

Week 6

## Char-Arrays & Strings - L2

Q. 4 Remove all adjacent duplicates in string.

i/p :- <sup>string</sup> 'abba' 'abbaca' o/p :- 'ca'

Eg :- 

a	b	b	a	c	a
0	1	2	3	4	5

  
↓  

a	b	b	a	c	a
---	---	---	---	---	---

 → 

c	a
---	---

 ⇒ final ans.

Q: 

a	z	x	x	z	y
---	---	---	---	---	---

 ⇒ 

a	z	z	y
---	---	---	---

 ⇒ 

a	y
---	---

 ⇒ final ans.

\* Approach → new string ← nayi string ke andar answer build karo.

str → 

a	z	x	x	z	y
---	---	---	---	---	---

create empty string.

string temp = "";

rightmost = a

a ≠ z → Insert z

z ≠ x → Insert x

x = x → x delete

z = rightmost = z → z, del

y ≠ a ? → Insert y.

different ⇒ push

same ⇒ pop

a	x	x			
---	---	---	--	--	--

rightmost = a

a ≠ z, ∴ Insert z

z ≠ x

∴ 

a	y
---	---

SC → O(n)

TC → O(n)

Q: Remove all Occurance of a Substring.

str → "abbcbadde**bab**bbbaae"

part → "cba"

o/p:- 'abbd**dd**bbbaae'

↑  
This can be called find substring, and we can use find function  
To remove, we can use erase.

∴ while (substring exist in badi string)

```
{  
    remove;  
}
```

HW → string ke andar ek pattern findout karna hai, toh uske  
liye kon kon se Algorithms exist karti hai? with TC.

→ (named algorithms)

\* Functions ko khud se likho → find and erase

```
while (s.find(part) != string::npos)
```

```
{
```

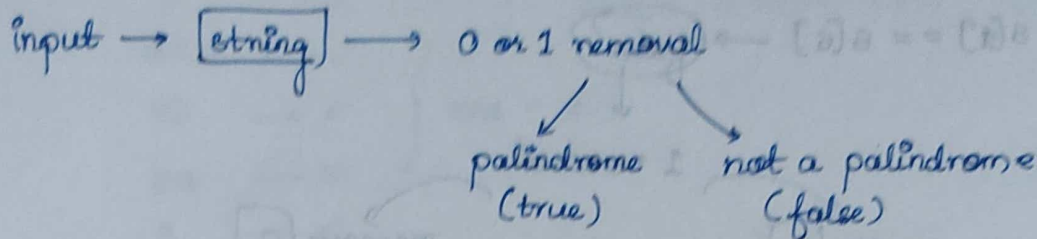
```
    s.erase(s.find(part), part.length());
```

```
}
```



## Q1 Valid Palindrome - II

→ Given a string s, return true, if the s can be palindrome after deleting the atmost one character from it.



Q: 'aba' → already palindrome → true

Q: 'ab/a' → 1 remove → Palindrome → true

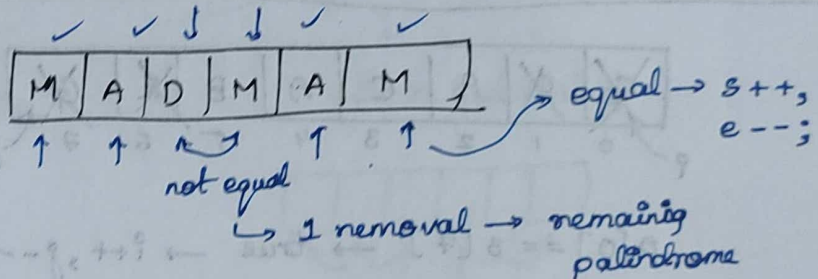
Q: 'abc' →

a → 'bc'  
 b → 'ac'  
 c → 'ab'  
 x → 'abc'

}

False, no palindrome

Eg:- 'MADAM'



