Compression for AIDA

The algorithm used is “bit-packed minimum+offset”.

The input data is in fixed and known length blocks of 16-bit words. The output data consists of a 16-bit word containing the number of bits each input word has been packed into, followed by a second 16-bit word containing the smallest value in the input block, followed by the compressed data.

The compression technique is to search for the minimum & maximum values in the input block; the difference between them is the range. Each input word will be expressed, on output, as an offset from the minimum. The range is the largest value that will occur. Each of the offsets can be stored in a number of bits no greater than that needed to store the range. During compression, each word offset is stored packed next to the preceding offset. High order zero-bits beyond that needed to store the range are not stored.

Examples

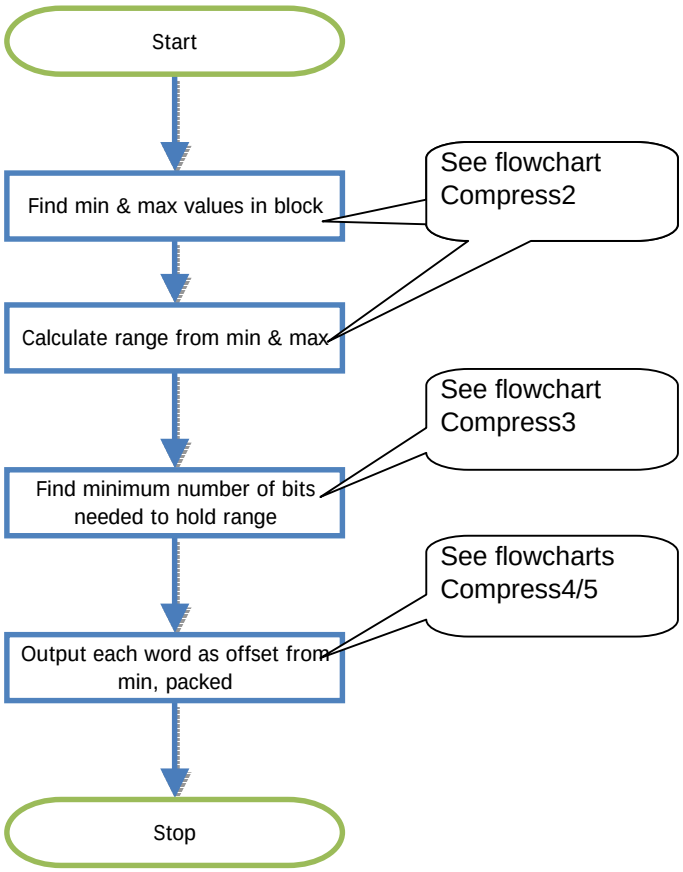
Minimum: 1216, Maximum: 1222
Range: 6, In binary: 0000 0000 0000 0110
Bits needed per compressed word: 3
Data: **1221**, 1220, **1218**, 1216, **1217**, … (Require 5 * 16 bits = 80 bits => 10 bytes)
Offsets: **5**, 4, **2**, 0, 1, … (Require 5*3 bits = 15 bits => 1 bytes + 7 (bits) = 2 bytes)
Binary: **101**, 100, **010**, 000, **001**, …
Packed: **10**100**101**, **001**000**0** …
Decimal: 165, 16 …
Bitstream: (shown in opposite order) …**001**000**0**|**10**100**101**

Minimum: 1216, Maximum: 1234
Range: 18, In binary: 0000 0000 0001 0010
Bits needed per compressed word: 5
Data: **1231**, 1220, **1233**, 1216, **1226**, … (Require 5 * 16 bits = 80 bits => 10 bytes)
Offsets: **15**, 4, **17**, 0, **10**, … (Require 5*5 bits = 25 bits => 3 bytes + 1 (bit) = 4 bytes)
Binary: **01111**, 00100, **10001**, 00000, **01010**, …
Packed: 100**01111**, 0**1000**100, **1010**0000, **0** …
Decimal: 143, 68, 160, 0 …
Bitstream: (shown in opposite order) …**0**|**1010**0000|0**1000**100|100**01111**

