



Strings

Today's checklist



1. Basics of Strings
2. String immutability in java
3. CharSequence Interface
4. Built-in stringBuilder functions
5. Problems based on strings
6. Sorting a string using in-built functions
7. Integer.parseInt function

What are strings and Why are they used?

```
int[] arr = { 10, 20, 30 };
```

```
char[] ch = { 'a', 'z', '#', '@', '8' };
```

```
char[] ch = new char[5];
```

Declaration of Strings and taking Input

String s;

String s = "Raghar";

next();

nextLine();

charAt() and length()

0 1 2 3 4 5
String s = "Raghav";

s[3] s.charAt(3)
✗ ✓

s.length s.length()
✗ ✓

Ques:

Q1 : Input a string and count all the vowels in the given string.

```
String s = "Raghav Garg is a teacher";
```

indexOf() and compareTo()

```
String s = "Raghav Garg";  
sout ( s.indexOf('v'));
```

'compareTo()' is used to compare 2 given strings lexicographically

abc def aab aet

aab, abc, aet, def

indexOf() and compareTo()

```
String a = "abc";
```

```
String b = "dbc";
```

```
System.out.println(a.compareTo(b));
```

→ $a.charAt(i) - b.charAt(i)$

a = "a b c"

b = "d b c"

'a' - 'd' is the ans ⇒ 97 - 100
↓ ↓
97 100

contains() and startsWith()

→ true/false

```
String s = "Raghar";
```

```
Sout(s.startsWith("Ra"));
```

toLowerCase() and concat()



```
String s = "Raghav Garg is 24 years Old";
```

substring(i) & substring(i,j)

String s = "abcd"; → this gets part of string from
i to j-1

Substrings of s are = a, ab, abc, abcd, b, bc, bcd,
c, cd, d, ""

sout (s.substring(2));
 ↓
 idx

Ques:

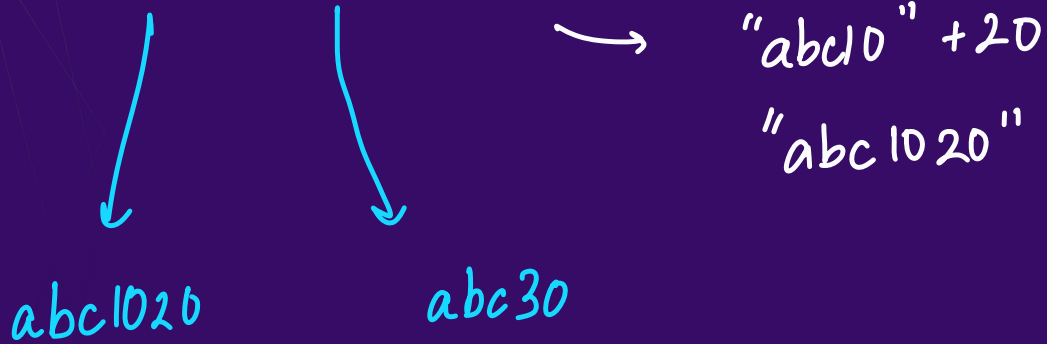
Q2 : Input a string and print all the substrings of that string.

String s = "abcd";

String + int / char / String



sout("abc"+10+20)



Sout (10 + 20 + "abc");

→ 30 + "abc"

→ "30abc"

Ques:

Q3 : Take integer input and convert it into a String.

Hint: $10 + \text{"abc"} \rightarrow \text{"10 abc"}$
 $10 + \text{"."} \rightarrow \text{"10."}$

Ans: $10 + \text{" "}$ or " " + 10

Ques:

Q4 : Return the total number of digits in a number without using any loop.

Hint : Try using inbuilt `Integer.toString()` function.

$n = 1256 \rightarrow \boxed{4} \text{ ans}$

String s = n + "";

sout(s.length());

Interning and **new** keyword

String s = "Raghar";

s = "Madhav";

String t = "Madhav";

String r = "Raghar"

r → "Raghar"

s → "Madhav"

t → "Madhav"

a → "Raghar"

So save space

String a = new String("Raghar");

String immutability in java



We cannot change individual characters in a string, we can but we will waste lot of time & space

```
String s = "hello";  
// heylo  
// 2nd index change to y  
s = s.substring(0,2) + 'y' + s.substring(3);
```

"hello"

"he" + "y"

"hey" + "lo"

s → "heylo"

Ques:

Q5 : Input a string and Update all the even positions in the string to character 'a'. Consider 0-based indexing.

String s = "Physics Wallah Skills";
 ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑

String t = "";

s = "Raghav";

str = "";



"" , "a" , "aa" , "aaa" , "aaah" , "aaaha"
"aaahar";

