Assignment Project Exam Help

Lists Recap, append and Exercise

https://tutorcs.com

WeChat: cstutorcs

Fariba Sadri

Recap on List Unification

% children_of(Mother, Father, [Child$_1$,..,Child$_k$])

children_of(elizabeth, philip,[charles, ann, edward, andrew]).
children_of(diana, charles, [harry, william]).
children_of(jane, bob, [jim]).
children_of(mary, peter, []).
children_of(mo, joe, [charles, james]).

Assignment Project Exam Help
https://tutorcs.com
WeChat: cstutorcs

Example Queries:

?-children_of(M, F, [ ]).
?-children_of(M, F, [C]).
?-children_of(M, F, [C|Cs]).
?-children_of(M, F, [C1, C2|Cs]).
?-children_of(M, F, [C1, C2, C3, C4]).
?-children_of(M, F, Children), length(Children, 4).

```
children_of(elizabeth, philip,[charles, ann, edward, andrew]).
children_of(diana, charles, [harry, william]).
children_of(jane, bob, [june]).
children_of(mary, peter, []).
children_of(mo, joe, [james, charles]).
```

Assignment Project Exam Help
https://tutorcs.com
WeChat: cstutorcs

More Example Queries:

```
?-children_of(M, bob, Cs).
?-children_of(M, F, [charles|Rest]).
?-children_of(M, F, [ann|Rest]).
?-children_of(M, F, X), member(charles, X).
?-children_of(M, F, X), member(charles, X),
      \+ member(ann, X).
```

Appending Lists: append/3
built-in predicate

append(L1, L2, L) :

L is the result of appending list L1 to the front of list L2.

Assignment Project Exam Help
https://tutorcs.com
WeChat: cstutorcs

e.g. **append([a,b],[c,d,e],[a,b,c,d,e])**

append([], [1,2], [1,2])

Definition of append

```
append([ ], L, L).
append([H|L1], L2, [H|L3]):-
        append(L1, L2, L3).
```

Assignment Project Exam Help
https://tutorcs.com
WeChat: cstutorcs

Use of append/3

?-append([1], [2, 3], [1, 2, 3]).

yes

?-append([1], [2, 3], X).

X = [1,2,3]

?-append(X, [2, 3], [1, 2, 3]).

X = [1]

?-append([1], X, [1, 2, 3]).

X = [2, 3]

Assignment Project Exam Help
https://tutorcs.com
WeChat: cstutorcs

?-append(X, Y, [1, 2, 3]) .

(find all splittings of a given list)

Assignment Project Exam Help
https://tutorcs.com
WeChat: cstutorcs

X = [ ], Y = [1, 2, 3] ;

X = [1] , Y = [2, 3] ;

X = [1, 2], Y = [3] ;

X = [1, 2, 3], Y = [ ] ;

No

?- append(F, [3|R], [1,2,3,4,5]).

(split at an element)

Assignment Project Exam Help

F=[1,2], R=[4,5]

https://tutorcs.com

WeChat: cstutorcs

Exercise

Define

last(E, L) where E is the last element of list L.

Assignment Project Exam Help

Do:

https://tutorcs.com

- one version with *append*, and

WeChat: cstutorcs

- one version without *append*.

Exercise

Let L be a list of tuples of the form

(Hospital_name, Type) giving the name of a hospital and its type (nhs or private). Assume all nhs hospitals come before the private ones in L.

Assignment Project Exam Help
https://tutorcs.com
WeChat: cstutorcs

Write a program for

hosp_list(L, NHS, Priv)

that takes such a list L, and produces a list NHS of the NHS hospital tuples and a list Priv of the private ones.

E.g. Given

L= [(st_thomas, nhs), (st_george, nhs),
(guy, nhs), (bupa, private), (harley, private)]

Assignment Project Exam Help
https://tutorcs.com
WeChat: cstutorcs

NHS will be

[(st_thomas, nhs), (st_george, nhs), (guy, nhs)]

and Priv will be

[(bupa, private), (harley, private)]

Do two versions:

Assignment Project Exam Help

1. Using *append*

https://tutorcs.com

2. Using aggregation

WeChat: cstutorcs

Edit the program So:

NHS will be a list of NHS hospitals, e.g.