Minimum: 1216, Maximum: 1713
Range: 497, In binary: 0000 0001 1111 0001
Bits needed per compressed word: 9
Data: **1231**, 1216, **1301**, 1700, **1529**, … (Require 5 * 16 bits = 80 bits => 10 bytes)
Offsets: **15**, 0, **85**, 484, **313**, …(Need 5*9 bits = 45 bits => 5 bytes + 5(bits) = 6 bytes)
Binary: **000001111**, 000000000, **001010101**, 111100100, **100111001**, …
Packed: **00001111**, 00000000, **01010**100, 00100**001, 1001**1111, **10011** …
Decimal: 15, 0, 84, 33, 159, 19 …
Bitstream: (shown in opposite order)
…**10011**|**1001**1111|00100**001**|**01010**100|00000000|**00001111**

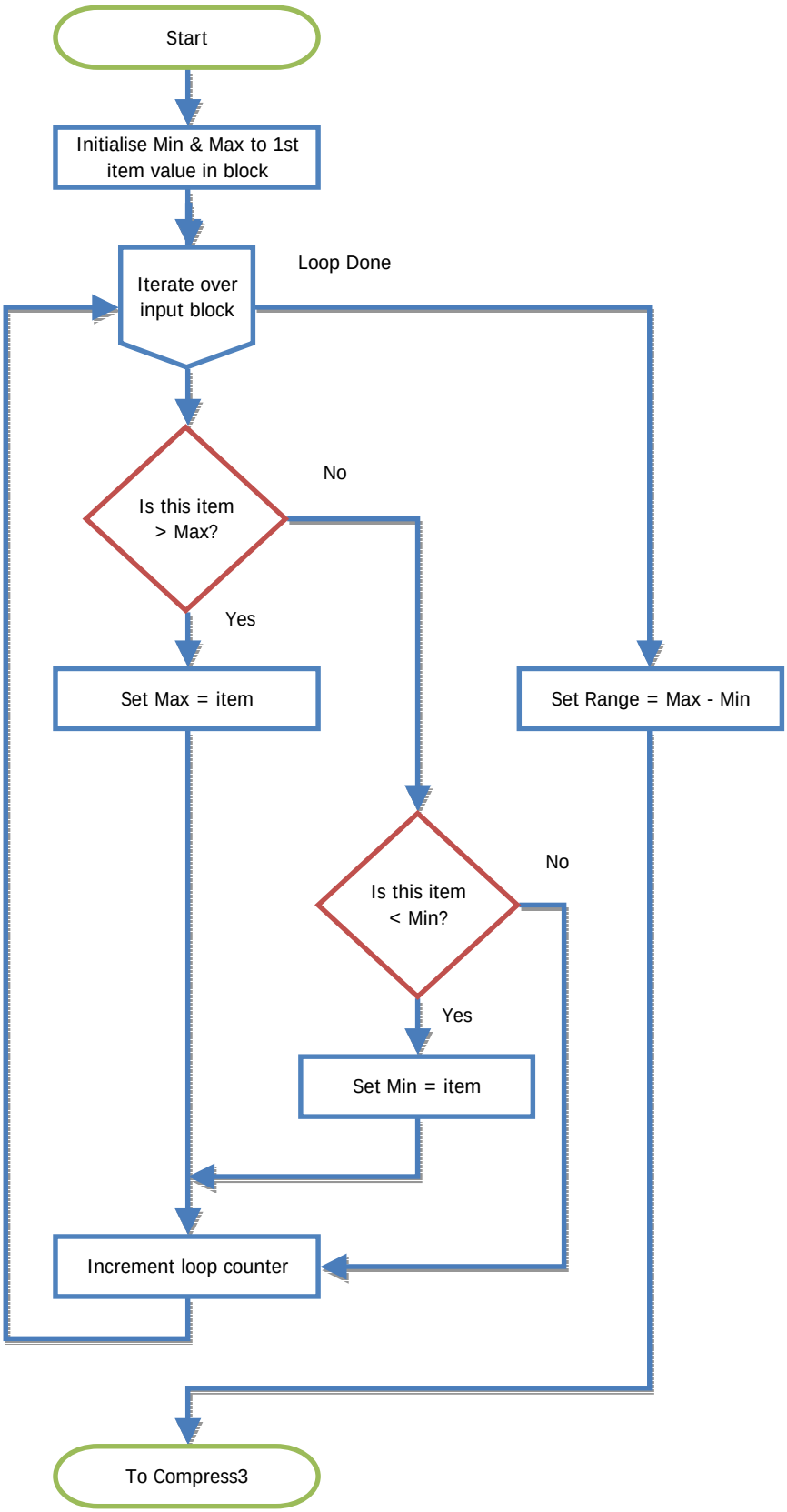
Simon Letts: August 2008

Compress1



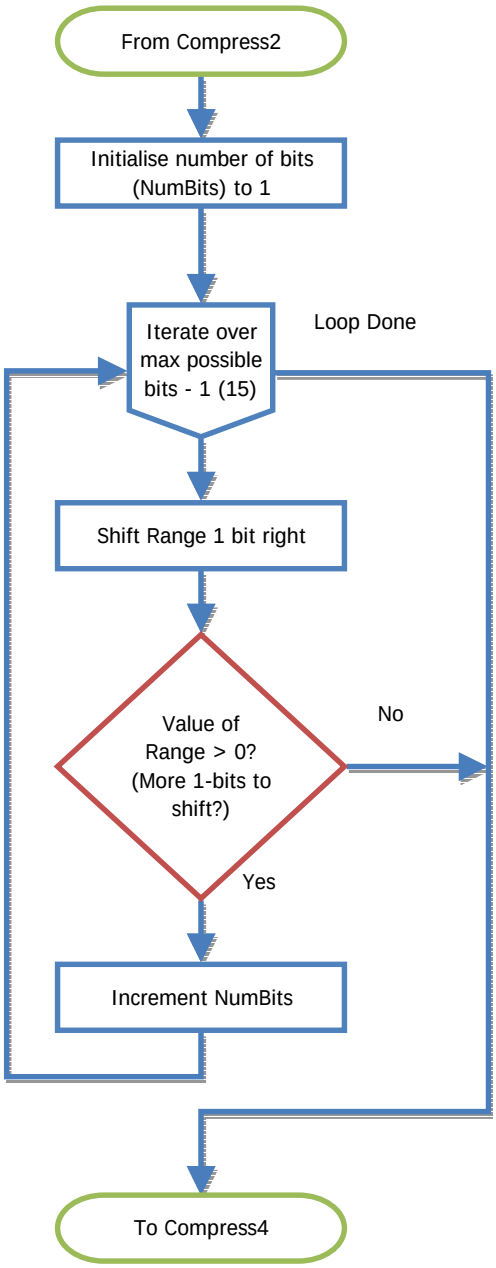
Simon Letts: August 2008

Compress2