Eg:-

M	A	D	A	R	R	A	C	A	M
---	---	---	---	---	---	---	---	---	---

$i=0$     1    2    3    4    5    6    7    8    9 =  $j$

$s[0] == s[9] \rightarrow \text{true} \rightarrow i++; j--;$

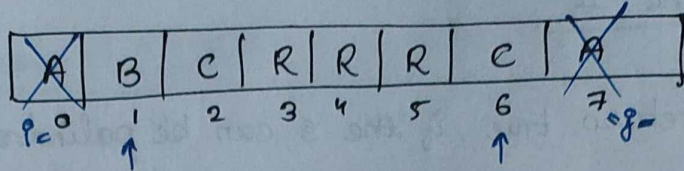
$s[1] == s[8] \rightarrow \text{true} \rightarrow i++; j--;$

$s[2] != s[7] \rightarrow \text{false} \rightarrow \text{only 1 allowed. (have 2 options.)}$

if D  
A R R A C  
X no

if C  
D A R R A  
X no

Exit → Overall ans = false;



$s[0] == s[7] \rightarrow \text{true} \rightarrow i++, j++$

$s[1] == s[6] \rightarrow \text{false}$

1 removal

remove [B]

C R R R C

(true) ✓

remove [C]

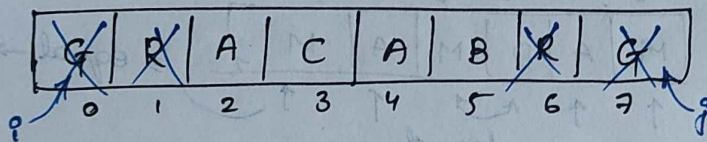
B C R R R

(false)

agar ek bhi true aayega, toh overall ans true  $\rightarrow$  OR

$\therefore$  inn dono answers ko 'OR' karo, then get ans.

0	0	$\rightarrow$	0
0	1	$\rightarrow$	1
1	0	$\rightarrow$	1
1	1	$\rightarrow$	1



$s[0] == s[7] \rightarrow \text{true} \rightarrow i++, j--$

$s[1] == s[6] \rightarrow \text{true} \rightarrow i++, j--$

$s[2] == s[5] \rightarrow \text{false}$

$i^{\text{th}}$  char

remaining  $\Rightarrow (i+1 \text{ se } j)$

C A B

false  $\rightarrow$  false || true

$j^{\text{th}}$  char

remaining  $\Rightarrow (i \text{ se } j-1)$

A C A

true

true



# Q 647. Palindromic Substrings

i/p → "aba"

count → kitna substring  
palindrome hai?

substring

a → ✓

aba → ✓

ab → ✗

ba → ✗

a → ✓

b → ✓

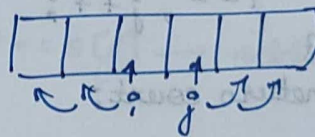
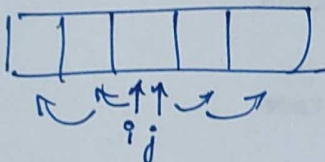
ans = 4.

Approach 1 :- Bachha jo sabhi substring nikalega, phir

Bruteforce se selection nahi hoga!!! Check  
palindrome.

Approach 2 :- 2 pointers, but different one.

length palindrome length  
Odd substring Even substring



if (match)  
→ count++

else  
→ no need to  
compare further

match

Eg :- odd  
a | b | c  
X → (i, j) → 1 2  
stop.

