5. Consider a simple grammar for arithmetic expressions:

$$E \to id \mid num \mid -E \mid E + E \mid (E)$$

where *num* is an integer constant and *id* is a variable. The task is to compile expressions to code for a simple stack-based machine with the following instruction set.

instruction	*meaning*
PUSHVAR k	push the value of the k-th variable on top of stack
PUSH *num*	push *num* on top of stack
ADD	replace the top two stack items with their sum
NEG	replace the top stack item with its negation

(a) What is the code for $x + 2$?

(b) What is the code for $-(-x + (14 + y))$?

(c) Suppose we now want to extend the language of integer expressions with multiplication

$$E \to E * E$$

but we cannot extend the machine with an instruction for multiplication. Can you implement this extended language directly with the machine instruction set presented above? If not, suggest a minimal extension to the instruction set that allows for the implementation of multiplication using an addition from the instruction set. Explain the semantics of your extensions and how you would use them to implement multiplication.

(2+4+6 p)