Indian Institute of Technology, Indore
Computer Science & Engineering
CS 354N: Assignment II-Prolog
Date- 14-01-2025

Some general instructions:

- Name your file in "Assignment_2_yourRollno.pdf" format.
- Submission of the assignment should be made using the Google Classroom platform only.
- Plagiarism in any form will not be tolerated.
- You are allowed to do only one submission before the deadline. Avoid the multiple submissions. In such case, only the last submitted file will be used for evaluation.
- Last date for submission of the assignment: **21-01-2025**
- Submit a single file (report) containing procedure (screenshot of main procedures/code/Results).

---

1) Write the rules to reverse a list.

2) Write the rules to find the K'th element of a list.
Example:
?- element_at(X,[a,b,c,d,e],3).
X = c

3) Write the rules to find out whether a list is a palindrome. A palindrome can be read forward or backward; e.g. [x,a,m,a,x].

4) Determine the prime factors of a given positive integer. Construct a list containing the prime factors and their multiplicity.
Example:
?- prime_factors_mult(315, L).
L = [[3,2],[5,1],[7,1]]

5) Determine whether two positive integer numbers are coprime. Two numbers are coprime if their greatest common divisor equals 1.
Example: ?- coprime(35, 64).
Yes

6) Transform a list, possibly holding lists as elements into a 'flat' list by replacing each list with its elements (recursively).
Example:
?- my_flatten([a, [b, [c, d], e]], X).
X = [a, b, c, d, e]

7) Eliminate consecutive duplicates of list elements. If a list contains repeated elements they should be replaced with a single copy of the element. The order of the elements should not be changed.

Example: ?- compress([a,a,a,a,b,c,c,a,a,d,e,e,e,e],X).
X = [a,b,c,a,d,e]

8) Generate the combinations of K distinct objects chosen from the N elements of a list. In how many ways can a committee of 3 be chosen from a group of 12 people? We all know that there are C(12,3) = 220 possibilities (C(N,K) denotes the well-known binomial coefficients). For pure mathematicians, this result may be great. But we want to really generate all the possibilities (via backtracking).

Example:
?- combination(3,[a,b,c,d,e,f],L).
L = [a,b,c] ; L = [a,b,d] ; L = [a,b,e] ;
...

9) Sort a list of lists according to length of sublists.

We suppose that a list (InList) contains elements that are lists themselves. The objective is to sort the elements of InList according to their length. E.g. short lists first, longer lists later, or vice versa.

Example:
?- lsort([[a,b,c],[d,e],[f,g,h],[d,e],[i,j,k,l],[m,n],[o]],L).
L = [[o], [d, e], [d, e], [m, n], [a, b, c], [f, g, h], [i, j, k, l]]

10) Sort a list of lists according to length frequency of sublists.

Suppose that a list (InList) contains elements that are lists themselves. But this time the objective is to sort the elements of InList according to their length frequency; i.e. in the default, where sorting is done ascendingly, lists with rare lengths are placed first, others with a more frequent length come later.

Example: ?- lfsort([[a,b,c],[d,e],[f,g,h],[d,e],[i,j,k,l],[m,n],[o]],L).
L = [[i, j, k, l], [o], [a, b, c], [f, g, h], [d, e], [d, e], [m, n]]

Note that in the above example, the first two lists in the result L have length 4 and 1, both lengths appear just once. The third and forth list have length 3; there are two list of this length. And finally, the last three lists have length 2. This is the most frequent length.