Number of Laser Beams in a Bank

Anti-theft security devices are activated inside a bank. You are given a **0-indexed** binary string array `bank` representing the floor plan of the bank, which is an `m x n` 2D matrix. `bank[i]` represents the i^{th} row, consisting of `'0'`s and `'1'`s. `'0'` means the cell is empty, while `'1'` means the cell has a security device.

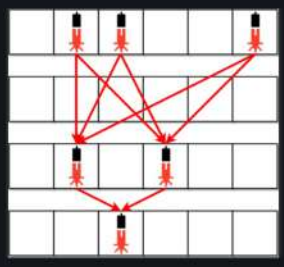
There is **one** laser beam between any **two** security devices **if both** conditions are met:

- The two devices are located on two **different rows**: r_1 and r_2, where $r_1 < r_2$.
- For **each** row i where $r_1 < i < r_2$, there are **no security devices** in the i^{th} row.

Laser beams are independent, i.e., one beam does not interfere nor join with another.

Return *the total number of laser beams in the bank.*

```
[ "0 1 1 0 0 1",
  "0 0 0 0 0 0",
  "0 1 0 1 0 0",
  "0 0 1 0 0 0" ]
```



1 denotes security device 0 means no device

APPROACH:

→ Count num of devices in each row

→ Multiply num of devices of first row with next and add to result

Edge-Case:

What if there is no device in any row??

Just skip that row and go to next row

DRY RUN:

```
[ "0 1 1 0 0 1",    CurrRowDevices = 3;  (prevRowDevices = 0)
  "0 0 0 0 0 0",    There is no device so Curr will be 3
  "0 1 0 1 0 0",    CurrRowDevices = 2; (prevRowDevices = 3) ans += curr*prev = (6)
  "0 0 1 0 0 0" ]   currRowDevices = 1; (prevRowDevices = 2) ans += (curr * prev) = 6+(1*2) = 8
```

Every Time we move to next row which has some devices then we will update prevRowDevices count!!!

```cpp
int numberOfBeams(vector<string>& bank) {
    int n = bank.size();
    int prevRowDevices = 0;
    int ans = 0;

    for(int i = 0; i < n; i++){
        int currRowDevices = 0;

        for(char ch: bank[i]){
            if(ch == '1') currRowDevices++;
        }

        ans += prevRowDevices * currRowDevices;
        if(currRowDevices != 0){
            prevRowDevices = currRowDevices;
        }
    }
    return ans;
}
```


Daily Coding Challenge Completed!
Completion Streak: 107 Days
Consistency is key, see you tomorrow!