Day 13 / 100 :

Topic - Array

1️⃣ Problem statement: **Maximize the count of exam** (medium)

A teacher is writing a test with n true/false questions, with 'T' denoting true and 'F' denoting false. He wants to confuse the students by maximizing the number of consecutive questions with the same answer (multiple trues or multiple falses in a row).

You are given a string answerKey, where answerKey[i] is the original answer to the ith question. In addition, you are given an integer k, the maximum number of times you may perform the following operation:

Change the answer key for any question to 'T' or 'F' (i.e., set answerKey[i] to 'T' or 'F'). Return the maximum number of consecutive 'T's or 'F's in the answer key after performing the operation at most k times.

Example 1:

Input: answerKey = "TTFF", k = 2
Output: 4
Explanation: We can replace both the 'F's with 'T's to make answerKey = "TTTT".
There are four consecutive 'T's.
Example 2:

Input: answerKey = "TFFT", k = 1
Output: 3
Explanation: We can replace the first 'T' with an 'F' to make answerKey = "FFFT".
Alternatively, we can replace the second 'T' with an 'F' to make answerKey = "TFFF".
In both cases, there are three consecutive 'F's.

Solutions :

Approach 1 - Sliding window

Intuition & Approach:

The given code uses a sliding window approach to solve the problem. Here's a step-by-step explanation of the intuition and approach:

1.
2. Initialize variables t and f to keep track of the counts of 'T's and 'F's encountered so far, and j to represent the start of the window.
3. Initialize ans to store the maximum length of consecutive 'T's or 'F's that satisfy the given constraint.
4. Iterate through each character of the input string s.
5. If the current character is 'T', increment t; otherwise, increment f.
6. Check if the number of 'T's (t) or 'F's (f) exceeds the given constraint (k). If so, move the window start j one position forward and decrease the count of the corresponding character ('T' or 'F') until the constraint is satisfied.
7. Update ans with the maximum length of consecutive 'T's or 'F's encountered so far.
8. Repeat steps 4-6 until all characters in s are processed.
9. Return the final value of ans, which represents the maximum length of consecutive 'T's or 'F's that satisfy the given constraint.
10. The idea behind this approach is to maintain a sliding window that contains only the allowed number of 'T's and 'F's. By moving the window start j forward whenever the constraint is violated, we can find the maximum length of consecutive 'T's or 'F's that satisfy the constraint.

Complexity:
Time complexity:O(n)
Space complexity:O(1)

```
class Solution {
public:
    int maxConsecutiveAnswers(string s, int k) {
    int t = 0, f = 0;  // Count of 'T's and 'F's encountered so far
    int j = 0;         // Pointer to the start of the window
    int ans = 0;       // Maximum length of consecutive 'T's or 'F's

    for (int i = 0; i < s.size(); i++) {
        if (s[i] == 'T')
            t++; // Increment 'T' count
        else
            f++; // Increment 'F' count

        // Check if the count of 'T's or 'F's exceeds the constraint
'k'
        while (t > k && f > k) s[j++]=='T' ? t--:f--;
            // Move the window start 'j' forward and decrease the
count of the corresponding character


        // Update 'ans' with the maximum length of consecutive 'T's
or 'F's encountered so far
        ans = max(ans, t + f);
    }

    return ans; // Return the maximum length of consecutive 'T's or
'F's
}

};
```

2 Problem statement: Zigzag Conversion (medium)

The string "PAYPALISHIRING" is written in a zigzag pattern on a given number of rows like this: (you may want to display this pattern in a fixed font for better legibility)

```
P   A   H   N
A P L S I I G
Y   I   R
```

And then read line by line: "PAHNAPLSIIGYIR"
Write the code that will take a string and make this conversion given a number of rows:
string convert(string s, int numRows);

Example 1:

Input: s = "PAYPALISHIRING", numRows = 3
Output: "PAHNAPLSIIGYIR"
Example 2:

Input: s = "PAYPALISHIRING", numRows = 4
Output: "PINALSIGYAHRPI"
Explanation:

```
P     I    N
A   L S  I G
Y A   H R
P     I
```

Solutions :

Approach 1 - 2 pointers

Explanation:

1. Start traversing the input string from left to right chracter by chracter
2. For the first numRows value, put the incoming chracter in the first column (starting from 0th row to last row)
3. For the next set of values, put the incoming chracter in the next column decrementing row value one by one
4. Repeat steps 2 and 3 until there are no more chracters left to be processed.

Complexity:
Time complexity: O(n)
Space complexity: O(n)

```
class Solution {
public:
    string convert(string s, int numRows) {
        vector<string> v(numRows,"");
        int counter = 1;
        string ans = "";
        int index = 0;

        if(numRows <= 1)
          return s;

      for(int i=0; i<s.length(); i++)
        {
            v[index] += s[i];
            if(index == 0)
              counter = 1;
            if(index == numRows - 1)
              counter = -1;

              index += counter;
        }

        for (int i = 0; i < numRows; i++)
         {
               ans += v[i];
         }
         return ans;
    }
};
```