

**4. Truth Tables**

**PROBLEM:** Have you been creating a lot of truth tables lately? You need to have a program to do that for you, but you must write it. Given a Boolean expression write a truth table for it.

**INPUT:** There will be 10 inputs. Each input line will be a string representing a Boolean expression. The table will be set up using the standard ACSL method of listing 0's first as shown below. The plus sign (+) will represent OR, the asterisk (\*) will represent AND, the pound sign (#) will represent XOR and the at symbol (@) will represent NOT. Single sets of parentheses will be used when required. As an example  $A + B * C$  produces the table below:

A	B	C	$B * C$	$A + B * C$
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	1	1
1	0	0	0	1
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

**OUTPUT:** Print the truth table for the Boolean expression. Each binary operation must have its own column. All columns must be labeled. The table must stay on the screen so that a proctor can grade it. The entries in each column must be aligned. Unary operation columns may be included. Grid lines are not required. Except for the last column, columns do not have to be in any order except that order of operations must be followed to get the correct answers.

**SAMPLE INPUT**

1.  $A + B * C$
2.  $@(A+B)$
3.  $(@A * B) \# (B + C)$

**SAMPLE OUTPUT**

1. As shown above

Output 2

A	B	$A + B$	@
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

Output 3.

A	B	C	$@A*B$	$B + C$	#
0	0	0	0	0	0
0	0	1	0	1	1
0	1	0	1	1	0
0	1	1	1	1	0
1	0	0	0	0	0
1	0	1	0	1	1
1	1	0	0	1	1
1	1	1	0	1	1