

Project 1

Task: 1

Division by 10^n . Example Input: 123

Step	ht	tt	t	h	d	o	acc	Comment
(1)	0	0	0	0	0	0	123	$ht = acc/100000$
(2)	0	0	0	0	0	0	123	$acc = acc - ht \cdot 100000$
(3)	0	0	0	0	0	0	123	$tt = acc/10000$
(4)	0	0	0	0	0	0	123	$acc = acc - tt \cdot 100000$
(5)	0	0	0	0	0	0	123	$t = acc/1000$
(6)	0	0	0	0	0	0	123	$acc = acc - t \cdot 1000$
(7)	0	0	0	0	0	0	123	$h = acc/100$
(8)	0	0	0	1	0	0	123	$acc = acc - h \cdot 100$
(9)	0	0	0	1	0	0	23	$d = acc/10$
(10)	0	0	0	1	2	0	23	$acc = acc - d \cdot 10$
(11)	0	0	0	1	2	0	3	$o = acc$
(12)	0	0	0	1	2	3	3	END

Double dabble. Example Input: 123 -> 01111011

Step	ht	tt	t	h	d	o	Input	Comment
(1)	0000	0000	0000	0000	0000	0000	01111011	Shift <<
(2)	0000	0000	0000	0000	0000	0000	11110110	Shift <<
(3)	0000	0000	0000	0000	0000	0001	11101100	Shift <<
(4)	0000	0000	0000	0000	0000	0011	11011000	Shift <<
(5)	0000	0000	0000	0000	0000	0111	10110000	Add 3 to o because o >=5
(6)	0000	0000	0000	0000	0000	1010	10110000	Shift <<
(7)	0000	0000	0000	0000	0001	1000	11000000	Add 3 to o because o >=5
(8)	0000	0000	0000	0000	0011	0000	11000000	Shift <<
(9)	0000	0000	0000	0000	0110	0001	10000000	Add 3 to d because d >=5
(10)	0000	0000	0000	0000	1001	0001	10000000	Shift <<
(11)	0000	0000	0000	0001	0010	0011	00000000	END
	0	0	0	1	2	3		

In C the smallest type is unsigned char which contains 8 bits. To implement double dabble algorithm you have to mask older bits by mask.

MASK = 0x00ff;

TASK 2

Look-up table

Lp.	Input	HEX[0]	HEX[1]	HEX[2]	HEX[3]	HEX[4]	HEX[5]	HEX[6]
(1)	0	0	0	0	0	0	0	0
(2)	1	0	1	1	0	0	0	0
(3)	2	1	1	0	1	1	0	1
(4)	3	1	1	1	1	0	0	1
(5)	4	0	1	1	0	0	1	1
(6)	5	1	0	1	1	0	1	1
(7)	6	1	0	1	1	1	1	1
(8)	7	1	1	1	0	0	0	0
(9)	8	1	1	1	1	1	1	1
(10)	9	1	1	1	1	0	1	1

