Prefix/Infix/Postfix Notation

One commonly writes arithmetic expressions, such as 3 + 4 * (5 - 2) in *infix* notation which means that the operator is placed in between the two operands. In this example, the answer is 15 because the order of operations is used which most people remember as PEMDAS. In other words, evaluation is dependent on the precedence of the operators and the location of parentheses. Two alternative formats, *prefix* and *postfix*, have been developed to make evaluation more mechanical and hence, solvable by a computer algorithm. The expression 3 + 2 * 4 equals 11 because we use the order of operations, but without that rule, it could also equal 5 * 4 = 20 by doing each operation left to right. As you will see later, there is only one possible way to evaluate a prefix or postfix expression and no order of operations rule is necessary.

Mathematical Expressions

Mathematical expressions include *unary* operators, *binary* operators, and even *ternary* operators based on the number of operands needed. On a calculator, the ± or √ keys are *unary* operators because they are performed on a single operand. For this topic we will use the following *binary* operators only: + (add), - (subtract), * (multiply), / (divide). and ^ (exponents). Answers do not have to be integers, but all operands will be one-digit integers so spacing is not confusing. Prefix notation places each operator before its operands and postfix places each operator after its operands. All operators are considered *binary* operators because they operate on exactly two numbers at a time. The example above becomes "+ 3 * 4 - 5 2*"* in prefix notation and "3 4 5 2 - * +" in postfix notation. In all notations, the relative order of the operands is the same. For example, because addition and subtraction are done left to right, two operands at a time, 3 + 9 + 7 = "3 9 + 7 +". The following are equivalent, but not exact translations: "3 9 7 + +", "9 3 + 7 +", or "3 7 + 9 +".

Because this is the Elementary Division and 3rd graders have not been exposed to much division or exponents, we will guarantee that all division is by 1 which does not change the number or by 2 which means to simply take half of the number and all powers are 1 which means to simply use the number itself or 2 which means to multiply the number by itself. Exponents can be written as either 4 ^ 2 or 4^2. Therefore, using the above concepts, here is the way to evaluate the following two expressions using PEMDAS:

A. 7 * 3 – 6 / 2 + 4 ^ 2 = 21 - 3 + 16 = 34

B. $(3^2 - 2 * 3 + 8 / 2)^2 = (9 - 6 + 4)^2 = 7^2 = 7 * 7 = 49$

Translating from Infix to Prefix or Postfix

A simple method of translating an expression from infix to prefix is to figure out the order in which each operation is done using PEMDAS. For each pair of operands, put the operator before the two operands so that the operands are in the same order. To translate from infix to postfix, put the operator after the two operands so that the operands are in the same order. The expression 3 + 2 *4 is written as "+ 3 * 2 4" in prefix or it is written as "3 2 4 * +" in postfix.

Therefore, in expression A from above, the steps would be as follows:

Prefix:	Postfix:
7 * 3 – 6 / 2 + 4 ^ 2	7 * 3 – 6 / 2 + 4 ^ 2
(* 7 3) – (/ 6 2) + (^ 4 2)	(7 3 *) – (6 2 /) + (4 2 ^)
(– * 7 3 / 6 2) + (^ 4 2)	(7 3 * 6 2 / –) + (4 2 ^)
+ – * 7 3 / 6 2 ^ 4 2	7 3 * 6 2 / – 4 2 ^ +

All operands must be in the same order as the original infix expression. Parentheses are provided for clarity only and are not in the final answer.

Here is another example from expression B that was used above:

Prefix:	Postfix:
$(3^2 – 2 * 3 + 8 / 2) ^ 2$	$(3^2 – 2 * 3 + 8 / 2) ^ 2$
((^ 3 2) – (* 2 3) + (/ 8 2)) ^ 2	((3 2 ^) – (2 3 *) + (8 2 /)) ^ 2
((– ^ 3 2 * 2 3) + (/ 8 2)) ^ 2	((3 2 ^ 2 3 * –) + (8 2 /)) ^ 2
(+ – ^ 3 2 * 2 3 / 8 2) ^ 2	(3 2 ^ 2 3 * – 8 2 / +) ^ 2
^ + – ^ 3 2 * 2 3 / 8 2 2	3 2 ^ 2 3 * – 8 2 / + 2 ^

Evaluating a Prefix or Postfix Expression

Some of the very first scientific calculators used postfix notation (also known as *Polish* and *Reverse Polish* notation for the Polish logician Jan Lukasiewicz) because of how easy it was to evaluate an expression. Every time you get to an operator, you simply perform that operation on the two previous results and keep the answer. In expression A above, the process in both postfix and postfix is as follows:

Prefix:	Postfix:
+ – * 7 3 / 6 2 ^ 4 2	7 3 * 6 2 / – 4 2 ^ +
+ – (7 * 3) (6 / 2) (4 ^ 2)	(7 * 3) (6 / 2) – (4 ^ 2) +
+ – 21 3 16	21 3 – 16 +
+ (21 – 3) 16	(21 – 3) 16 +
+ 18 16	18 16 +
18 + 16 = 34	18 + 16 = 34

Therefore, evaluating from prefix or postfix uses the same steps in reverse by doing two operands and an operator at a time. Here is the process for expression B:

Prefix:	Postfix:
^ + – ^ 3 2 * 2 3 / 8 2 2	3 2 ^ 2 3 * – 8 2 / + 2 ^
^ + – (3 ^ 2) (2 * 3) (8 / 2) 2	(3 ^ 2) (2 * 3) – (8 / 2) + 2 ^
^ + – 9 6 4 2	9 6 – 4 + 2 ^
^ + (9 – 6) 4 2	(9 – 6) 4 + 2 ^
^ + 3 4 2	3 4 + 2 ^
^ (3 + 4) 2	(3 + 4) 2 ^
^ 7 2	7 2 ^
7 ^ 2 = 49	7 ^ 2 = 49

