→ Recursion

↳ Recursion with Arraylist

Get Kpc (Day 25)

Problem | Submissions | Leaderboard | Discussions

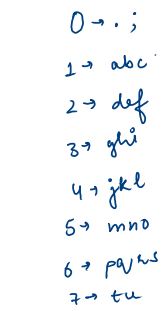
1. You are given a string str. The string str will contains numbers only, where each number stands for a key pressed on a mobile phone.
2. The following list is the key to characters map : 0 -> .; 1 -> abc 2 -> def 3 -> ghi 4 -> jkl 5 -> mno 6 -> pqrs 7 -> tu 8 -> vwx 9 -> yz
3. Get the list of all words that could be produced by the keys in str.

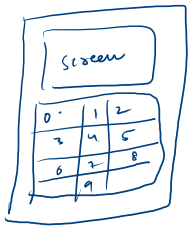
Sample Input 0

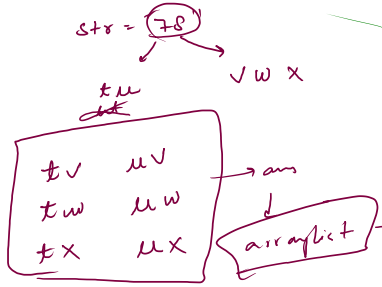
str → 78

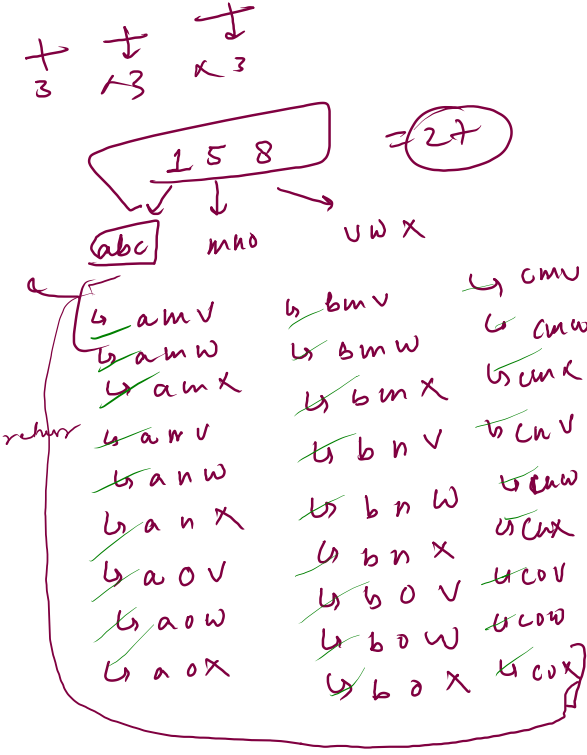
Sample Output 0

```
[tv, tw, tx, uv, uw, ux]
```









Imp:

