NAME: ABHAS NATH ROLL_NO.:LCI2024006
BRANCH:CSAI

TASK-7

QUESTION:
MAKE SUBARRAY

Make Subarray

There is a canvas with N total cells, and each cell can be painted white or black. Currently, the state of the board can be represented by a binary string S, where $S_i = 1$ if and only if the i-th call in the canvas is painted black, and $S_i = 0$ otherwise.

You want the black cells to form a contiguous subarray, i.e. if cell x and cell y are coloured black, then all cells z between the 2 must also be black. Note that if there are no black cells, this condition is satisfied.

You can **only** change the colour of a cell from white to black (**but not the other way around**). Find the minimum number of changes needed.

Input Format

- The first line of input will contain a single integer T, denoting the number of test cases.
- Each test case consists of multiple lines of input.
  - The first line of each test case contains N - the number of cells.
  - The second line contains S - a binary string of size N.

Output Format

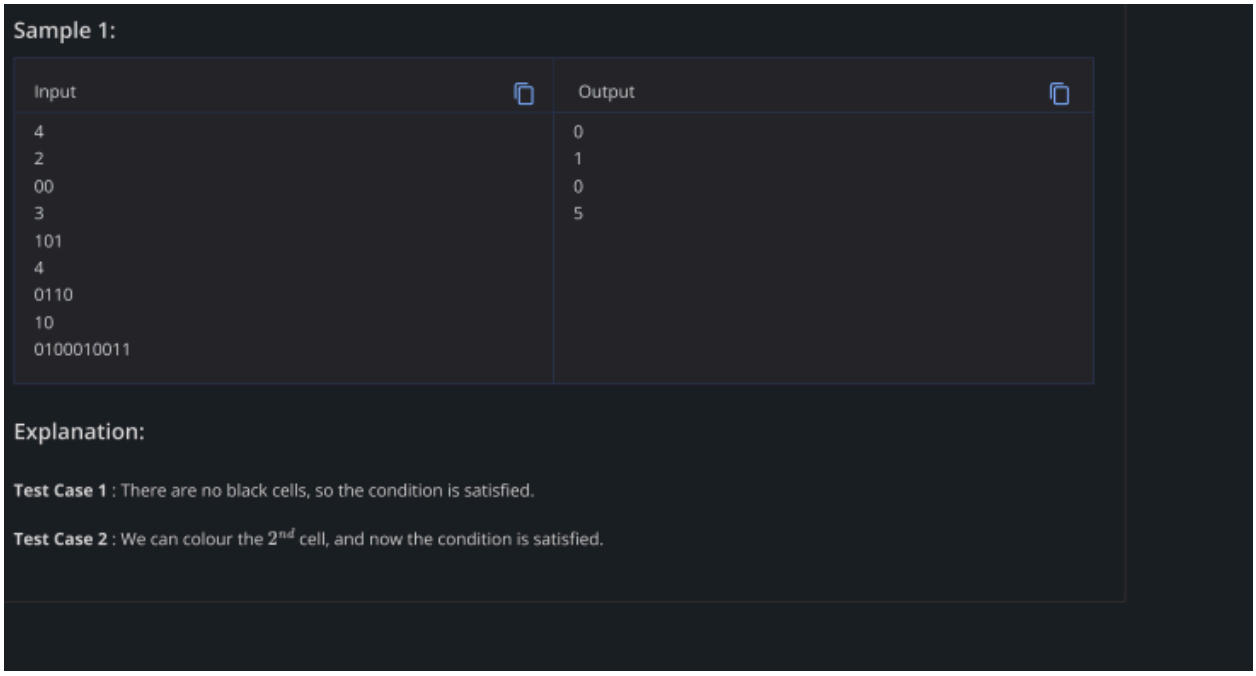
For each test case, output on a new line the minimum number of cells that need to be changed from white to black, such that the black cells form a contiguous subarray.

Constraints

- $1 \leq T \leq 100$
- $1 \leq N \leq 100$
- $S_i \in \{0, 1\}$

Sample 1:

SAMPLE TEST CASES:



Sample 1:

