CSCI 3005 – Spring 2019 – Programming Assignment 2

While studying with a group of friends for an *Algorithms* exam, all of you decide to order a large pizza. However, the group is having a hard time reaching consensus on what toppings to include (or not). Each group member wants certain toppings and wants others left out. Of course, they all understand that since there is only going to be one pizza, no one is likely to have all their requirements satisfied. Therefore, the group decides to write a program that will determine if there is a pizza that will make everyone happy.

The pizza parlor you are calling offers the pizza toppings below; you can include or omit any of them:

Code	Topping
A	Anchovies
B	Black Olives
C	Canadian Bacon
D	Diced Garlic
E	Extra Cheese
F	Fresh Broccoli
G	Green Peppers
H	Ham
I	Italian Sausage
J	Jalapeno Peppers
K	Kielbasa
L	Lean Ground Beef
M	Mushrooms
N	Nonfat Feta Cheese
O	Onions
P	Pepperoni

To help define the problem, each of your friends writes down his/her preferences on a single line. For example, the line:

```
+O-H+P
```

reveals that someone will accept a pizza with onion or pepperoni but will reject a pizza with ham. Similarly, the line

```
-E+A+J
```

indicates that someone else will accept a pizza that omits extra cheese and includes anchovies or jalapenos. To help reach consensus, someone's request can include as many desirable toppings as needed but is limited to one undesirable topping.

Your assignment is to solve the problem by using a backtracking approach that systematically tries various combinations of toppings until a satisfactory combination is found or it is determined that one does not exist, while pruning non-promising partial solutions when possible. The resulting pizza is subject to the following constraints:

- The pizza must contain at least one topping ordered by each person
- Each topping in the pizza must have been requested by someone
- The pizza must exclude any topping disliked by someone

Your solution will be implemented in the form of a `Toppings` class, with the following public methods:

`Toppings(String dataFile)`: a constructor that accepts a data file containing a list of 1 to 20 topping request lines, using the format above.

`boolean canBeSatisfied()`: determines whether a combination of toppings exists that satisfies everyone's requests.

`String getToppings()`: returns a String containing the toppings in the solution in alphabetical order (or "NO PIZZA!!" if no solution exists). So, a pizza with onion, anchovies, fresh broccoli and Canadian bacon would be represented by:

```
ACFO
```

It is possible that for a group of requests, there is more than one combination of toppings that will satisfy everyone's requests. In that case, the `getToppings` method should return the first solution discovered by a depth-first backtracking expansion of the possible solution space that considers the toppings in alphabetical code order.

The source file `ToppingsTest.java` and sample input files are available for testing. Submit your solution source file to *Mimir* for final testing.