Performance of Strings

↓

Poor Performance because of immutability

equals() vs ==

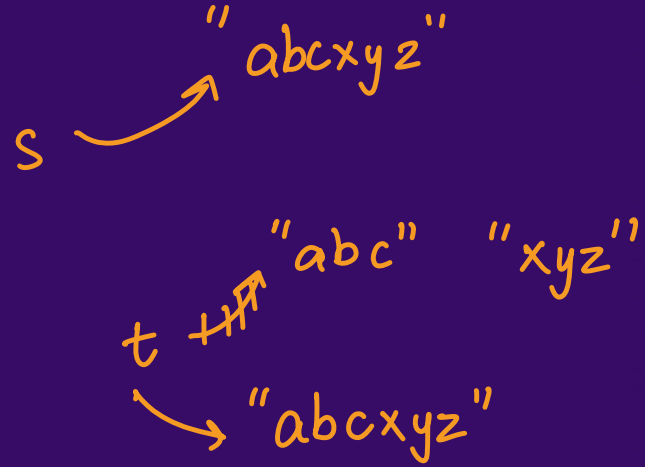
↓
equality operator

$a = b$
↓
assignment
a has now the
value of b

$a == b$
↓
comparison
true/false

equals() vs ==

```
String s = "abcxyz";  
String t = "abc";  
t = t + "xyz";  
System.out.println(s==t);
```



StringBuilder

```
String s = "Raghav";
```

```
StringBuilder sb = new StringBuilder("abc");
```

StringBuilder - Input

```
String s = sc.nextLine();
```

```
StringBuilder sb = new StringBuilder(sc.nextLine());
```


setCharAt()

```
StringBuilder sb = new StringBuilder("abcd");  
sb.setCharAt(1, 'g');
```

Ques:

Q6 : Input a string and toggle all the characters of it.
(Replace small case with capital case & vice versa)

String s = "RaGhaV";
↓
rAgHAr

'A' → 65

'a' = 97

'B' → 66

'b' = 98

⋮

⋮

'z' → 90

'Z' = 122

'a' - 'A' = 32

'n' - 'N' = 32

append()

+ → string

```
sb = "abc";
```

```
sb.append("xyz");
```

insert() and deleteCharAt()



inserts a char, int, string
.. at a particular index
↳ Shifts the rest of
elements.



you give idx, that particular
character is removed from
the String

reverse()



H.W. Take input a `StringBuilder` & reverse it without using builtin function.

Ques:

Q7 : Reverse each word in a given sentence.

(Ex : "i am raghav garg" -> i ma vahgar grag)

Sb = 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
I ma vahgar garg
 i j

reverse(sb, i, j-1);

Ques:

Q7 : Reverse each word in a given sentence.

(Ex : i am raghav garg -> i ma vahgar grag)

s cisyhp hallaw skills
i i

physics

Sorting a string

↓
String → Immutable
X

"raghav"
↓
aaghrv

Ques:

Q8 : Given two strings s and t, return true if t is an anagram of s, and false otherwise.

s = anagram	t = nagaram
↓ sort	↓ sort
aaagmnrr	aaagmnrr

Ques:

Q9 : Given a string consisting of lowercase English alphabets. Print the character that is occurring most number of times.

Hint: there are only 26 alphabets

Make a frequency array

`s = "abaaacrmrra";`

0	1	2	4								23	24	25
45	1	1				..	1	1	2				
a	b	c	d	e	f		m	n	r		x	y	z

Ques:

Q10 : Given two strings s and t , determine if they are isomorphic.

Example 1:

Input: $s = \text{"egg"}, t = \text{"add"}$

Output: true

Example 2:

Input: $s = \text{"foo"}, t = \text{"bar"}$

Output: false

Example 3:

Input: $s = \text{"paper"}, t = \text{"title"}$

Output: true

$e \rightarrow a$

$g \rightarrow d$

$f \rightarrow b$
 $o \rightarrow a$
 $o \rightarrow r$

$p \rightarrow t$

$a \rightarrow i$

$e \rightarrow l$

$r \rightarrow e$

True

$s = eat$
 $t = man$

$e \rightarrow m$
 $a \rightarrow a$
 $t \rightarrow n$

[Leetcode 205]

Ques:

Q10 : Given two strings s and t , determine if they are isomorphic.

$s = \text{race}$

↓ ↓ ↓ ↓

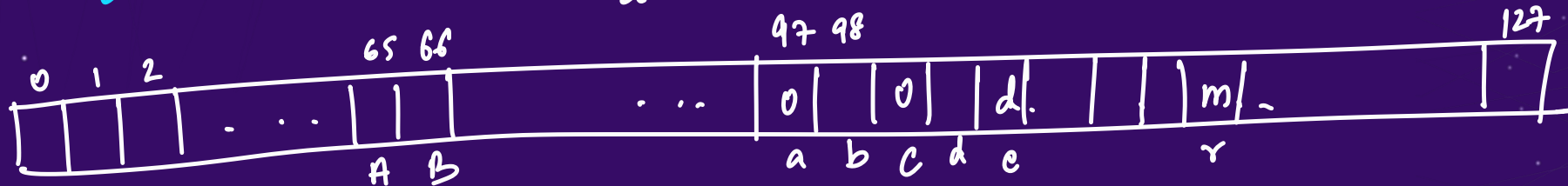
$t = \text{mood}$

$r \rightarrow m$

$a \rightarrow o$

$c \rightarrow$ ↗

$d \rightarrow e$



Ques:

String array
↑



Q11 : Given **n string** consisting of digits from 0 to 9. Return the string which has maximum value. (the no.s are true)

arr = { "999" , "1018" , "0078" } ;

000881



$s = "aaaabbbccdddefff"$
 $ans = "a4b3c2d2e3f3"$

S = 0 1 2 3 4 5 6 7 8 9 10 11 12 13
 a a a a b b c a a d d e f f

j
↑

ans = a4b2ca2d2ef2

↓

j-i

[Leetcode 443]

◀ **THANK YOU** ▶