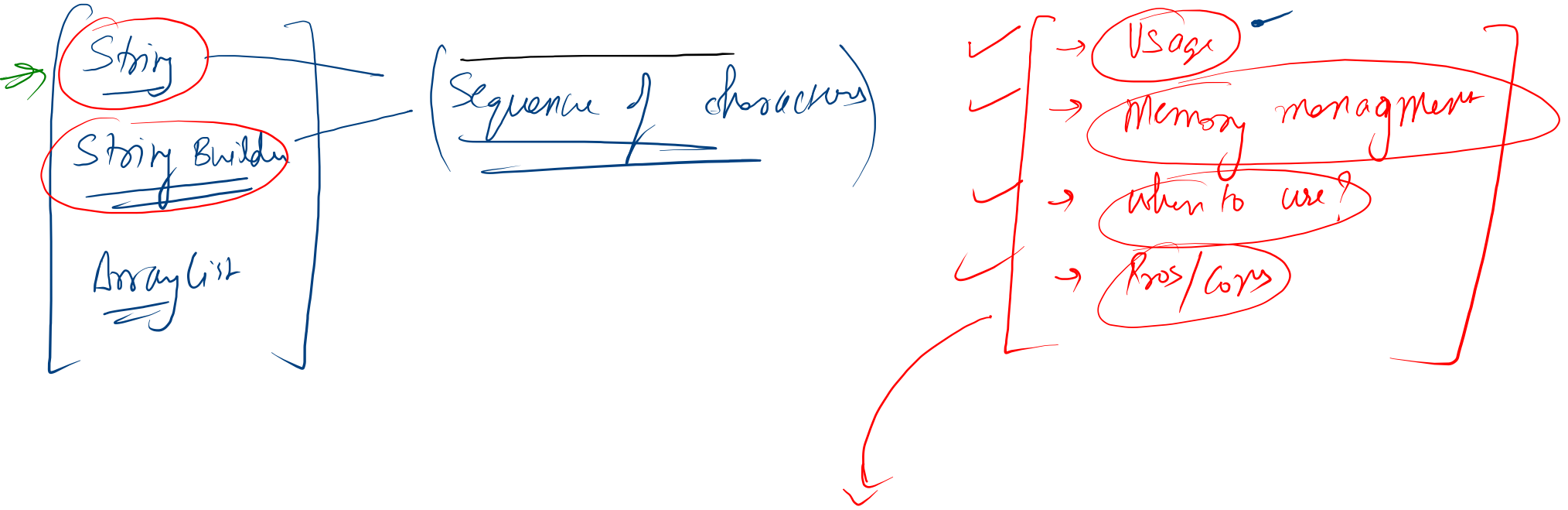
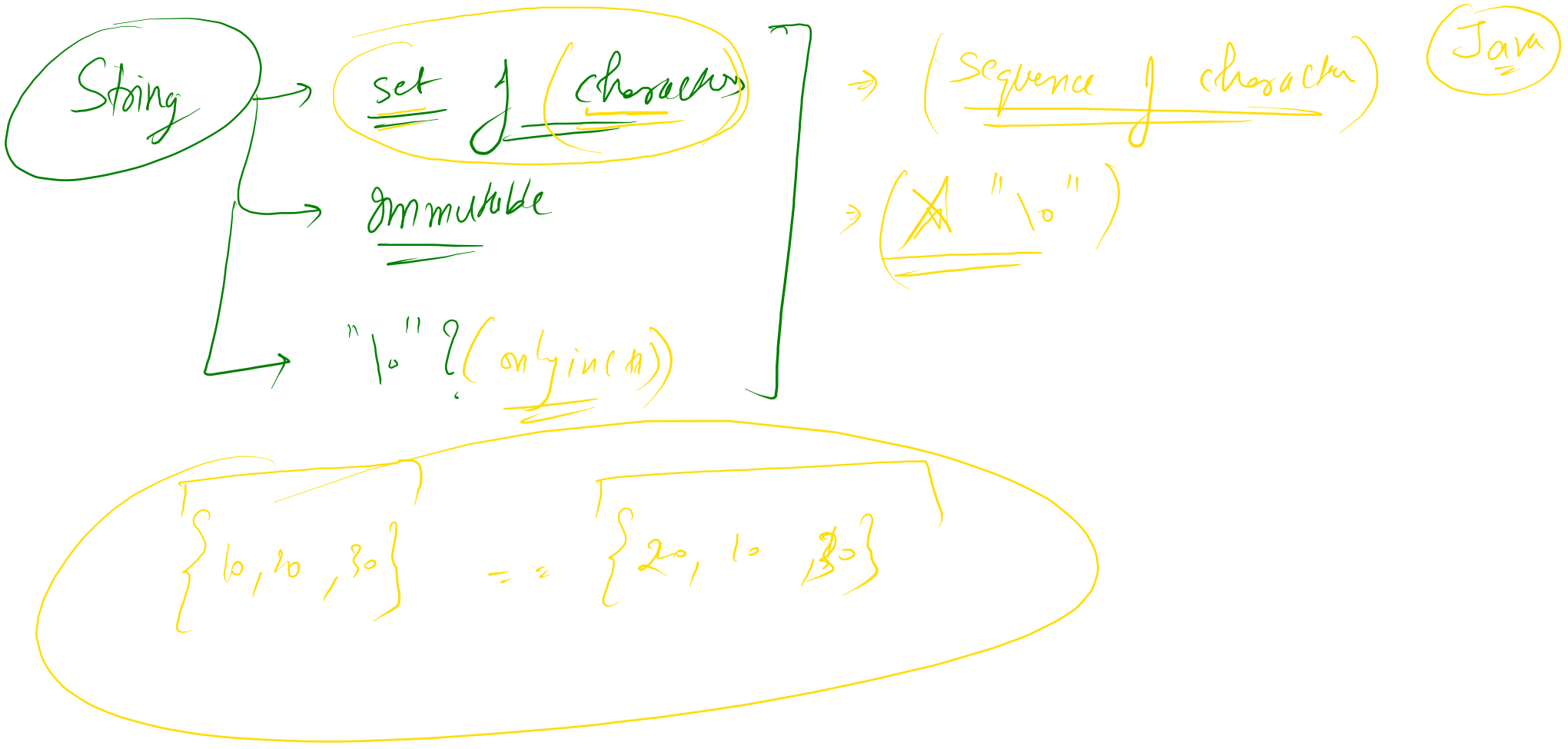

