

AQA Computer Science A-Level
4.3.3 Reverse Polish
Concise Notes



Specification:

4.3.3.1 Reverse Polish – infix transformations

Be able to convert simple expressions in infix form to Reverse Polish notation (RPN) form and vice versa

Be aware of why and where it is used



Reverse Polish Notation

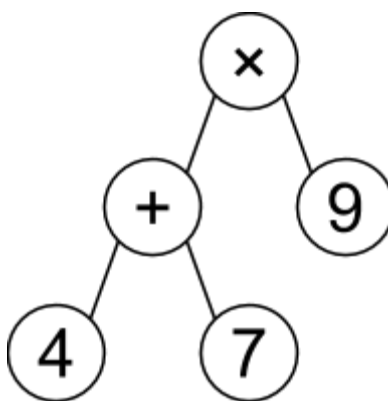
- A way of **writing expressions**
- Uses **postfix** notation
- Operators are placed **after** the operands on which they operate

Prefix	Infix	Postfix (Reverse Polish)
$+ 2 3 = 5$	$2 + 3 = 5$	$2 3 + = 5$

Converting between Infix and Postfix

- Infix expressions can be **converted** into reverse Polish (and vice versa) by **traversing** an **expression tree**
 - To return an **infix** expression, use an **in order** traversal
 - To return a **postfix** expression, carry out **postorder** traversal
- Simpler expressions can be converted **by observation**

Expression Tree



Infix
In order traversal

$$(4 + 7) \times 9 = 99$$

Postfix (Reverse Polish)
Postorder traversal

$$4 7 + 9 \times = 99$$



Why is reverse Polish notation used?

- Reverse Polish notation **eliminates the need for brackets**, simplifying expressions
- Reverse Polish notation is **well suited to manipulation by a stack**
- This makes reverse Polish a popular choice when working with **computers**

Where is reverse Polish notation used?

- Reverse Polish notation is used in **interpreters** which are based on stacks:
 - Bytecode
 - PostScript

