DECLARATIVE PROGRAMMING – 2020
HOMEWORK - I

1. (a) Construct a valid argument in English the validity of which can be captured in first-order predicate logic but not in propositional logic. (5 points)

   PS: You are advised to keep the argument as simple as possible.

(b) Translate the argument into propositional logic (L_1). (5 points)

   PS: a. Specify the key clearly in each translation.
   b. Let the translations be as expressive as possible (i.e., make every possible logical operator explicit).

(c) Re-translate the translation in (b) into Prolog. (20 points)

(d) Translate the argument into first-order predicate logic (L_3). (5 points)

   PS: a. Specify the key clearly in each translation.
   b. Let the translations be as expressive as possible (i.e., make every possible logical operator explicit).

(e) Re-translate the translation in (d) into Prolog. (25 points)

(f) Show that the argument loses its validity when translated into propositional logic with the appropriate Prolog query. (10 points)

   PS: Any of the messages below is to be considered as an indication of invalidity:

   i. "false."
   ii. "ERROR: toplevel: Undefined procedure".

(g) Show that the argument retains its validity when translated into first-order predicate logic with the appropriate Prolog query. (10 points)

   PS: A positive response is an indication of validity.

2. (a) Write a Prolog program to determine whether or not a given list of characters is a palindrome. (15 points)

   PS: A palindrome is a word or group of words that is the same when you read it forwards from the beginning or backwards from the end (e.g., "anastas mum satsana", "level").

   (b) Demonstrate with two queries that your program can distinguish between palindromes and non-palindromes. (5 points)

   PS: ?- palindrome([a, n, a, s, t, a, s, m, u, m, s, a, t, s, a, n, a]).
   true.

   ?- palindrome([k, a, r, p, u, z]).
   false.