String
StringBuilder
ArrayList
Sequence of characters
Usage
Memory management
When to use?
Pros/Cons


String
set of characters
immutable
"\0" ? (original)
→ (sequence of characters)
→ ("\0")
Java
{10, 10, 30} == {20, 10, 30}

String

Usage

- `String s = "msg";`
- `String s = new String("msg");`

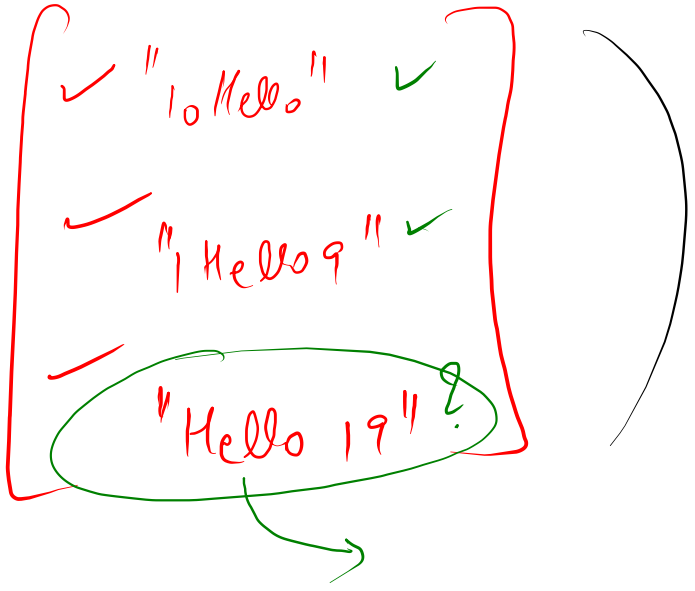
✓ [Golden Rule]

↳ any variable concatenated with String results in a String Type

(— + String → String)

18 + "hello" → "18 hello"

```java
// Golden Rule
System.out.println(1+9+"Hello");
System.out.println(1+"Hello"+9);
System.out.println("Hello"+1+9);
```



("substring")

s = "H E L L O"

0 1 2 3 4

substring(idx)

↳ get all chars from idx → len

s.substring(0) → "HELLO"
= = (1) → "ELLO"
= = (2) → "LLO"
= = (3) → "LO"
= = (4) → "O"
= = (5) → ""

