K-GOODNESS STRING

M1 – Jarugula Hari Krishna
M2 – Ravi Kakarla
M3 – Mohammed Sameer
M4 – Priyan Kishore M S
M5 – Sandeep Kumar

Index

Abstract:

A collaborative project to learn open-source applications and implement them.
K-goodness String is an interactive coding question that we have used to analyse it and use it as an aid to learn other open-source applications such as git and GitHub.
We have also used other open-source applications such as vim and libre office to complete this project.

Problem:

Charles defines the goodness score of a string as the number of indices i such that $S_i \neq S_{N-i+1}$ where $1 \leq i \leq N/2$ (1-indexed). For example, the string CABABC has a goodness score of 2 since $S_2 \neq S_5$ and $S_3 \neq S_4$.

Charles gave Ada a string S of length N, consisting of uppercase letters and asked her to convert it into a string with a goodness score of K. In one operation, Ada can change any character in the string to any uppercase letter. Could you help Ada find the minimum number of operations required to transform the given string into a string with goodness score equal to K?

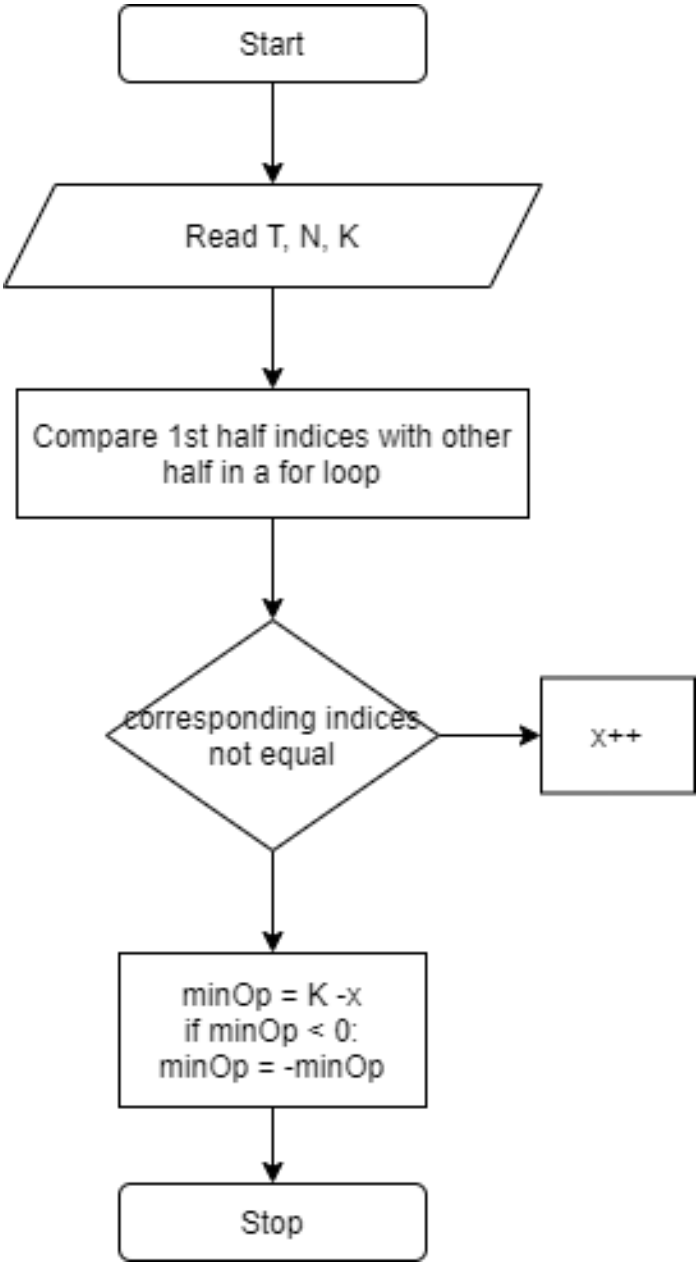
Input
The first line of the input gives the number of test cases, T. T test cases follow.

The first line of each test case contains two integers N and K. The second line of each test case contains a string S of length N, consisting of uppercase letters.

Output
For each test case, output one line containing Case #x: y, where x is the test case number (starting from 1) and y is the minimum number of operations required to transform the given string S into a string with goodness score equal to K.

Limits
Memory limit: 1 GB.
$1 \leq T \leq 100$.
$0 \leq K \leq N/2$.
Test Set 1
Time limit: 20 seconds.
$1 \leq N \leq 100$.
Test Set 2
Time limit: 40 seconds.
$1 \leq N \leq 2 \times 105$ for at most 10 test cases.
For the remaining cases, $1 \leq N \leq 100$.

