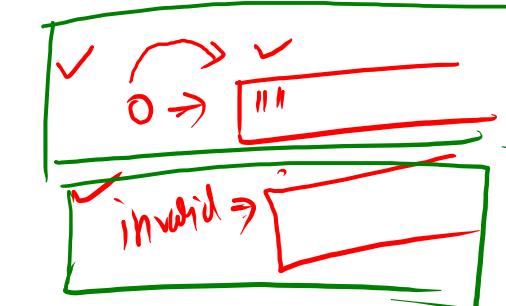
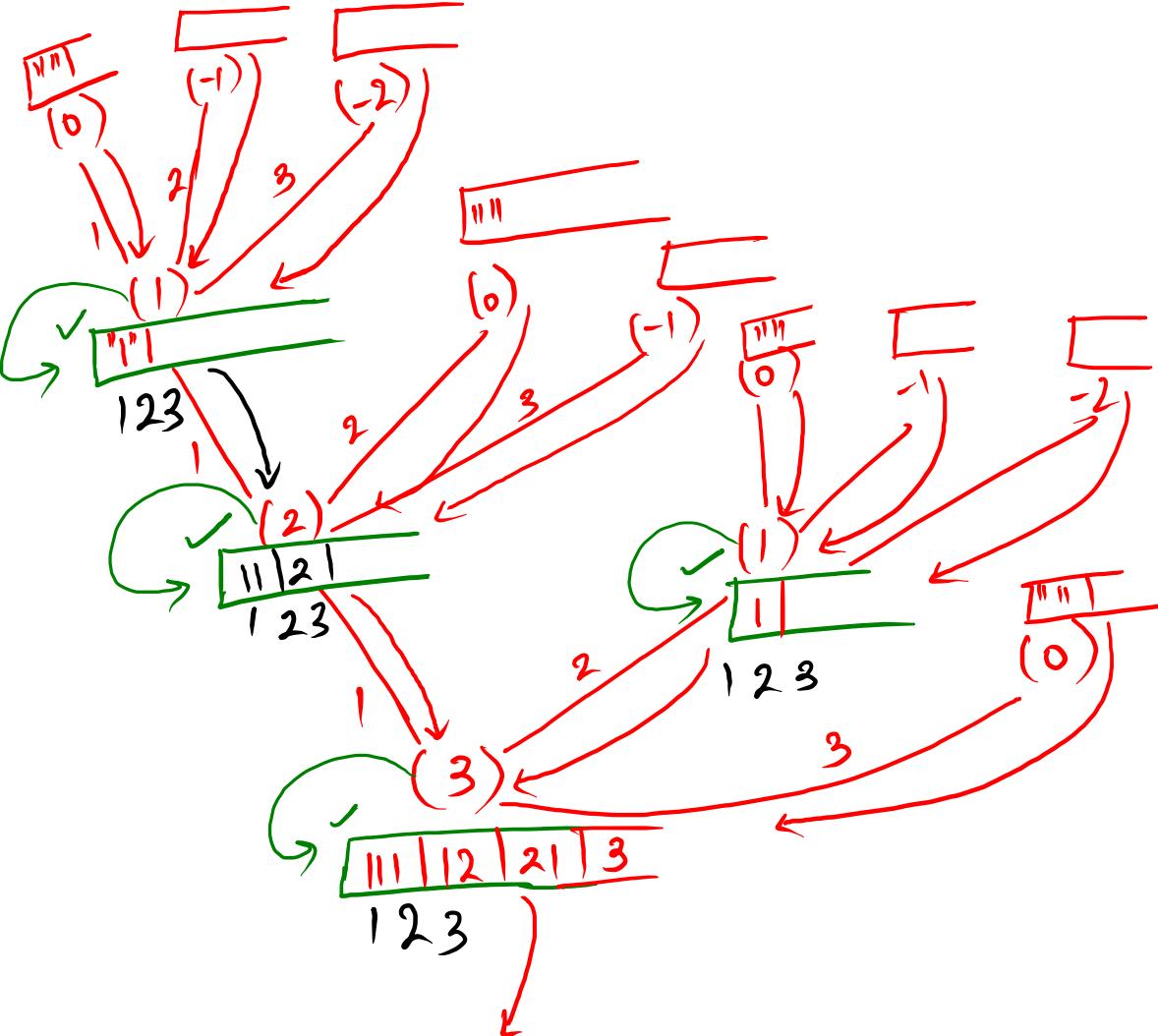


