

Test specifications				Comments				
Test number	Test steps	Expected Result	Actual Result					
1	INP X // 2 INP Y // 4 OUT X OUT Y	0000 0000 0000 0010 0000 0000 0000 0100	Registers: // 0x00000000 0000000000000010 0000000000000100 Output: // 0x00000000 0000000000000010 0000000000000100 Flags: // 0x00000000 0 0 0 0				flags = zero, negative, carry, overflow	
2	INP X // input for 2 ADD X, #5 OUT X	0000 0000 0000 0111	Registers: // 0x00000000 0000000000000111 0000000000000000 Output: // 0x00000000 0000000000000111 Flags: // 0x00000000 0 0 0 0				Unsigned numbers, so limits: -2 ¹⁵ = 1000000000000000 (C2) si 2 ¹⁵ -1 = 0111 1111 1111 1111	
3	INP X // 3 = 0000 0000 0000 0011 INP Y // 4 = 0000 0000 0000 0100 ADD X, Y OUT X	0000 0000 0000 0111	Registers: // 0x00000000 0000000000000111 0000000000000100 Output: // 0x00000000 0000000000000111 Flags: // 0x00000000 0 0 0 0					

4	INP X // $2^{15}-1 = 0111\ 1111\ 1111\ 1111$ INP Y // $2^{15}-1 = 0111\ 1111\ 1111\ 1111$ ADD X, Y OUT X // inspect FLAGS	1111 1111 1111 1110 0101	Registers: // 0x00000000 11111111111111110 01111111111111111 Output: // 0x00000000 11111111111111110 Flags: // 0x00000000 0 1 0 1				
5	MOV X, #3 MOV Y, #2 SUB X, Y OUT X // inspect FLAGS	0000 0000 0000 0001 0000	Registers: // 0x00000000 00000000000000001 00000000000000010 Output: // 0x00000000 00000000000000001 Flags: // 0x00000000 0 0 1 0	Carry flag unexpectedly set			
6	MOV X, #1 SUB X, #1 OUT X // inspect FLAGS	0000 0000 0000 0000 1000	Registers: // 0x00000000 00000000000000000 00000000000000000 Output: // 0x00000000 00000000000000000 Flags: // 0x00000000 1 0 1 0	Carry flag unexpectedly set			
7	MOV X, #3 MOV Y, #5 SUB X, Y OUT X // inspect FLAGS	1111 1111 1111 1110 0100	Registers: // 0x00000000 11111111111111110 0000000000000101 Output: // 0x00000000 11111111111111110 Flags: // 0x00000000 0 1 0 0				

8	MOV X, #3 // 0000 0000 0000 0011 LSL X, #1 OUT X // inspect FLAGS	0000 0000 0000 0110 0000	Registers: // 0x00000000 00000000000000110 0000000000000000 Output: // 0x00000000 0000000000000110 Flags: // 0x00000000 0 0 0 0				
8	INP X // 1111 0000 0000 0001 LSL X, #1 OUT X // inspect FLAGS	1110 0000 0000 0010 0110	Registers: // 0x00000000 1110000000000010 0000000000000000 Output: // 0x00000000 1110000000000010 Flags: // 0x00000000 0 1 1 0				
9	INP X // 1111 0000 0000 0001 LSR X, #1 OUT X // inspect FLAGS	0111 1000 0000 0000 0010	Registers: // 0x00000000 0111100000000000 0000000000000000 Output: // 0x00000000 0111100000000000 Flags: // 0x00000000 0 0 1 0				
10	MOV X, #6 // 0000 0000 0000 0110 LSR X, #1 OUT X // inspect FLAGS	0000 0000 0000 0011 0000	Registers: // 0x00000000 0000000000000011 0000000000000000 Output: // 0x00000000 0000000000000011 Flags: // 0x00000000 0 0 0 0				

11	INP X // 32774 = 1000 0000 0000 0110 RSL X, #1 OUT X // inspect FLAGS	0000 0000 0000 1101 0010	Registers: // 0x00000000 0000000000001101 0000000000000000 Output: // 0x00000000 0000000000001101 Flags: // 0x00000000 0 0 1 0				
12	MOV X, #7 // 0000 0000 0000 0111 RSR X, #1 OUT X // inspect FLAGS	1000 0000 0000 0011 0110	Registers: // 0x00000000 1000000000000011 0000000000000000 Output: // 0x00000000 1000000000000011 Flags: // 0x00000000 0 1 1 0				
13	MOV X, #3 MUL X, #2 OUT X // inspect FLAGS	0000 0000 0000 0110 0000	Registers: // 0x00000000 0000000000000110 0000000000000000 Output: // 0x00000000 0000000000000110 Flags: // 0x00000000 0 0 0 0				
14	MOV X, #3 MUL X, #0 OUT X // inspect FLAGS	0000 0000 0000 0000 1000	Registers: // 0x00000000 0000000000000000 0000000000000000 Output: // 0x00000000 0000000000000000 Flags: // 0x00000000 1 0 0 0				

