na ([b,a],N).
N = 0 ;
no
listCount_na ([b,[a,[a],c],a],N).
N = 2 ;
no
listCount_na ([b,[a,[a],c],[a]],N).
N = 3 ;
no
```

Exercise 5

Write the predicate `listCount_wa(InList,Count)` using an accumulator to assert the same predicate as in Exercise 6. Use cut,`!`, and not `not`, `\+`, to eliminate multiple answers.