Output
point all ways to climb n stairs

given
at any time

1 km 2 km 3 km

Idea



```

public static ArrayList<String> getStairPaths(int n) {
    public static ArrayList<String> getStairPaths(int n) {

        ArrayList<String> mypath = new ArrayList<>();

        1 ArrayList<String> path1len = getStairPaths(n-1); // 1 Len
        for(String path : path1len){
            mypath.add("1"+path);
        }

        2 ArrayList<String> path2len = getStairPaths(n-2); // 2 Len
        for(String path : path2len){
            mypath.add("2"+path);
        }

        3 ArrayList<String> path3len = getStairPaths(n-3); // 3 Len
        for(String path : path3len){
            mypath.add("3"+path);
        }

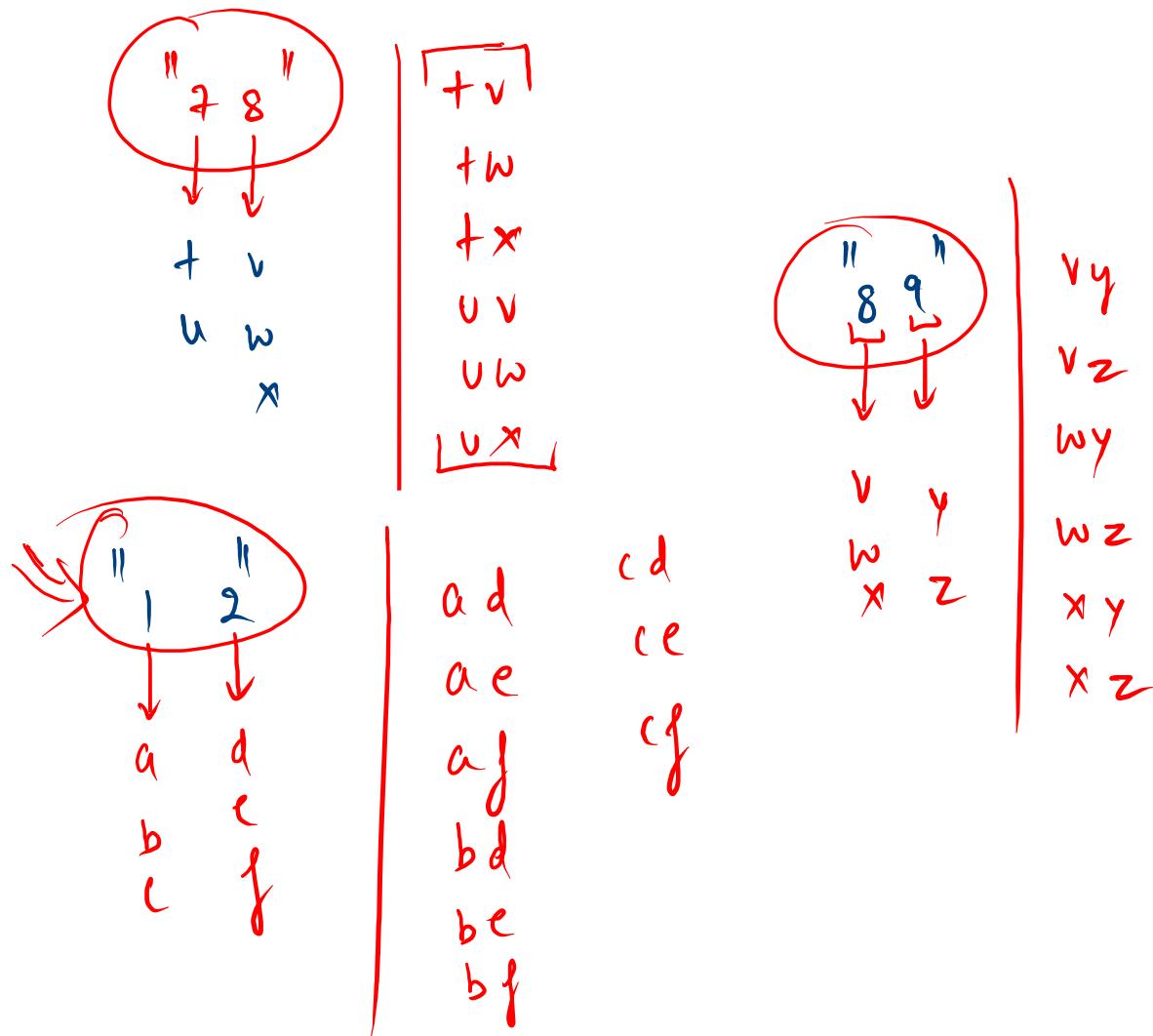
        return mypath;
    }
}

