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. Plateau program (max sequence length) (a combinatorial algorithm)

The array a(1..n) contains sorted integers. Write a function maxlen(a,n) that returns the length of the longest sequence of identical numbers (for example, if a= 1, 1, 1, 2, 3, 3, 5, 6, 6, 6, 6, 7, 9 then maxlen returns 4 because the longest sequence 6, 6, 6, 6 contains 4 numbers. Write a demo main program for testing the work of maxlen. Explain your solution, and insert comments in your program. The time complexity of the solution should be in O(n).

The answer is listed on the pages 2 through 5.

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),
    !.
```

Auxiliary functions (in separate file "functions.c"):

```
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 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),
```