BATTLE POWERSET

Let S be a set of integers. For example:

$$S = \{1, -2, 17\}$$

The *powerset* of S is the set of all subsets of S:

$$\mathcal{P}(S) = \{\{\}, \{1\}, \{-2\}, \{17\},$$

$$\{1, -2\}, \{1, 17\}, \{-2, 17\}, \{1, 2, 17\}\}$$

Note in particular that:

- $S \in \mathcal{P}(S)$ for all S.
- $\{\} \in \mathcal{P}(S)$ for all S.
- $\mathcal{P}(\{\}) = \{\{\}\}$.

BATTLE POWERSET (contd.)

A set of sets of integers can be represented as a text file. Each line represents a set of integers. The empty set can be represented as an empty line. For example, the set

$$A = \{\{1, 2\}, \{\}, \{4\}, \{1, -2, -17\}, \{5, 6\}\}$$

can be represented as follows:

```
1 2

4
1 -2 -17
5 6
```

BATTLE POWERSET (contd.)

The same set can also be represented as a file which looks like this:

```
(define A '(
(1 2)
()
(4)
(1 -2 -17)
(5 6)
))
```

Use whichever format you prefer.

BATTLE POWERSET (contd.)

Write a program which decides whether or not a set of sets of integers is a powerset. The program prints "good" if the file represents a powerset and "bad" otherwise.

BATTLE POWERSET (contd.)

Here is an example of “good” input:

```
1
1 -2
-2
17 -2
17
17 -2 1

1 17
```

```
(define B '(
(1)
(1 -2)
(-2)
(17 -2)
(17)
(17 -2 1)
()
(1 17)
))
```

BATTLE POWERSET (contd.)

Here is an example of “bad” input:

```
1
1 -2
-2
17 -2 1
17 5
17

1 17
```

```
(define C '(
(1)
(1 -2)
(-2)
(17 -2 1)
(17 5)
(17)
()
(1 17)
))
```

BATTLE POWERSET (contd.)

Hint: Implement a set ADT. A set is like a list but the order of elements does not matter and there can be no duplicates. Make sure that your ADT is general enough to handle sets of sets.

BATTLE POWERSET (contd.)

Who's cuisine reigns supreme? Will it be the *Pundits of Perl*...or the *Champions of Scheme*?