Stack Machine Programmer

`program.c`, `program.C`, `program.java`

Many ciphers can be computed much faster using various machines and automata. The trouble with such machines is that someone has to write programs for them. Just imagine, how easy it would be if we could write a program that would be able to write another programs. In this contest problem, we will (for a while) ignore the fact that such a "universal program" is not possible. And also another fact that most of us would lose our jobs if it existed.

Your task is to write a program that will automatically generate programs for the stack machine defined in problem `execute`.

Input Specification

The input contains several test cases. Each test case starts with an integer number N ($1 \leq N \leq 5$), specifying the number of inputs your program will process. The next N lines contain two integer numbers each, V_i and R_i. V_i ($0 \leq V_i \leq 10$) is the input value and R_i ($0 \leq R_i \leq 20$) is the required output for that input value. All input values will be distinct.

Each test case is followed by one empty line. The input is terminated by a line containing one zero in place of the number of inputs.

Output Specification

For each test case, generate any program that produces the correct output values for *all* of the inputs. It means, if the program is executed with the stack initially containing only the input value V_i, after its successful execution, the stack must contain one single value R_i.

Your program must strictly satisfy all conditions described in the specification of the problem `execute`, including the precise formatting, amount of whitespace, maximal program length, limit on numbers, stack size, and so on. Of course, the program must not generate a failure.

Print one empty line after each program, including the last one.

Sample Input

```
3
1 3
2 6
3 11

1
1 1

2
2 4
10 1

0
```

Output for Sample Input

```
DUP
MUL
NUM 2
ADD
END

END

NUM 3
MOD
DUP
MUL
END
```