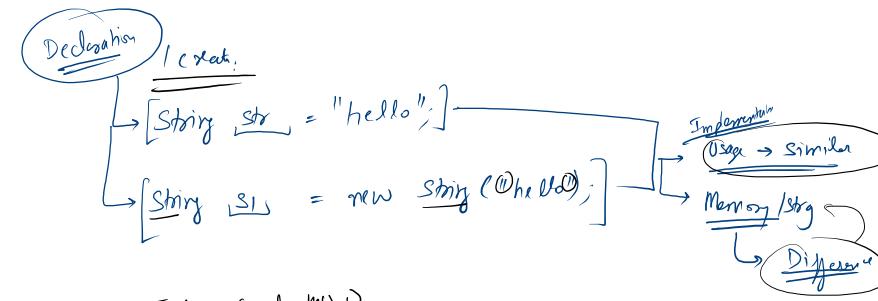


'ch' \Rightarrow char

Scanner scn = new Scanner (System.in)

input

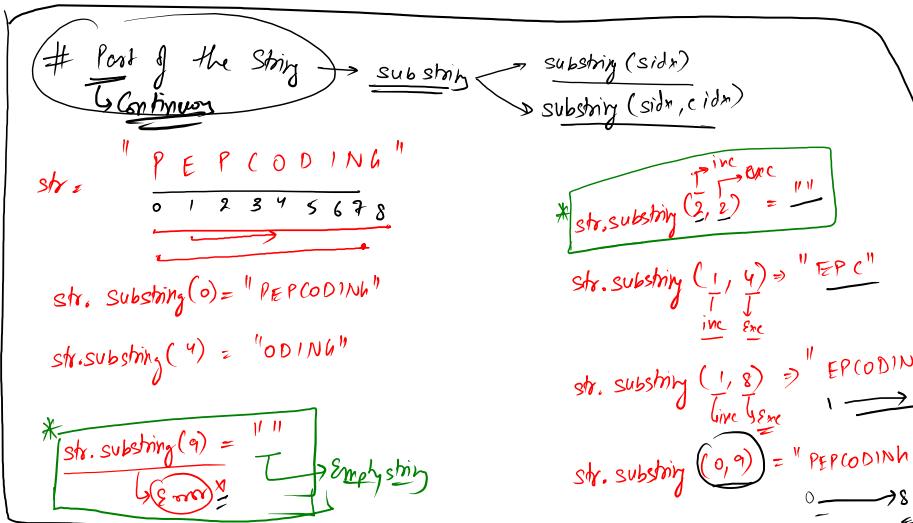
scn.next() → word
scn.nextLine() → lined



Index → (0 > length() - 1)
str = "h e l l o"
0 1 2 3 4

str.charAt(idx)
ex: str.charAt(2) = 'l'
str.charAt(4) = 'o'

NOTE: no. of characters → str.length()



No ↗ re id



0 ↗ Show
0, 1, 2, 3, 4, 5, 6, 7, 8 ↗ id

= "A B C D"

array → Subarray

String → Substrin

"A"

"AB"

"ABC"

"ABCD"

"B"

"B C"

"B C D"

"C"

"C D"

"D"

palindromic

substrin

palindromic

nonon

abc d cba

① Generate all substrings

② Palindrome

abcc

sp,ep

ss(0,1) - $\frac{a}{0}$

ss(0,2) $\frac{a b}{0 1}$

ss(0,3) $\frac{a b c}{0 1 2}$

ss(0,4) $\frac{a b c c'}{0 1 2 3}$

sp=0, ep=1 → length()

(a b c c')
0 1 2 3

~~substrings (inc, exc)~~

sp,ep
ss(1,2) $\frac{b}{1}$

ss(1,3) $\frac{b c}{1 2}$

ss(1,4) $\frac{b c c'}{1 2 3}$

sp,ep
 $\frac{c}{2} ss(2,3)$

$\frac{c c^*}{2 3} ss(2,4) \frac{c^*}{3} ss(3,4)$

sp=1
ep=2 → length()

sp=2
ep=3 → length()

sp=3
ep=4 → length()