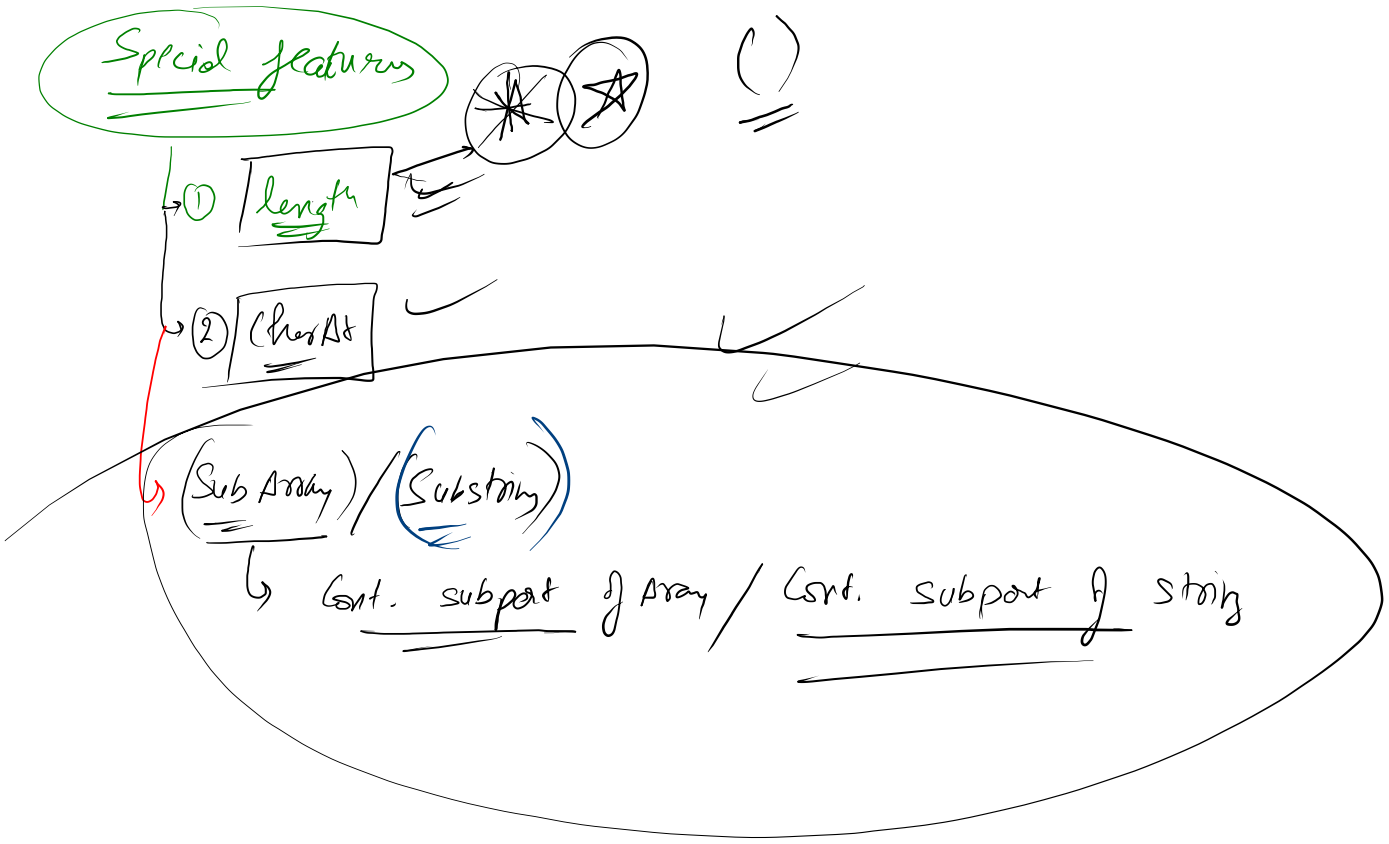
s.substring(0,0) → ""
= = (0,1) → "H"
= = (0,2) → "HE"
= = (0,3) → "HEL"
= = (0,4) → "HELL"
= = (0,5) → "HELLO"

substring(si, ei)

ei → exc

si → inc

s.substring(1,1) → ""


Special features
1 length
2 charAt
SubArray / Substring
Const. subpart of Array / Const. subpart of string

1) Substrings
2) Check palindrom

a	b	c	c
0	1	2	3

a
b
c
c c
c

(0,1) = "a"

(0,2) = "ab"

(0,3) = "abc"

(0,4) = "abcc"

(i,j)

(1,2) = "b"

(1,3) = "bc"

(1,4) = "bcc"

(i,j)

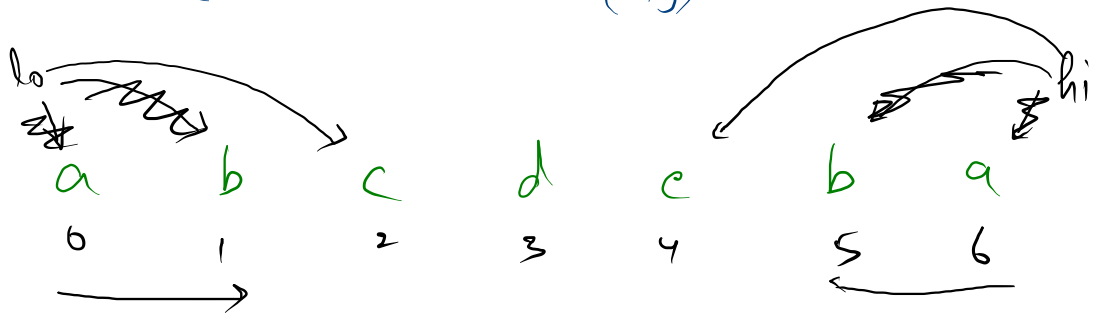
(2,3) = "c"

(2,4) = "cc"

(i,j)

(3,4) = "c"

