cps721: Assignment 3 (100 points).
Due date: Electronic file - Thursday, November 9, 2017, 17:00pm (sharp).
YOU SHOULD NOT USE ";" , "!" AND "–>" IN YOUR PROLOG RULES.

> You must work in groups of TWO, or THREE. You can**not** work alone. You can discuss this assignment only with your CPS721 group partners or with the CPS721 instructor. By submitting this assignment you acknowledge that you read and understood the course Policy on Collaboration in homework assignments stated in the CPS721 course management form.

This assignment asks you to build a simple natural language system for answering "who" and "what" questions about noun phrases, much like the example in class, but instead of using the datebase about parks, hats, people, etc, you have to create a database about people, their relatives, friends, the cities they live in, and the countries of those cities, and so on. Use the following predicates:

1. Before we start using any English words, build in Prolog a simple database of facts (e.g., about 10 facts of each type) with the following predicates:

- `male(P):P` is a male.
- `female(P):P` is a female.
- `person(P):P` is a person.
- `home(P,C):P` lives in city `C`.
- `city(C):C` is a city.
- `country(C):C` is a country.
- `location(City,Country):City` is located in `Country`.
- `population(X,Y)`: The size of city `X` is `Y`. (For the purposes of this assignment, every city with population less than 50,000 is considered to be small, and every city with population larger than or equal to 50,000 is defined to be big.)
- `married(P1,P2):P1` is a person married to a person `P2`.
- `parent(P,C):P` is a parent of `C`.
- `friend(P1,P2):P1` is a person who is a friend of a person `P2`.

Keep your database (and Prolog rules implementing the predicates $father, mother, brother, sister, ancestor$, $grandmother, grandfather, uncle, auntie, relative$) in the file **nlu.pl** The facts included in your database can be imaginary; make it up so that most queries will get positive answers and will retrieve some data from your data base.

Show that your database works properly by formulating the testing queries in Prolog (similar to what you did in the first assignment), and obtaining suitable answers. (These queries should *not* use English noun phrases!) It is up to you to formulate a range of queries to convince yourself and the TA that your data base and programs are designed correctly, but at least 10 queries are required. Keep your queries and answers computed by Prolog in the file **nlu.txt**

Once you have this working, you are ready to consider English noun phrases and the people, cities and countries they refer to. Here are some examples of the kinds of queries your system should be able to handle:

1. what([a,small,city,in,canada], X).

2. what([a,largest,city,in,canada], B).

3. who([any,friend,from,toronto,of,a,single,man,from,a,large,city], Name).

4. who([a,married,person,from,toronto,with,a,relative,from,montreal], Who).

5. who([a,grandmother,from,canada,with,a,husband,from,toronto], W).

6. who([a,father,from,a,small,city,in,canada,with,a,wife,from,montreal], P).

7. who([a,canadian,child,from,a,small,city,with,a,sister], P).

8. who([an,american,woman,with,a,canadian,ancestor], W).

9. who([a,married,person,with,a,single,american,friend], W).

10. who([a,brother,of,a,married,woman,from,a,largest,city,in,usa], W).

2. Build a lexicon, as we did in class, of articles, adjectives, proper nouns, common nouns, and prepositions. To cover the above examples, you will need at least these words :

articles: `a, an, any`

common nouns: `man, woman, father, mother, wife, husband, child, grandfather, grandmother,brother,sister,uncle,auntie,parent,ancestor,relative,city, country,...`

prepositions: `with, of, from, in, ...`

proper nouns: `toronto, losAngeles, england, montreal, canada, ...`

adjectives: `big,small,largest,smallest,single,married,american,canadian,chinese,...`

For the purposes of this assignment, a Canadian (American) is defined to be anyone living in a city of Canada (USA). You may actually need more words than those mentioned explicitly above.

