Solving system of linear congruences

R_i = remainder for the i_th congruence
m_i = modulo for the i_th congruence, m_i ≠ m_j
M = ∏(m_i)
n_i = M/m_i for the i_th congruence
n_i bar = inverse of n_i modulo m_i
Product = (R_i)*(n_i)*(n_i bar)

	(B9)	(C9)	(D9)	(E9)	(F9)	(G9)	(H9)	(I9)
	System of congruences		**R_i**	**mod**	**m_i**		**M**	**VERIFY**
			(enter)		(enter)		∏(m_i)	
1		X =	4	(mod	3)		**78540**	1
2		X =	3	(mod	4)			3
3		X =	7	(mod	5)			2
4		X =	5	(mod	7)			5
5		X =	4	(mod	11)			4
6		X =	3	(mod	17)			3
7		X =		(mod	1)			0
8		X =		(mod	1)			0
9		X =		(mod	1)			0
10		X =		(mod	1)			0

Chinese Remainder Algorithm

	R_i	**m_i**		**n_i**	**n_i bar**	**Product**	Column (cell)	Content
	(B26)	(C26)	(D26)	(E26)	(F26)	(G26)		
1	4	3		26180	-1	-104720	R_i (B27)	D12
2	3	4		19635	-1	-58905	m_i C27)	F12
3	7	5		15708	2	219912	n_i (E27)	H12/C27
4	5	7		11220	-1	-56100	n_i bar (F27)	from Euclidean Algorithm
5	4	11		7140	1	28560	Product (G27)	B27*E27*F27
6	3	17		4620	4	55440		
7	0	1		78540	0	0		
8	0	1		78540	0	0		
9	0	1		78540	0	0		
10	0	1		78540	0	0		
				Total:		84187	>>>>>> **x =**	**5647** **mod M**

Finding the inverse of n_i modulo m_i for the i_th congruence **(n_i)*(n_i bar) + (m_i)*y = 1**

	n_i	**m_i**	**quotient (n_i)/(m_i)**	**n_i bar**	**y**	**– quotient – (n_i)/(m_i)**	column (cell)	content
	(B45)	(C45)	(D45)	(E45)	(F45)	(G45)		
1	**26180**	**3**	0	**-1**	8727	0		
	3	26180	8726	8727	-1	-8726	n_i (B47)	MOD(C46,B46)
	2	3	1	-1	1	-1		
	1	2	2	1	0	-2	quotient (D46)	FLOOR(C46/B46,1)
	0	1	#DIV/0!	0	1	#DIV/0!	n_i bar (E46)	IF(B46=0,0,F46*G46+F47)
	#DIV/0!	0	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	y (F46)	IF(B46=0,1,E47)
	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!		
	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!		
	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!		
	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!		
	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!		
	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!		
	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!		
	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!		