2. (10 points) Cache

Suppose we have a direct mapped cache with **block size four** that is initially empty.

(a) (4 points) Suppose we have the sequence of load words and store words given in the table. Fill in the hit/miss column of the table.

Line #	Instruction	Data memory address (hex)	Data memory address (binary) Tag	Index	Block	Byte	**H**it/ **M**iss
1	LDUR	0x020	000	0001	00	000	M
2	STUR	0x028	000	0001	01	000	H
3	LDUR	0x438	010	0001	11	000	M
4	LDUR	0x430	010	0001	10	000	H
5	LDUR	0x428	010	0001	01	000	H
6	LDUR	0x420	010	0001	00	000	H
7	STUR	0x278	001	0011	11	000	M
8	LDUR	0x220	001	0001	00	000	M
9	LDUR	0x228	001	0001	01	000	H
10	LDUR	0x230	001	0001	10	000	H

(b) (4 points) Given below are empty tables of cache. Fill in each table with the contents of the cache after the instructions at line 3, 7 and 10 have been executed. You must state what data is loaded from Memory. For example, if memory address '0x020' is in the cache, write 'M[020]' under index 0001 and block 00. Note that '0x020' is a hexidecimal number, not a decimal number.

Be sure to fill in the Tags and Dirty bits. If a row is empty (i.e., hasn't been accessed and not valid), you may leave the row blank.

Cache after executing Lines 1, 2, and 3:

			Block			
Index	**Tag**	**D**	**00**	**01**	**10**	**11**
0000						
0001	010	0	M[420]	M[428]	M[430]	M[438]
0010						
0011						

Cache after executing Lines 1 through 7:

			Block			
Index	**Tag**	**D**	**00**	**01**	**10**	**11**
0000						
0001	010	0	M[420]	M[428]	M[430]	M[438]
0010						
0011	001	1	M[260]	M[268]	M[270]	M[278]

Cache after executing Lines 1 through 10:

Index	Tag	D	Block 00	Block 01	Block 10	Block 11
0000						
0001	001	0	M[220]	M[228]	M[230]	M[244]
0010						
0011	001	1	M[260]	M[268]	M[270]	M[278]

(c) (2 points)

Assume the cache uses a write-back scheme, where data written to cache is only written to memory when the cache block is replaced. At this point, the entire cache block is written back to memory.

Circle the line number of the instructions above that *cause* a cache block to be written to memory. You need to circle the **line numbers** and not the number of instructions that cause a write-back.

#1 #2 #3 #4 #5 #6 #7 #8 #9 #10