

- A. Explain, in mathematical terms, the logic in the computation of s_2 on lines 5 – 7. *Hint:* Think about the bit-level representation of -16 and its effect in the *andq* instruction of line 6.**

The *leaq* instruction of line 5 computes the value $8n + 22$, which is then rounded down to the nearest multiple of 16 by the *andq* instruction of line 6. The resulting value will be $8n + 8$ when n is odd and $8n + 16$ when n is even, and this value is subtracted from s_1 to give s_2 .

- B. Explain, in mathematical terms, the logic in the computation of p on lines 8-10.**

The three instructions in this sequence round s_2 up to the nearest multiple of 8. They make use of the combination of biasing and shifting that we saw for dividing by a power of 2 in Section 2.3.7.

C.

- D.** We can see that s_2 is computed in a way that preserves whatever offset s_1 has with the nearest multiple of 16. We can also see that p will be aligned on a multiple of 8, as is recommended for an array of 8-byte elements.