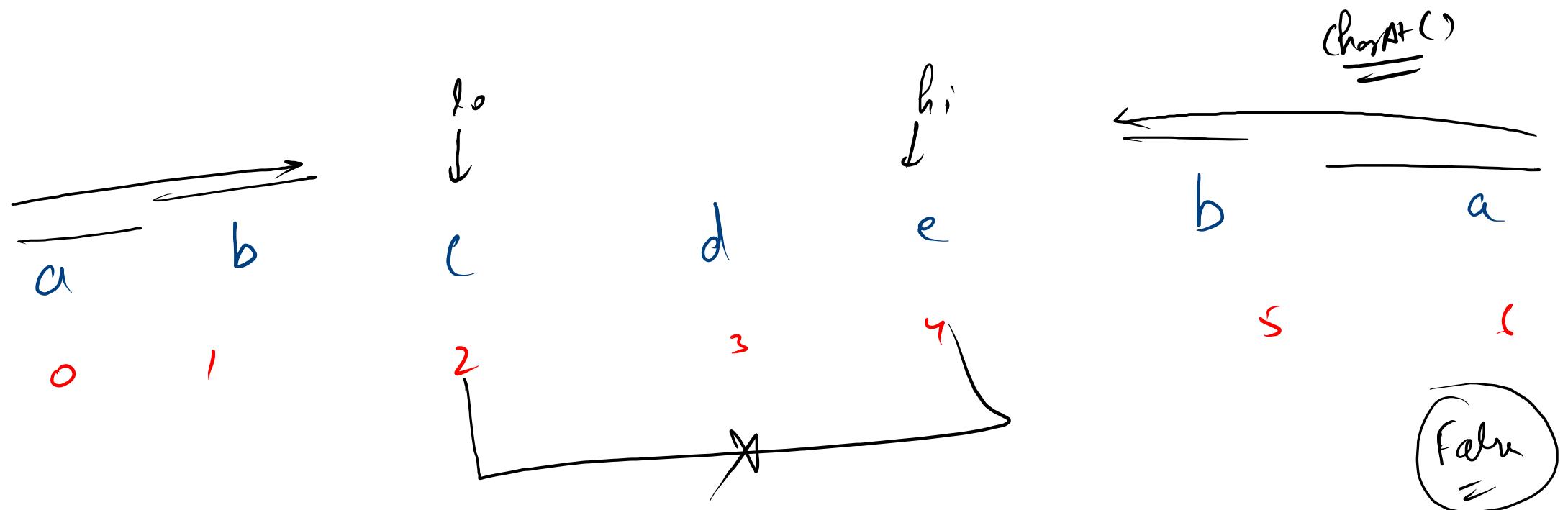
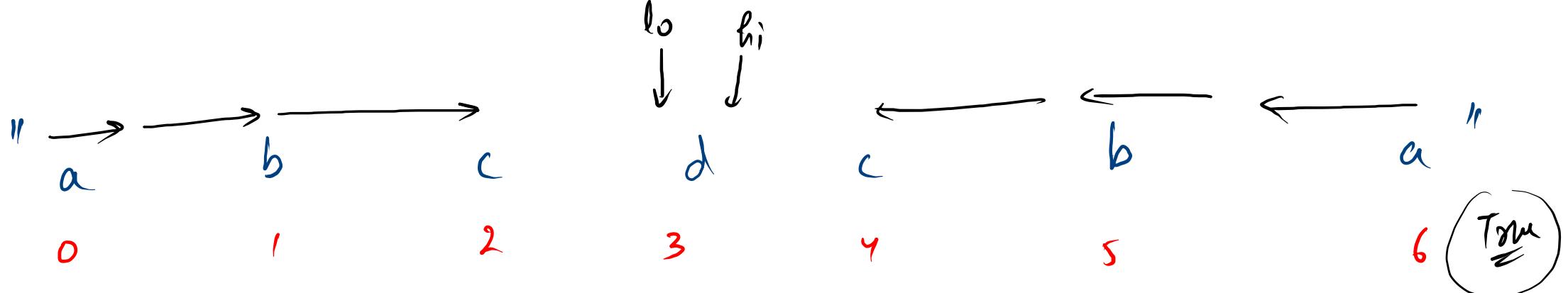
sp
 ↓
 "a b c c"
 0 1 2 3 4
 ↑
 ep

