Day-5 [Strings]

→ NULL terminated character array.

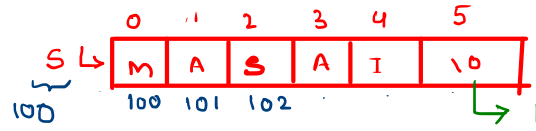
String s = "MASAI";

Size of char = 1B

8 bits

↳ $2^8 = 256$

(0 to 255)



① 0 | 1

print(s) ↳ 100 M A S A I

② ASCII → American Standard code for Information Interchange

↳ 65 → A 97 → a

90 → Z 122 → z

65 66 69

A B C D

26 ✓

Day-5 [Strings]

→ NULL terminated character array.

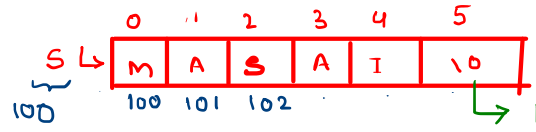
String s = "MASAI";

Size of char = 1B

8 bits

↳ $2^8 = 256$

(0 to 255)



① 0 | 1

print(s) ↳ 100 M A S A I

② ASCII → American Standard code for Information Interchange

↳ 65 → A 97 → a

90 → Z 122 → z

65 66 69

A B C D

26 ✓

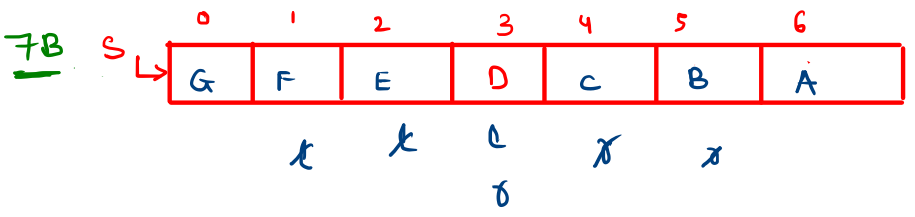
(2 ptrs + swap)

O(n) + O(n) ✓

function storev(string s)
{

✓1) Implement StringReverse function

String S = "ABCDEFG" i/p → O(n)



while (l < r)
{

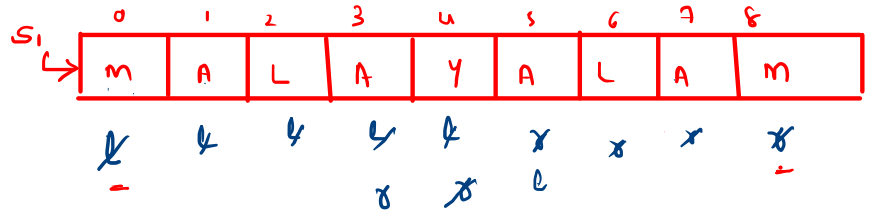
}

}

2-ptr :-

2)Check given string is palindrome or not

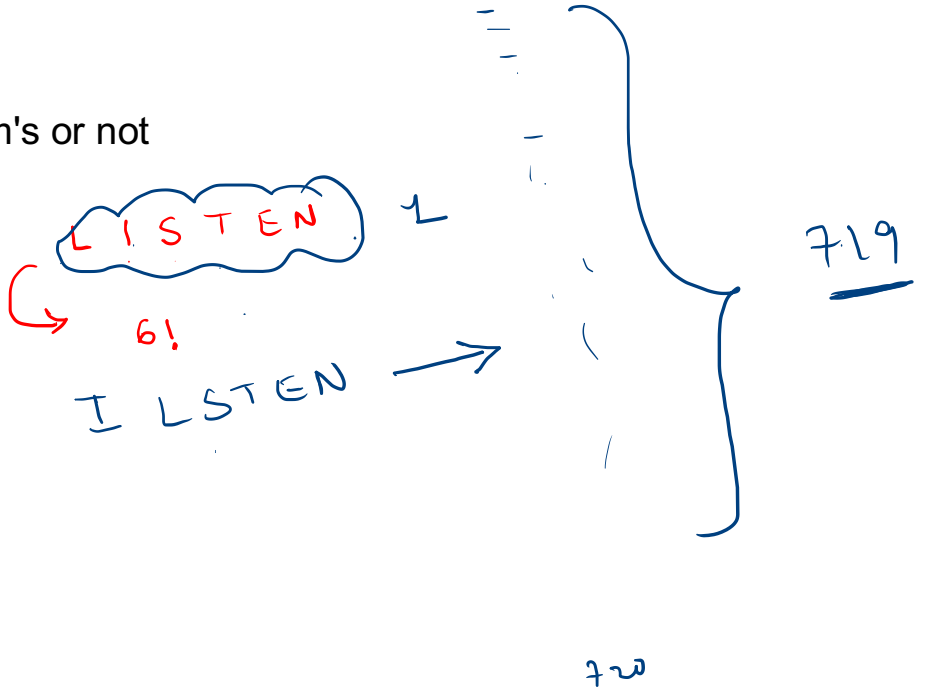
Eg. String s_1 = " MALAYALAM"




```
while(l<r)
{
    if(s1[l] != s1[r])
        return false
    l++
    r--
}
return true
```

s_1 → A B B A → even

3)Check given two strings are Anagram's or not



S_1 = "ABCDE" S_2 = "DCBAE"

