

CSC 600-01 (SECTION 1)
Homework 3 - Logic Programming in Prolog
prepared by Ilya Kopyl

CSC 600 HOMEWORK 3 - LOGIC PROGRAMMING IN PROLOG

March 13, 2018

*Homework is prepared by: Ilya Kopyl.
It is formatted in LaTeX, using TeXShop editor (under GNU GPL license).
Diagrams are created in LucidChart online editor (lucidchart.com).*

1. Write a PROLOG program that investigates family relationships using lists. The facts should be organized as follows:

```
m([first_male_name, second_male_name, ..., last_male_name]).  
f([first_female_name, second_female_name, ..., last_female_name]).  
family( [father, mother, [child1, child2, ..., child_n]] ).
```

Write rules that define the following relationships:

```
male(X)  
female(X)  
father, mother, parent  
siblings1, siblings2  
brother1, brother2  
sister1, sister2  
cousins  
uncle, aunt  
grandchild, grandson, granddaughter  
greatgrandparent  
ancestor
```

For each of these rules show an example of its use.

The answer is listed on the pages TBD through TBD.

The code listing of maxlen function:

```
somePredicate(A, B) :-  
    arbitraryPredicate(A, _, 1, 2),  
    predicateWithAtom(someAtom),  
    anotherPredicate(B, someAtom, myPredicate(A, _)),  
    findall(X, ('testString'(X), myPredicate(A, X)), L1),  
    member(A, L1),  
    !.
```

The code listing of main program:

```
somePredicate(A, B) :-  
    arbitraryPredicate(A, _, 1, 2),  
    predicateWithAtom(someAtom),  
    anotherPredicate(B, someAtom, myPredicate(A, _)),  
    findall(X, ('testString'(X), myPredicate(A, X)), L1),  
    member(A, L1),  
    !.
```

The result of the program execution:

Standard output:

2. Integer plot function (find a smart way to code big integers)

Write a program `BigInt(n)` that displays an arbitrary positive integer `n` using big characters of size 7x7, as in the following example for `BigInt(170)`:

Standard output :

Write a demo main program that illustrates the work of `BigInt(n)` and prints the following sequence of big numbers 1, 12, 123, 1234,..., 1234567890, one below the other.

The answer is listed on the pages 7 through 9.

The code listing of the two-dimensional array that stores bit pattern of each `BigInt` digit. It is declared in the global space (outside of any function).

```
somePredicate(A, B) :-  
    arbitraryPredicate(A, _, 1, 2),  
    predicateWithAtom(someAtom),  
    anotherPredicate(B, someAtom, myPredicate(A, _)),  
    findall(X, ('testString'(X), myPredicate(A, X)), L1),  
    member(A, L1),  
    !.
```

Main program, excluding the declaration of BIG_DIGITS array:

```
somePredicate(A, B) :-  
    arbitraryPredicate(A, _, 1, 2),  
    predicateWithAtom(someAtom),  
    anotherPredicate(B, someAtom, myPredicate(A, _)),  
    findall(X, ('testString'(X), myPredicate(A, X)), L1),  
    member(A, L1),  
    !.
```

3. Array processing (elimination of three largest values) (one of many array reduction problems)

The array $a(1..n)$ contains arbitrary integers. Write a function $\text{reduce}(a, n)$ that reduces the array $a(1..n)$ by eliminating from it all values that are equal to three largest different integers. For example, if $a=(9, 1, 1, 6, 7, 1, 2, 3, 3, 5, 6, 6, 6, 6, 7, 9)$ then three largest different integers are 6, 7, 9, and after reduction the reduced array would be $a=(1, 1, 1, 2, 3, 3, 5)$, $n=7$. The time complexity of the solution should be in $O(n)$.

The answer is listed on the pages 11 through 13.

The code listing of the entire program for problem #3:

```
somePredicate(A, B) :-  
    arbitraryPredicate(A, _, 1, 2),  
    predicateWithAtom(someAtom),  
    anotherPredicate(B, someAtom, myPredicate(A, _)),  
    findall(X, ('testString'(X), myPredicate(A, X)), L1),  
    member(A, L1),
```