Flowchart:


Start
Read T, N, K
Compare 1st half indices with other half in a for loop
corresponding indices not equal
x++
minOp = K -x
if minOp < 0:
minOp = -minOp
Stop

Sample Input/Output

Sample Input
2
5 1
ABCAA
4 2
ABAA

Sample Output

Case #1: 0
Case #2: 1

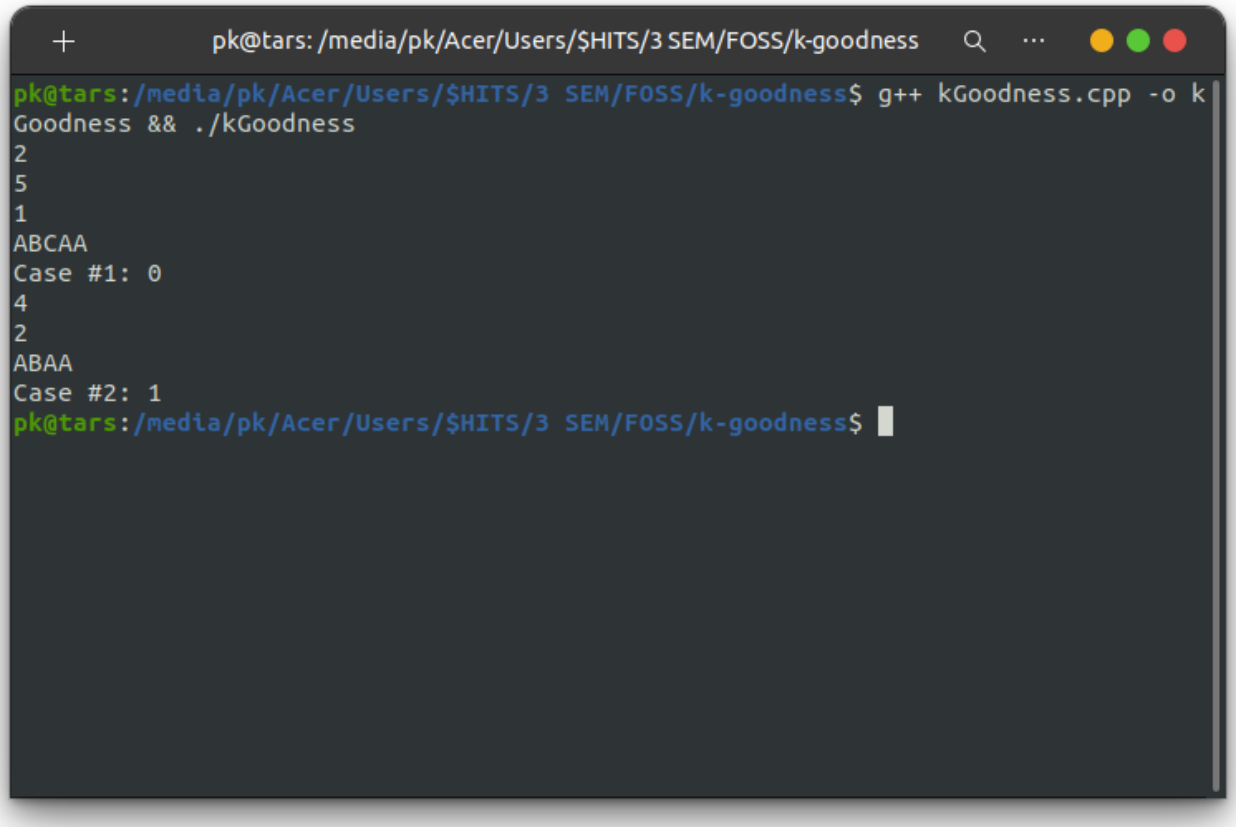
In Sample Case #1, the given string already has a goodness score of 1. Therefore, the minimum number of operations required is 0.

In Sample Case #2, one option is to change the character at index 1 to B in order to have a goodness score of 2. Therefore, the minimum number of operations required is 1.

Program:

```cpp
#include <iostream>
#include <string>
using namespace std;

int main()
{
    int T; //No. of test cases
    string s;

    cin >> T;
    for (int i = 1; i <= T; i++)
    {
        int N, K; //Length of string and goodness score
        cin >> N >> K;
        cin >> s;
        int x = 0;
        for (int i = 0; i < N / 2; i++) //Checking indices from first half wit
h the other half of the stiring
        {
            if (s[i] != s[N - i - 1]) //if the corresponding indices are not e
qual, x is increased by 1
                x++;
        }
        /*
        Case 1: X=K,
        The string already has a goodness score of K. Therefore number of oper
ations required is 0.

        Case 2: X>K,
        The string has a goodness score of X which is greater than K, so the m
inimum operations to change the string with
        goodness score of K is X - K.

        Case 3: X<K,
        The string has a goodness score of X which is lesser than K, so the mi
nimum operations to change the string with
        goodness score of K is K - X.
        */
        int minOp = K - x;
        if (minOp < 0)
            minOp = -minOp;
        printf("Case #%d: %d\n", i, minOp);
    }
    return 0;
}
```

Output:


pk@tars: /media/pk/Acer/Users/$HITS/3 SEM/FOSS/k-goodness
pk@tars:/media/pk/Acer/Users/$HITS/3 SEM/FOSS/k-goodness$ g++ kGoodness.cpp -o k
Goodness && ./kGoodness
2
5
1
ABCAA
Case #1: 0
4
2
ABAA
Case #2: 1
pk@tars:/media/pk/Acer/Users/$HITS/3 SEM/FOSS/k-goodness$