# Worksheet 6 Reverse Polish Notation Answers

# Task 1

1. Convert to Reverse Polish Notation by numbering: (e – f) + a^b \* g

( e – f ) + a ^ b \* g

 1 3 2 9 4 6 5 8 7

e f – a b ^ g \* +

1. Convert to Reverse Polish Notation by bracketing: g \* (f + e) / h

g \* ( f + e ) / h

( ( g \* ( f + e ) ) / h )

( ( ( g ( f e + ) \* ) h / )

g f e + \* h /

1. Convert to Reverse Polish Notation by any method: 12 \* v – 8 \* v - 5

( ( 12 \* v ) – ( 8 \* v ) ) – 5

12 v \* 8 v \* - 5 -

1. Check your answer by bracketing

12 v \* 8 v \* - 5 -

( ( 12 v \*) ( 8 v \* ) - ) 5 - )

12 \* v – 8 \* v - 5

# Task 2

1. Create a binary tree for 6 \* a \* (~4 + 8 \* a), with the second \* at the root. (~ is unary minus)
	1. Show pre-order traversal (Polish Notation, prefix)
	2. Show in-order traversal (infix)
	3. Show post-order traversal (Reverse Polish Notation, post-fix)

Add brackets as required

(6 \* a) \* (-4 + (8 \* a))



Pre-order traversal: \* \* 6 a + - 4 \* 8 a

In-order traversal: 6 \* a \* - 4 + 8 \* a

Post-order traversal: 6 a \* 4 – 8 a \* + \*

# Task 3

6. Convert the following expression from RPN to infix using the scanning method.

b c \* a c \* - b a c \* / +

(b \* c)

(b\*c)(a \* c)

(b \* c) – (a \* c)

(b \* c) – (a \* c) b (a \* c)

(b \* c) – (a \* c) (b / (a \* c))

(b \* c) – (a \* c) + (b / (a \* c))

7. Convert the following expression from RPN to infix using the bracketing method.

a b c + c a ^ - /

a (b c +) (c a ^) - /

a ((b c +) (c a ^) -) /

(a ((b c +) (c a ^) -) /)

(a ((b + c) - (c ^ a)) /)

(a / ((b + c) - (c ^ a)))

a / ((b + c) - (c ^ a))

# Task 4

8. Evaluate RPN by stack: 3 2 \* 4 5 \* +

* + If the token is an operand, push it on the stack
	+ If the token is operator, pop required number of operands from stack; perform the operation; push the result

Note: when popping operands, pop right operand first and left operand second

|  |  |  |
| --- | --- | --- |
| **Stack** | **Pop, execute, push** | **3 2 \* 4 5 \* +** |
| 23 | 6 = 3 \* 2 | 3 2 \* |
| 6 |  | 3 2 \* 4 5 |
| 546 | 20 = 4 \* 5 | 3 2 \* 4 5 \* |
| 206 |  | 3 2 \* 4 5 \*  |
| 26 | 26 = 6 + 20 | 3 2 \* 4 5 \* + |

26 = (3 \* 2) + (4 \* 5)