Artificial Intelligence

Asssignment 4

Name : Swati Anil Sonone

Div : CS\_D

Roll no. : 53

Problem Statement : CryptArithmetic Problem

Code :

import java.util.ArrayList;

import java.util.ArrayList;

import java.util.HashMap;

public class Arithmetics {

    private String s;

    private String sol;

    private HashMap<Character,Integer> theMap = new HashMap<>();

    private ArrayList<Character> CharacterList = new ArrayList<>();

    private ArrayList<Character> NonZeroChar = new ArrayList<>();

    Arithmetics(String s){

        this.s = s;

        this.theMap = new HashMap<>();

        this.CharacterList = new ArrayList<>();

        this.NonZeroChar = new ArrayList<>();

        solve();

    }

    void solve(){

        String[] arr = this.s.split(" [\\+,=] ");

        int maxString = 0;

        for(String string : arr){

            this.NonZeroChar.add(string.charAt(0));

            if(maxString<string.length()){

                maxString = string.length();

            }

        }

        char[][] charMatrix = new char[arr.length][maxString];

        for(int i=0 ;i<arr.length;i++){

            char[] stringToArr = arr[i].toCharArray();

            for(int j= stringToArr.length-1;j>=0;j--){

                charMatrix[i][maxString-(stringToArr.length - j)] = stringToArr[j];

                this.theMap.put(stringToArr[j], -1);

            }

        }

        for (int j = charMatrix[0].length - 1; j >= 0; j--) {

            for (int i = 0; i < charMatrix.length; i++) {

                if (Character.isAlphabetic(charMatrix[i][j])) {

                    this.CharacterList.add(charMatrix[i][j]);

                }

                if (i == charMatrix.length - 1) {

                    this.CharacterList.add('=');

                }

            }

        }

        for(int i=0;i< CharacterList.size();i++){

            System.out.print(CharacterList.get(i)+" ");

        }

        System.out.println();

        for(int i=0;i< NonZeroChar.size();i++){

            System.out.print(NonZeroChar.get(i)+" ");

        }

        if(this.theMap.isEmpty()||this.CharacterList.isEmpty()){

            return;

        }

        if(!findAlphametics(0,0,0)){

            return;

        }

        String sol;

        sol = this.s;

        for (Character c : this.theMap.keySet()) {

            sol = sol.replace(c, Character.forDigit(this.theMap.get(c), 10));

        }

        this.sol = sol;

    }

    private boolean findAlphametics(int index, int sum, int carry) {

        if (this.CharacterList.get(index + 1) == '=') {

            sum += carry;

            int value = sum % 10;

            if (this.theMap.get(this.CharacterList.get(index)) == -1) {

                if (!this.theMap.containsValue(value) && !(this.NonZeroChar.contains(this.CharacterList.get(index)) && value == 0)) {

                    this.theMap.put(this.CharacterList.get(index), value);

                    if (this.CharacterList.size() == index + 2) {

                        return sum <= 9;

                    }

                    if (findAlphametics(index + 2, 0, sum / 10)) {

                        return true;

                    }

                    this.theMap.put(this.CharacterList.get(index), -1);

                } else {

                    return false;

                }

            } else {

                if (this.theMap.get(this.CharacterList.get(index)) == value) {

                    if (this.CharacterList.size() == index + 2) {

                        return sum <= 9;

                    }

                    if (findAlphametics(index + 2, 0, sum / 10)) {

                        return true;

                    }

                } else {

                    return false;

                }

            }

            return false;

        }

        if (this.theMap.get(this.CharacterList.get(index)) == -1) {

            for (int i = 0; i <= 9; i++) {

                if (!this.theMap.containsValue(i) && !(this.NonZeroChar.contains(this.CharacterList.get(index)) && i == 0)) {

                    this.theMap.put(this.CharacterList.get(index), i);

                    if (findAlphametics(index + 1, sum + i, carry)) {

                        return true;

                    }

                }

            }

            this.theMap.put(this.CharacterList.get(index), -1);

            return false;

        } else {

            if (findAlphametics(index + 1, sum + this.theMap.get(this.CharacterList.get(index)), carry)) {

                return true;

            }

        }

        return false;

    }

    public static void main(String[] arg){

        String s = "HOW + MUCH = POWER";

        Arithmetics a= new Arithmetics(s);

        System.out.println(a.theMap);

        System.out.println(a.s);

        System.out.println(a.sol);

    }

}

Output

