Operations and results	Description of step
`$ n = 00000000 01000000 00011011 01111001`	Initial input

`$ m = ~(1 & (n >> 31)) + 1`
`$ n = (~m & n) | (m & ~n)`
`00000000 01000000 00011011 01111001`
	Find the absolute value

`$ n >> 16` `00000000 00000000 00000000 01000000`	Shift right by half the block: 16 bits
`$ !!(00000000 00000000 00000000 01000000)` `1`	Check if the result is nonzero
`$ 1 << 4` `16`	Check how many bits we now know we need at least by shifting our result left by 4. If there was a 1, it's 16 — otherwise it's 0.
`$ n = n >> 16` `00000000 00000000 00000000 01000000`	Shift our tracked number (initially the original input) by the above:
`$ n >> 8` `00000000 00000000 00000000 00000000`	We start iteration 2 and shift right by half the block: 8 bits
`$ !!(00000000 00000000 00000000 00000000)` `0`	Check if the result is nonzero
`$ 0 << 3` `0`	How many extra bits do we need? If there was a 1, it's 8 more bits — otherwise it's 0
`$ n = n >> 0` `00000000 00000000 00000000 01000000`	Shift our tracked number by the above:
`$ n >> 4` `00000000 00000000 00000000 00000100`	We start iteration 3 and shift right by half the block: 4 bits
`$ !!(00000000 00000000 00000000 00000100)` `1`	Check if the result is nonzero
`$ 1 << 2` `4`	How many extra bits do we need? If there was a 1, it's 4 more bits — otherwise it's 0
`$ n = n >> 4` `00000000 00000000 00000000 00000100`	Shift our tracked number by the above:
`$ n >> 2` `00000000 00000000 00000000 00000001`	We start iteration 4 and shift right by half the block: 2 bits
`$ !!(00000000 00000000 00000000 00000001)` `1`	Check if the result is nonzero
`$ 1 << 1` `2`	How many extra bits do we need? If there was a 1, it's 2 more bits — otherwise it's 0
`$ n = n >> 2` `00000000 00000000 00000000 00000001`	Shift our tracked number by the above:
`$ n >> 1` `00000000 00000000 00000000 00000000`	We start iteration 5 and shift right by half the block: 1 bit
`$ !!(00000000 00000000 00000000 00000000)` `0`	Check if the result is nonzero. This is
`$ 0 << 0` `0`	How many extra bits do we need? If the result was a 1, it's 1 more bit — otherwise it's 0
`$ n = n >> 0` `00000000 00000000 00000000 00000000`	Shift our tracked number by the above:
`$ 16 + 0 + 4 + 2 + 0 + 1` `23`	Sum up all our findings and add 1 more bit for negative values. We thus need 23 bits.