Solution Idea:

This problem is a nice example of segment tree with lazy propagation. All you need to solve this kind of problem is to do some experiment with the formula and convert them into a suitable form. From which we can drive the segment tree solution.

In this problem given an integer array A=[a1,a2,a3,a4…]. You need to perform 3 kinds of operation on it.

type 0: Set all the element in the interval l to r to x.
type 1: Increase all the element in the interval l to r by x.
type 2: Print the value of square sum of the interval l to r. For example let l=1 and r=3. then you need to print a1^2 + a2^2 + a3^2.

At first think we have a pre-calculated square sum array. Then if we perform type 1 operation which is increase it by x then what scenario will happen?

Our sum now become (a1+x)^2 + (a2+x)^2 + (a3+x)^2. Let expand this equation-

(a1+x)^2 + (a2+x)^2 + (a3+x)^2
= a1^2+ 2*a1*x + x^2 + a2^2+ 2*a2*x + x^2 + a3^2+ 2*a3*x + x^2
= (a1^2 + a2^2 + a3^2) + 3*x^2 + 2*x*(a1 + a2 + a3)

for a range l to r the generalized equation is –

(a_l^2 + a_l+1^2 +….+ a_r^2) + (r-l+1)*x^2 + 2*x*(a_l + a_l+1 +….+ a_r)

We can store the information about sum, square_sum, type_0 lazy and type_1 lazy of an interval in each segment tree node. And perform type 1 operation then we can calculate then sum value for a node over a range by multiplication and square sum value by above equation.

For every type 0 operation we can simply put the value of sum and square sum by using some basic calculation and arithmetic multiplication operation.

And for every type 2 operation we just need to get the sum of square sum value over a range l to r.

So we can perform each of 3 operation using a segment tree with lazy propagation.