for → iterating all the characters represented by "no." → 8

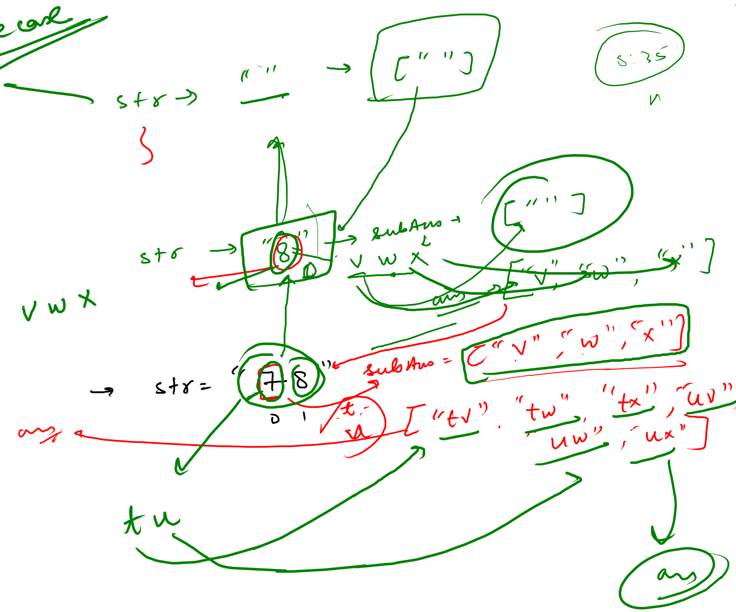
↳ iterating over all the subAns

KPC

getKPC

base case

arr = { "", "", "abc", "def" }
↓ string



7 → tu, 8 → vwx

5-6 min

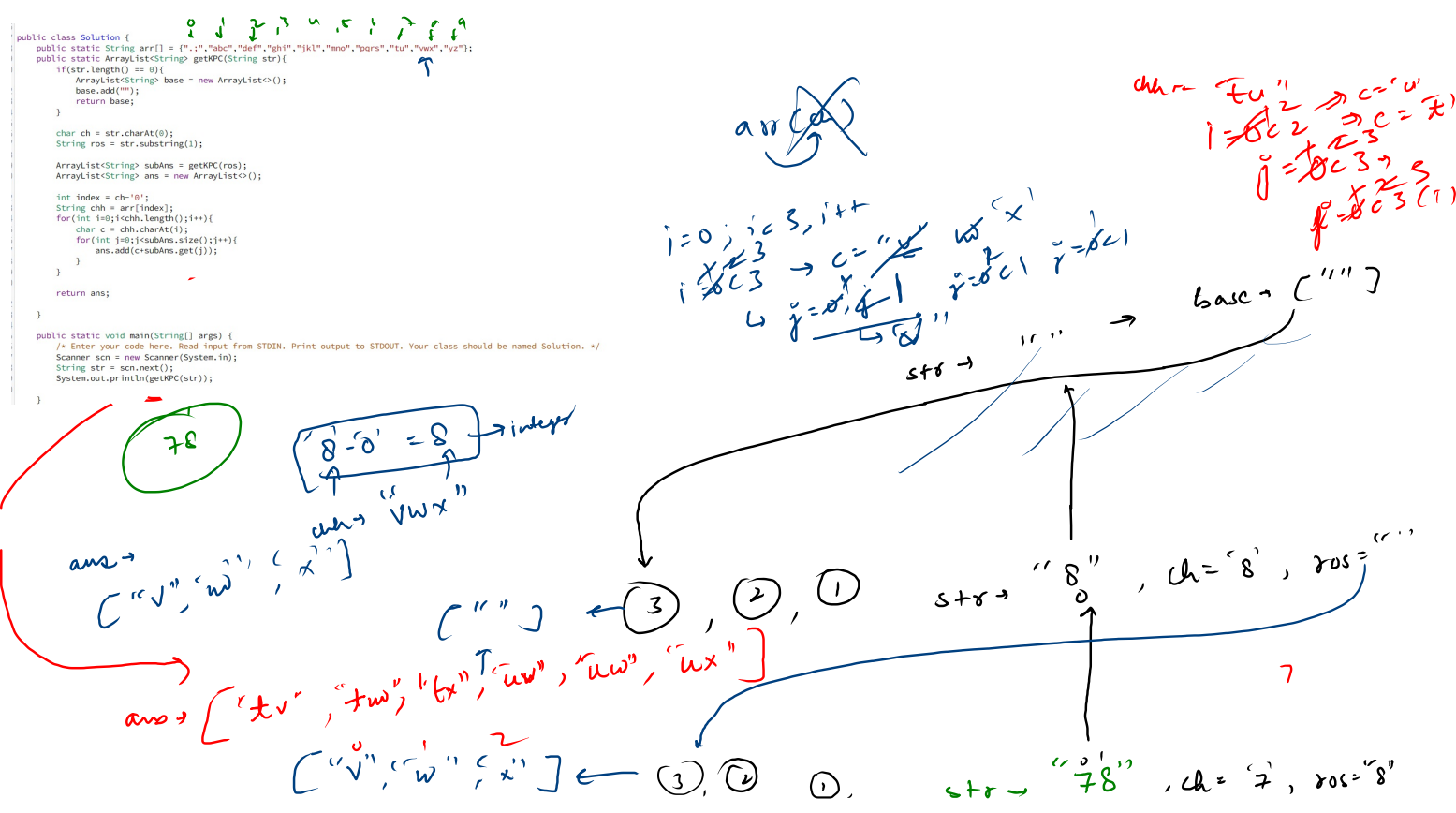
↳ Dr work

→ Make an array, which represents characters present in keypad

arr → ______

→ fn → find all the combination of given string

```java
public class Solution {
    public static String arr[] = {".;","abc","def","ghi","jkl","mno","pqrs","tu","vwx","yz"};
    public static ArrayList<String> getKPC(String str){
        if(str.length() == 0){
            ArrayList<String> base = new ArrayList<>();
            base.add("");
            return base;
        }

        char ch = str.charAt(0);
        String ros = str.substring(1);

        ArrayList<String> subAns = getKPC(ros);
        ArrayList<String> ans = new ArrayList<>();

        int index = ch-'0';
        String chh = arr[index];
        for(int i=0;i<chh.length();i++){
            char c = chh.charAt(i);
            for(int j=0;j<subAns.size();j++){
                ans.add(c+subAns.get(j));
            }
        }

        return ans;

    }

    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be named Solution. */
        Scanner scn = new Scanner(System.in);
        String str = scn.next();
        System.out.println(getKPC(str));

    }
```



9:10 PM

Code + Dry Run

```java
public class Solution {
    public static String arr[] = {".;","abc","def","ghi","jkl","mno","pqrs","tu","vwx","yz"};