```


0 -> .;
 1 -> abc
 2 -> def
 3 -> ghi
 4 -> jkl
 5 -> mno
 6 -> pqrs
 7 -> tu
 8 -> vwx
 9 -> yz

<u>abc</u>	<u>def</u>	<u>ghi</u>
1	2	3
jkl	mno	pqrs
4	5	6
tu	vwx	yz
7	8	9
	!	o

<u>abc</u>	<u>def</u>	<u>ghi</u>
1	2	3
jkl	mno	pqrs
4	5	6
tu	vwx	yz
7	8	9
	!	o





```
static String keypad[] = {".;","abc","def","ghi","jkl","mno","pqrs","tu","vwx","yz"};
```

0 1 2 3 4 5 6 7 8 9

Keypad[4]

can be used in any game

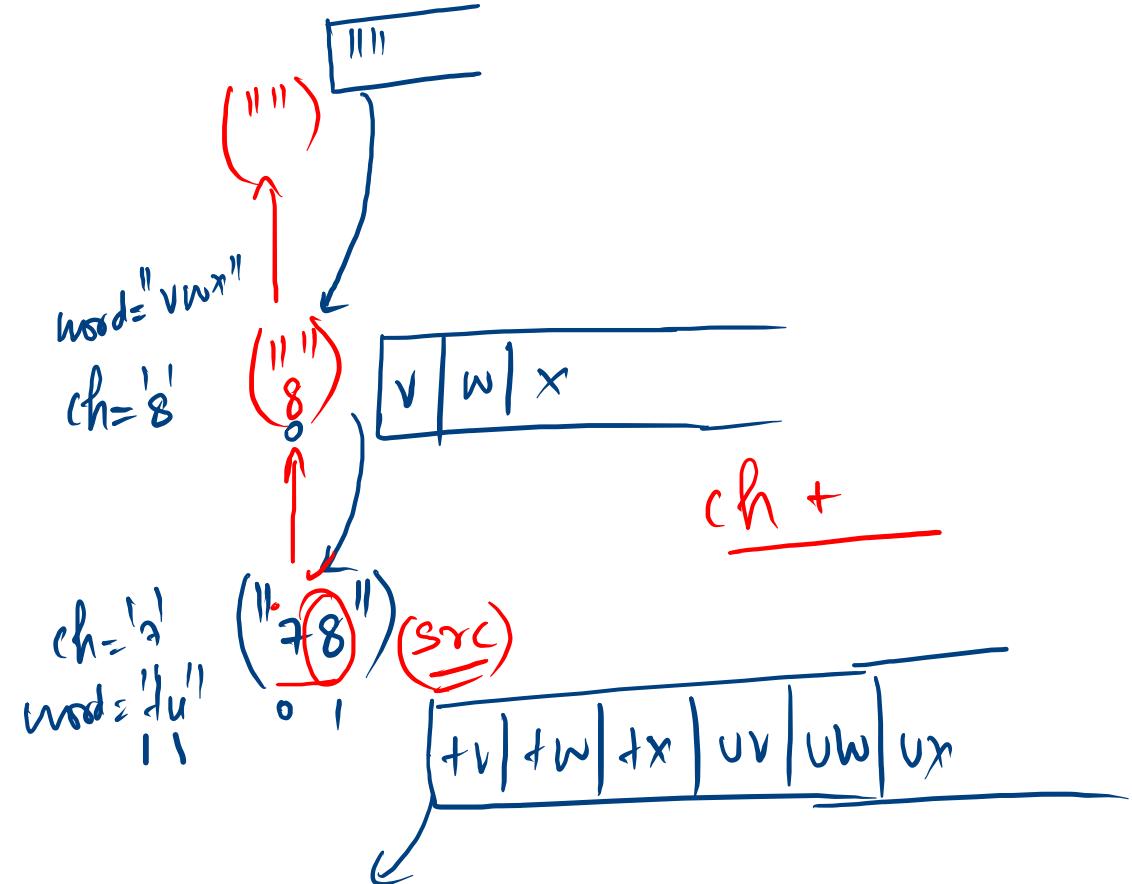
Keypad[i] → characters represented by ith key

```

public static String keypad[] = {".;","abc","def","ghi","jkl","mno","pqrs",("tu"),("vwx"),"yz"};
      0   1   2   3   4   5   6   7   8   9

