

Programming languages (TC-2006)

Homework 16

In this homework, you will practice what you know of the Prolog language. Please consider that the purpose of this homework is to allow you to practice and identify strengths and weaknesses. Then, implement these programs as requested and avoid using any built-in predicates that already do what you are requested to implement.

1 `duplicate` (10%)

Write a program in Prolog that duplicates the elements in a list.

2 `sum` (10%)

Write a program in Prolog that sums all the elements in a list but ignores sublists.

3 `toBinaryString` (10%)

Write a program in Prolog that receives a positive integer and returns a string with the binary representation of such an integer. Please note that the result of `X` is a string, not a list. For completing this assignment you might find useful the predicate `string_concat/3`.

4 `isort` (20%)

Implement insertion sort in Prolog.

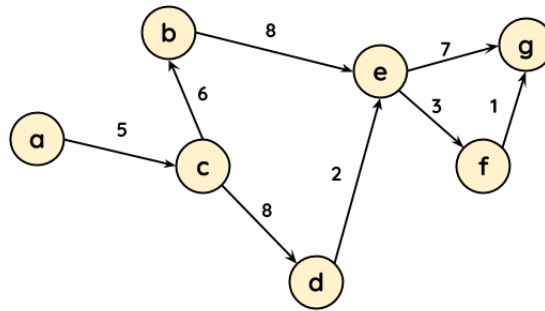
5 `magicSquare` (25%)

A 3×3 magic square is a 3×3 square grid filled with distinct numbers (from 1 to 9) such that the numbers in each row, and in each column, as well as the numbers in the main and secondary diagonals, all add up to 15.

Write a program in Prolog that prints a solution to the 3×3 magic square. For this assignment you might find the predicates `between` and `all_different/1` useful when solving writing this program. To use `all_different/1` you will need to include `use_module(library(bounds))` in your `p1` file.

6 `path` (25%)

Write a program in Prolog that checks if there is at least one path between two given nodes in the following directed graph.



Such a directed graph can be represented in Prolog as:

```

edge(a, c, 5).
edge(c, b, 6).
edge(c, d, 8).
edge(b, e, 8).
edge(d, e, 2).

```

The program must return a valid path (an ordered sequence of the nodes) and the overall cost of such a path. Please note that your program can return a suboptimal path (finding the shortest one is not requested by the assignment).

Deliverables



Prepare a PL file that contains the functions requested (in its corresponding module) and submit it to Canvas. **Please, do not submit other formats but PL.** To prepare your PL file, use the code template distributed along with this document. The template contains some test cases for each program to help you verify that your codes work as requested.



I promise to apply my knowledge, strive for its development, and not use unauthorized or illegal means to complete this activity, following the Tecnológico de Monterrey Student Code of Honor.