References

http://cs-study.blogspot.com/2012/11/infix-to-postfix-conversion.html
http://mathworld.wolfram.com/ReversePolishNotation.html
http://www.codechannels.com/video/mycodeschool/university/infix-prefix-and-postfix/
www.youtube.com/watch?v=jos1Flt21is

Sample Problems

Evaluate the postfix expression 3 4 + 7 2 – *	3 4 + = 3 + 4 = 7 and 7 2 – = 7 – 2 = 5. Therefore, 7 5 * = 7 * 5 = 35.
Evaluate the following postfix expression: 7 4 – 2 ^ 3 1 + * 6 –	First, 7 4 – = 7 – 4 = 3 and 3 1 + = 3 + 1) = 4. Then 3 2 ^ = 3 ^ 2 = 9 and 9 4 * = 36. Finally, 36 6 – = 36 – 6 = 30.
Translate the following infix expression into prefix. (6 + 4) / (7 – 5) * 3 ^ 2	(6 + 4) / (7 – 5) * 3 ^ 2 (+ 6 4) / (– 7 5) * (^ 3 2) (/ + 6 4 – 7 5) * (^ 3 2) * / + 6 4 – 7 5 ^ 3 2
Translate the following infix	The expression converts as follows:

expression into postfix. **(**9 – 8 / 2 + 5) ^ 2	**(**9 – 8 / 2 + 5) ^ 2 (9 – (8 2 /) + 5) ^ 2 ((9 8 2 / –) + 5) ^ 2 (9 8 2 / – 5 +) ^ 2 9 8 2 / – 5 + 2 ^
Evaluate the following postfix expression: 6 2 – 2 ^ 2 / 9 3 1 / – + 2 *	Since, 6 2 – = 6 – 2 = 4 and 3 1 / = 3 / 1 = 3, the expression is 4 2 ^ 2 / 9 3 – + 2 *. Since 4 2 ^ = 4 ^ 2 = 16 and 9 3 – = 9 – 3 = 6, the expression is 16 2 / 6 + 2 *. Since 16 / 2 = 8, it becomes 8 6 + 2 *. Therefore, 8 6 + = 8 + 6 = 14. Finally, 14 2 * = 14 * 2 = 28.
Evaluate the following prefix expression: + ^ / + 7 5 – 6 4 2 3	+ ^ / + 7 5 – 6 4 2 3 + ^ / (+ 7 5) (– 6 4) 2 3 + ^ / 12 2 2 3 + ^ (12 / 2) 2 3 + ^ 6 2 3 + (6 ^ 2) 3 + 36 3 = 36 + 3 = 39
Evaluate the following prefix expression: – ^ / + + 4 6 2 – 4 1 2 9	Convert to infix: – ^ / + + 4 6 2 – 4 1 2 9 – ^ / + (4 + 6) 2 (4 – 2) 2 9 – ^ / + 10 2 2 2 9 – ^ / (+ 10 2) 2 2 9 – ^ / 12 2 2 9 – ^ (12 / 2) 2 9 – ^ 6 2 9 = – (6 ^ 2) 9 – 36 9 = 36 – 9 = 27
Evaluate the following prefix expressions: – + ^ / 6 2 2 7 ^ – 5 2 1	– + ^ / 6 2 2 7 ^ – 5 2 1 – + ^ (6 / 2) 2 7 ^ (5 – 2) 1 – + ^ 3 2 7 ^ 3 1 – + (3 ^ 2) 7 (3 ^ 1) – + 9 7 3 – (9 + 7) 3 = – 16 3 = 13
Translate the following infix	((7 + 5) / (6 – 4)) ^ 2 + 3

expression to postfix: ((7 + 5) / (6 – 4)) ^ 2 + 3	((7 5 +) / (6 4 –)) ^ 2 + 3 (7 5 + 6 4 – /) ^ 2 + 3 (7 5 + 6 4 – / 2 ^) + 3 7 5 + 6 4 – / 2 ^ 3 +
Translate the following infix expression into prefix. ((5 – 3) * 5 + (9 – 3) / 3) ^ 2	((5 – 3) * 5 + (9 – 3) / 3) ^ 2 ((– 5 3) * 5 + (– 9 3) / 3) ^ 2 ((* – 5 3 5) + (/ – 9 3 3)) ^ 2 (+ * – 5 3 5 / – 9 3 3) ^ 2 ^ + * – 5 3 5 / – 9 3 3 2
Translate the following prefix expression to a postfix expression: ^ + * – 5 3 5 / – 9 3 3 2	^ + * – 5 3 5 / – 9 3 3 2 ^ + * (5 – 3) 5 / (9 – 3) 3 2 ^ + ((5 – 3) * 5) ((9 – 3) / 3) 2 ^ (((5 – 3) * 5) + ((9 – 3) / 3)) 2 (((5 – 3) * 5) + ((9 – 3) / 3)) ^ 2 (((5 3 –) * 5) + ((9 3 –) / 3)) ^ 2 ((5 3 – 5 *) + (9 3 – 3 /)) ^ 2 (5 3 – 5 * 9 3 – 3 / +) ^ 2 5 3 – 5 * 9 3 – 3 / + 2 ^
Translate the following postfix expression to a prefix expression: 7 5 + 6 4 – / 2 ^ 3 +	7 5 + 6 4 – / 2 ^ 3 + (+ 7 5) (– 6 4) / 2 ^ 3 + (/ (+ 7 5) (– 6 4)) 2 ^ 3 + (^ (/ (+ 7 5) (– 6 4)) 2) 3 + + ^ / + 7 5 – 6 4 2 3