15	MOV X, #3 INP Y // -1 MUL X, Y OUT X // inspect FLAGS	1000 0000 0000 0011 0100	Registers: // 0x00000000 1000000000000011 1000000000000001 Output: // 0x00000000 1000000000000011 Flags: // 0x00000000 0 1 0 1	Overflow shouldn't activate here				
16	MOV X, #12 DIV X, #2 OUT X // inspect FLAGS	0000 0000 0000 0110 0000	Registers: // 0x00000000 0000000000000110 0000000000000000 Output: // 0x00000000 0000000000000110 Flags: // 0x00000000 0 0 0 0					
17	MOV X, #1 DIV X, #0 OUT X	Execution should stop	Registers: // 0x00000000 xxxxxxxxxxxxxxxx 0000000000000000 Output: // 0x00000000 xxxxxxxxxxxxxxxx Flags: // 0x00000000 x x 0 0					
18	MOV X, #6 MOD X, #2 OUT X // inspect FLAGS	0000 0000 0000 0000 1000	Registers: // 0x00000000 0000000000000000 0000000000000000 Output: // 0x00000000 0000000000000000 Flags: // 0x00000000 1 0 0 0					

19	MOV X, #6 MOD X, #5 OUT X // inspect FLAGS	0000 0000 0000 0001 0000	Registers: // 0x00000000 0000000000000000 0000000000000000 Output: // 0x00000000 0000000000000001 Flags: // 0x00000000 0 0 0 0				
20	MOV X, #111 // 0000 0000 0110 1111 MOV Y, #0 AND X, Y OUT X // inspect FLAGS	0000 0000 0000 0000 1000	Registers: // 0x00000000 0000000000000000 0000000000000000 Output: // 0x00000000 0000000000000000 Flags: // 0x00000000 1 0 0 0				
21	MOV X, #111 // 0000 0000 0110 1111 INP Y // 1903 = 0000 0111 0110 1111 ORR X, Y OUT X // inspect FLAGS	0000 0111 0110 1111 0000	Registers: // 0x00000000 0000011101101111 0000011101101111 Output: // 0x00000000 0000011101101111 Flags: // 0x00000000 0 0 0 0				
22	MOV X, #111 // 0000 0000 0110 1111 INP Y // 1903 = 0000 0111 0110 1111 XOR X, Y OUT X // inspect FLAGS	0000 0111 0000 0000 0000	Registers: // 0x00000000 0000011100000000 0000011101101111 Output: // 0x00000000 0000011100000000 Flags: // 0x00000000 0 0 0 0				

23	MOV X, #111 // 0000 0000 0110 1111 NOT X OUT X // inspect FLAGS	1111 1111 1001 0000	Registers: // 0x00000000 11111111110010000 0000000000000000 Output: // 0x00000000 11111111110010000 Flags: // 0x00000000 0 1 0 0					
24	MOV X, #3 CMP X, #3 // inspect FLAGS	1000	Flags: // 0x00000000 1 0 1 0	Carry flag unexpectedly active				
25	MOV X, #3 CMP X, #4 // inspect FLAGS	0100	Registers: // 0x00000000 0000000000000011 0000000000000000 Output: // 0x00000000 11111111110010000 Flags: // 0x00000000 0 1 0 0					
26	MOV X, #3 INC X OUT X	0000 0000 0000 0100	Registers: // 0x00000000 0000000000000100 0000000000000000 Output: // 0x00000000 0000000000000100 Flags: // 0x00000000 0 0 0 0					

