Usage

Tool is developing on the following github repository:
`https://github.com/pablogventura/relationaldef`.

To run it:

```
$ python3 main.py < example.model
```

Input

Tool receives input from stdin in this format.

First line describes model universe. Must be a list of elements separated by spaces. Then must be an empty line.

Next lines describes relations. For each relation must be a first line containing relation name, number of tuples and arity separated by spaces followed by a line for each tuple containing elements separated by spaces. At the end of relation, must be a empty line. So it is able to start another.

Relations which names started in "T" will be target relations, and the rest will be base relations.

For example, a relational model **A**, where $A = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ with relations $R0^2, R1^2, T0^3$ is represented by:

```
0 1 2 3 4 5 6 7 8 9

R0 2 2
6 4
3 1

R1 2 2
0 3
2 2

T0 3 3
0 0 2
9 4 8
3 9 2
```

Output

Tool will return to stdout if target relations are open-definable by base relations. In the case it is not definable, returns an counterexample. Also returns spectrum, diversity, number of Minion calls and number of isomorphisms and automorphisms checked at the last moment.

For example, for the earlier example, output is:

```
Spectrum = [3, 2]
NOT DEFINABLE
Counterexample:
Automorphism(
    0->1
    1->0
    2->2
  from:
    RelationalModel(universe=[0, 1, 2],
      relations={'T0': {(0, 0, 2)},
                 'R1': {(2, 2)},
                 'R0': set()}))

State before abort:
  Diversity = 0
  #Auts = 2
  #Isos = 0
  1 calls to Minion
```

Thus, T0 is not open-definable. Counterexample is an automorphism of submodel generated by $\{0, 1, 2\}$.