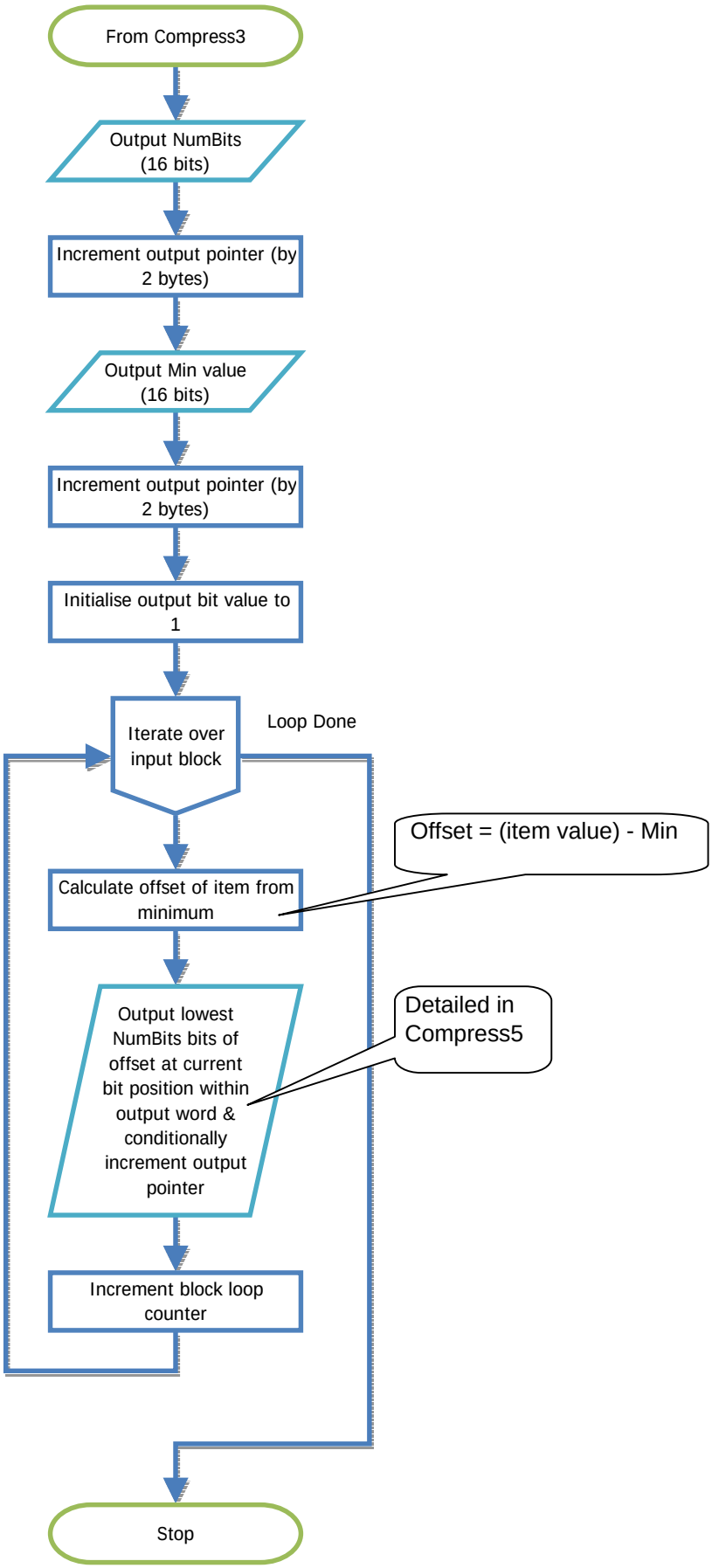
Simon Letts: August 2008

Compress3



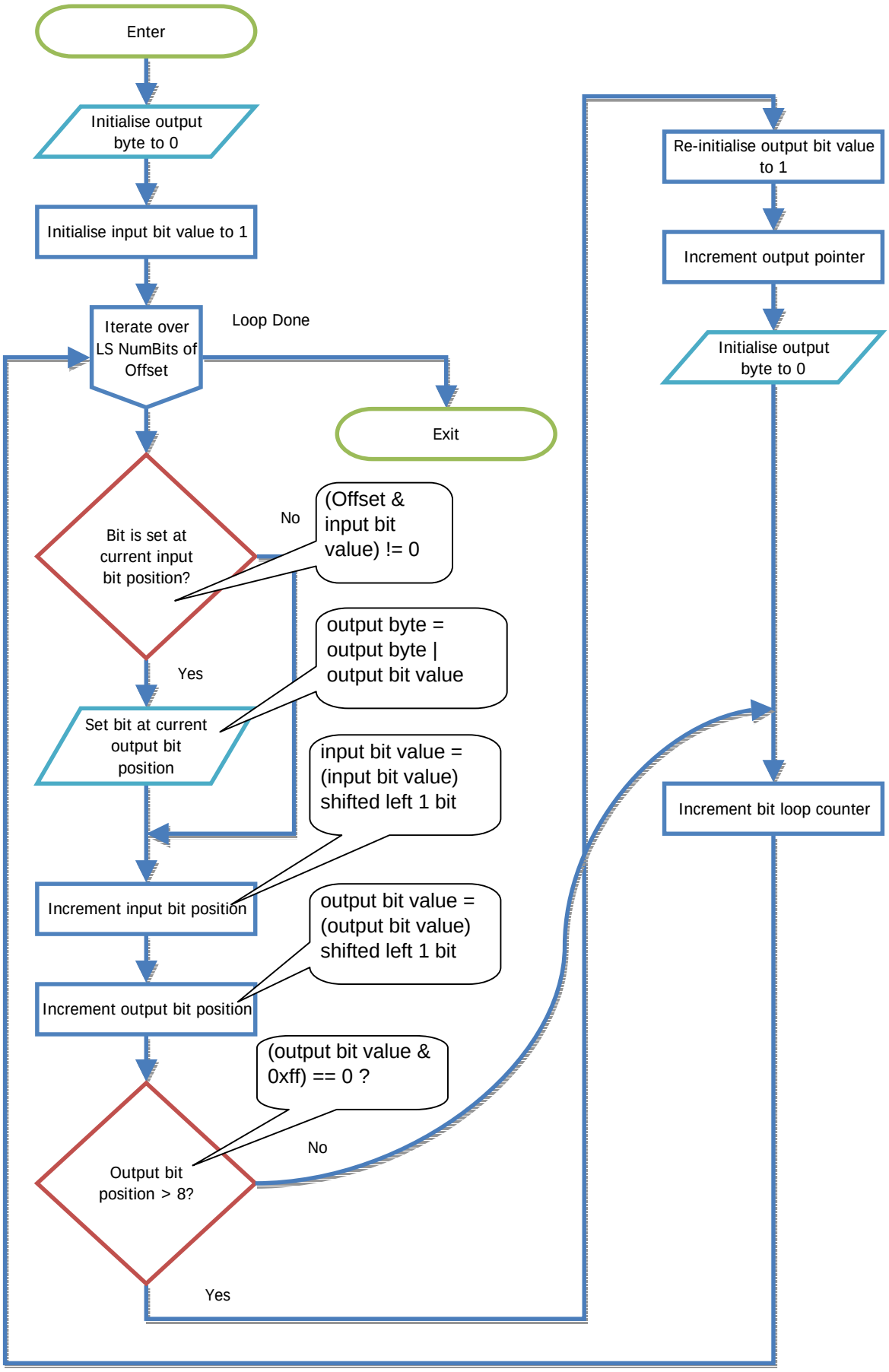
Simon Letts: August 2008

Compress4



Simon Letts: August 2008

Compress5



DeCompression for AIDA

The algorithm used is “bit-packed minimum+offset”.

The input data consists of a 16-bit word containing the number of bits each word has been packed into, followed by a second 16-bit word containing the smallest value in the output block, followed by the compressed data. The reconstructed data is in fixed and known length blocks of 16-bit words.

The technique is to read successive ***n*** bits of data, where ***n*** is the packed length that was read from the start of the input stream. Each ***n***-bit word represents an offset from the minimum