27	INP X // 0111 1111 1111 1111 INC X // inspect FLAGS	0101	Registers: // 0x00000000 1000000000000000 0000000000000000 Output: // 0x00000000 1000000000000000 Flags: // 0x00000000 0 1 0 1				
28	MOV X, #3 DEC X OUT X	0000 0000 0000 0010	Registers: // 0x00000000 0000000000000010 0000000000000000 Output: // 0x00000000 0000000000000010 Flags: // 0x00000000 0 0 0 0				
29	MOV X, #1 DEC X // inspect FLAGS	1000	Registers: // 0x00000000 0000000000000000 0000000000000000 Output: // 0x00000000 0000000000000010 Flags: // 0x00000000 1 0 1 0	Carry flag unexpectedly active			
30	MOV X, #0 DEC X // inspect FLAGS	0100	Registers: // 0x00000000 1111111111111111 0000000000000000 Output: // 0x00000000 xxxxxxxxxxxxxxxx Flags: // 0x00000000 0 1 0 0				

31	MOV X, #4 MOV Y, #0 BRA somebranch somebranch: MOV Y, #1 OUT Y	0000 0000 0000 0001	Registers: // 0x00000000 0000000000000100 0000000000000001 Output: // 0x00000000 0000000000000001 Flags: // 0x00000000 0 0 0 0				
32	MOV X, #4 MOV Y, #0 BRA somebranch somebranch: MOV Y, X OUT Y	0000 0000 0000 0100	Registers: // 0x00000000 0000000000011000 0000000000011000 Output: // 0x00000000 0000000000011000 Flags: // 0x00000000 0 0 0 0				
32	MOV X, #33 CMP X, #33 BRZ somebranch MOV X, #5 somebranch: MOV X, #1 OUT X	0000 0000 0000 0001	Registers: // 0x00000000 0000000000000001 0000000000000000 Output: // 0x00000000 0000000000000001 Flags: // 0x00000000 0 0 0 0				
33	MOV X, #33 CMP X, #34 BRZ somebranch MOV X, #2 somebranch: ADD X, #1 OUT X	0000 0000 0000 0011	Registers: // 0x00000000 0000000000000011 0000000000000000 Output: // 0x00000000 0000000000000011 Flags: // 0x00000000 0 0 0 0				

33	MOV X, #33 SUB X, #34 BRN somebranch MOV X, #5 somebranch: MOV X, #2 OUT X	0000 0000 0000 0010	Registers: // 0x00000000 0000000000000010 0000000000000000 Output: // 0x00000000 0000000000000010 Flags: // 0x00000000 0 0 0 0				
34	MOV X, #3 SUB X, #2 BRN somebranch ADD X, #2 somebranch: OUT X	0000 0000 0000 0011	Registers: // 0x00000000 0000000000000011 0000000000000000 Output: // 0x00000000 0000000000000011 Flags: // 0x00000000 0 0 0 0				
34	INP X // 1111 0000 0000 0001 LSL X, #1 MOV Y, #1 BRC somebranch MOV Y, #5 somebranch: MOV Y, #3 OUT Y	0000 0000 0000 0011	Registers: // 0x00000000 1111111111111110 0000000000000011 Output: // 0x00000000 0000000000000011 Flags: // 0x00000000 0 0 0 0				
35	MOV X, #4 JMP addfunc OUT X HLT addfunc: ADD X, X ADD X, X RET	0000 0000 0001 0000	Registers: // 0x00000000 0000000000010000 0000000000000000 Output: // 0x00000000 0000000000010000 0000000000010000 Flags: // 0x00000000 0 0 0 0				

36	MOV X, #100 LOG X OUT X	0000 0000 0000 0110	Registers: // 0x00000000 0000000000000110 0000000000000000 Output: // 0x00000000 0000000000000110 Flags: // 0x00000000 0 0 0 0				
37	MOV X, #1 LOG X OUT X	0000 0000 0000 0000	Registers: // 0x00000000 0000000000000000 0000000000000000 Output: // 0x00000000 0000000000000000 Flags: // 0x00000000 1 0 0 0				
38	MOV X, #32 LOG X OUT X	0000 0000 0000 0101	Registers: // 0x00000000 0000000000000101 0000000000000000 Output: // 0x00000000 0000000000000101 Flags: // 0x00000000 0 0 0 0				
39	MOV X, #27 RTT X OUT X	0000 0000 0000 0011	Registers: // 0x00000000 0000000000000011 0000000000000000 Output: // 0x00000000 0000000000000011 Flags: // 0x00000000 0 0 0 0				

40	MOV X, #30 RTT X OUT X	0000 0000 0000 0011	Registers: // 0x00000000 0000000000000011 0000000000000000 Output: // 0x00000000 0000000000000011 Flags: // 0x00000000 0 0 0 0					
----	------------------------------	---------------------	--	--	--	--	--	--