How is the answer 3? Could someone explain how they get this answer?

One cons that points to two cons object, one of which holds the two values, one of which holds the next and prev pointers. So in total 3 cons.

Correct. If you have n pieces of information, you need n-1 cons cells at a minimum. If you want to keep your n items in a list, you need n cons cells, one for each item.

So just to be clear this would be the code for such task:

(cons (cons val1 val2) (cons prev next))

A drawing of this would look something like this, correct? Where each box is a call to cons.

[ val1 | val2 ] <-- [   ] --> [ prev | next ]

I believe that's correct