Indian Institute of Engineering Science & Technology, Shibpur
Department of Computer Science & Technology,
8th Semester Artificial Intelligence Laboratory,
Logic Programming Assignments - II

1. Write recursive programs in Prolog to
   a) add
   b) subtract
   c) multiply
   d) divide
   two integers.

2. Write Prolog programs to
   a) split a list of integers such that one contains positive integers and other contains negative integers.
   b) count number of integers > 100 in a list
   given integers.

3. Let L and L1 denote two lists of terms. Write Prolog programs to realize the following:
   a) replace the first occurrence of X in L with Y, giving
   the result in L1.
   b) delete n^{th} element in L, leaving the rest in L1.
   c) replace n^{th} element in L by X, giving
   the result in L1.

4. Let L be a list of terms. Write Prolog program for the following definitions.
   a) cutlast(L, L1) that defines L1 to be obtained from L with last element removed.
   b) trim(N, L, L1) that defines L1 to be obtained from L with first N elements removed.
   c) trimlast(N, L, L1)defines that L1 to be obtained from L with last N elements removed.

5. Write a Prolog program for double(List, ListList), where every element in List appears twice in ListList, e. g., double([1,2,3], [1,1,2,2,3,3]) is true.

6. Write Prolog programs to:
   a) calculate factorial(N) (i) without using accumulator
      (ii) using accumulator.
   b) reverse a list (i) without using accumulator
      (ii) using accumulator.
   c) remove duplicate elements from a list (i) without using accumulator
      (ii) using accumulator.
   d) perform Quick sort (i) without using accumulator
      (ii) using accumulator.

7. Write Prolog programs to implement:
   a) Bubble Sort.
   b) Heap Sort.

8. Define a predicate to "flatten" a list by constructing a list containing no lists as elements, but containing all of the atoms of the original list. For example, consider the following goal and its corresponding answer.

   ?- flatten ([a, [b, c], [[d], [ ], e]], L).

   {L= [a, b, c, d, e]}

9. Let L denote a list of integers.
   a) Write a Prolog program **sum (L, S)** that defines S to be the sum of all elements in L.
   b) Write a Prolog program **mean (L, X)** that defines X to be the mean value of elements in L, by referring to the already developed procedure **sum** and **length**.
   c) Write a Prolog program that performs the same task as the one in part (b) without explicitly referring to **sum** and **length**.

10. Let L1, L2 and L denote lists of terms. Write Prolog programs to realize the following.
    a) Interleave alternate elements of L1 and L2 into L. For example,
       if L1= [a, b, c] and
       L2= [1, 2], then L= [a, 1, b, 2, c].

    b) Transpose L1, L2 into L. That is, if L1= [a, b, c] and L2= [1, 2, 3], then L= [(a, 1), (b, 2), (c, 3)].

    c) Suppose that L1 and L2 are lists of numeric values. Write a prolog program Inner_product (L1, L2, X) that defines X to be inner product of two vectors L1, L2.

11. Write Prolog programs for
    a) Deleting middle element of an odd numbered list.
    b) Deleting two middle elements of an even numbered list.
    c) Reversing the elements of a list from 1 to middle-1 elements and middle+1 to last element of an odd numbered list.

12. There is an old song that goes as follows:

        99 bottles of coke on the wall
        99 bottles of coke
        You take one down and pass it around
        98 bottles of coke on the wall

    and so on, until the last verse

        1 bottle of coke on the wall
        1 bottle of coke
        You take one down and pass it around
        No bottle of coke on the wall.

    Write a Prolog program *cola* that receives a positive integer and prints the Lyrics of the song. The program should print all the verses, and when it gets to the last verse, it must print 1 *bottle*, not 1 bottles and *no bottle* rather than 0 bottles.

13. In a lost-world language, a poem can have any number of verses, each of which takes the following form:

A	B	B	C
D	E	E	C
F	F	G	
H	I	I	C

where the same letter represents rhymed words. For example,

anun kura tama su
unuri bimo co kuru
sonen ariten sicom
kana te shime xanadu.

Design a database to store a number of lost-world words and write a Prolog program to produce a poem of a given number of verses.

14. The tower of Hanoi is a game played with three poles and a set of discs. The discs are graded in diameter, and fit onto the poles by means of a hole cut through the centre of each disc. Initially all the discs are on the left-hand pole. The object of the game is to move all the discs onto the centre pole. The right-hand pole can be used as a "spare" pole, a temporary resting place for discs. Each time a disc is moved from one pole to another, two constraints must be observed: only the top disc on a pole can be moved, and no disc may be placed on top of a smaller one.

Write a Prolog program to implement the tower of Hanoi problem as stated above, i. e., given N, no of discs on the left pole, generate the moves that will ultimately transfer all N discs to the centre pole.

15. One way of representing the positive whole numbers is a Prolog terms involving the integer 0 and the successor functor s with one argument. Thus, we represent0 by itself, 1 by s(0), 2 by s(s(0)) and so on. Write definitions of standard arithmetic operations addition, multiplication and subtraction, given the above representation of numbers. For example, the predicate plus may be defined to exhibit the following behavior.

?- plus(s(s(0)), s(s(s(0))), X).

{ X= s(s(s(s(s(0)))))}

that is, 2+3 = 5. Also define the predicate "less than". Also define arithmetic operations, like integer division, remainder of integer division, absolute value and square root.

16. Three musicians of a multinational band take turns playing solo in a piece of music: each plays only once. The pianist plays first. John plays saxophone plays before the Australian. Mark comes from the United States and plays before the violinist. One soloist comes from Japan and one is Sam.

Write a PROLOG program to find out who comes from which country, plays what instrument, and in which order

17. The Ackerman number used in mathematical logic can be calculated using the formula shown below:

$$A(M, N) = \begin{cases} N+1, & \text{if } M=0 \\ A(M-1, 1), & \text{if } N=0 \\ A(M-1, A(M, N-1)), & \text{otherwise.} \end{cases}$$

Write a PROLOG program to determine the Ackerman number given M and N

18. A positive integer is represented by a list of decimal digits. Its next higher permutation is defined to be the next greater integer composed of exactly the same digits. For example, the next higher permutation of 123542 is 124235.

Write a) a declarative Prolog program

b) an efficient procedural Prolog program

that receive a list of decimal digits and return its next higher permutation in a list.

19. Write Prolog programs to implement preorder, inorder and postorder traversals of a binary tree.

20. Given preorder and inorder traversals of a binary tree. Write a Prolog program to obtain postorder traversal of the binary tree.

21. Given postorder and inorder traversals of a binary tree. Write a Prolog program to obtain preorder traversal of the binary tree.

22. a) Write PROLOG programs
    i) To insert an element in a binary search tree.
    ii) To find the minimum element in a binary search tree.
    iii) To delete a leaf node from a binary search tree.
    iv) To linearize a binary tree into a list.

    b) Write TWO DIFFERENT PROLOG programs, using the above programs and any other programs (if needed), to sort an unordered list into an ordered list.

23. Given is a list of arcs in the form [arc(3,2), arc(4,1),…..]. Write PROLOG programs to perform the following tasks:

    i) Test if the arcs in the given list can be rearranged into a continuous path of the form [arc(3,2), arc(2,4), arc(4,1),…], and if so, return the continuous path.
    ii) Test if the arcs in the given list form a cycle, that is, a continuous path of which last arc can be joined to the first one.

24. Write a Prolog program to implement Automatic Theorem Proving in Propositional Logic.

25. Write a Prolog Program to implement unification and generate most general unifier from two sets of clauses.

26. Write a Prolog program to implement Monkey Banana problem in Prolog.