Part 1

1. What is the hit rate? Why is it that much? and 2. How many compulsory misses? Why is it that much?

0.000 hit rate because the first 256 misses are compulsory. There are 256 (64*4) words in our matrix, but only 128 cache lines. Since the cache is too small, each one is accessed only after the rest of the row has been cleared from the cache. After this, each miss is a capacity miss since it has been loaded before but since been overwritten due to the cache being completely full.

3. How are the compulsory misses affected when Block Size is changed from 4 bytes to 8 bytes? Why does it change?

The amount of compulsory misses decreases to 128 since 2 words are loaded at once instead of one into the cache. Thus the amount of compulsory misses is cut in half, but we still miss these accesses to capacity misses because the cache is too small and the loop is in a bad order.

4. Change the block size back to 4 again, and change the mapping from "direct mapping" to "2ways set associative"? Explain why the hit rate does or doesn't change.

it does not change because all our misses are compulsory and capacity misses, which are not helped by 2-way set associative caches.

5. What cache size improves the hit rate? Why is the hit rate improved? Try with both "direct mapping and "2-ways set associative".

With the cache size 1024 Bytes the hit rate is improved to 0.9 because the cache now can fit the whole matrix, and only needs to miss the first loop through (these are compulsory misses). There is no difference between direct mapping and 2-way set associative since none of the misses are conflict misses.

6. At this "optimal" cache size (question 5), and "direct mapping", what hit rates do you obtain by changing the Block size? Try with the sizes 4 bytes, 8 bytes and 16 bytes.

By increasing the block size, the amount of compulsory misses is decreased since more words are loaded for each access. With 8 bytes, the amount of compulsory misses is 128 and the hit rate 0.95 , and at 16 bytes, 64 misses with a hit rate of 0.975.

7. Show how the achieved hit rate (as shown by simulator) can be computed (by hand) for different block sizes for the "optimal" cache size you found in question 5. Hint: what is the size of a single element in the array?

The matrix is 256 words. The cache block size decides how many words are loaded at once for each access. So a cache with block size 4B loads each word one at a time and so needs 256/1 = 256 misses, whereas a cache with block size 8B loads words two at a time and so

needs 256/2 = 128 misses. Similarly, a cache with block size 16B loads words four at a time and needs 256/4 = 64 misses

Part 2

1. Does the hit rate improve? Why (show computation)?

The hit rate does not improve since the cache still fills up and we still evict each word before it is used again.

2. Change the Block Size to 8 bytes? Does the hit rate improve? Why (show computation)?

Yes, the hit rate is now 0.5 because we load two words at a time so every other access is a cache hit - 2560/2 = 1280 misses, 1280/2560=0.5 hit rate. However we still evict every line before that line is accessed again.

3. Change the Block Size to 16 bytes? Does the hit rate improve? Why (show computation)?

Yes, the hit rate is now 0.75 because we load four words at a time so every other access is a cache hit - 2560/4 = 640 misses and 2560-640 =1920 hits, 1920/2560=0.75 hit rate

4. Change the Cache Size to 1024 bytes and Reset the Block Size to 4 bytes. What is the hit rate? Compute this hit rate by hand

The hit rate is 0.9 since we load the entire matrix into the cache during the first iteration of the loop, meaning 1 out of 10 iterations contains 1 compulsory miss per word in the matrix: 1/10 = 0.1 miss rate 1-0.1=0.9 hit rate

5. Change Block Size to 8 bytes. What is the hit rate? Compute this hit rate by hand

Now, we load two words at a time into the cache so we have half the number of compulsory misses in the first iteration - (½)/10=0.05 miss rate, 1-0.05=0.95 hit rate

6. Change Block Size to 16 bytes. What is the hit rate? Compute this hit rate by hand

Now, we load four words at a time into the cache so we have one quarter the number of compulsory misses in the first iteration - (¼)/10=0.025 miss rate, 1-0.025=0.975 hit rate

7. Can a perfect hit rate of 1.0 be achieved without changing the program? Why?

No, we still need to load the matrix into the cache, meaning in the best case we load the entire matrix into the cache at once which entails 1 miss out of 2560 = 1/2560=0.00039 miss rate, 1-0.00039 = 0.99961 hit rate.