count = 0  
count++  
= 1

a = a  
i == j

Eg :- a | b | c  
i j

count = 1  
if (i == j)  
c++

count = 1 + 1 = 2

then is a == c → false  
ruk jao

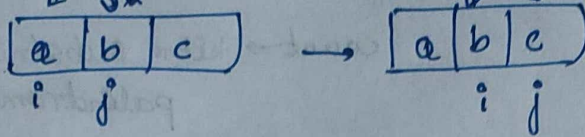
a | b | c  
0 1 2  
X  
i j

i == j  
→ c++

count = 3

Eg

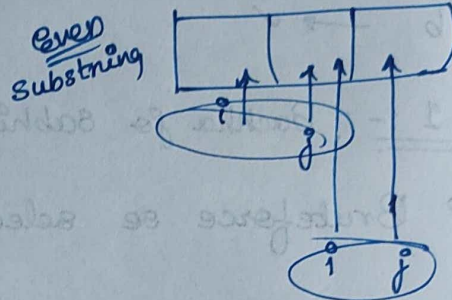
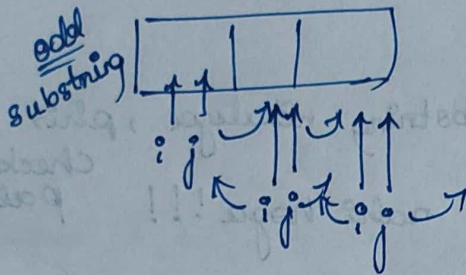
Even length



$\therefore \text{count} = 0$

Hence, total count =  $3 + 0 = \boxed{3}$

\* \* \*



Code :-

```
int expand (string s, int i, int j)
{
    int count = 0;
    while (i >= 0 && j < s.length() && s[i] == s[j])
    {
        count++;
        i--; j++;
    }
    return count;
}
```

```
int countSubstrings (string s)
{
    int totalCount = 0;
    for (int center = 0; center < s.length(); center++)
    {
        int oddAns = expand (s, center, center);
        int evenAns = expand (s, center, center+1);
        totalCount = totalCount + oddAns + evenAns;
    }
    return totalCount;
}
```



M	A	D	A	M
---	---	---	---	---

centre = 0

odd Ans = expand (s, 0, 0);

M	A	D	A	M
0	1	2	3	4

count = 0

s[i] == s[j] → (M) ⇒ odd count = 1, i=0, j=4

even Ans = expand (s, 0, 1);

count = 0,

s[0] == s[j] → X → even count = 0

∴ total = 0 + 1 + 0 → total = 1

centre = 1

odd Ans = expand (s, 1, 1) → add count = 1 (A)

~~even Ans~~

i = 0, j = 2

↓  
s[i] == s[j] → M ≠ D.

even Ans = expand (s, 1, 2)

→ X

total = 1 + 1 ⇒ total = 2

centre = 2

odd Ans = expand (s, 2, 2) → odd count = 1 (D)

now, i = 1, j = 3

↪ s[i] == s[j] → odd count = 2 (ADA)

now, i = 0, j = 4

↪ s[i] == s[j] → odd count = 3 (MADAM)

even Ans = expand (s, 2, 3)

↪ s[i] ≠ s[j] → even = 0.

∴ total = (2) + 3 → total = 5

centre = 3

centre = 3

~~odd Ans~~

M	A	D	A	M
0	1	2	3	4

odd Ans = expand (3, 3, 3)  $\rightarrow$  odd Ans = 1

even Ans = expand (3, 3, 4)  $\rightarrow$  0

$\therefore$  total = 5 + 1  $\Rightarrow$  6

centre = 4

odd Ans = expand (3, 4, 4)  $\rightarrow$  odd Ans = 1

even Ans = expand (3, 4, 5)  $\rightarrow$  0

$\therefore$  total = 6 + 1  $\rightarrow$  total = 7

— x — x —

total = 1 + 1 = 2

(a) 1 2 = 1 + 1 = 2

(a a) 2 = 2 + 1 = 3

new, 1 = 0, 0 = 1, even

(M A D A M) 5 = 5 + 1 = 6

even Ans = expand (3, 3, 2)

0 = new  $\leftarrow$  [0]  $\neq$  [0]

total = 3 + (2) = 5

0 = new