

Dry run of Plug-In problem

$s.length() = 14$.

		0 1 2 3 4 5 6 7 8 9 10 11 12 13 14		
	s =	a b a c a b a a b a c a b a a		
eq →				
i	s[i]	s[i+1]	same not	str s
0	a	b	X	a
1	b	a	X	ab
2	a	c	X	aba
3	c	a	X	abac
4	a	b	X	abaca
5	b	a	X	abacab
6	a	a	✓	abacab
8	b	a	X	abacabb
9	a	c	X	abacabba
10	c	a	X	abacabbac
11	a	b	X	abacabbaca
12	b	a	X	abacabbacab
13	a	a	✓	abacabbacab

New changed become true

New changed become true

New s = abacabbacab

0 1 2 3 4 5 6 7 8 9 10
a b a c a b b a c a b

Again continue same process.

loop will always work until we have satisfied the inside break condition.

Dry run for stack approach

eg: → meallazy

current char	ans before	Action	ans after
m	" "	push	m
e	m	push	me
a	me	push	mea
l	mea	push	meal
l	meal	pop	mea
a	mea	pop	me
z	me	push	mez
y	mez	push	mezy

```

for (char c : s) {
    if (!ans.empty() && ans.back() == c) {
        ans.pop_back();
    } else {
        ans.push_back(c);
    }
}

```

This means if ans is not empty & current character c is same as last character in ans, then remove last character from ans.

else,

If the stack is empty OR

current character is different, keep the character.

* Each character is pushed once, can be popped once.
 so Time complexity = $O(n)$.