1. **A)** Blocks of main memory= Size of main memory (2^{16}) divided by size of a cache block (2^5). $2^{16}/2^5= 2^{11}$ which equals 2048.
Final result: There are 2048 blocks of main memory.

B) In total we have 16 bits, and we know that the cache size is 256 bytes which means that the block should be $2^8/2^5=2^3$. We also know that each block contains 32 bytes which is 2^5 so the offset is 5. That leaves us with the tag which is the total number of bits – what's left which is 16-(5+3)= 8.

Tag= 8	Block= 3	Offset= 5

C) First we convert the hexadecimal number to binary so 43B2 = 0100001110110010. The last 5 bits are the offset bits. And since 3-bit block of cache is 101 in binary, then the memory will map to block 5.
Final result: block 5.

2. **A)** Size of main memory is 4k*8 words each = $2^{10}*4 = 2^{12}$ blocks. Now 8 = 2^3, therefore $2^3*2^{12} = 2^{15}$ words. So we need 15 bits to address each location. We already know what our block is, and we also know that we have 4 sets which means that we have 2 bits in the index and the rest is for the tag.

Tag= **10**	Index= **2**	Block= **3**

B)

32-39	
8-15	40-47
16-23	48-55
24-31	

We have 4 sets. Starting from location 8 till 51. 8 in binary is 000000000001000 so the index is (01) which is a miss. Everything else (9-15) are hits. Next is 16 which in binary is 000000000010000 and the index is (10) which is a miss and everything else (17-23) are hits. Next is 24 which in binary is 000000000011000 and index is (11) and that's a miss and everything else is a hit (25-31). The next 32 is 000000000100000 in binary and the index is (00) which is a

miss and the rest (33-39) are hits. Next is 40 which is 000000000101000 with index (01) and is also a miss but the rest are hits (41-47). The next is 48 which in binary is 11000 and index is (10)which is a miss and the rest (48-51) are hits.
1st iteration we have 38 hits
2nd iteration we have 44 hits
3rd iteration we have 44 hits
Total misses: 6 misses
Total hits: 132-number of miss(6)= 126
Hit ration: number of hits/ total = 0.9545%

3. A) Since the maximum size of virtual address space is 32MB, we know that 1MB is 2^{20}, therefore 32MB is $2^5*2^{20} = 2^{25}$. So, the total number of bits required is 25 bits.
Final answer: 25 bits are required for each virtual address.

B)Since the maximum physical memory size of the machine is 2MB, we know that 1MB is 2^{20} therefore, 2MB is $2^{20}*2^1$ which is 32.
Final answer: 21 bits are required for each physical address.

C) We know that the page size is 1024 bytes = 2^{10} and virtual memorize size is 32MB which is 2^{25}. Therefore, the number of page entries is the size of virtual memorize divided by the page size. $2^{25}/2^{10} = 2^{15}$. The page number is therefore 15 and it contains the offset which is the remaining bits. Virtual address is 1524 which in binary is equivalent to 10111110100_2. Since the binary number is 11 bits and our virtual address is 25 bits, we need to add extra 0's to fill in. Virtual address is now 0000000000000010111110100. As we said, we take the first 15 bits (000000000000001),and the remaining bits are offset (0111110100). 000000000000001 is really 1, meaning that virtual page 1 is going to map to **frame 2** as given to us. Frame 2 in binary is 000000000000010 and when we add it with the offset we get 0000000000000100111110100 which when we convert to decimal we get 2548_{10}
Final answer: 2548_{10} is the physical address it'll map to (frame 2).

4. A) 5ns (TLB access)+ 25ns (Memory reference)+ 12ns (Cache access) = 42ns therefore when TLB misses, then page table can be accessed.

B) 5ns (TLB access)+25ns (Memory reference)+200ms (Disk reference)+5ns(TLB access when restarted)+12ns(Cache access)= 47ns+200ms.