Logic Programming

1. Encode the following facts and rules in pyDatalog:http://localhost:8888/notebooks/Downloads/New folder/pyDataLog (1).ipynb - Encode-the-following-facts-and-rules-in-pyDatalog:

 Bear is bighttp://localhost:8888/notebooks/Downloads/New folder/pyDataLog (1).ipynb - %EF%82%B7-Bear-is-big

 Elephant is big¶

 Cat is smallhttp://localhost:8888/notebooks/Downloads/New folder/pyDataLog (1).ipynb - %EF%82%B7-Cat-is-small

 Bear is brownhttp://localhost:8888/notebooks/Downloads/New folder/pyDataLog (1).ipynb - %EF%82%B7-Bear-is-brown

 Cat is blackhttp://localhost:8888/notebooks/Downloads/New folder/pyDataLog (1).ipynb - %EF%82%B7-Cat-is-black

 Elephant is grayhttp://localhost:8888/notebooks/Downloads/New folder/pyDataLog (1).ipynb - %EF%82%B7-Elephant-is-gray

 An animal is dark if it is blackhttp://localhost:8888/notebooks/Downloads/New folder/pyDataLog (1).ipynb - %EF%82%B7-An-animal-is-dark-if-it-is-black

 An animal is dark if it is brownhttp://localhost:8888/notebooks/Downloads/New folder/pyDataLog (1).ipynb - %EF%82%B7-An-animal-is-dark-if-it-is-brown

Write a query to find which animal is dark and big.

2. Write a recursive program to find factorial of a number using pyDatalog.

3. Implement using pyDatalog:http://localhost:8888/notebooks/Downloads/New folder/pyDataLog (1).ipynb - Implement-using-pyDatalog:

Assume given a set of facts of the form father(name1,name2) (name1 is the fatherhttp://localhost:8888/notebooks/Downloads/New folder/pyDataLog (1).ipynb - Assume-given-a-set-of-facts-of-the-form-father(name1,name2)-(name1-is-the-father

of name2).http://localhost:8888/notebooks/Downloads/New folder/pyDataLog (1).ipynb - of-name2).

a. Define a predicate brother(X,Y) which holds iff X and Y are brothers.http://localhost:8888/notebooks/Downloads/New folder/pyDataLog (1).ipynb - a.-Define-a-predicate-brother(X,Y)-which-holds-iff-X-and-Y-are-brothers.

b. Define a predicate cousin(X,Y) which holds iff X and Y are cousins.http://localhost:8888/notebooks/Downloads/New folder/pyDataLog (1).ipynb - b.-Define-a-predicate-cousin(X,Y)-which-holds-iff-X-and-Y-are-cousins.

c. Define a predicate grandson(X,Y) which holds iff X is a grandson of Y.http://localhost:8888/notebooks/Downloads/New folder/pyDataLog (1).ipynb - c.-Define-a-predicate-grandson(X,Y)-which-holds-iff-X-is-a-grandson-of-Y.

d. Define a predicate descendent(X,Y) which holds iff X is a descendent of Y.http://localhost:8888/notebooks/Downloads/New folder/pyDataLog (1).ipynb - d.-Define-a-predicate-descendent(X,Y)-which-holds-iff-X-is-a-descendent-of-Y.

e. Consider the following genealogical tree:http://localhost:8888/notebooks/Downloads/New folder/pyDataLog (1).ipynb - e.-Consider-the-following-genealogical-tree:

ahttp://localhost:8888/notebooks/Downloads/New folder/pyDataLog (1).ipynb - a

/ \http://localhost:8888/notebooks/Downloads/New folder/pyDataLog (1).ipynb - /-\

b chttp://localhost:8888/notebooks/Downloads/New folder/pyDataLog (1).ipynb - b-c

/ \ |http://localhost:8888/notebooks/Downloads/New folder/pyDataLog (1).ipynb - /-\--|

d e fhttp://localhost:8888/notebooks/Downloads/New folder/pyDataLog (1).ipynb - d-e-f

What are the answers generated by your definitions for the queries:

4. Write a program to check prime numbers in python logic programming.