ss(0,1) → "a"
 ss(0,2) → "ab"
 ss(0,3) → "abc"
 ss(0,4) → "abcc"
 ss(1,2) → "b"
 ss(1,3) → "bc"
 ss(1,4) → "bcc"
 "abcc" → length=4

ss(2,3) → "c"
 ss(2,4) → "cc"
 ss(3,4) → "c"

```

public static void solution(String str){
  for(int sp = 0 ; sp <= str.length()-1 ; sp++){
    for(int ep = sp+1 ; ep <= str.length() ; ep++){
      String ss = substring(sp,ep);
    }
  }
}
  
```



input \Rightarrow string

str= wwwwaadexxxxxx
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14

1

C1 \Rightarrow res = "Wadex"

$\text{res} = \text{"W4a3dex6"}$

Count=XXXXXYYZ

wwwaaadexxxxxxx
 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14

res = "w4a3dex1"

Count = 11234123486

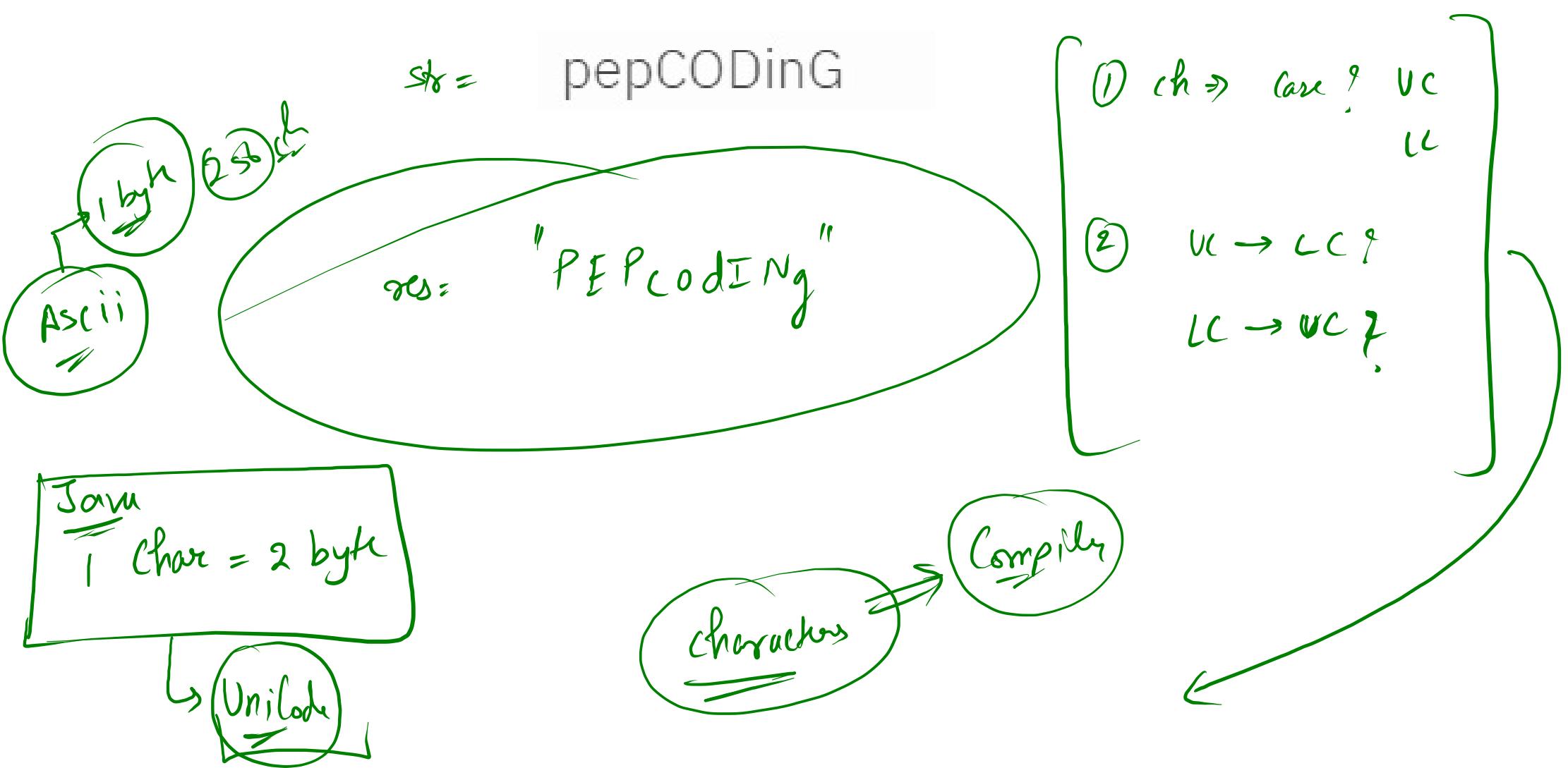
ab
 c
 012
 abc

```

public static String compression2(String str){
  String res = str.charAt(0)+"";
  int count = 1;

  for(int idx = 1 ; idx < str.length() ; idx++){
    char curr = str.charAt(idx);
    char prev = str.charAt(idx-1);

    if(curr == prev){
      count = count+1;
    }else{
      if(count > 1){
        res = res + count;
        count = 1;
      }
      res = res + curr;
    }
    if(count > 1){
      res = res + count;
    }
  }
  return res;
}
  