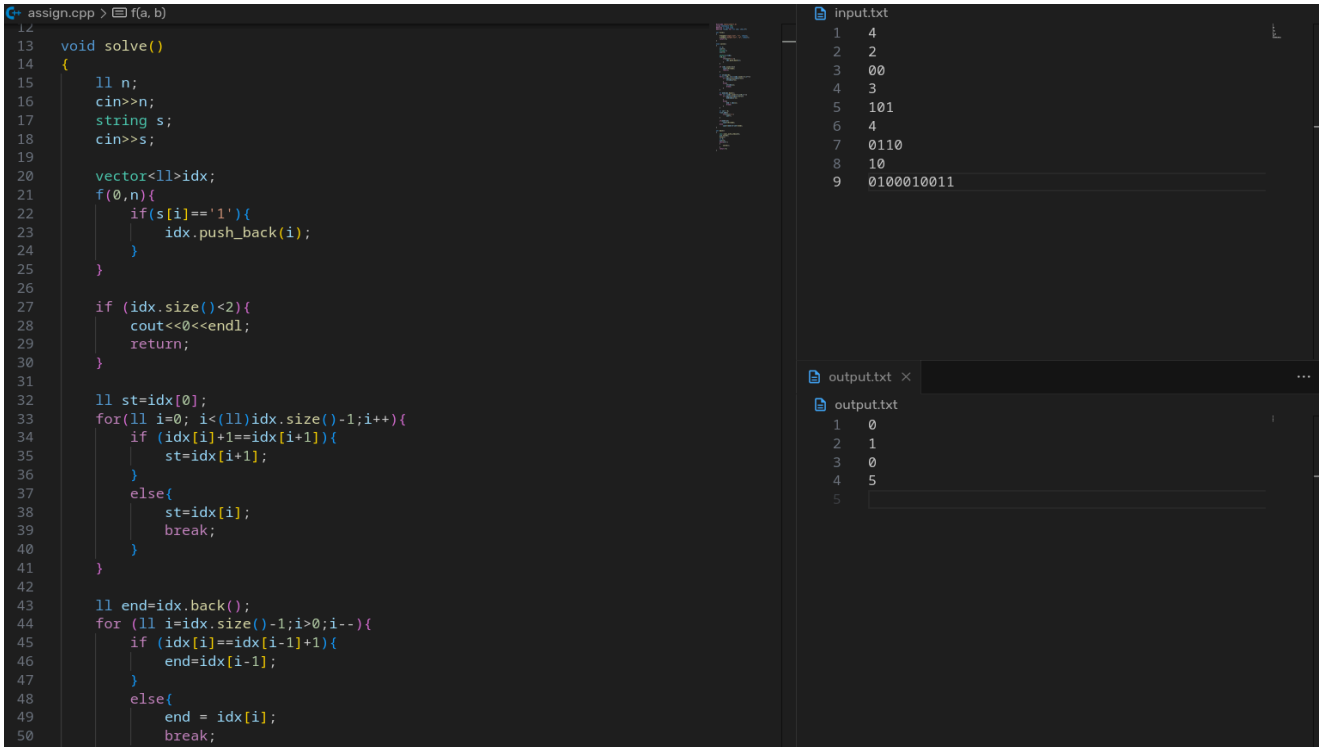
Input	Output
4 2 00 3 101 4 0110 10 0100010011	0 1 0 5

Explanation:

Test Case 1 : There are no black cells, so the condition is satisfied.

Test Case 2 : We can colour the 2^{nd} cell, and now the condition is satisfied.

CODE & OUTPUT:



assign.cpp > f(a, b)

```cpp
void solve()
{
    ll n;
    cin>>n;
    string s;
    cin>>s;

    vector<ll>idx;
    f(0,n){
        if(s[i]=='1'){
            idx.push_back(i);
        }
    }

    if (idx.size()<2){
        cout<<0<<endl;
        return;
    }

    ll st=idx[0];
    for(ll i=0; i<(ll)idx.size()-1;i++){
        if (idx[i]+1==idx[i+1]){
            st=idx[i+1];
        }
        else{
            st=idx[i];
            break;
        }
    }

    ll end=idx.back();
    for (ll i=idx.size()-1;i>0;i--){
        if (idx[i]==idx[i-1]+1){
            end=idx[i-1];
        }
        else{
            end = idx[i];
            break;
```

input.txt

```
4
2
00
3
101
4
0110
10
0100010011
```

output.txt

output.txt

```
0
1
0
5
```

```
42
43      ll end=idx.back();
44      for (ll i=idx.size()-1;i>0;i--){
45          if (idx[i]==idx[i-1]+1){
46              end=idx[i-1];
47          }
48          else{
49              end = idx[i];
50              break;
51          }
52      }
53
54      ll cnt = 0;
55      f(st,end){
56          if(s[i]=='1')
57              cnt++;
58      }
59
60      if(end<=st)
61          cout<<0<<endl;
62      else
63          cout<<end-st-cnt<<endl;
64  }
65
66  int main()
67  {
68      ios::sync_with_stdio(0);
69      cin.tie(0);
70      file();
71      ll t;
72      cin>>t;
73      while(t--)
74      {
75          solve();
76      }
77      return 0;
```

```
4   3
5   101
6   4
7   0110
8   10
9   0100010011
```

output.txt

output.txt

```
1   0
2   1
3   0
4   5
5
```