

# PROBLEM 1: ALGOC - A Language to Generate Only Constants

## File Names

Source File: algoc.pas or algoc.c or algoc.cpp

Input File: algoc.in

Output File: algoc.out

## Statement of the Problem

An obscure Computer Science professor wants to become famous developing a new programming language, ALGOC – A Language to Generate Only Constants. This language is very simple, and has only four constructs:

- PLUSONE - creates the constant 1 (positive).
- MINUSONE - creates the constant -1 (negative).
- INC - adds one to the constant being generated.
- DUP - multiplies the constant being generate by two.

A program in this language is a sequence of these constructs, one in each line, executed sequentially. The professor wants to keep the programs written in this language simple, small and fast. To achieve his goal, he adds the following constraints:

- Every program must begin either with PLUSONE or with MINUSONE.
- A given constant  $C$  must be generated with the smallest number of instructions possible.
- If a constant  $C$  can be generated by different programs (all with equal number of instructions), then the fastest program should be used. For this purpose, suppose that the instruction DUP is executed in  $T$  nanoseconds and the instruction INC in  $2T$  nanoseconds.

You were hired by the professor to write several sample programs, so that he can use them in various conferences to demonstrate his powerful new language. The professor will give you a few constants and your task is to write programs to generate those constants, obeying the constraints above.

## Input Format

The input file may contain several instances of the problem. Each instance of the problem is just one line containing the numeric constant to be generated. All numbers are **nonzero** integers between -32768 and 32767. A line containing the integer zero terminates the input file.

## Output Format

For each instance of the problem, your program should print one line saying **Constant XXX**, where XXX is the constant for that instance, followed by the most efficient program to generate that constant, one instruction per line.

Each output of an instance is terminated with a blank line.

## Sample Input

```
3
-5
1
7
0
```

## Sample Output

Constant 3  
PLUSONE  
DUP  
INC

Constant -5  
MINUSONE  
DUP  
DUP  
INC  
DUP  
INC

Constant 1  
PLUSONE

Constant 7  
PLUSONE  
DUP  
INC  
DUP  
INC