```



32

$$\underline{LC \rightarrow UC} (ch)$$

$$UC = 'A' + (ch - 'a')$$

$$65 + (109 - 97)$$

$$65 + 12 \Rightarrow 77(M)$$

$$\underline{UC \rightarrow LC} (ch)$$

$$LC = 'a' + (ch - 'A')$$

$$LC = 97 + (74 - 65) \\ = 98 \Rightarrow 102(j)$$

$ch \rightarrow 2\text{ bytes}$

$\text{int} \rightarrow 4\text{ bytes}$

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0 000	000	NUL (null)	32	20 040	000	 	Space	64	40 100	000	@	0	96	60 140	000	`	'
1	1 001	001	SOH (start of heading)	33	21 041	001	!	!	65	41 101	001	A	A	97	61 141	001	a	@
2	2 002	002	STX (start of text)	34	22 042	002	"	"	66	42 102	002	B	B	98	62 142	002	b	b
3	3 003	003	ETX (end of text)	35	23 043	003	#	#	67	43 103	003	C	C	99	63 143	003	c	c
4	4 004	004	EOT (end of transmission)	36	24 044	004	$	\$	68	44 104	004	D	D	100	64 144	004	d	d
5	5 005	005	ENQ (enquiry)	37	25 045	005	%	%	69	45 105	005	A	E	101	65 145	005	e	e
6	6 006	006	ACK (acknowledge)	38	26 046	006	&	&	70	46 106	006	F	F	102	66 146	006	f	f
7	7 007	007	BEL (bell)	39	27 047	007	'	'	71	47 107	007	G	G	103	67 147	007	g	g
8	8 010	010	BS (backspace)	40	28 050	010	((72	48 110	010	H	H	104	68 150	010	h	h
9	9 011	011	TAB (horizontal tab)	41	29 051	011))	73	49 111	011	I	I	105	69 151	010	i	i
10	A 012	012	LF (NL line feed, new line)	42	2A 052	012	*	*	74	4A 112	012	J	J	106	70 152	016	j	j
11	B 013	013	VT (vertical tab)	43	2B 053	013	+	+	75	4B 113	013	K	K	107	6B 153	017	k	k
12	C 014	014	FF (NP form feed, new page)	44	2C 054	014	,	,	76	4C 114	014	L	L	108	6C 154	018	l	l
13	D 015	015	CR (carriage return)	45	2D 055	015	-	-	77	4D 115	015	M	M	109	6D 155	019	m	m
14	E 016	016	SO (shift out)	46	2E 056	016	.	.	78	4E 116	016	N	N	110	6E 156	010	n	n
15	F 017	017	SI (shift in)	47	2F 057	017	/	/	79	4F 117	017	O	O	111	6F 157	011	o	o
16	10 020	020	DLE (data link escape)	48	30 060	020	0	0	80	50 120	020	P	P	112	70 160	012	p	p
17	11 021	021	DC1 (device control 1)	49	31 061	021	1	1	81	51 121	021	Q	Q	113	71 161	013	q	q
18	12 022	022	DC2 (device control 2)	50	32 062	022	2	2	82	52 122	022	R	R	114	72 162	014	r	r
19	13 023	023	DC3 (device control 3)	51	33 063	023	3	3	83	53 123	023	S	S	115	73 163	015	s	s
20	14 024	024	DC4 (device control 4)	52	34 064	024	4	4	84	54 124	024	T	T	116	74 164	016	t	t
21	15 025	025	NAK (negative acknowledge)	53	35 065	025	5	5	85	55 125	025	U	U	117	75 165	017	u	u
22	16 026	026	SYN (synchronous idle)	54	36 066	026	6	6	86	56 126	026	V	V	118	76 166	018	v	v
23	17 027	027	ETB (end of trans. block)	55	37 067	027	7	7	87	57 127	027	W	W	119	77 167	019	w	w
24	18 030	030	CAN (cancel)	56	38 070	030	8	8	88	58 130	028	X	X	120	78 170	020	x	x
25	19 031	031	EM (end of medium)	57	39 071	031	9	9	89	59 131	029	Y	Y	121	79 171	021	y	y
26	1A 032	032	SUB (substitute)	58	3A 072	032	:	:	90	5A 132	030	Z	Z	122	7A 172	022	z	z
27	1B 033	033	ESC (escape)	59	3B 073	033	;	:	91	5B 133	031	[[123	7B 173	023	{	{
28	1C 034	034	FS (file separator)	60	3C 074	030	<	<	92	5C 134	032	\	\	124	7C 174	024	|	
29	1D 035	035	GS (group separator)	61	3D 075	031	=	=	93	5D 135	033]]	125	7D 175	025	}	}
30	1E 036	036	RS (record separator)	62	3E 076	032	>	>	94	5E 136	034	^	_	126	7E 176	026	~	_
31	1F 037	037	US (unit separator)	63	3F 077	033	?	?	95	5F 137	035	_	-	127	7F 177	027		DEL

