AIML LAB 2

U18CO021: SAHIL BONDRE

Q.1 Find the last element of a list.

```
?- last([1,2,3,4,5,6,7],X).
X = 7.
```

Q.2 Find the K'th element of a list.

```
?- nth1(2, [1,2,3,4,5], X).
X = 2.
```

Q.3 Find the number of elements in list.

```
?- proper_length([1,2,3,4],X).
X = 4.
```

Q.4 Find out whether a list is a palindrome.

```
palindrome([H|T]):-reverse([H|T],[H|T]).
```

```
?- palindrome([1,2,3,a,b,b,a,3,2,1]).
true.

?- palindrome([1,2,3]).
false.
```

Q.5 Eliminate consecutive duplicates of list elements

```
remove_dups([X],[X]).
remove_dups([X,Y|Z],N):- X = Y, remove_dups([Y|Z],N).
remove_dups([X,Y|Z],[X|N]):- X \= Y, remove_dups([Y|Z],N).
```

```
?- remove_dups([1,2,2,3,3,4,5,2,2],X).
X = [1, 2, 3, 4, 5, 2] .
```

Q.6 Duplicate the elements of a list a given number of times.

```
replicate(X1,N,X2) :- replicate(X1,N,X2,N).
replicate([],_,[],_).
replicate([_|Z],N,Y,0) :- replicate(Z,N,Y,N).
replicate([X|Z],N,[X|Y],K) :- K > 0, K1 is K - 1,
replicate([X|Z],N,Y,K1).
```

```
?- replicate([1, 2, 3, 4], 2, X).
X = [1, 1, 2, 2, 3, 3, 4, 4] .
```

Q.7 Drop every N'th element from a list.

```
drop(X1,N,X2) :- drop(X1,N,X2,N).
drop([],_,[],_).
drop([_|Z],N,Y,1) :- drop(Z,N,Y,N).
drop([X|Z],N,[X|Y],K) :- K > 1, K1 is K - 1, drop(Z,N,Y,K1).
```

```
?- drop([1,2,3,4],3,X).
X = [1, 2, 4] .
```

Q.8 Rotate a list N places to the left

```
split(X1,0,[],X1).
split([X|Z],N,[X|Y],Z1) :- N > 0, N1 is N - 1, split(Z,N1,Y,Z1).
rotate([],_,[]) :- !.
rotate(X1,N,X2) :- length(X1,NX1), N1 is N mod NX1, split(X1,N1,Y1,Y2),
append(Y2,Y1,X2).
```

```
?- rotate([1,2,3,4],2,X).
X = [3, 4, 1, 2] .
```

Q.9 Extract a given number of randomly selected elements from a list.

```
remove_at(X,[X|X1],1,X1).
```

```
remove_at(X,[Y|X1],N,[Y|Y1]) :- N > 1,
   N1 is N - 1, remove_at(X,X1,N1,Y1).
rand_select(_,0,[]).
rand_select(X1,N,[X|Z1]) :- N > 0,
    length(X1,M),
    I is random(M) + 1,
    remove_at(X,X1,I,Y1),
    N1 is N - 1,
    rand_select(Y1,N1,Z1).
```

```
?- rand_select([1,2,3,4,5,6,7],3,X).
X = [6, 2, 5] .
```

Q.10 Generate a random permutation of the elements of a list.

```
rand_perm(X,Y) :- length(X,N), rand_select(X,N,Y).
```

```
?- rand_perm([1,2,3,4,5],X).
X = [2, 1, 4, 5, 3] .
```