Prolog Assignment

1. Write a predicate abs(*integer1,integer2*) which returns true if the second integer is

the absolute value of the first.

Examples:

abs(3,3) evaluates as true.

abs(-3,3) evaluates as true.

abs(4,-4) evaluates as false.

abs(-6,X) unifies as X = 6.

abs(X,Y):-

number(X), X < 0 , Y is -X.

abs(X,X):-

number(X), X >= 0.

1. Write a predicate oddList(List1,List2) which returns true if List2 contains the elements of List1 that occur in the odd-numbered positions (assuming 0-indexing).

Examples:

oddList([],[]) evaluates as true.

oddList([10],Y) unifies as Y = [].

oddList([1,2,3,4,5,6,7],[2,4,6]) evaluates as true.

oddList([8,3,1,6,9,12,15,21],X) unifies as X = [3,6,12,21].

oddList([],[]).

oddList([\_], X):-

X = [].

oddList([\_,X|T],[X|T2]):-

oddList(T,T2).

1. Write a predicate evenList(List1,List2) which returns true if List2 contains the elements of List1 that occur in the even-numbered positions (assuming 0-indexing).

Examples:

evenList([],[]) evaluates as true.

evenList([10],Y) unifies as Y = [10].

evenList([1,2,3,4,5,6,7],[1,3,5,7]) evaluates as true.

evenList([8,3,1,6,9,12,15,21],X) unifies as X = [8,1,9,15].

evenList([],[]).

evenList([Y], X):-

X is Y.

evenList([X, \_|T], [X|T2]):-

evenList(T, T2).

1. Write a predicate mergeLists(List1,List2,List3) which returns true if List3 is List containing the elements of List1 in sequence followed by the elements of List2 insequence

Examples:

mergeLists([1,2,3],[4,5,6,7][1,2,3,4,5,6,7]) returns true.

mergeLists([1],[2],[1,2]) returns true.

mergeLists([1,2],[1,2],X) unifies as X = [1,2,1,2].

mergeLists(A, [], A).

mergeLists([], B, B).

mergeLists([H1|T1], [H2|T2], R):-

H1 =< H2,

mergeLists(T1, [H2|T2], M),

R = [H1|M].

mergeLists([H1|T1], [H2|T2], R):-

H1 > H2,

mergeLists(T2, [H1|T1], M),

R = [H2|M].

1. Write a predicate mergeSort(List1,List2) which returns true if List2 consists of the sorted elements of List1.

Examples:

mergeSort([8,5,7,3,9],[3,5,7,8,9]) returns true

split(L,[],L,0).

split([H|T],[H|A],B,N):-

N1 is N-1,

split(T, A, B, N1).

divide(L, A, B):-

length(L, Len),

Half is Len//2,

split(L, A, B, Half).

mergeLists(A, [], A).

mergeLists ([], B, B).

mergeLists ([H1|T1], [H2|T2], R):-

H1 =< H2,

mergeLists (T1, [H2|T2], M),

R = [H1|M].

mergeLists ([H1|T1], [H2|T2], R):-

H1 > H2,

mergeLists (T2, [H1|T1], M),

R = [H2|M].

mergeSort([], []).

mergeSort([X], [X]).

mergeSort([H1, H2], [H1, H2]):-

H1 =< H2.

mergeSort([H1, H2], [H2, H1]):-

H1 > H2.

mergeSort(L, R):-

divide(L, A, B),

mergeSort(A, Asort),

mergeSort(B, Bsort),

merge(Asort, Bsort, R).

1. There is a street with three neighboring houses that all have a different color, namely red, blue, and green. People of different nationalities live in the different houses and they all have a different pet. Here are some more facts about them:

-The Englishman lives in the red house.

-The jaguar is the pet of the Spanish family.

-The Japanese lives to the right of the snail keeper.

-The snail keeper lives to the left of the blue house.

Who keeps the zebra? Don’t work it out for yourself: define a predicate zebra/1 that tells you the nationality of the owner of the zebra!

(Hint: Think of a representation for the houses and the street. Code the four constraints in Prolog. You may find member/2 and sublist/2 useful.)

1. The formal language Even is very simple: it consists of all strings containing an even number of a’s, and nothing else. Note that the empty string ϵ belongs to Even. Write a program that generates Even .

What are the two basic activities of test driven development

-use automated tests

-write the tests before you implement the code

What are the six step we can identify as a practice of test driven development

-write an automated test

-run the test – it fails

-write the implementation

-run the test – it may pass

- refactor only once

-run the test, it passes

What is refactoring

-behavior preserving code transformation

Why do we refactor code

-to simplify and restructure code to make it more efficient and easier to maintain

What happens if a test passes

-you are done

What happens if a test fails

-write more code

Failing the plan is planning to fail