There is a bookstore owner that has a store open for n minutes. Every minute, some number of customers enter the store. You are given an integer array customers of length n where customers[i] is the number of the customer that enters the store at the start of the ith minute and all those customers leave after the end of that minute.

On some minutes, the bookstore owner is grumpy. You are given a binary array grumpy where grumpy[i] is 1 if the bookstore owner is grumpy during the ith minute, and is 0 otherwise.

When the bookstore owner is grumpy, the customers of that minute are not satisfied, otherwise, they are satisfied.

The bookstore owner knows a secret technique to keep themselves not grumpy for minutes consecutive minutes, but can only use it once.

Return *the maximum number of customers that can be satisfied throughout the day*.

**Example 1:**

Input: customers = [1,0,1,2,1,1,7,5], grumpy = [0,1,0,1,0,1,0,1], minutes = 3  
Output: 16  
Explanation: The bookstore owner keeps themselves not grumpy for the last 3 minutes.   
The maximum number of customers that can be satisfied = 1 + 1 + 1 + 1 + 7 + 5 = 16.

**Example 2:**

Input: customers = [1], grumpy = [0], minutes = 1  
Output: 1

**Constraints:**

* n == customers.length == grumpy.length
* 1 <= minutes <= n <= 2 \* 104
* 0 <= customers[i] <= 1000
* grumpy[i] is either 0 or 1.