1. C=2^21 bytes B=2^6 bytes line = C/B=32768
    a. For A j misses every one in eight so 2048/8=256 misses, i misses every access so that is 1024 misses all together that is 262144 misses. For B j is 2048 misses and i is 1024/8 = 128 misses all together B misses 262144. Thus, in total there are 524288 misses.

    b. Due to the size of j and i for A and B there is no difference between a direct mapped and fully associative caches. The total number of misses is 524288.
    c. Because the number of set 2048 is larger than j is has the same number of misses for both A and B the same is true for i. In i's case because the distance between A[i][j] and A[i+1][j] is greater than the block size we are automatically getting another miss. Thus in total we are getting 524288 misses.
2. S
    a. See table

	A	B	C
I	NA	1	32
J	1	65536	NA
K	256	NA	1
L	8192	128	NA

    b. In a fully associative cache A has 131072 misses and B has 524288 misses.
    c. In a direct mapped cache, A has 131072 misses and B has 524288 misses. This is because due to the access patterns of the data there is nothing taking advantage of temporal or spatial locality beyond the last index.
    d. I would order them by k I j l as j and l are used on accessing both of A and B but I and k only work with one. I would put k on top as it is the smaller loop thus keeping the untouched variable in memory for longer.