package com.ankur.voice\_boston;

import java.io.PrintStream;

import java.util.HashMap;

import java.util.Iterator;

import java.util.Map;

import java.util.Map.Entry;

import java.util.Set;

public class ExpressionFormatter

{

private static Map<String, String> numberMap;

private static Map<String, String> operatorMap = new HashMap();

private String input;

static

{

numberMap = new HashMap();

operatorMap.put(",", "");

operatorMap.put("dot", ".");

operatorMap.put("point", ".");

operatorMap.put("comma", ".");

operatorMap.put("divided by", "/");

operatorMap.put("divide", "/");

operatorMap.put("divided", "/");

operatorMap.put("by", "/");

operatorMap.put("buy", "/");

operatorMap.put("shared by", "/");

operatorMap.put("slash", "/");

operatorMap.put("at", "+");

operatorMap.put("and", "+");

operatorMap.put("plus", "+");

operatorMap.put("add", "+");

operatorMap.put("had", "+");

operatorMap.put("x", "+");

operatorMap.put("place", "+");

operatorMap.put("bless", "+");

operatorMap.put("glad", "+");

operatorMap.put("ad", "+");

operatorMap.put("ed", "+");

operatorMap.put("left", "+");

operatorMap.put("let", "+");

operatorMap.put("day", "+");

operatorMap.put("earth", "+");

operatorMap.put("that", "+");

operatorMap.put("last", "+");

operatorMap.put("act", "+");

operatorMap.put("as", "+");

operatorMap.put("subtract", "-");

operatorMap.put("minus", "-");

operatorMap.put("take away", "-");

operatorMap.put("take-away", "-");

operatorMap.put("takeaway", "-");

operatorMap.put("dash", "-");

operatorMap.put("wire", "-");

operatorMap.put("my net", "-");

operatorMap.put("mynet", "-");

operatorMap.put("mire", "-");

operatorMap.put("times", "\*");

operatorMap.put("multiply", "\*");

operatorMap.put("multiplied by", "\*");

operatorMap.put("multiplied", "\*");

operatorMap.put("asterisk", "\*");

operatorMap.put("into", "\*");

operatorMap.put("in to", "\*");

operatorMap.put("in too", "\*");

operatorMap.put("in two", "\*");

operatorMap.put("in toe", "\*");

operatorMap.put("in tow", "\*");

operatorMap.put("indoor", "\*");

operatorMap.put("in 2", "\*");

operatorMap.put("it back by", "\*");

operatorMap.put("might lead by ", "\*");

operatorMap.put("make it back by", "\*");

operatorMap.put("to the power of", "^");

operatorMap.put("to the power off", "^");

operatorMap.put("2 the power of", "^");

operatorMap.put("tothepowerof", "^");

operatorMap.put("2thepowerof", "^");

operatorMap.put("indice", "^");

operatorMap.put("to factor", "^");

operatorMap.put("factor", "^");

operatorMap.put("to the poor of", "^");

operatorMap.put("to the par of", "^");

operatorMap.put("to the part of", "^");

// operatorMap.put("power", "^");

operatorMap.put("lock", "&");

operatorMap.put("logarithm", "&");

operatorMap.put("log", "&");

//operatorMap.put("logarithm", "&");

operatorMap.put("root", "@");

operatorMap.put("note", "@");

operatorMap.put("under root", "@");

operatorMap.put("root", "@");

operatorMap.put("squareroot", "@");

operatorMap.put("squareroot", "@");

operatorMap.put("squareroute", "@");

operatorMap.put("root", "@");

operatorMap.put("rude", "@");

operatorMap.put("route", "@");

operatorMap.put("mod", "%");

operatorMap.put("percentage", "%");

operatorMap.put("what is", "");

operatorMap.put(" ", "");

numberMap.put("pi", String.valueOf(3.141592653589793D));

numberMap.put("pie", String.valueOf(3.141592653589793D));

numberMap.put("newyork", "0");

numberMap.put("feel", "0");

numberMap.put("deal", "0");

numberMap.put("xero", "0");

numberMap.put("hero", "0");

numberMap.put("aero", "0");

numberMap.put("nero", "0");

numberMap.put("youtube", "0");

numberMap.put("zero", "0");

numberMap.put("not", "0");

numberMap.put("nought", "0");

