Basically, the “fun” function will take in a pointer, and 4 to it and then will return 3 times that value and will subtract 1. With no precedence rule, based on the two operations:

sum1 = (i / 2) + fun(&i); = 46

sum2 = fun(&j) + (j / 2); = 48

It would seem like sum1 and sum2 would end up with the same values. But, because the precedence rule in C is evaluation from left to right, it would actually change the values. For sum1, the value of i is not affected because the fun(&i) happens after (i/2). In sum2; however, fun(&j) is called before (j/2). This means that fun(&j) will return 41 and the value of j will now be 14. So 41 + 7 equals 48. Just to reiterate, if there were no precedence rules, then the values of i and j would not be different in both cases resulting in the same value for each operation/expression.