 $ch = 'm'$ $if (ch >='a' \& \& ch <='z')$

}

 $ch \in LC$ $ch \in UC$

}

 $ch \in UC$

}

 $if (ch >='0' \& \& ch <='9')$

}

 $ch \in digits$

}

 $ch \in digits$

}

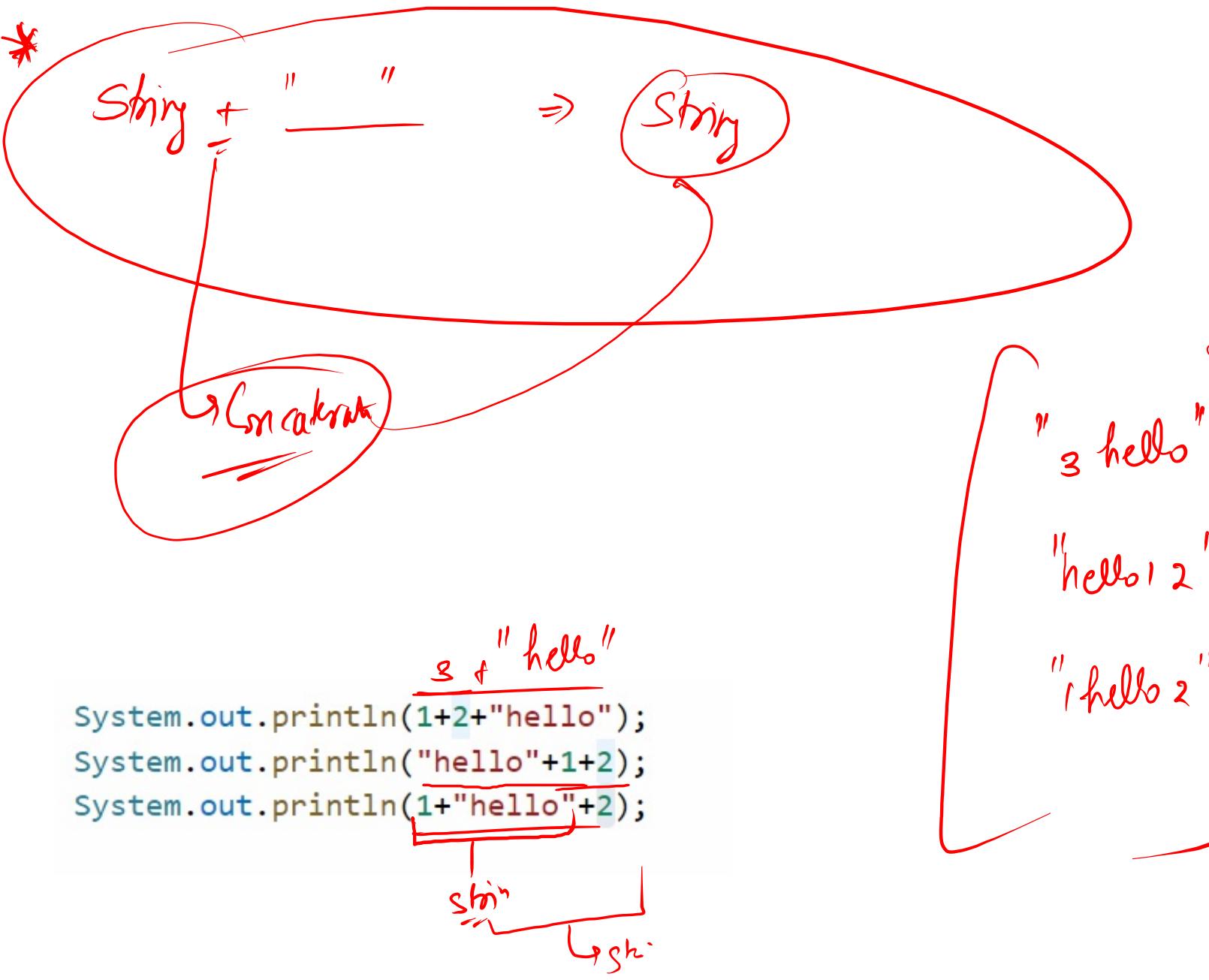
* \rightarrow $ch \rightarrow operation$ \rightarrow $ch \rightarrow unicode$

