

Assignment Project Exam Help Lists Recap, append and Exercise

<https://powcoder.com>

Add WeChat powcoder

Fariba Sadri

Recap on List Unification

```
% children_of(Mother, Father, [Child1,...,Childk])  
children_of(elizabeth, philip,[charles, ann, edward, andrew]).  
children_of(diana, charles, [harry, william]).  
children_of(jane, philip, [abigail, james]).  
children_of(mary, peter, []).  
children_of(mo, joe, [charles, james]).
```

Assignment Project Exam Help
<https://powcoder.com>

Example Queries:

Add WeChat powcoder

```
?-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://powcoder.com>

More Example Queries:

Add WeChat powcoder

?-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://powcoder.com>

Add WeChat powcoder

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]):-

Assignment Project Exam Help

append(L1, L2, L3).

<https://powcoder.com>

Add WeChat powcoder

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://powcoder.com>

Add WeChat powcoder

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

(find all splittings of a given list)

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

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

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

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

No

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

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

(split at an element)

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

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Exercise

Define

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

Do:

- one version with *append*, and
- one version without *append*.

Exercise

Let L be a list of tuples of the form

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

Write a program for

$\text{hosp_list}(L, \text{NHS}, \text{Priv})$

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)]

NHS will be

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

and Priv will be

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

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Do two versions:

Assignment Project Exam Help

1. Using *append*

<https://powcoder.com>

2. Using aggregation

Add WeChat powcoder

Edit the program So:

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

[st_thomas, st_george, guy]

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

[bupa, harley].

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

List Processing Styles

E.g. The Bubble Sort Algorithm

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

Assignment Project Exam Help

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

<https://powcoder.com>

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

Add WeChat powcoder

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

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

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

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

Bubble

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

`bubble(L, SL) :-`

Assignment Project Exam Help

`append(L1, [X, Y | Rest], L),`

<https://powcoder.com>

`X > Y,`

Add WeChat powcoder

`append(L1, [Y, X | Rest], NewL),`

`bubble(NewL, SL).`

`sorted(L) :-`

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

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).

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

`rev([],[]).`

Assignment Project Exam Help

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

<https://powcoder.com>

`rev(T,RT),`

Add WeChat powcoder

`append(RT,[H],R).`

reverse [1, 2, 3]

reverse [2, 3] add 1 at the end

reverse [3] add 2 at the end, add 1 at the end

reverse [] add 3 at the end, add 2 at the end,

add 1 at the end
Add WeChat powcoder

[]

[3]

[3, 2]

[3, 2, 1]

Reverse with Accumulator

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

Assignment Project Exam Help

```
h_rev([], Acc, Acc).
```

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

<https://powcoder.com>
Add WeChat powcoder

Reverse with Accumulator

reverse [1, 2, 3]

List	Accumulator	Result
[1,2, 3]	https://powcoder.com	
[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

<https://powcoder.com>

Add WeChat powcoder

With Direct Recursion

sumList([],0).

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

[4, 6, 8]

4 [6, 8] Assignment Project Exam Help

4 6 [8] <https://powcoder.com>

4 6 8 Add WeChat powcoder

4 6 8 0

4 6 8

4 14

18

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://powcoder.com>

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

Add WeChat powcoder

```
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://powcoder.com>

Add WeChat powcoder