You should **not** introduce any new predicates into your database, i.e., your atomic statements can use only the 11 predicates mentioned above in Part 1, and the heads of your rules in Part 1 can use only the predicates mentioned there. Your lexicon should include all the words mentioned above (in total, **30 words or more**, apart from articles and proper nouns). The word "any" should be treated as an article. Notice that some words are ambiguous, and for this reason, you need several rules for them in your lexicon: one rule per possible meaning. (For comparison, consider the 4 rules for the preposition "with" discussed in class.) Remember that it is easy to defeat a language understanding program by using a word that it does not know about. Vocabulary is important in these systems. Make yours as smart as you can. Three (3) best Prolog programs that will do significatly more work than specified explicitly in this assignment will get bonus marks (e.g., implement 15 or more additional common noun and adjectives, or use in your lexicon nouns, prepositions and adjectives from the language other than English). Bonus marks will be given at the discretion of the TA: minor variations will not be awarded any extra marks. Demonstrate your creativity to get bonus marks!

3. Copy the noun phrase parser/interpreter given in class (or write your own), and define the `what` predicate used above. The parser must be also in the same file **nlu.pl**

4. Test the `what` and `who` predicates on a variety of noun phrases, like those above, showing that it is capable of identifying the entities being referred to by your noun phrases. It is up to you to choose noun phrases for testing, but you must convincingly demonstrate that your program works properly (try at least 10 new different noun phrases in addition to the phrases given to you). Remember that testing your program is very important part of the software development cycle. *You lose marks if you do not test your program as required.* Copy all results of your tests into **nlu.txt**

5. Do not attempt this work until the previous parts of your assignment are complete. The English noun phrases described above are quite limited. Generalize your program to handle some additional features of English:

- Handle the article "the" properly. The trick here is that a noun phrase like "the friend of a parent from a small city with a married child from chicago" should succeed in naming a person if there is a *unique* person of the appropriate kind, but if the parent mentioned in the phrase happened to have several friends, then the query should fail. Similarly, "the largest city in Canada" should retrieve the unique city that has the largest population.

Handing in solutions. Upload to D2L a ZIP file that includes the following: (a) your database, lexicon and parser in a single file (the name of the file must be **nlu.pl**), (b) your session with Prolog, showing the queries you submitted and the answers returned (the name of the file must be **nlu.txt**). It is up to you to formulate a range of queries that demonstrates that your program is working properly.

How to submit this assignment. Read regularly *Frequently Answered Questions* and replies to them that are linked from the Assignments Web page at

`http://www.scs.ryerson.ca/~mes/courses/cps721/assignments.html`

If you write your code on a Windows machine, make sure you save your files as plain text that one can easily read on Linux machines. Before you submit your Prolog code electronically make sure that your files do not contain any extra binary symbols: it should be possible to load `nlu.pl` into a recent release 6 of ECLiPSe Prolog, compile your program and ask testing queries. TA will mark your assignment using ECLiPSe Prolog. If you run any other version of Prolog on your home computer, it is your responsibility to make sure that your program will run on ECLiPSe Prolog (release 6 or any more recent release), as required. For example, you can run a command-line version of *eclipse* on moon remotely from your home computer to test your program (read handout about running *ECLiPSe Prolog*). To submit files electronically do the following. First, create a **zip** archive on `moon`:

```
zip yourLoginName.zip nlu.pl nlu.txt
```

where `yourLoginName` is the login name of the person who submits this assignment from a group. Remember to mention at the beginning of each file *student, section numbers* and *names* of all people who participated in discussions (see the course management form). You may be penalized for not doing so. Second, upload **your ZIP** file only (**No individual files!**) **yourLoginName.zip** into the "Assignment 4" folder on D2L.

Revisions: If you would like to submit a revised copy of your assignment, then run simply the submit command again. (The same person must run the submit command.) A new copy of your assignment will override the old copy. You can submit new versions as many times as you like and you do not need to inform anyone about this. Don't ask your team members to submit your assignment, because TA will be confused which version to mark: only one person from a group should submit different revisions of the assignment. The time stamp of the last file you submit will determine whether you have submitted your assignment on time.