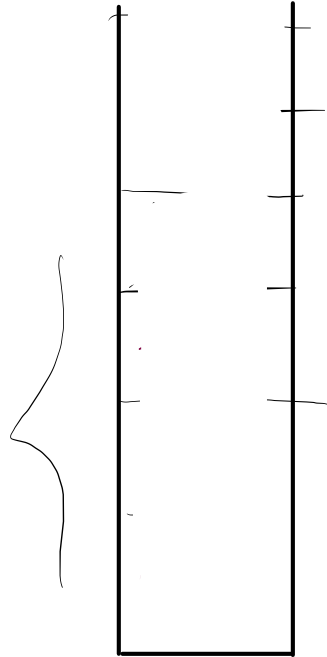


$n=5$

5
4
3
2
1

```
public static void func(int n){  
    1. if(n == 0)  
        return;  
  
    2. System.out.println(n);  
    3. func(n-1);  
}
```

5
4
3
2
1



'AmdPiotVa' \rightarrow aMDpIOTvA

'F' - 'A' = 'f' - 'a'
D - A = d - a

CH - 'A' = ch - 'a'

ASCII

L to U

U to L

CH = ch - 'a' + 'A'

ch = CH - 'A' + 'a'

```
11 public static void main(String[] args) {
12     String s = "aJiTyyYMn";
13     String ans = "";
14
15     for(int i = 0; i < s.length(); i++){
16         char c = s.charAt(i);
17
18         if(c >= 'A' && c <= 'Z'){
19             // upper to lower
20
21             ans += (char)(c - 'A' + 'a');
22         }
23         else{
24             //lower to upper
25
26             ans += (char)(c + 'A' - 'a');
27         }
28     }
29     System.out.println(ans);
30 }
31 }
```

1758. Minimum Changes To Make Alternating Binary String

Easy 577 21 Add to List Share

You are given a string `s` consisting only of the characters `'0'` and `'1'`. In one operation, you can change any `'0'` to `'1'` or vice versa.

The string is called alternating if no two adjacent characters are equal. For example, the string `"010"` is alternating, while the string `"0100"` is not.

Return the **minimum** number of operations needed to make `s` alternating.

0 1 0 1
1 0 1 0

→ 0 1 0 1 0 1 0
0 0 1 0 1 0
C++
min (2 ,

1 0 1 0 1 0 1
0 0 1 0 1 0
C++
4)

0 0 1 1 0 1 0