You are given a **0-indexed** binary string target of length n. You have another binary string s of length n that is initially set to all zeros. You want to make s equal to target.

In one operation, you can pick an index i where 0 <= i < n and flip all bits in the **inclusive** range [i, n - 1]. Flip means changing '0' to '1' and '1' to '0'.

Return *the minimum number of operations needed to make* s *equal to* target.

**Example 1:**

Input: target = "10111"  
Output: 3  
Explanation: Initially, s = "00000".  
Choose index i = 2: "00000" -> "00111"  
Choose index i = 0: "00111" -> "11000"  
Choose index i = 1: "11000" -> "10111"  
We need at least 3 flip operations to form target.

**Example 2:**

Input: target = "101"  
Output: 3  
Explanation: Initially, s = "000".  
Choose index i = 0: "000" -> "111"  
Choose index i = 1: "111" -> "100"  
Choose index i = 2: "100" -> "101"  
We need at least 3 flip operations to form target.

**Example 3:**

Input: target = "00000"  
Output: 0  
Explanation: We do not need any operations since the initial s already equals target.

**Constraints:**

* n == target.length
* 1 <= n <= 105
* target[i] is either '0' or '1'.