1.)   a.) $2^{12} \times 2^{16} = 2^{28}$ logical addresses $\implies$ 28 bits.

b.) Since frames and pages have the same size, each frame is $2^{12} = 4096$ bytes.

c.) The number of frames is given by $2^{32}/2^{12} = 2^{20}$, thus 20 bits are used for each frame.

d.) Since there are 2^{16} pages, the page table has 2^{16} entries.

e.) Each page requires 20 bits plus the valid/invalid bit, thus each entry has 21 bits.

2.)   a.) You must access the page table, and then the actual memory, thus it would take $0.2 + 0.2 = 0.4$ microseconds.

b.) If the reference is found in an associative register, it takes 10 nanoseconds to access it, and if it is not, it takes 20 nanoseconds, thus the effect memory reference time is $0.75 \times 10 + 0.25 \times 20 = 7.5 + 5 = 12.5$ nanoseconds.

3.)   a.) FIFO Fault Order: FFFFFFFFHHHFFHHFFFF, thus 14 faults.

b.) LRU Fault Order: FFFFFFFFHHHFFHHFFFF, thus 14 faults.

c.) Optimal Fault Order: FFFFFHHFHHHFHHHFHFH, thus 9 faults.

4.) Assuming $M < N$, or else there would never be any page faults.

a.) If the string has N distinct page numbers, then there will be at least N faults given a fault occurs every time a new page number is encountered. After all distinct page numbers have been encountered however, it is possible that the string only contains page numbers that already exist in the M frames, thus the lower bound for the number of faults is N.

b.) Alternatively, you could choose a string that repeatedly contains page numbers that are not currently in the M pages, thus it would incur P faults, which is clearly the maximum possible for a string of length P.