```

abc 1	def 2	ghi 3
jkl 4	mno 5	pqrs 6
tu 7	vwx 8	yz 9
.	0	



ch → int
ch → (ch - '0')

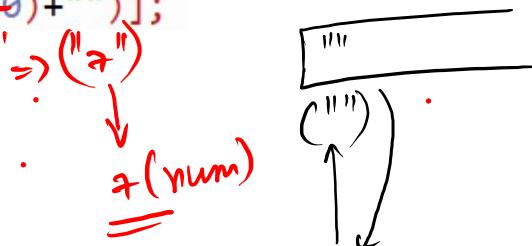
Integer.parseInt(str.charAt(0) + "") → num

ch(int)
ch(ascii)

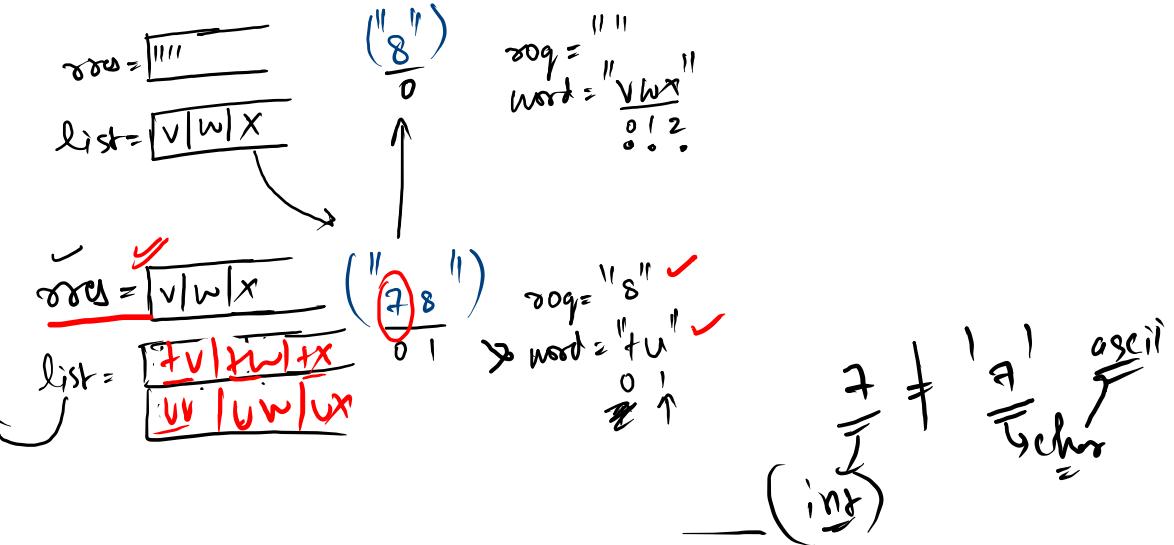
public static String keypad[] = {".", "abc", "def", "ghi", "jkl", "mno", "pqrs", "tuv", "vwx", "yz"};

Keypad[?]