(i,j)



```java
public static boolean isPalindrome(String s){
    int lo = 0 , hi = s.length()-1;
    while(lo < hi){
        if(s.charAt(lo) != s.charAt(hi)){
            return false; //  non-palindromic string
        }
        lo++;
        hi--;
    }

    return true; //  palindromic string
}
public static void printAllPalindromicSubstring(String str){
    for(int si = 0 ; si < str.length() ; si++){
        for(int ei = si+1 ; ei <= str.length() ; ei++){
            String ss = str.substring(si,ei);
            if(isPalindrome(ss)){
                System.out.println(ss);
            }
        }
    }
}
```

si	ei
0	1, 2, 3, 4
1	2, 3, 4
2	3, 4
3	4

str => "abcc"
0123

len => 4

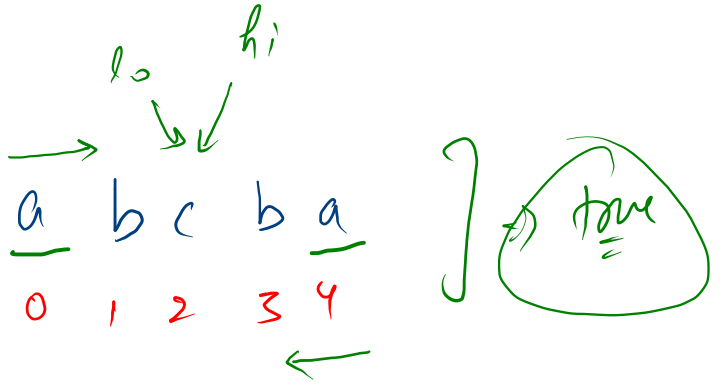
ss(0,1) => "a"
ss(0,2) => "ab"
ss(0,3) => "abc"
ss(0,4) => "abcc"

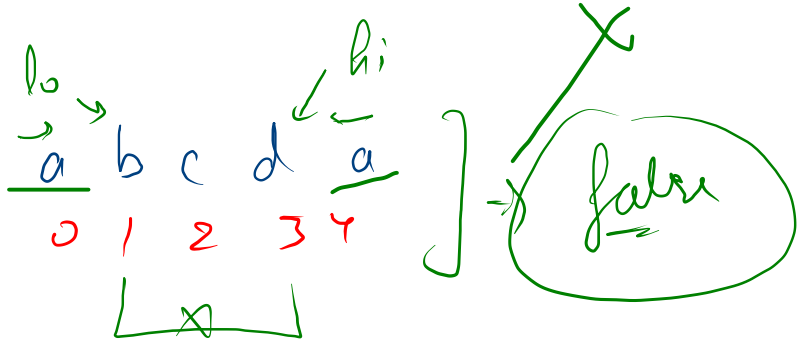
(1,2) => "b"
(1,3) => "bc"
(1,4) => "bcc"

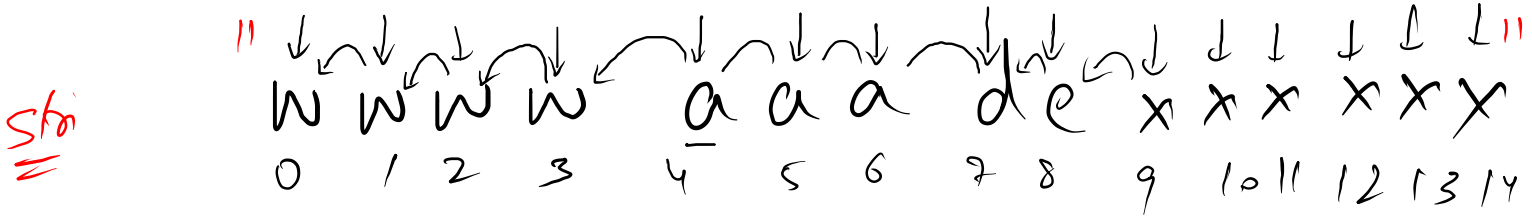
(2,3) = "c"
(2,4) = "cc"

(3,4) => "c"

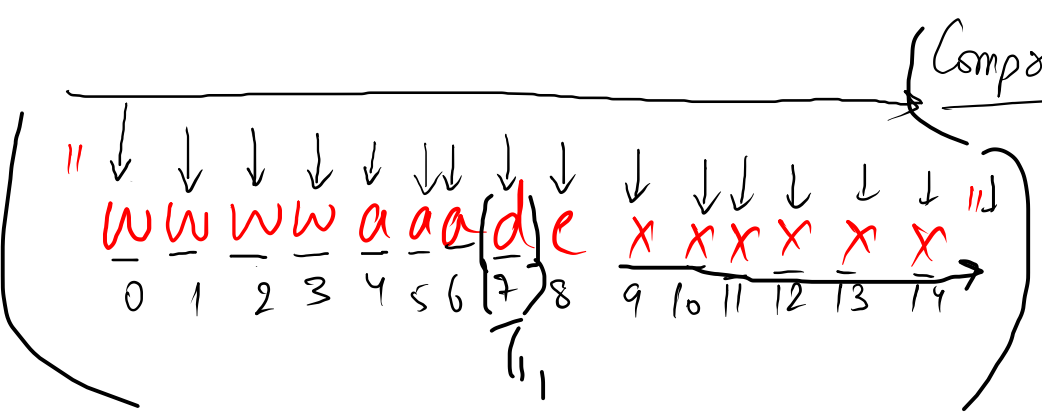
```java
public static boolean isPalindrome(String s){
    int lo = 0 , hi = s.length()-1;
    while(lo < hi){
        if(s.charAt(lo) != s.charAt(hi)){
            return false; //  non-palindromic string
        }
        lo++;
        hi--;
    }

    return true; //  palindromic string
}
public static void printAllPalindromicSubstring(String str){
    for(int si = 0 ; si < str.length() ; si++){
        for(int ei = si+1 ; ei <= str.length() ; ei++){
            String ss = str.substring(si,ei);
            if(isPalindrome(ss)){
                System.out.println(ss);
            }
        }
    }
}
```







