

CS2104 Programming Language Concepts

Lab 3 : Logic Programming Exercise

Please also attach short report (in pdf form) on your solution.
Submit both your report and prolog program in a zip file

(Deadline : 28th October 2016)

- Q1 Design a naive sorting algorithm which make use of two predicates, `permute(Xs,Ys)`, where `Ys` is a permutation of `Xs`; and `ordered(Xs)` which ensures that `Xs` is a sorted list of integers. This solution to sorting is simple but naive since it has exponential complexity to the size of the input list.

```
permute::(List[Int],List[Int]) -> Bool
ordered::(List[Int]) -> Bool
sorted::(List[Int],List[Int]) -> Bool
```

- Q2 The Dutch national flag problem is to take a list of elements that are either red, white, or blue and return a list with all red elements first, followed by all white elements, and then all blue elements last (the order in which they appear on the Dutch national flag). We represent the property of being red, white, or blue with three predicates, `red(x)`, `white(x)`, and `blue(x)`. You may assume that every element of the input list satisfies exactly one of these three predicates. An append predicate is provided for your convenience.

```
append::(List[Int],List[Int],List[Int]) -> Bool
white ::(Int) -> Bool
dutch_flag::(xs:List[Int],rs:List[Int]) -> Bool.
```

- Q3 Consider a maze problem that is being constructed from three predicates, `entry(P)` to denote an entry point `P` of the maze, `exit(P)` to denote an exit point of the maze, and `next(x,y)` to denote an immediate path from point `X` to point `Y`.

- (i) Write a solution

```
mazepath :: (X:Point,Y:Point,P:List[Point]) -> Bool
```

which will find all paths `P` from point `X` to point `Y`.
Your initial solution may assume that the maze is acyclic,

- (ii) Write a better solution that will ensure that a finite path is returned even if there are cycles in the graph.

```
mazepath2 :: (Point,PointList[Point]) -> Bool.
```

Q4 Tom asked his Granny how old she was. Rather than giving him a straight answer, she replied:

"I have 6 children, and there are 4 years between each one and the next. I had my first child (your Uncle Peter) when I was 19. Now the youngest one (Your Auntie Jane) is 19 herself. That's all I'm telling you!"

Your initial solution may be in the following format

```
granny_age :: (Age: Int, Peter_age: Int, Jane_age: Int) -> Bool
```

Generalise your solution to

```
granny_age_N :: (Age: Int, Peter_age: Int, Jane_age: Int, N: Int)
```

where N is the number of children that granny has, and where Peter_age is age of the oldest child, while Jane_age is the age of the youngest child.

Q5 A kid goes into a grocery store and buys four items. The cashier charges \$7.11. The kid pays and is about to leave when the cashier calls the kid back, and says "Hold on, I multiplied the four items instead of adding them; I'll try again... Gosh, with adding them the price still comes to \$7.11"! What were the prices of the four items?

```
grocery :: (Items: List[Int]) -> Bool
```