

Liam Chapman, Calum Hunter - Cosc326 - Etude 13 - Counting up.

Compiles with `javac CountingUp.java`

Runs with `java CountingUp < test.txt`

Expected output: the result from each inputted line, one line at a time.

The first thing our method does (not including the checks in the main to insure the correct values are inputted) is check the size of K , if it is less than half n , make $k = n - k$. We do this to make it easier to loop as $n = 100$, $k = 3$ is the same as $n = 100$, $k = 97$. Then we loop through from n to k (n —each iteration) adding the new n values to an ArrayList (be the top half) and adding the new n values minus k to another ArrayList (be the bottom half). We do this because it eliminates a majority of the equation (a.k.a eliminating common factors). After this we do another check to see if there are any common factors left in the top and bottom halves, eliminating them as we go to decrease the equation size. Next we split apart the top and bottom halves into their prime factors which makes it easier for again eliminating common factors, which is what we do next. After all the possible common factors have been eliminated from the top and bottom halves we then calculate the sum of them by timesing all remaining values in our ArrayLists, dividing the top by the bottom, resulting in the answer.

The reason behind getting rid of all the common factors is so we can times them all together without getting in number overflows (making sure we don't go over $2^{63}-1$).