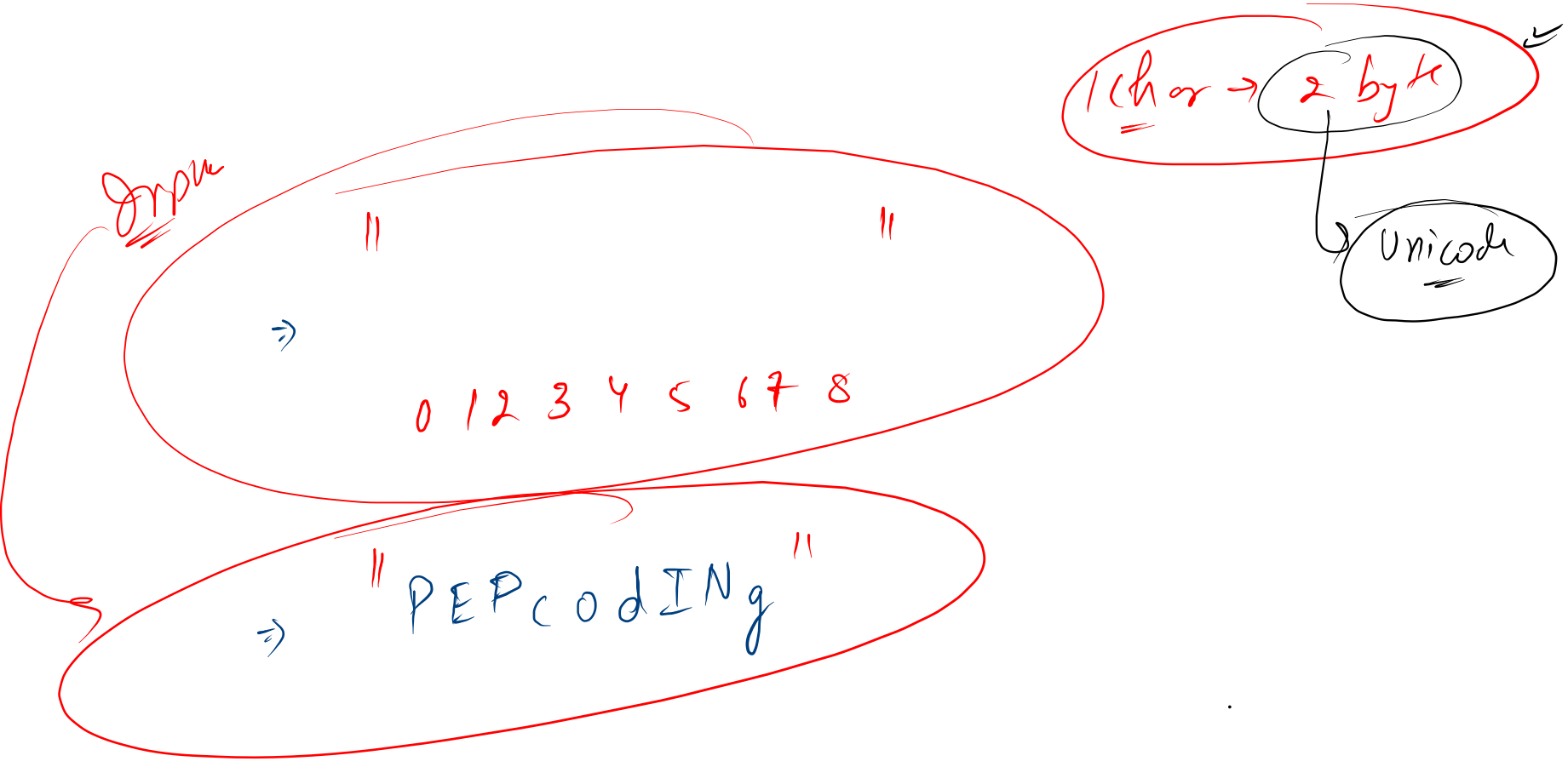
res = "wadex" → Compression



res = " w4a3dex6 "

Count = 1 2 3 4 1 2 3 1 2 3 4 1 6

```java
public static String compression1(String str){
    String res = "";
    // logic
    for(int idx = 0 ; idx < str.length() ; idx++){
        if(idx == 0){
            res = res + str.charAt(0);
        }else{
            char curr = str.charAt(idx);
            char prev = str.charAt(idx-1);

            if(curr != prev){
                res = res + curr;
            }
        }
    }
    return res;
}
```


1 char → 2 byte
Unicode
Input
" "
" "
0 1 2 3 4 5 6 7 8
" PEPcodING "