numberMap.put("nil", "0");

numberMap.put("zilch", "0");

numberMap.put("one", "1");

numberMap.put("won", "1");

numberMap.put("none", "1");

numberMap.put("done", "1");

numberMap.put("when", "1");

numberMap.put("son", "1");

numberMap.put("on", "1");

numberMap.put("lone", "1");

numberMap.put("wane", "1");

numberMap.put("throne", "1");

numberMap.put("brain", "1");

numberMap.put("vine", "1");

numberMap.put("glan", "1");

numberMap.put("month", "1");

numberMap.put("one", "1");

numberMap.put("two", "2");

numberMap.put("too", "2");

numberMap.put("to", "2");

numberMap.put("tow", "2");

numberMap.put("you", "2");

numberMap.put("google", "2");

numberMap.put("tube", "2");

numberMap.put("do", "2");

numberMap.put("toe", "2");

numberMap.put("to", "2");

numberMap.put("three", "3");

numberMap.put("tree", "3");

numberMap.put("free", "3");

numberMap.put("thrie", "3");

numberMap.put("t", "3");

numberMap.put("p", "3");

numberMap.put("c", "3");

numberMap.put("e", "3");

numberMap.put("eat", "3");

numberMap.put("we", "3");

numberMap.put("leave", "3");

numberMap.put("turry", "3");

numberMap.put("threw", "3");

numberMap.put("leave", "3");

numberMap.put("please", "3");

numberMap.put("deal", "3");

numberMap.put("the deal", "3");

numberMap.put("four", "4");

numberMap.put("for", "4");

numberMap.put("phor", "4");

numberMap.put("fore", "4");

numberMap.put("or", "4");

numberMap.put("force", "4");

numberMap.put("fall", "4");

numberMap.put("bore", "4");

numberMap.put("bored", "4");

numberMap.put("phone", "4");

numberMap.put("voiles", "4");

numberMap.put("five", "5");

numberMap.put("wife", "5");

numberMap.put("knife", "5");

numberMap.put("funny", "5");

numberMap.put("six", "6");

numberMap.put("sex", "6");

numberMap.put("sicks", "6");

numberMap.put("sick", "6");

numberMap.put("fix", "6");

numberMap.put("sics", "6");

numberMap.put("hey", "6");

numberMap.put("seven", "7");

numberMap.put("sven", "7");

numberMap.put("saving", "7");

numberMap.put("sern", "7");

numberMap.put("devon", "7");

numberMap.put("heaven", "7");

numberMap.put("eight", "8");

numberMap.put("hate", "8");

numberMap.put("ate", "8");

numberMap.put("eat", "8");

numberMap.put("it", "8");

numberMap.put("wait", "8");

numberMap.put("weight", "8");

numberMap.put("faith", "8");

numberMap.put("nine", "9");

numberMap.put("nein", "9");

numberMap.put("fine", "9");

numberMap.put("wine", "9");

numberMap.put("shine", "9");

numberMap.put("ten", "10");

numberMap.put("then", "10");

}

public ExpressionFormatter(String paramString)

{

input = paramString;

}

private void formatNumbers()

{

Iterator localIterator = numberMap.entrySet().iterator();

while (true)

{

if (!localIterator.hasNext())

return;

Map.Entry localEntry = (Map.Entry)localIterator.next();

if (!this.input.contains((CharSequence)localEntry.getKey()))

continue;

System.out.println("[VOICECALCULATOR] Replaced '" + (String)localEntry.getKey() + "' with '" + (String)localEntry.getValue() + "'.");

this.input = this.input.replaceAll((String)localEntry.getKey(), (String)localEntry.getValue());

}

}

private void formatOperators()

{

Iterator localIterator = operatorMap.entrySet().iterator();

while (true)

{

if (!localIterator.hasNext())

return;

Map.Entry localEntry = (Map.Entry)localIterator.next();

if (!this.input.contains((CharSequence)localEntry.getKey()))

continue;

System.out.println("[VOICECALCULATOR] Replaced '" + (String)localEntry.getKey() + "' with '" + (String)localEntry.getValue() + "'.");

this.input = this.input.replaceAll((String)localEntry.getKey(), (String)localEntry.getValue());

}

}

public String format()

{

formatOperators();

formatNumbers();

return this.input;

}

}