O. Xum

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

You have a blackboard and initially only an **odd** number x is written on it. Your goal is to write the number 1 on the blackboard.

You may write new numbers on the blackboard with the following two operations.

- You may take two numbers (not necessarily distinct) already on the blackboard and write their sum on the blackboard. The two numbers you have chosen remain on the blackboard.
- You may take two numbers (not necessarily distinct) already on the blackboard and write their bitwise XOR on the blackboard. The two numbers you have chosen remain on the blackboard.

Perform a sequence of operations such that at the end the number 1 is on the blackboard.

Input

The single line of the input contains the odd integer x ($3 \le x \le 999,999$).

Output

Print on the first line the number q of operations you perform. Then q lines should follow, each describing one operation.

- The "sum" operation is described by the line "$a + b$", where a, b must be integers already present on the blackboard.
- The "xor" operation is described by the line "a ^ b", where a, b must be integers already present on the blackboard.

The operation symbol (+ or ^) must be separated from a, b by a whitespace.

You can perform at most $100,000$ operations (that is, $q \le 100,000$) and all numbers written on the blackboard must be in the range $[0, 5 \cdot 10^{18}]$. It can be proven that under such restrictions the required sequence of operations exists. You can output any suitable sequence of operations.

Examples

input Copy

```
3
```

output Copy

```
5
3 + 3
3 ^ 6
3 + 5
3 + 6
8 ^ 9
```

input Copy

```
123
```

output Copy

```
10
123 + 123
123 ^ 246
141 + 123
246 + 123
264 ^ 369
121 + 246
367 ^ 369
30 + 30
```

```
60 + 60
120 ^ 121
```