

**AQA Computer Science A-Level**  
**4.3.3 Reverse Polish**  
Past Paper Questions

## June 2011 Comp 3

**5** Reverse Polish Notation is an alternative to standard infix notation for writing arithmetic expressions.

**5 (a)** Convert the following Reverse Polish Notation expressions to their equivalent infix expressions.

Reverse Polish Notation	Equivalent Infix Expression
45 6 +	
12 19 + 8 *	

*(2 marks)*

**5 (b)** State **one** advantage of Reverse Polish Notation over infix notation.

.....

.....

*(1 mark)*

- 5 (c) The pseudo-code algorithm in **Figure 3** can be used to calculate the result of evaluating a Reverse Polish Notation expression that is stored in a string. The algorithm is designed to work only with the single digit denary numbers 0 to 9. It uses procedures and functions listed in **Table 1**, two of which operate on a stack data structure.









**Figure 3**

```
StringPos ← 0
Repeat
  StringPos ← StringPos + 1
  Token ← GetCharFromString(InputString, StringPos)
  If Token = '+' Or Token = '-' Or Token = '/' Or Token = '*'
  Then
    Op2 ← Pop()
    Op1 ← Pop()
    Case Token Of
      '+': Result ← Op1 + Op2
      '-': Result ← Op1 - Op2
      '/': Result ← Op1 / Op2
      '*': Result ← Op1 * Op2
    EndCase
    Push(Result)
  Else
    IntegerVal ← ConvertToInteger(Token)
    Push(IntegerVal)
  EndIf
Until StringPos = Length(InputString)
Output Result
```

5 (c) Complete the table below to trace the execution of the algorithm when `InputString` is the string: `64+32+*`

In the `Stack` column, show the contents of the stack once for each iteration of the `Repeat..Until` loop, as it would be at the end of the iteration.

The first row and the leftmost column of the table have been completed for you.

StringPos	Token	IntegerVal	Op1	Op2	Result	Stack
0	-	-	-	-	-	
1						
2						
3						
4						
5						
6						
7						

(5 marks)

Final output of algorithm: .....

(1 mark)

5 (d) A programmer is going to implement the algorithm from **Figure 3** in a programming language that does not provide built-in support for a stack data structure.

The programmer intends to simulate a stack by using a fixed length array of 20 integers named `StackArray` with indices running from 1 to 20 and an integer variable `TopOfStackPointer` which will be initialised to 0.

Write a pseudo-code algorithm for the `Push` operation to push a value stored in the variable `ANumber` onto the stack.

Your algorithm should cope appropriately with any potential errors that might occur.

.....

.....

.....

.....

.....

.....

.....

.....

.....

(4 marks)

# Spec Qs Paper 1

**0 5**

Convert the following Reverse Polish Notation expressions to their equivalent infix expressions.

**0 5** . **1** 3 4 \*

**[1 mark]**

**0 5** . **2** 12 8 + 4 \*

**[1 mark]**

Reverse Polish Notation is an alternative to standard infix notation for writing arithmetic expressions.

**0 5** . **3**

State **one** advantage of Reverse Polish Notation over infix notation.

**[1 mark]**