Dec	Char	Dec	Chr	Dec	Chr	Dec	Chr
0	NUL (null)	32	Space	64	@	96	`
1	SOH (start of heading)	33	!	65	A	97	a
2	STX (start of text)	34	"	66	B	98	b
3	ETX (end of text)	35	#	67	C	99	c
4	EOT (end of transmission)	36	$	68	D	100	d
5	ENQ (enquiry)	37	%	69	E	101	e
6	ACK (acknowledge)	38	&	70	F	102	f
7	BEL (bell)	39	'	71	G	103	g
8	BS (backspace)	40	(	72	H	104	h
9	TAB (horizontal tab)	41	)	73	I	105	i
10	LF (NL line feed, new line)	42	*	74	J	106	j
11	VT (vertical tab)	43	+	75	K	107	k
12	FF (NP form feed, new page)	44	,	76	L	108	l
13	CR (carriage return)	45	-	77	M	109	m
14	SO (shift out)	46	.	78	N	110	n
15	SI (shift in)	47	/	79	O	111	o
16	DLE (data link escape)	48	0	80	P	112	p
17	DC1 (device control 1)	49	1	81	Q	113	q
18	DC2 (device control 2)	50	2	82	R	114	r
19	DC3 (device control 3)	51	3	83	S	115	s
20	DC4 (device control 4)	52	4	84	T	116	t
21	NAK (negative acknowledge)	53	5	85	U	117	u
22	SYN (synchronous idle)	54	6	86	V	118	v
23	ETB (end of trans. block)	55	7	87	W	119	w
24	CAN (cancel)	56	8	88	X	120	x
25	EM (end of medium)	57	9	89	Y	121	y
26	SUB (substitute)	58	:	90	Z	122	z
27	ESC (escape)	59	;	91	[	123	{
28	FS (file separator)	60	<	92	\	124	|
29	GS (group separator)	61	=	93	]	125	}
30	RS (record separator)	62	>	94	^	126	~
31	US (unit separator)	63	?	95	_	127	DEL

```java
for(int i = 0 ; i <= 9 ; i++){
    int enc = i+'0';
    System.out.println(i+" --> "+enc);
}
```

'A' - UCL = 'a' - LCL

'A' - 'M' = 'a' - 'm'

(1) UCL ⟶ LCL UCL ⇒ 'D'

LCL ⇒ 'a' + (UCL - 'A') / 3 ⇒ 100 (char) ⇒ 'd'

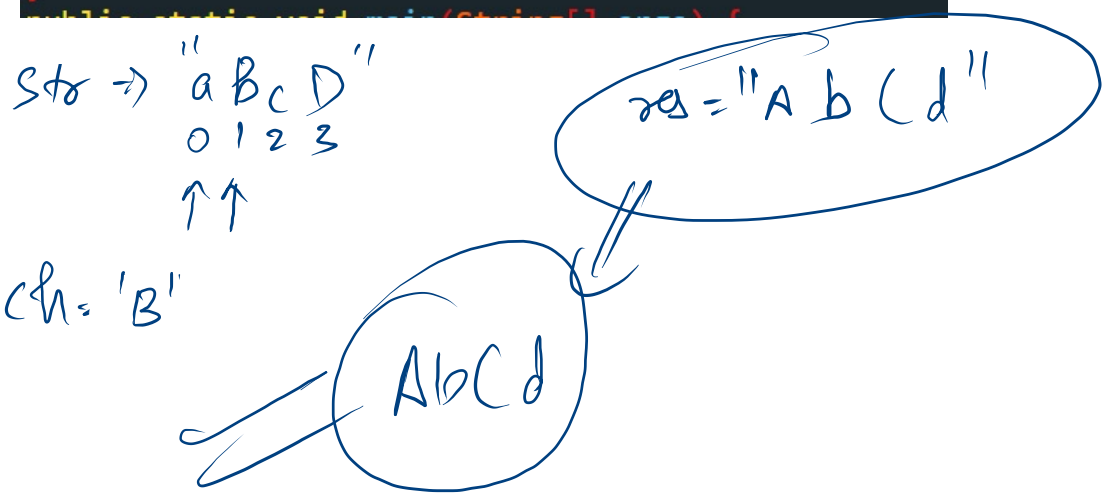
97 + 3

(2) LCL ⟶ UCL LCL ⇒ 'q'

