

CSCE 625**Programming Assignment #8****due: Friday, April 24 (by start of class)**PROLOG

1. Write Prolog clauses that can be used to compute the **intersection** of two lists.

```
?- intersection([1,2,3,4,5,6],[2,4,6,8],V).
```

```
V = [2, 4, 6]
```

2. Meal planning. Write a Prolog program that can be used to determine whether a list of foods is a good meal. For example,

```
?- good_meal([hamburger,salad,banana,water]).
```

```
true
```

```
?- good_meal([coffee,pie]).
```

```
false
```

Here are some properties to help define a good meal. A good meal should be nutritious, by which we mean that it must contain at least 1 fruit_or_vegetable, and at least 1 source of protein. Also, the total calories should be between 400 and 600. (You will have to write set of clauses to add up the calories of each of the items in a meal list.) You can define the calories for each item like this (you can look up representative values on a web site like Wikipedia):

```
calories(water,0).
```

```
calories(hamburger,354).
```

```
calories(carrot,25).
```

```
calories(salad,100).
```

```
calories(banana,105).
```

```
calories(apple,95).
```

```
calories(peanuts,828).
```

```
calories(chicken_soup,87).
```

```
calories(lasagna,166). % traditional meat lasagna
```

```
calories(apple_pie,67).
```

```
calories(bean,41).
```

```
calories(peas,118).
```

```
calories(milk,8).
```

```
calories(orange_juice,39).
```

```
calories(coke,140).
```

```
calories(cookie,132).
calories(chicken_soup,87).
calories(naan,78).
calories(potato_soup,149).
% you may add any of your favorite foods too...
```

You can define the properties of individual food items like this:

```
meat(hamburger).
vegetable(carrot).
fruit(apple).
fruit(banana).
drink(water).
drink(coke).
...
```

Remember that things like orange_juice and apple_pie can also be considered fruits.

Finally, write a rule that defines a vegetarian meal as one that does not contain any meat. Note that other sources of protein besides meats include legumes, such as peas, beans, and peanuts. Milk too.

Show an example of a good meal that is vegetarian, and one that is non-vegetarian (as a Prolog query). Also show the total calorie counts for these meals.

(hint: My implementation has facts and rules for: meat(X), legume(X), protein(X), drink(X), fruit(X), vegetable(X), fruit_or_vegetable(X), total(+Meal,-Cals), contains_protein(M), contains_fruit_or_vegetable(M), nutritious(M), and good_meal(M).)

3. Write Prolog clauses that can be used to calculate **square roots** by Newton-Raphson iteration. (see http://en.wikipedia.org/wiki/Newton%27s_method)

?- sqrt(10,X).

X = 3.162288819352989

Hint: write a clause for sqrt(+In,-Out), which calls a helper function that looks like this: sqrt(+Target,+CurrentEst,+Tolerance,-FinalAnswer), which is a recursive predicate that keeps calling itself until Current Est is close enough to the correct answer. Target is the input value you are taking the square root of, like 9. FinalAnswer is the output value, like 3.000015..., i.e. the square root that you are trying to compute and that ultimately gets returned. CurrentEst is another input argument that starts out with a random guess of the answer (e.g. initialized to Target), and that keeps getting closer to the true square root with each step in the recursion. Finally, Tolerance is used to define the base case, i.e. when the square of CurrentEst is close enough (absolute value of difference is within Tolerance) to the Target value, return it as FinalAnswer.

4. Solve the **5-queens problem** in Prolog.

?- queens(A,B,C,D,E).

A = 5 , B = 3 , C = 1 , D = 4 , E = 2;

...

(there are 10 solutions; show them all)

Each variables above are the Columns of the queens in Rows A-E. Just for fun, I wrote a print_queens(A,B,C,D,E) predicate to visualize what this looks like using format():

```
- - - - Q
- - Q - -
Q - - - -
- - - Q -
- Q - - -
```

What to Turnin

1. Submit your Prolog **source code** via the Turnin program on CSNet.
2. Also include a **transcript** showing your solutions to the problems above.