Kick Start 2021 - Round A

Analysis: K-Goodness String

As per the given definition of operation, Ada can only change the goodness score by one in a single move. Therefore to get a string with the required goodness score in the minimum number of operations Ada can either increase or decrease the goodness score by one in each step. Let us assume there are X indices($\leq \mathbf{N}/2$ (1-indexed)) i in the given string $\mathbf{S}$ such that $\mathbf{S}_i \neq \mathbf{S}_{\mathbf{N}-i+1}$. We have 3 cases now.

- Case 1: $X = \mathbf{K}$,
  In this case, Ada already has the string which has a goodness score of $\mathbf{K}$. Therefore number of operations required is 0.
- Case 2: $X > \mathbf{K}$,
  In this case, Ada can change $(X - \mathbf{K})$ letters at indices i ($1 \leq i \leq \mathbf{N}/2$ (1-indexed)) that satisfy $\mathbf{S}_i \neq \mathbf{S}_{\mathbf{N}-i+1}$ to the letter at $\mathbf{S}_{\mathbf{N}-i+1}$. Then she will have a string with a goodness score of $\mathbf{K}$. Therefore the minimum number of operations is $(X - \mathbf{K})$.
- Case 3: $X < \mathbf{K}$,
  In this case, Ada can change $(\mathbf{K} - X)$ letters at indices i ($1 \leq i \leq \mathbf{N}/2$ (1-indexed)) that satisfy $\mathbf{S}_i = \mathbf{S}_{\mathbf{N}-i+1}$ to any other uppercase letter other than the one at $\mathbf{S}_{\mathbf{N}-i+1}$. As a result, she will have a string with a goodness score of $\mathbf{K}$. Therefore the minimum number of operations is $(\mathbf{K} - X)$.

All the above described operations can be done in $O(\mathbf{N})$ complexity. Therefore the overall complexity is $O(\mathbf{N})$.

Sample Code (C++)

```
int kGoodnessString(string s, int k) {
  int minOperations = 0, x = 0;
  for(int i = 0; i < s.size() / 2; i++) {
     if(s[i] != s[s.size() - i - 1]) {
         x++;
     }
  }

  if(x == k) {
      minOperations = 0;
  }
  else if(x > k) {
      minOperations = x - k;
  }
  else {
      minOperations = k - x;
  }
  return minOperations;
}
```