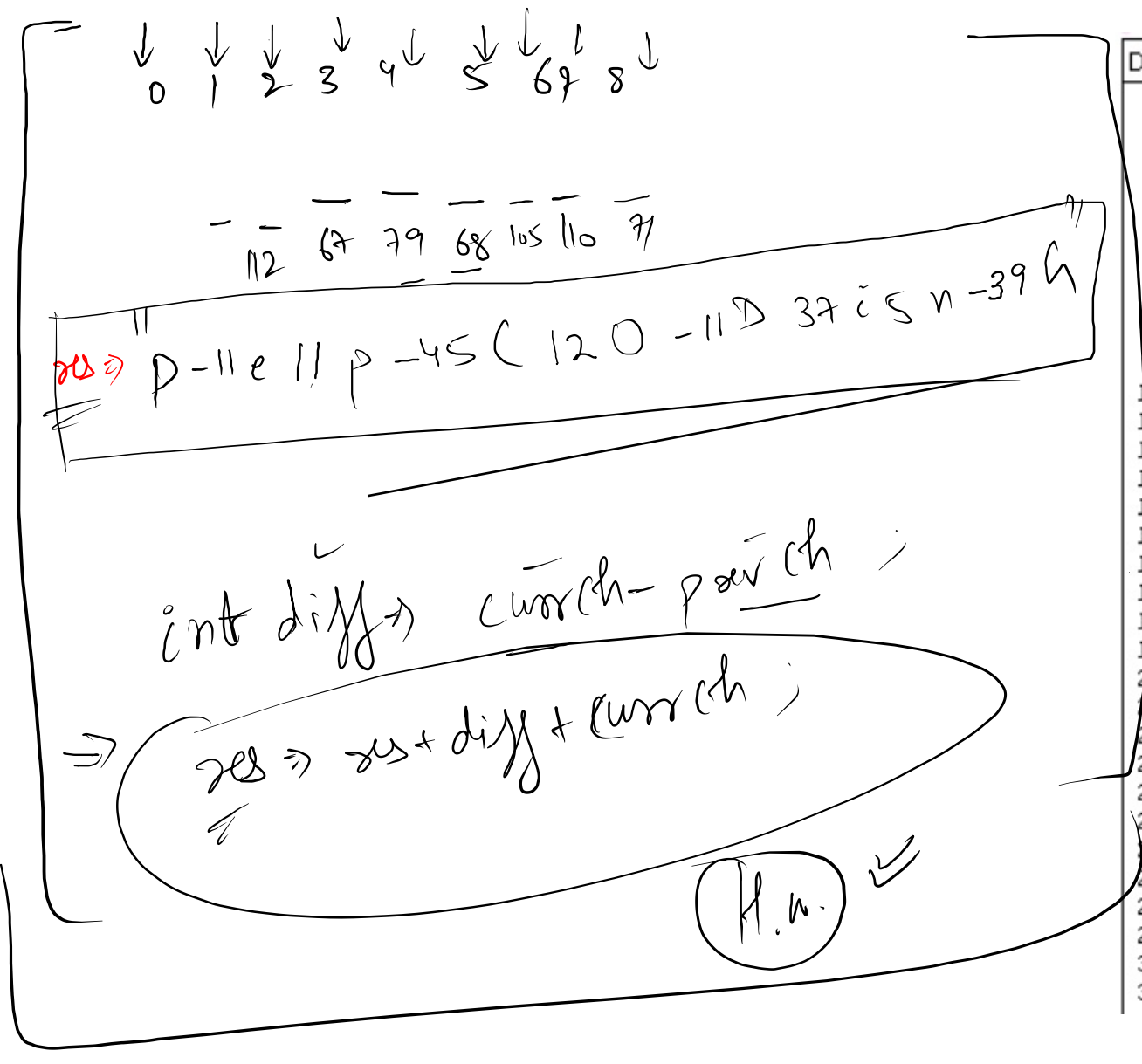
UCL ⇒ 'A' + (LCL - 'a') / 16 ⇒ 81 (char) ⇒ 'Q'

65 + 16

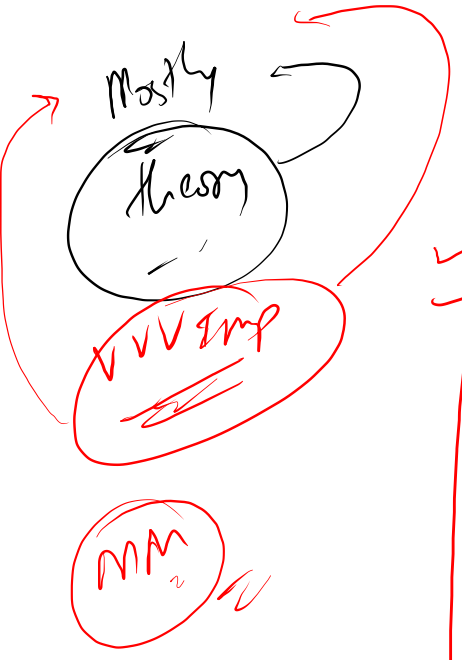
```java
public static String toggleCase(String str){
    String res = "";

    for(int idx = 0 ; idx < str.length() ; idx++){
        char ch = str.charAt(idx);

        if(ch >= 'A' && ch <= 'Z'){
            // upper case -> lower case
            char lcl = (char)('a'+(ch-'A'));
            res = res + lcl;
        }else if(ch >= 'a' && ch <= 'z'){
            // lower case -> upper case
            char ucl = (char)('A' + (ch - 'a'));
            res = res + ucl;
        }
    }

    return res;
}
```

