



Assignment 1

CSI2120 Programming Paradigms

Winter 2018

Due on February 26th before 11:00 pm in Virtual Campus
6 marks

There are [10 points] in this assignment. The assignment is worth 6% of your final mark.

For Questions 1 and 2 please refer to the database on the last page of this assignment.

Question 1. [2.5 points]

Find the query to obtain the following answer using the `findall` predicate in a) to d) and the `setof` predicate in e):

- a) Obtain a list of persons who work in a city other than the one where they live:
`L = [suzy, paul].`
- b) Obtain a list of Ontario companies
`L = [shopify, rossvideo, uber, google].`
- c) Obtain the list of unemployed persons:
`L = [robert, tim].`
- d) Obtain the list of people working in Ottawa:
`L = [Tom, Jane, Marie, Suzy, Paul, Marie].`
- e) The previous list contains twice the name of Marie, why? Show how one can use `setof` to eliminate this problem.
`L = [jane, marie, paul, suzy, tom].`

Question 2. [2 points]

The predicate below could read as follows:

“A person is busy if they employed by two companies or if their employer is in Toronto.”

`busy (X): - employee (X, C1), employee (X, C2), compare (<, C1, C2).`

`busy (X): - employee (X, C), company (C, Toronto).`

Draw the complete search tree for the following query:

?- busy(X) .

The Prolog search tree shows, for each node, the current goals to be proven. The branches of the tree must show the unification performed in order to pass from a parent node to its child node.

Question 3. [1.5 points]

Give a predicate that calculates the distance between two geographic locations specified by latitude and longitude in radians.

?- distance(45.421016 -75.690018, 45.4222, -75.6824, D) .

This distance can be found as follows (great circle interpolation):

Input: lat_1, lon_1 and lat_2, lon_2 in radians

Output: *distance* in kilometers

$$d_{radians} = 2 * asin \left(\sqrt{ \left(\sin \left(\frac{lat_1 - lat_2}{2} \right) \right)^2 + \cos(lat_1) * \cos(lat_2) * \left(\sin \left(\frac{lon_1 - lon_2}{2} \right) \right)^2 } \right)$$

$$distance = 6371.0 * d_{radians}$$

Note that the GPS location use angles in degree but the above formula expects angles in radians. The angles can be converted as usual:

$$angle_{radians} = pi * \frac{angle_{degrees}}{180}$$

Question 4. [2 points]

Write a predicate `absDiff` that calculates the absolute difference of the corresponding elements of two lists. The result is also a list.

```
?- absDiff([1,3,5,6],[3,5,2,1],L).  
L=[2,2,3,5].
```

This predicate must also work if the two input lists are not the same length. We ask you to create two versions of this predicate:

- a) Assume that the missing elements in the shorter list have the value of 0. The result will be of length equal to the longer list.

```
?- absDiffA([1,3,5,6,2,5],[3,5,2,1],L).  
L=[2,2,3,5,2,5].
```

```
?- absDiffA([1,3,5,6],[3,5,2,1,2,5],L).  
L=[2,2,3,5,2,5].
```

- b) Ignore the extra values of the longer list. The result will be of length equal to the shorter list.

```
?- absDiffB([1,3,5,6,2,5],[3,5,2,1],L).  
L=[2,2,3,5].  
?- absDiffB([1,3,5,6],[3,5,2,1,2,5],L).  
L=[2,2,3,5].
```

Question 5. [2 points]

The predicate `flower(name, color)` is true if a flower exists in a given color.

Write a predicate `bouquet` producing a list of 3 different flowers (i.e., of different names) of which the first two must be red or, if this is not possible, then the three flowers simply have to have different colors.

Note: It is essential to use a cut (!) to avoid solutions with 3 flowers of different colors are produced in cases where there are red flowers.

```
flower(rose,red).  
flower(marigold,yellow).  
flower(tulip,red).  
flower(daffodil,yellow).  
flower(rose,yellow).  
flower(maigold,red).  
flower(rose,white).  
flower(cornflower,purple).
```

Appendix Database for Question 1 and 2.

```
city(ottawa, ontario) .
city(toronto, ontario) .
city(kingston, ontario) .
city(gatineau, quebec) .
city(montreal, quebec) .
company(shopify, ottawa) .
company(rossvideo, ottawa) .
company(dium, gatineau) .
company(uber, toronto) .
company(deepmind, montreal) .
company(google, toronto) .
person(annie, gatineau) .
person(paul, gatineau) .
person(suzy, gatineau) .
person(robert, gatineau) .
person(tom, ottawa) .
person(tim, kingston) .
person(joe, montreal) .
person(jane, ottawa) .
person(marie, ottawa) .
person(jack, toronto) .
person(simon, toronto) .
employee(annie, dium) .
employee(tom, shopify) .
employee(jane, shopify) .
employee(marie, shopify) .
employee(joe, deepmind) .
employee(jack, google) .
employee(simon, google) .
employee(suzy, shopify) .
employee(paul, rossvideo) .
employee(marie, rossvideo) .
employee(simon, uber) .
```