$int = 4bytes$

$ch = 2bytes$

$int \rightarrow 4bytes$

$int \rightarrow 4bytes$



```
System.out.println(1+2+"hello");
```

```
System.out.println("hello"+1+2);
```

```
System.out.println(1+"hello"+2);
```

a b c

$$\text{abc} = \overline{0} \quad \overline{1} \quad \overline{2} \quad \overline{3}$$

$$0 \quad \perp \quad 2$$

$$b - c = abc \checkmark$$

$$a \swarrow \quad c - b = acb \checkmark$$

$$b \swarrow \quad a - c = bac \checkmark$$

$$c \swarrow \quad c - a = bca \checkmark$$

$$a \swarrow \quad b - b = cab \checkmark$$

$$b \swarrow \quad a = cba \checkmark$$

$$\begin{array}{r} 3 \\ | \\ 2 \\ | \\ 1 \end{array} \quad \begin{array}{r} 3 \\ | \\ 1-0 \\ | \\ 0-1 \\ | \\ 0-0 \end{array}$$

$$\begin{array}{r} "abc" \\ | \\ b \\ | \\ 0 \\ | \\ 0 \end{array}$$

acb ✓

$$\begin{array}{r} 3 \\ | \\ 2 \\ | \\ 1 \end{array} \quad \begin{array}{r} 5 \\ | \\ 1-2 \\ | \\ 0-p \\ | \\ 0-0 \end{array}$$

$$\begin{array}{r} "abc" \\ | \\ b \\ | \\ 1 \\ | \\ 0 \end{array}$$

cba ✓

$$\begin{array}{r} 3 \\ | \\ 2 \\ | \\ 1 \end{array} \quad \begin{array}{r} 4 \\ | \\ 1-1 \\ | \\ 0-1 \\ | \\ 0-0 \end{array}$$

$$\begin{array}{r} "abc" \\ | \\ a \\ | \\ 0 \\ | \\ 0 \end{array}$$

bca

$$\begin{array}{r} \rightarrow 3 \\ | \\ \rightarrow 2 \\ | \\ \rightarrow 1 \end{array} \quad \begin{array}{r} 0 \\ | \\ 0-0 \\ | \\ 0-0 \\ | \\ 0-0 \end{array}$$

$$\begin{array}{r} "abc" \\ | \\ b \\ | \\ 1 \\ | \\ 0 \\ | \\ 0 \end{array}$$

"abc"

$$\begin{array}{r} 3 \\ | \\ 2 \\ | \\ 1 \end{array} \quad \begin{array}{r} 1 \\ | \\ 0-1 \\ | \\ 0-0 \\ | \\ 0-0 \end{array}$$

$$\begin{array}{r} "abc" \\ | \\ "dc" \\ | \\ "q" \\ | \\ 0 \end{array}$$

"bac"

$$\begin{array}{r} 3 \\ | \\ 2 \\ | \\ 1 \end{array} \quad \begin{array}{r} 2 \\ | \\ 0-2 \\ | \\ 0-0 \\ | \\ 0-0 \end{array}$$

$$\begin{array}{r} "abc" \\ | \\ "ab" \\ | \\ "b" \\ | \\ 0 \end{array}$$

cab

num

0

1

2

③
3

4

5

(tmp) (val)

3	3
2	1 - 0
1	0 - 1
0	0 - 0

tmp mStrn val

3	abc	3
	012	

2	bc	1
	01	

1	(b) 0	0
	" "	

0	" "	0
	" "	

]a[bc

]b[c

]b[c

lp = " "
rp = "bc"

lp = "b"
rp = " "

perm = "" + 'a' + 'c' + 'b' = "acb"

]b[
]b[
lp = " "
rp = " "] → " "

str = "abc"

abc

abc

```
public static void solution(String str){
    int n = str.length();
    int totalPerm = fact(n);

    for(int num = 0 ; num < totalPerm ; num++){
        int tmp = n;
        String mStr = str;
        String perm = "";
        int val = num;

        while(tmp != 0){
            int rem = val % tmp;
            val = val/tmp;
            perm = perm + mStr.charAt(rem);
        }

        String lp = mStr.substring(0,rem);
        String rp = mStr.substring(rem+1);
        mStr = lp+rp;
        tmp--;
    }
    System.out.println(perm);
}
```

$n^2 n!$

$\frac{n!}{(0,0)}$

$\frac{0+1}{1}$