3/20

ASSIGNMENT 5

This assignment is about writing logic programs using Prolog language. All predicates must be written in Prolog that runs on the SWI-Prolog software. You can only use those Prolog language features discussed in class. The predicates must be placed in a file named program. The program must be well documented. Test your predicates thoroughly with all input cases.

1) Write a predicate power that raises a number to a power. `Example: power(2, 6, X) returns X = 64`

2) Write a predicate minmaxArray that finds the minimum and maximum values of an integer array. Example: `minmaxArray([5, 7, 3, 10, 6, 1, 4], X, Y) returns X = 1, Y = 10`

3) Write a predicate sumEven that finds the sum of all even numbers in an integer array. `Example: sumEven([5, 7, 3, 10, 6, 1, 4], X) returns X = 20`

4) Write a predicate positive that finds all positive numbers in an integer array. `Example: positive([5, -2, -8, 6, 2, 0, -1], X) returns X = [5, 6, 2]`

5) Write a predicate secondLast that finds the element before the last element of a list. `Example: secondLast([c, [a, b], d, [b, c], a], X) returns X = [b, c]`

6) Write a predicate completeReverse that reverses a list at all levels. `Example: completeReverse([a, [b, c], [a, [c, [b, d, e], a], b], e], X) returns X = [e, [b, [a, [e, d, b], c], a], [c, b], a]`

7) Write a predicate dremove that removes duplicates from a list. `Example: dremove([c, [a, b], c, d, [a, b]], X) returns X = [c, [a, b], d]`

8) Write a predicate intersect that finds the intersection of two lists. The intersection here means the common elements of the two lists. `Example: intersect([a, [a, b], c, d], [b, [a, b], d], X) returns X = [[a, b], d]`

9) Write a predicate descendSort that sorts an integer array in the descending order. `Example: descendSort([5, -2, -8, 6, 2, 0, -1], X) returns X = [6, 5, 2, 0, -1, -2, -8]`

10) Write a predicate prime that decides whether a positive integer is a prime number or not. `Example: prime(43) returns true, prime(24) returns fail`