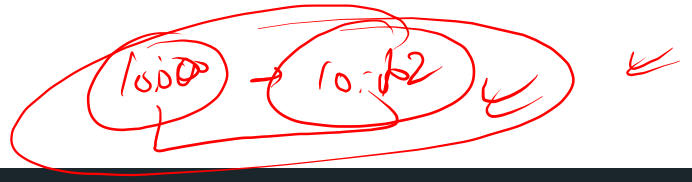


Dec	Char	Dec	Chr	Dec	Chr	Dec	Chr
0	NUL (null)	32	Space	64	@	96	`
1	SOH (start of heading)	33	!	65	A	97	a
2	STX (start of text)	34	"	66	B	98	b
3	ETX (end of text)	35	#	67	C	99	c
4	EOT (end of transmission)	36	$	68	D	100	d
5	ENQ (enquiry)	37	%	69	E	101	e
6	ACK (acknowledge)	38	&	70	F	102	f
7	BEL (bell)	39	'	71	G	103	g
8	BS (backspace)	40	(	72	H	104	h
9	TAB (horizontal tab)	41	)	73	I	105	i
10	LF (NL line feed, new line)	42	*	74	J	106	j
11	VT (vertical tab)	43	+	75	K	107	k
12	FF (NP form feed, new page)	44	,	76	L	108	l
13	CR (carriage return)	45	-	77	M	109	m
14	SO (shift out)	46	.	78	N	110	n
15	SI (shift in)	47	/	79	O	111	o
16	DLE (data link escape)	48	0	80	P	112	p
17	DC1 (device control 1)	49	1	81	Q	113	q
18	DC2 (device control 2)	50	2	82	R	114	r
19	DC3 (device control 3)	51	3	83	S	115	s
20	DC4 (device control 4)	52	4	84	T	116	t
21	NAK (negative acknowledge)	53	5	85	U	117	u
22	SYN (synchronous idle)	54	6	86	V	118	v
23	ETB (end of trans. block)	55	7	87	W	119	w
24	CAN (cancel)	56	8	88	X	120	x
25	EM (end of medium)	57	9	89	Y	121	y
26	SUB (substitute)	58	:	90	Z	122	z
27	ESC (escape)	59	;	91	[	123	{
28	FS (file separator)	60	<	92	\	124	|
29	GS (group separator)	61	=	93	]	125	}
30	RS (record separator)	62	>	94	^	126	~
31	US (unit separator)	63	?	95	_	127	DEL



Dec	Char		Dec	Chr	Dec	Chr	Dec	Chr
0	NUL	(null)	32	Space	64	@	96	`
1	SOH	(start of heading)	33	!	65	A	97	a
2	STX	(start of text)	34	"	66	B	98	b
3	ETX	(end of text)	35	#	67	C	99	c
4	EOT	(end of transmission)	36	$	68	D	100	d
5	ENQ	(enquiry)	37	%	69	E	101	e
6	ACK	(acknowledge)	38	&	70	F	102	f
7	BEL	(bell)	39	'	71	G	103	g
8	BS	(backspace)	40	(	72	H	104	h
9	TAB	(horizontal tab)	41	)	73	I	105	i
10	LF	(NL line feed, new line)	42	*	74	J	106	j
11	VT	(vertical tab)	43	+	75	K	107	k
12	FF	(NP form feed, new page)	44	,	76	L	108	l
13	CR	(carriage return)	45	-	77	M	109	m
14	SO	(shift out)	46	.	78	N	110	n
15	SI	(shift in)	47	/	79	O	111	o
16	DLE	(data link escape)	48	0	80	P	112	p
17	DC1	(device control 1)	49	1	81	Q	113	q
18	DC2	(device control 2)	50	2	82	R	114	r
19	DC3	(device control 3)	51	3	83	S	115	s
20	DC4	(device control 4)	52	4	84	T	116	t
21	NAK	(negative acknowledge)	53	5	85	U	117	u
22	SYN	(synchronous idle)	54	6	86	V	118	v
23	ETB	(end of trans. block)	55	7	87	W	119	w
24	CAN	(cancel)	56	8	88	X	120	x
25	EM	(end of medium)	57	9	89	Y	121	y
26	SUB	(substitute)	58	:	90	Z	122	z
27	ESC	(escape)	59	;	91	[	123	{
28	FS	(file separator)	60	<	92	\	124	|
29	GS	(group separator)	61	=	93	]	125	}
30	RS	(record separator)	62	>	94	^	126	~
31	US	(unit separator)	63	?	95	_	127	DEL





Introduction To Strings, Stringbuilders And Arraylists		10	✓Auth	0	✓Public	✓Sol	1		
Print All Palindromic Substrings	Easy	10	✓Auth	0	✓Public	✓Sol	2		
String Compression	Easy	10	✓Auth	0	✓Public	✓Sol	3		
String - Interning And Immutability		10	✓Auth	0	Public	✓Sol	4		
Stringbuilder - Usage And Performance		10	✓Auth	0	Public	✓Sol	5		
Toggle Case	Easy	10	✓Auth	0	✓Public	✓Sol	6		
String With Difference Of Every Two Consecutive Charac...	Easy	10	✓Auth	0	✓Public	✓Sol	7		
Print All Permutations Of A String Iteratively	Easy	10	✓Auth	0	Public	✓Sol	8		
Introduction To Arraylists - Usage And Demo		10	✓Auth	0	Public	✓Sol	9		
Remove Primes	Easy	10	✓Auth	0	Public	✓Sol	10		