

Day-5 Strings

→ NULL terminated character array.

String `s = "MASAI";`

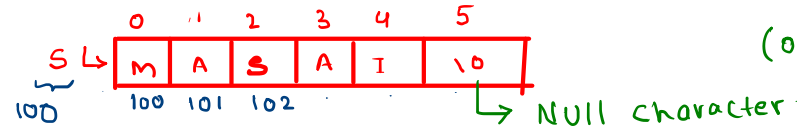
Size of char = 1B

8 bits

→ $\frac{8}{2} = 256$

(0 to 255)

①



0/1

point(s)

→ 100 M A S A I

②

ASCII → American Standard code for
Information Interchange.

65 66 67
A B C D

26 ✓

→ 65 → A 97 → a
90 → Z 122 → z

Day-5 Strings

→ NULL terminated character array.

String `s = "MASAI";`

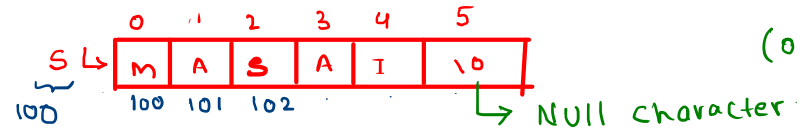
Size of char = 1B

8 bits

→ $\frac{8}{2} = 256$

(0 to 255)

①



0/1

point(s)

→ 100 M A S A I

②

ASCII → American Standard code for Information Interchange.

65 66 67
A B C D
26 ✓

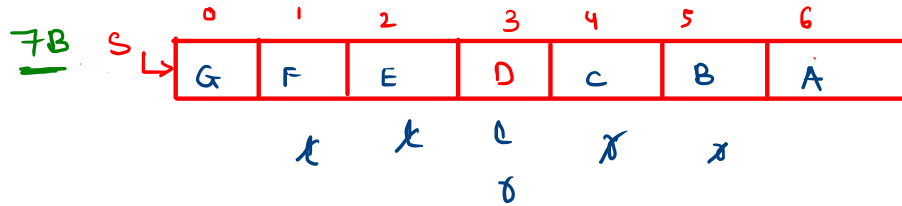
→ 65 → A 97 → a
90 → Z 122 → z

(2 ptr + swap)

$O(n)$
+
 $O(n)$ ✓

✓ 1) Implement StringReverse function

String s = "ABCDEF G" $\rightarrow O(n)$
i/p



while (l < r)

{

}

function storev(string s)
{

}

2-pts :-

2) Check given string is palindrome or not

```
while (l < r)
{
```

```
    if (s[l] != s[r])
```

```
        return false
```

```
    l++
```

```
    r--
```

```
}
```

```
return true ✓
```

Ex:-

String $s_1 = \text{"MALAYALAM"}$

	0	1	2	3	4	5	6	7	8
s_1	m	A	L	A	Y	A	L	A	m
	✓	✓	✓	✓	✓	✗	✗	✗	✗
	-			✗	✗	l			-

$s_1 \rightarrow$ A B B A \rightarrow even
✓ ✓ ✓ ✓
r l

* * *

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



LISTEN
6!
I LSTEN

719
72

Approach-1 [Sorting]

$S_1 = "A B C D E"$

$S_2 = "D C B A E"$

⑤ $\underline{\text{len}(S_1)} \neq \underline{\text{len}(S_2)} \Rightarrow \text{false}$

$(n \log n + n) \Rightarrow O(n \log n) \checkmark$

+
 $O(n) \checkmark$

	0	1	2	3	4
S_1	A	B	C	D	E
	0	1	2	3	4
S_2	A	B	C	D	E

Diagram illustrating the comparison of two strings S_1 and S_2 . S_1 is "A B C D E" and S_2 is "D C B A E". The indices 0 to 4 are shown above each character. A red arrow points from the 'A' at index 0 of S_1 to the 'A' at index 4 of S_2 . A red double-headed arrow is between the 'C' at index 2 of S_1 and the 'C' at index 3 of S_2 . A red arrow points from the 'D' at index 3 of S_1 to the 'D' at index 0 of S_2 . A red arrow points from the 'E' at index 4 of S_1 to the 'E' at index 4 of S_2 .

for ($i=0$; $i < S_1.length$; $i++$)

{ if ($S_1[i] \neq S_2[i]$)

return false

}

return true

$$O(n) + O(1)$$

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

Approach-2 [Count Array of size 256]

JS

Count \rightarrow

A	B	C	D	E	F	...	Y	Z
0	1	2	3	4	5		24	25
0	0	0	0	0	0	...	0	0

$$\begin{aligned} (64) - 65 &= 2 \\ 68 &\uparrow \\ \text{Base} &= 3 \end{aligned}$$

$$\begin{aligned} 65 &\leftarrow A \rightarrow 0 \\ 66 &\leftarrow B \rightarrow 1 \\ 67 &\leftarrow C \rightarrow 2 \\ 68 &\leftarrow D \rightarrow 3 \end{aligned}$$

JS

JS

Google

loop₁ $\rightarrow S_1 = "A B A B C D" \rightarrow$

A	2	C	1
B	2	D	1

<increment>

loop₂ $\rightarrow S_2 = "A C B B D A" \rightarrow$

A	2	C	1
B	2	D	1

<decrement>

loop₃ \Rightarrow

```

for (i = 0; i < 26; i++)
{
    if (Count[i] != 0)
        return false
}
return true

```

reproton true ✓

$$S_1 = "aabbcdcaab..." \Rightarrow \frac{n}{100} \quad \frac{O(n)}{\lambda}$$

$$S_2 = " \quad " \Rightarrow 100$$

$$\hookrightarrow \frac{0-25}{a \text{ to } 2}$$

$$\Rightarrow \textcircled{26}$$

$$\frac{O(25)}{\lambda} \hookrightarrow \underline{O(1)}$$

$O(n)$ + $O(n)$

Remove Duplicates from string →

$S = "a b a d c c e"$ $\Rightarrow a b d c e \Rightarrow \underline{a b d c e}$

a b c d e $\Rightarrow a b c d e$

JS → Set \Rightarrow collection of un-ordered distinct elements.

$s = \{ a b d c e \} \Rightarrow \underline{\underline{a b d c e}}$

Hashset Approach

Sorting Approach