Dhaval Makwana – 18bce108

Artificial Intelligence

Practical - 7

Definition:

Write a PROLOG program on lists

i. Reversing a list.

ii. Finding the position of given element in the list a) from beginning. b) from end

Code 1: Reversing a list.

```
domains
   list=symbol*
predicates
   rev(list)
   findrev(list,list,list)
clauses
   rev(L):-
      findrev(L,[],List2),
      write(\"\\nReverse Of Given List : \",List2).
   findrev([],List1,List1).
   findrev([X|Tail],List1,List2):-
      findrev(Tail,[X|List1],List2).
```

<u>OUT PUT</u>

```
Goal: rev([a,b,c,d,e])
Reverse Of Given List : [\"e\",\"d\",\"c\",\"b\",\"a\"]
-------------------------------------------------------------------
Goal: rev([])
Reverse Of Given List : []
-------------------------------------------------------------------
```

```
Goal: rev([y,o,g,e,s,h,p,a,t,e,l])

Reverse Of Given List :
[\"l\",\"e\",\"t\",\"a\",\"p\",\"h\",\"s\",\"e\",\"g\",\"o\",\"y\"]

-------------------------------------------------------------------
```

Code 2:

a) from beginning

```
list_delete(X,[X|L1], L1).

list_delete(X, [Y|L2], [Y|L1]) :- list_delete(X,L2,L1).

list_perm([],[]).

list_perm(L,[X|P]) :- list_delete(X,L,L1),list_perm(L1,P).
```

<u>Output</u>

```
| ?- list_perm([a,b,c,d],X).

X = [a,b,c,d] ? a

X = [a,b,d,c]

X = [a,c,b,d]

X = [a,c,d,b]

X = [a,d,b,c]

X = [a,d,c,b]

X = [b,a,c,d]

X = [b,a,d,c]

X = [b,c,a,d]

X = [b,c,d,a]

X = [b,d,a,c]

X = [b,d,c,a]

X = [c,a,b,d]

X = [c,a,d,b]

X = [c,b,a,d]

X = [c,b,d,a]

X = [c,d,a,b]
```

```
X = [c,d,b,a]

X = [d,a,b,c]

X = [d,a,c,b]

X = [d,b,a,c]

X = [d,b,c,a]

X = [d,c,a,b]

X = [d,c,b,a]
```

b) from end

```
list_concat([],L,L).

list_concat([X1|L1],L2,[X1|L3]) :- list_concat(L1,L2,L3).

list_shift([Head|Tail],Shifted) :- list_concat(Tail, [Head],Shifted).
```

Output:

```
| ?- [list_basics].

compiling D:/TP Prolog/Sample_Codes/list_basics.pl for byte code...

D:/TP Prolog/Sample_Codes/list_basics.pl compiled, 14 lines read - 2334 bytes
written, 25 ms

(16 ms) yes

| ?- list_append(a,[e,i,o,u],NewList).

NewList = [a,e,i,o,u]

yes

| ?- list_append(e,[e,i,o,u],NewList).

NewList = [e,i,o,u]

yes

| ?- list_append([a,b],[e,i,o,u],NewList).

NewList = [[a,b],e,i,o,u]

yes

| ?-
```