value, also previously read from the input stream. Each output word is constructed by adding the offset to the minimum.

Examples

Bit length: 3, Minimum: 1216
Bitstream following: (shown R to L) …**001**000**0|10**100**101**
Packed: (shown L to R) **10**100**101**, **001**000**0** …
Decimal: 165, 16 …
Binary: ((***n*** = 3)-bit words) **101**, 100, **010**, 000, **001**, …
Offsets: **5**, 4, **2**, 0, 1, …
Data: (1216 + offset) **1221**, 1220, **1218**, 1216, **1217**, …

Bit length: 5, Minimum: 1216
Bitstream following: (shown R to L) …**0|1010**0000|0**1000**00|100**01111**
Packed: (shown L to R) 100**01111**, 0**1000**00, **1010**0000, **0** …
Decimal: 143, 68, 160, 0 …
Binary: ((***n*** = 5)-bit words) **01111**, 00100, **10001**, 00000, **01010**, …
Offsets: **15**, 4, **17**, 0, **10**, …
Data: (1216 + offset) 1231, 1220, 1233, 1216, 1226, …

Bit length: 9, Minimum: 1216
Bitstream following: (shown R to L)
…**10011|1001**1111|00100**001|01010**00|0000000**0|00001111**
Packed: (L to R) **00001111**, 0000000**0**, **01010**00, 00100**001, 1001**1111, **10011** …
Decimal: 15, 0, 84, 33, 159, 19 …
Binary: ((***n***=9)-bits) **000001111**, 000000000, **001010101**, 111100100, **100111001**, …
Offsets: **15**, 0, **85**, 484, **313**, …
Data: (1216 + offset) **1231**, 1216, **1301**, 1700, **1529**, …

N.B. The flow chart below shows an implementation of this algorithm but is not optimised, i.e. there are faster ways of producing the correct output from an input stream.

Simon Letts: August 2008

Explode (Decompress)