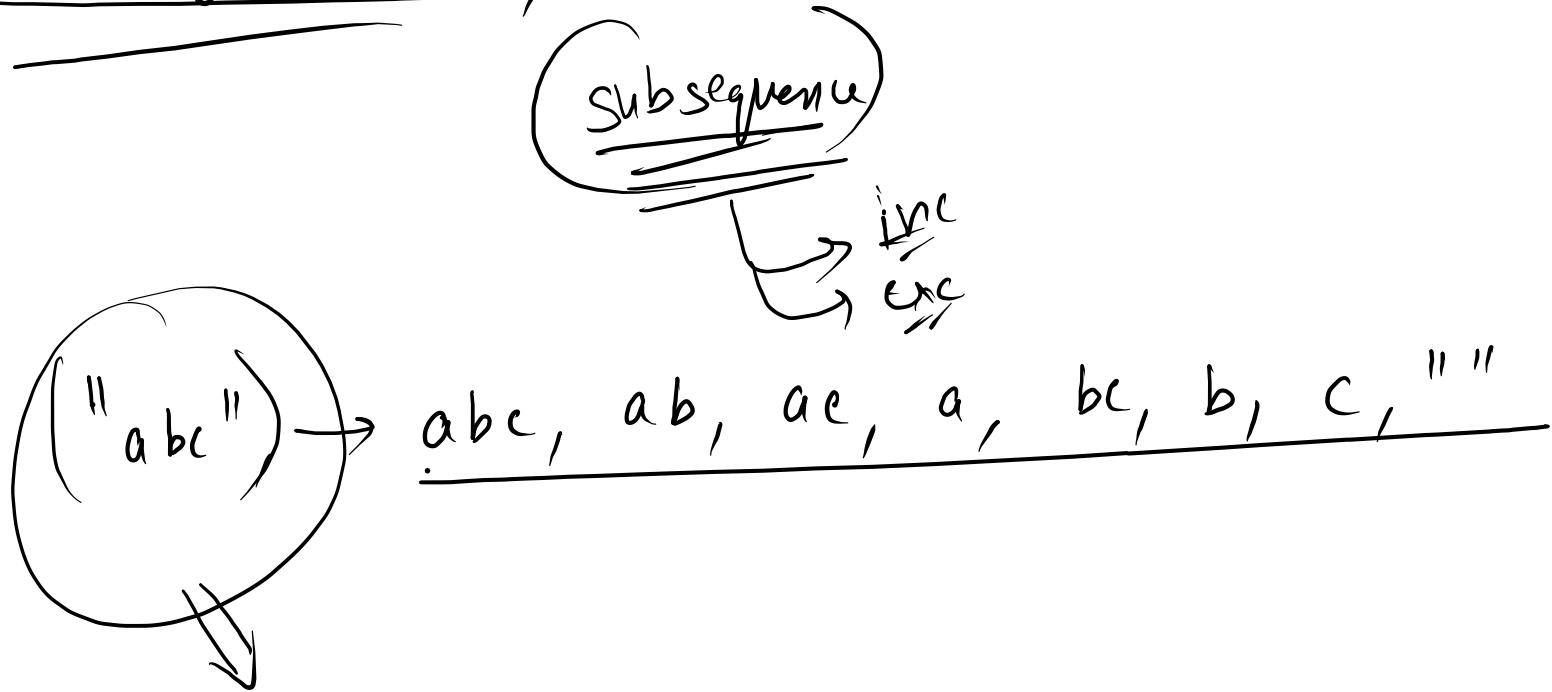
✓ String word = keypad[Integer.parseInt(str.charAt(0) + "")];



```
public static ArrayList<String> getKPC(String str) {
    String roq = str.substring(1); 1
    ArrayList<String> rres = getKPC(roq); 2
    String word = keypad[str.charAt(0)]; 3
    ArrayList<String> list = new ArrayList<>(); 4
    for(int idx = 0 ; idx < word.length() ; idx++){
        for(String comb : rres){
            list.add(word.charAt(0)+comb);
        }
    }
    return list;
}
```



