A. On average, how many instructions are executed in the inner loops of the two programs?
The inner loop of the code using the conditional jump has 9 instructions, all of which are executed when the array element is zero or negative, and 8 of which are executed when the array element is positive. The average is 8.5. The inner loop of the code using the conditional move has 8 instructions, all of which are executed every time.

B. On average, how many bubbles would be injected into the inner loops of the two programs?
The loop-closing jump will be predicted correctly, except when the loop terminates. For a very long array. this one misprediction will have a negligible effect on the perfomance. The only other source of bubbles for the jump-based code is the conditional jump, depending on whether or not the array element is positive. This will cause two bubbles, but it only 50% of the time, so the average is 1.0. There are no bubbles in the conditional move code.

C. What is the average number of clock cycles required per array element for the two programs?
Our conditional jump code requires an average of $8.5 + 1.0 = 0.5$ cycles per array element (9cycles in the best case and 10 cycles in the worst), while our conditional move code requires 8.0 cycles in all cases.