```java
/**
 * Name         :   Muhammad Ahmed bin Anwar Bahajjaj
 * Matric. No   :   A0140051A
 * PLab Acct.   :   -
 */

import java.util.*;

class Main {    //Ranking
    private void run() {
        Scanner sc = new Scanner(System.in);
        ArrayList<Student> list = new ArrayList<>();
        int num = sc.nextInt();
        for (int i = 0; i < num; i++) {
            list.add(new Student(sc.next(), sc.nextInt()));
        }
        InputComparator inComp = new InputComparator();
        ScoreComparator scComp = new ScoreComparator();
        Collections.sort(list, scComp);
        for (int i = 0; i < list.size(); i++) {
            if (i == 0) {
                list.get(i).setRank(i+1);
            } else if (list.get(i).getScore() == list.get(i-1).getScore()) {
                list.get(i).setRank(list.get(i-1).getRank());
            } else {
                list.get(i).setRank(i+1 );
            }
        }
        Collections.sort(list, inComp);
        for (Student stu : list) {
            System.out.println(stu);
        }
    }

    public static void main(String[] args) {
        Main newRanking = new Main();
        newRanking.run();
    }
}

class Student {
    private static int x = 0;
    private String _name;
    private int _score;
    private int _rank;
    private int _order;

    public Student(String name, int score) {
        _name = name;
        _score = score;
        _order = x;
        x++;
    }

    public int getScore() {
        return _score;
    }

    public int getOrder() {
        return _order;
    }

    public int getRank() {
        return _rank;
    }

    public void setRank(int rank) {
        _rank = rank;
    }

    @Override
    public String toString() {
        return _name + " " + _rank;
    }

    @Override
    public boolean equals(Object other) {
        if (other instanceof Student) {
            Student b = (Student) other;
            return this._name.equals(b._name) &&
                this._score == b._score;
        }
        else
            return false;
    }
}

class InputComparator implements Comparator<Student> {
    @Override
    public int compare(Student x, Student y){
        return Integer.compare(x.getOrder(), y.getOrder());
    }
}

class ScoreComparator implements Comparator<Student> {
    @Override   //Higher Score will be 'First'
    public int compare(Student x, Student y){
        return Integer.compare(y.getScore(), x.getScore());
    }
}

public class Generate {
    private static ArrayList<String> output = new ArrayList<>();
    private static ArrayList<String> sequence = new ArrayList<>();
    private static String original;

    private void run() {
        Scanner sc = new Scanner(System.in);
        original = sc.next();
        boolean[] used = new boolean[original.length()];
        for (int i = 0; i < used.length; i++) {
            used[i] = false;
        }
        for (int i = 0; i < original.length(); i ++) {
            used[i] = true;
            permute(original.substring(i, i+1), used);
            used[i] = false;
        }
        sequence  = subset(original);
        Collections.sort(sequence);
        Collections.sort(output);
        for (String out : output) {
            System.out.println(out);
        }
        for (String seq : sequence) {
            if (!seq.equals(""))
                System.out.println(seq);
        }
    }

    private ArrayList<String> subset(String s) {
        if (s.length() == 0) {
            ArrayList<String> answer = new ArrayList<>();
            answer.add("");
            return answer;
        }
        ArrayList<String> sub = subset(s.substring(1));
        ArrayList<String> answer = new ArrayList<>(sub);
        for (String x:sub) {
            answer.add(s.charAt(0) + x);
        }
        return answer;
    }

    private void permute(String curr, boolean[] used){
        if (curr.length() == used.length){
            output.add(curr);
            return;
        }
        for (int i = 0; i < used.length; i++) {
            if (!used[i]) {
                used[i] = true;
                permute(curr + original.charAt(i), used);
                used[i] = false;
            }
        }
    }

    public static void main(String[] args) {
        Generate newGenerate = new Generate();
        newGenerate.run();
    }
}

public class Boxes {
    private void run() {
        Scanner sc = new Scanner(System.in);
        int op = sc.nextInt(), rows = sc.nextInt(), cols = sc.nextInt();
        String x, y;
        HashMap<String, Boolean> gridBox = new HashMap<>();
        HashMap<String, Boolean> gridRow = new HashMap<>();
        HashMap<String, Boolean> gridCol = new HashMap<>();
        for (int i = 0; i < op; i++) {
            switch(sc.next()) {
                case "SIT":
                    x = sc.next();
                    y = sc.next();
                    gridBox.put(x + " "+ y, true);
                    gridRow.put(x, true);
                    gridCol.put(y, true);
                    break;
                case "BOX":
                    x = sc.next();
                    y = sc.next();
                    System.out.println(search(gridBox, x + " " + y));
                    break;
                case "ROW":
                    x = sc.next();
                    System.out.println(search(gridRow, x));
                    break;
                case "COL":
                    y = sc.next();
                    System.out.println(search(gridCol, y));
                    break;
            }
        }
    }

    public String search(HashMap<String, Boolean> grid, String key) {
        if (grid.containsKey(key)) {
            return "Y";
        } else {
            return "N";
        }
    }

    public static void main(String[] args) {
        Boxes newBoxes = new Boxes();
        newBoxes.run();
    }
}
```