1. Problem 1.1
2. Problem 1.2
    (a) The PC contains 300, and the address of the first instruction. This address is then given to the MAR, so that the value in 300 may be loaded to the MBR, and subsequently loaded into the IR. The PC is then incremented.
    (b) The first 4 bits in the IR indicate that the AC is to be loaded from memory. The remaining 12 bits are the address which is to be loaded from. The address is then given to the MAR, and the contents of the address are written to the MBR, which are then written to the AC.
    (c) 301 is then given to the MAR, and the instruction is loaded into the MBR, and written to the IR.
    (d) The next instruction says to add the contents of the next memory address to the AC, so the address is given to the MAR, and the contents are given to the MBR. The contents are then added to the AC, and are put back in the AC.
    (e) The next location is given to the MAR, and the instruction is written to the MBR, which then writes the instruction to the IR.
    (f) The memory address of the instruction is given to the MAR, and the contents of the AC are written to the MBR, and the written to the memory address in the MAR.
3. Problem 1.7
   The DMA is given priority to main memory over the processor because the process of moving blocks of data should not be interrupted, else errors may occur. In order to ensure there is no interrupt issued, the DMA must have priority over the processor.
4. Problem 1.10.
    (a) the line: $a[i] = a[i] * j$ is an example of spatial locality, because it involves the storage of blobs in succession
    (b) the two 'for' loops are examples of temporal locality, because they repeat the same instruction several times
5. Problem 1.11
    (a) equation 1.1 :
    (b) equation 1.2 : $C_s = \frac{\Sigma C_n S_n}{\Sigma S_n}$
6. Problem 1.14
   In a stack architecture, the PC can be substituted with the top of the stack. The top of the stack will act as the PC, and popping the top off the stack will be the same as moving the instruction from the PC to the IR.