Assignment Project Exam Help

[st_thomas, st_george, guy]

https://tutorcs.com

and Priv will be a list of private hospitals, e.g.

WeChat: cstutorcs

[bupa, harley].

List Processing Styles

E.g. The Bubble Sort Algorithm

[1, 2, 4, 6, 3, 5]

[1, 2, 4, 6, 3, 5]

[1, 2, 4, 3, 6, 5]

[1, 2, 4, 3, 6, 5]

[1, 2, 3, 4, 6, 5]

[1, 2, 3, 4, 6, 5]

[1, 2, 3, 4, 5, 6]

Assignment Project Exam Help

https://tutorcs.com

WeChat: cstutorcs

Bubble

```
bubble(L, L) :- sorted(L).

bubble(L, SL) :-
        append(L1, [X, Y|Rest], L),
        X>Y,
        append(L1, [Y, X|Rest], NewL),
        bubble(NewL, SL).

sorted(L) :-
        \+ (append(L1, [X, Y|Rest], L), X>Y).
```

Assignment Project Exam Help
https://tutorcs.com
WeChat: cstutorcs

Bubble with a cut

```
bubble(L, SL) :-
        append(L1, [X, Y|Rest], L),
        X>Y,
        !,
        append(L1, [Y, X|Rest], NewL),
        bubble(NewL, SL).
bubble(L, L).
```

Assignment Project Exam Help
https://tutorcs.com
WeChat: cstutorcs

Direct recursion or using an accumulator: Example - Reverse a List

```
rev([],[]).

rev([H|T],R) :-
        rev(T,RT),
        append(RT,[H],R).
```

Assignment Project Exam Help
https://tutorcs.com
WeChat: cstutorcs

reverse [1, 2, 3]

reverse [2, 3]

reverse [3]

reverse []

[]

[3]

[3, 2]

[3, 2, 1]

add 1 at the end

add 2 at the end, add 1 at the end

add 3 at the end, add 2 at the end, add 1 at the end

Assignment Project Exam Help
https://tutorcs.com
WeChat: cstutorcs

Reverse with Accumulator

```
rev2(L, Inv) :-      h_rev(L, [], Inv).

h_rev([], Acc, Acc).
h_rev([H|T], Acc, Inv) :- h_rev(T, [H|Acc], Inv).
```

Assignment Project Exam Help
https://tutorcs.com
WeChat: cstutorcs

Reverse with Accumulator

reverse [1, 2, 3]

Assignment Project Exam Help
https://tutorcs.com
WeChat: cstutorcs

List	Accumulator	Result
[1,2, 3]	[]	
[2, 3]	[1]	
[3]	[2,1]	
[]	[3,2,1]	[3,2,1]

Direct recursion or using an accumulator

E.g. Summing the elements of a list

[4, 6, 8] -----> 18

Assignment Project Exam Help
https://tutorcs.com
WeChat: cstutorcs

With Direct Recursion

```
sumList([],0).
sumList([N|L],S) :- sumList(L,SumL), S is N+SumL.
```

```
[4, 6, 8]
4      [6, 8]
4      6      [8]
4      6      8      []
4      6      8      0
4      6      8
4      14
4
18
```

Assignment Project Exam Help
https://tutorcs.com
WeChat: cstutorcs

With an accumulator

```
summing(L, S) :- sum_acc(L, 0, S).

sum_acc([], S, S).
```

Assignment Project Exam Help

Can also be written as:

https://tutorcs.com

```
sum_acc([], SumSoFar, S) :- S= SumSoFar.
```

WeChat: cstutorcs

```
sum_acc([E|Rest], SumSoFar, S) :-
        NewSum is SumSoFar+E,
        sum_acc(Rest, NewSum, S).
```

list	sum so far	final sum
[4, 6, 8]	0	
[6, 8]	4	
[8]	10	
[]	18	18

Assignment Project Exam Help
https://tutorcs.com
WeChat: cstutorcs