Recurison of the way up



("", "abc") ("", "ab")

\checkmark \circled{d} x

"", "ab"

\checkmark

$(\text{"", "ac})(\text{"", "a})$ ("", "bc) ("", "b)

x

x

("", "c) ("", ")

x

"", "c" o , "", "a"

x

"", "c" o , "", "b"

x

"", "c" o , $\text{"", "}"$

x

\checkmark \circled{b} x

x

"", "bc" o , "", "a"

x

"", "bc" o , $\text{"", "}"$

x

\checkmark \circled{b} x

x

"", "bc" o , $\text{"", "}"$

x

\checkmark \circled{a} x

x

("", "abc") o , "", "ay"

012

qsf

abc
 ab
 ac
 a
 bc
 b
 c
 "", "

~~so s, or ch~~
 ~~so s, ay~~
 ch
 (str, ay)

abc

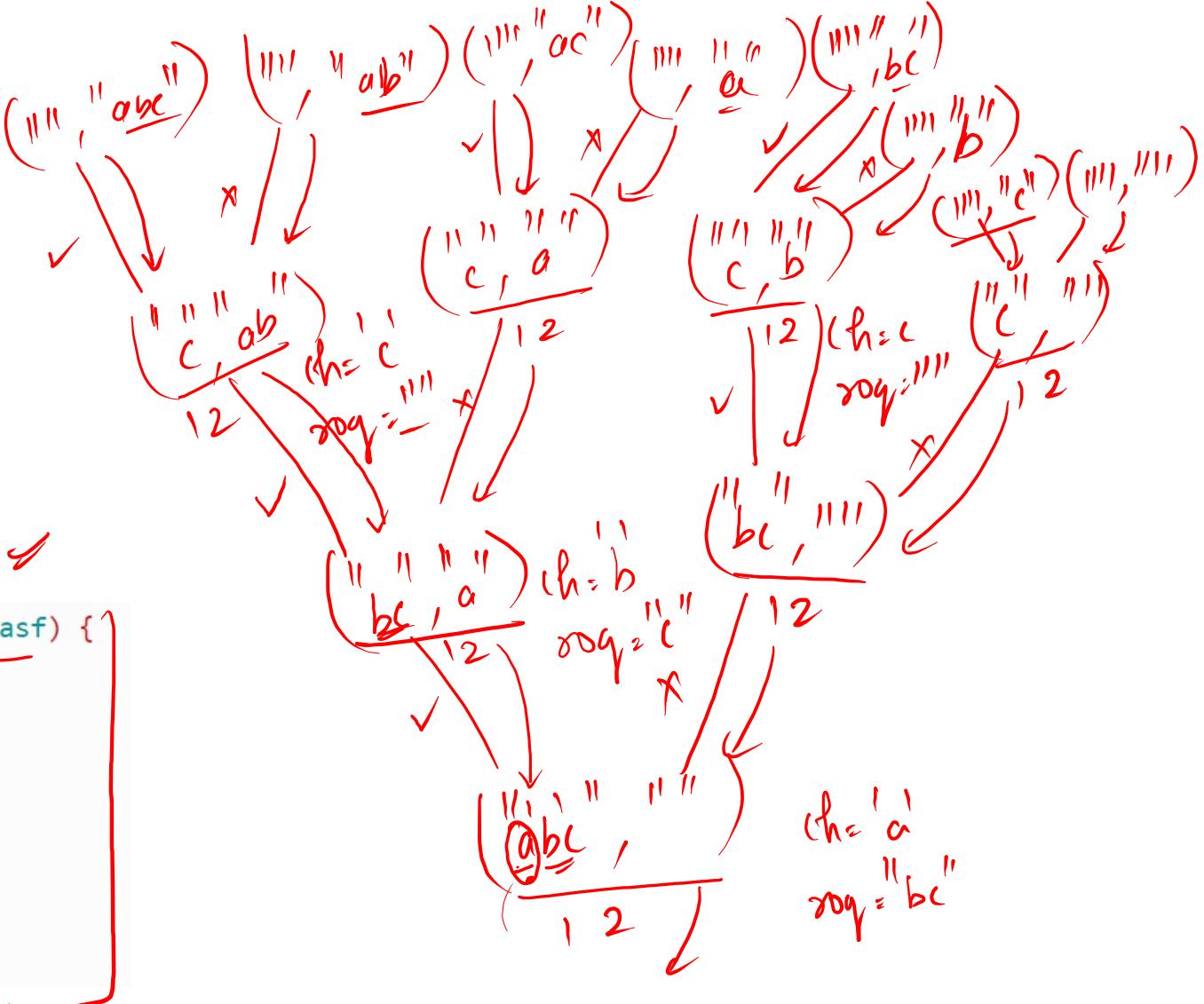
ab

ac

a

b
c
b
c
"

```
public static void printSS(String ques, String asf) {  
    if(ques.length() == 0){  
        System.out.println(asf);  
        return;  
    }  
    char ch = ques.charAt(0);  
    String roq = ques.substring(1);  
    printSS(roq, asf+ch); -1  
    printSS(roq, asf); -2  
}
```

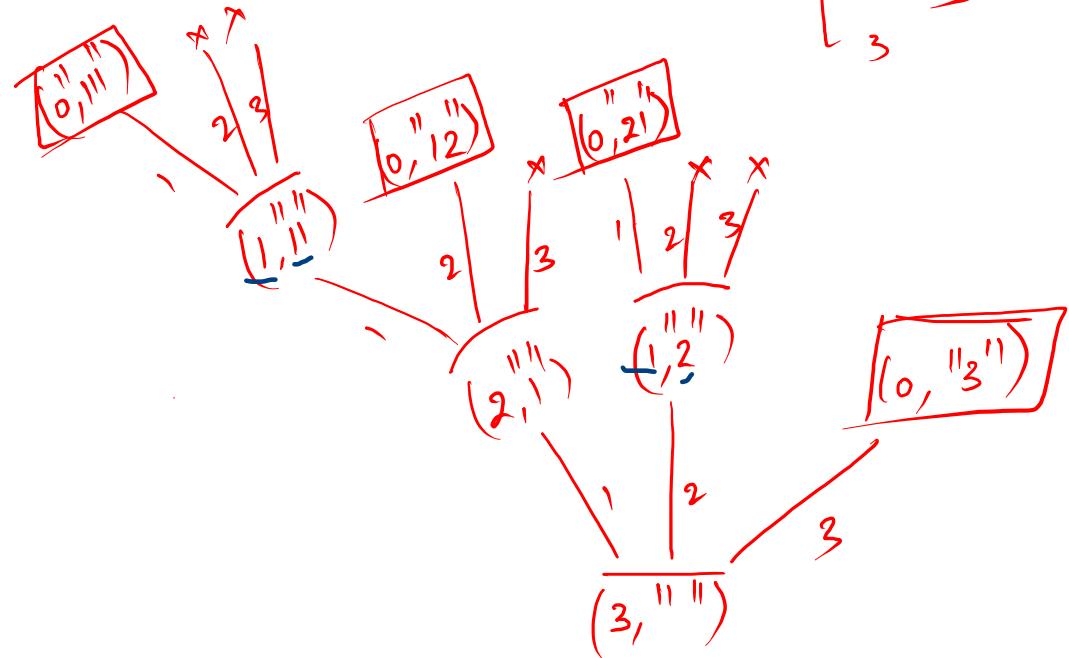


3

1 len
2 len
3 len

Sample

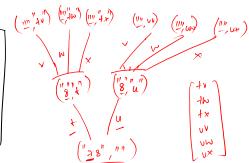
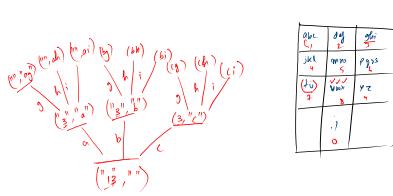
111
12
21
3



