1. There are a lot of ways to do this. The most obvious is to A) design a State Diagram and then B) turn it into gates. Another way is to use an ad hoc method specific to this problem.

State diagram:

(Converting that to gates not shown)

Ad hoc:


Here we just check if the current inputs are the same, we store that and use it to see if the last one was the same too (the AND gate). It may not seem like the last flip-flop is needed, but if you don't have it, your output will depend directly on the input and that's a no-no.


The blue + sign is an 8-bit adder with a 9-bit output (so including the Cout as the MSb).
The green + is, similarly, a 9-bit adder with 10 bits of output.
The four last values are added together and then shifted right by 2 (by taking the 8 most-significant bits), which is the same as dividing by 4 . One could find ways to round to the nearest rather than rounding down (most easily by setting two of the carry-in bits to be a one...).
3. There is an error in the diagram: There is no way to start at memory location 25 . This can be fixed in a number of ways including changing the 25 going into the adder to a 24 .

There are a number of correct answers. Here is one.

4. No answer supplied. We'll take anything reasonable.

