**Approach**: The idea is to first find the wallet balance of each friend, assuming that the wallet balance of all friends are made the same. As we can see, the summation of the wallet balance of all the friends remains the same after any number of transactions, therefore the wallet balance of each friend, if the wallet balance of all the friends are the same will be equal to the average wallet balance of the **'N'** friends.

So our first idea is to check whether the average balance is an integer. If it is not an integer, then we will return -1. As we can transact in integer values only. The other condition to check is that for each friend, the difference of his initial wallet balance and the average wallet balance should be a multiple of **‘K’**. If it is not, then we will return -1.Otherwise, it is guaranteed that an answer exists.

This works because we are allowed to transact in multiples of ‘**K’**only. This means that the wallet balance of a friend increases or decreases in multiples of '**K'**only. Therefore, it is necessary that the difference between initial wallet balance and the average wallet balance of friends is multiple of**‘K’.**

To find the transactions, we can iterate through all the friends having an initial balance greater than the average balance and count how many transactions are required for each friend so that their wallet balance becomes exactly equal to the average balance. Our final answer will be the total number of transactions that are carried out.

**Algorithm:**

1. Let ‘BALANCESUM’ denote the sum of the wallet balance of all the friends.
2. If ‘BALANCESUM’ is not divisible by ‘K’, then we will return -1.
3. Otherwise, we will set ‘AVERAGEBALANCE’ as the quotient of the division of ‘BALANCESUM’ and ‘N’.
4. Define the ‘TRANSACTIONCOUNT’ variable to store the total count of transactions. Initialize it as 0.
5. Iterate from ‘i’ = 0 to ‘N - 1’.
   * Define ‘BALANCECHANGE’ as the difference between ‘AVERAGEBALANCE’ and ‘ARR[i]’.
   * If ‘BALANCECHANGE’ is not divisible by ‘K’, then we will return -1. As it is impossible to make the wallet balance of all friends equal.
   * If ‘BALANCECHANGE’ is positive, then we will add the quotient of the division of ‘BALANCECHANGE’ by ‘K’ to ‘TRANSACTIONCOUNT’. We are not adding transactions for negative ‘BALANCECHANGE’, because, in one transaction, the balance of one friend decreases by ‘K’, and the balance of some other friend decreases by ‘K’ simultaneously. So we can consider all the positive ‘BALANCECHANGE’, or we can consider only the negative ‘BALANCECHANGE’.
6. Return the ‘TRANSACTIONCOUNT’ variable.