$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 \\ 2 & 1 \\ 3 \end{bmatrix}$

```

P   S   v   psp (int n, String str) {
if (n == 0) {
    System.out.println(str);
    return;
}
if (n - 1 >= 0) {
    psp (n - 1, str + "1");
}
if (n - 2 >= 0) {
    psp (n - 2, str + "2");
}
if (n - 3 >= 0) {
    psp (n - 3, str + "3");
}
}
  
```

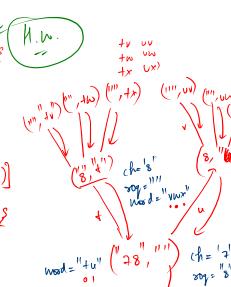


```

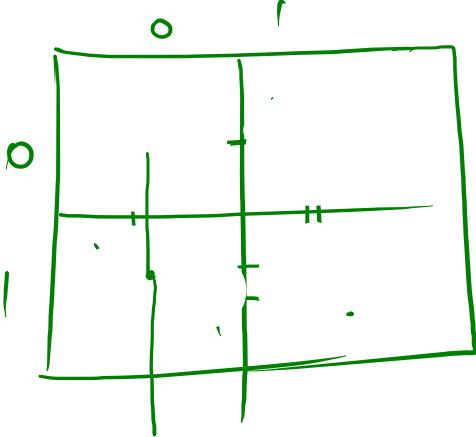
public void printKPC (String str, String obj) {
    if (str.length() == 0) {
        System.out.println(obj);
        return;
    }
    char ch = str.charAt(0);
    String word = str.substring(1);
    String[] keypad = { "abc", "def", "ghi", "jkl", "mno", "pqrs", "tuv", "wxyz" };
    for (int i = 0; i < keypad.length(); i++) {
        printKPC (word, obj + keypad[i].charAt(i));
    }
}

```

3



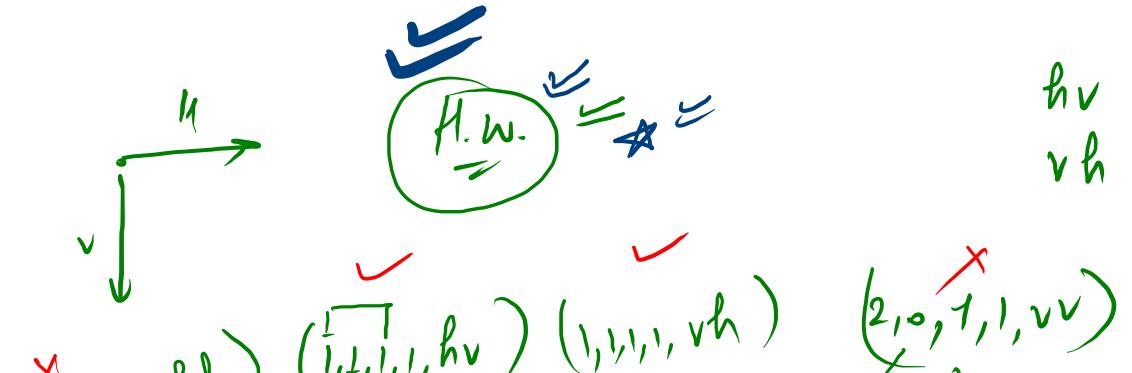
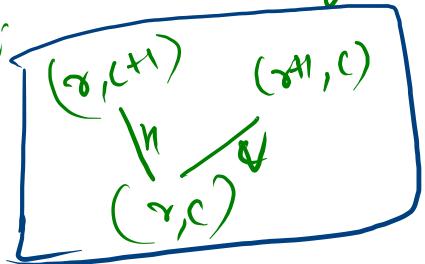
3



```

if (sr == dr && dc == dc) {
    System.out.println(ps);
    return;
}

```



```

if (sr > dr || sc > dc) {
    return;
}

```