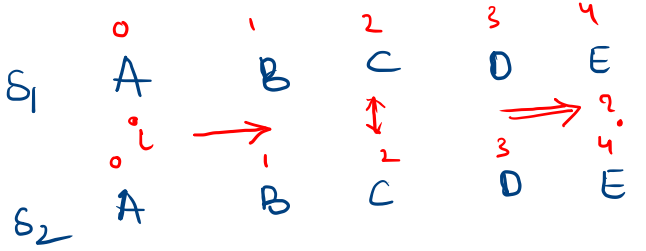
Approach-1 [Sorting]

(5) len(S_1) != len(S_2) ⇒ false

(nlogn + n) ⇒ O(nlogn)

+

O(n)

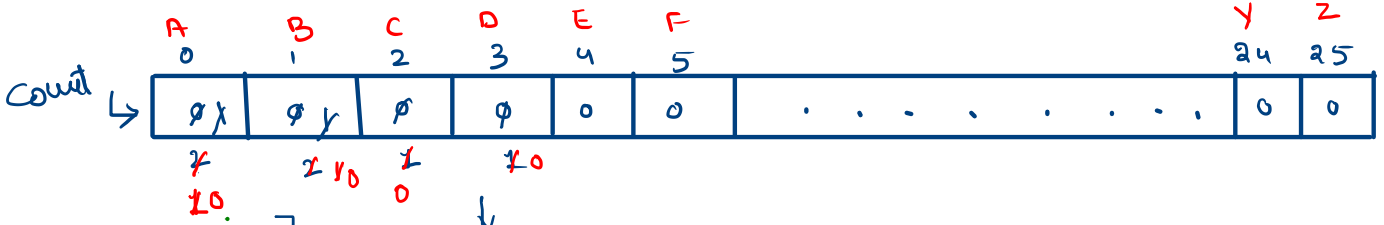



```
for(i=0; i<s1.length(); i++)
{
    if(a[i] != s2[i])
        return false
}
return true
```

O(n) + O(1)

$S_1, S_2 \Rightarrow$ A to Z $\Rightarrow$ Count[26]

Approach-2 [Count Array of size 256]



65 ← A → 0

66 ← B → 1

67 ← C → 2

68 ← D → 3

(C) 68 − 65 = 2

Base = 3

JS

JS

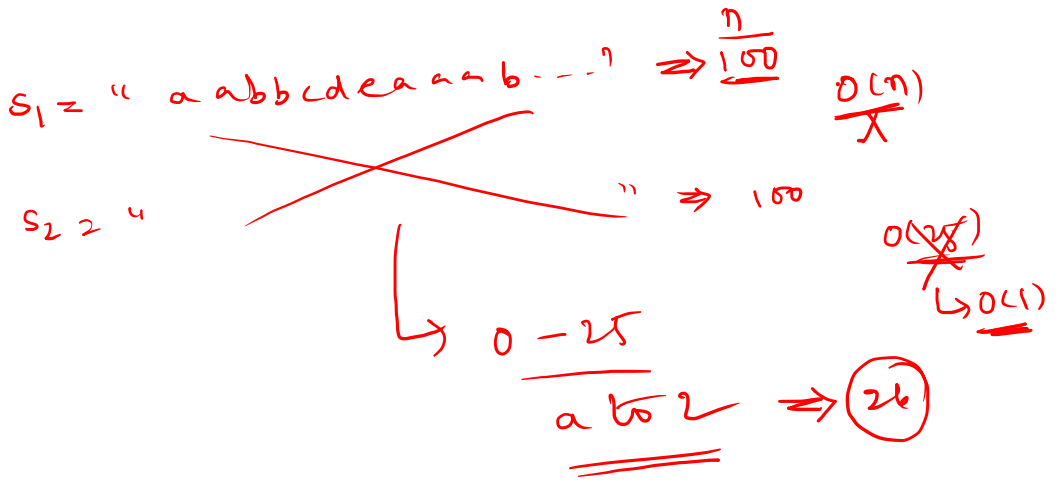
Google

loop1 ↳ S_1 = "A B A B C D" → A 2 C 1 B 2 D 1 <increment>

loop2 ↳ S_2 = "A C B B D A" → A 2 C 1 B 2 D 1 <decrement>

loop3 ⇒

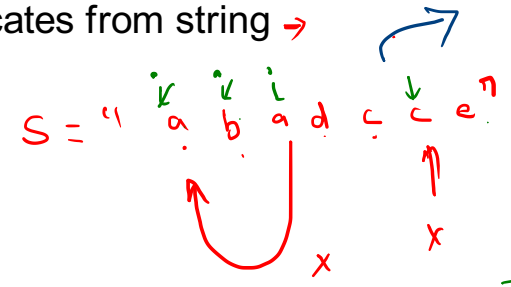
```
for(i=0; i<26; i++)
{   if(.count[i] != 0)
        → return false
}
return true ✓
```


s_1 = " a a b b c d e a a a b . . . "
n
100
O(n)
X
s_2 = "
" ⇒ 100
O(26)
O(1)
0 - 25
abc
⇒ 26

O(n) + O(n)

a d b c d e ⇒ a b c d e

Remove Duplicates from string →



⇒ a b d c e ⇒ abdce

JS → Set ⇒ collection of un-ordered distinct elements.

s = { a b d c e } ⇒ abdce

Set

Hashset Approach

Sorting Approach