    public static ArrayList<String> getKPC(String str){
        if(str.length() == 0){
            ArrayList<String> base = new ArrayList<>();
            base.add("");
            return base;
        }

        char ch = str.charAt(0);
        String ros = str.substring(1);

        ArrayList<String> subAns = getKPC(ros);
        ArrayList<String> ans = new ArrayList<>();

        int index = ch-'0';
        String chh = arr[index];
        for(int i=0;i<chh.length();i++){
            char c = chh.charAt(i);
            for(int j=0;j<subAns.size();j++){
                ans.add(c+subAns.get(j));
            }
        }

        return ans;

    }

    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be named Solu
        Scanner scn = new Scanner(System.in);
        String str = scn.next();
        System.out.println(getKPC(str));

    }
}
```

Get Maze Paths (Day 25)

Problem | Submissions | Leaderboard | Discussions

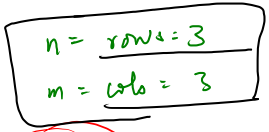
1. You are given a number n and a number m representing number of rows and columns in a maze.
2. You are standing in the top-left corner and have to reach the bottom-right corner. Only two moves are allowed 'h' (1-step horizontal) and 'v' (1-step vertical).
3. Get the list of all paths that can be used to move from top-left to bottom-right.

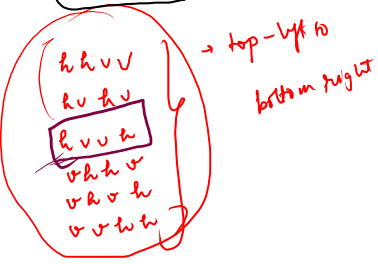
Sample Output 0

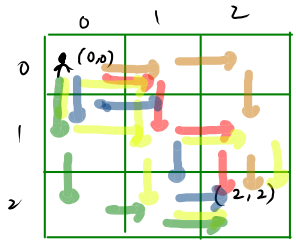
```
[hhvv, hvhv, hvvh, vhhv, vhvh, vvhh]
```

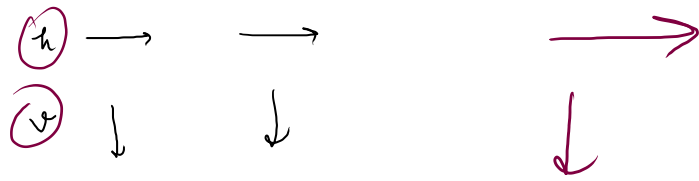
Sample Input 0

```
3
3
```

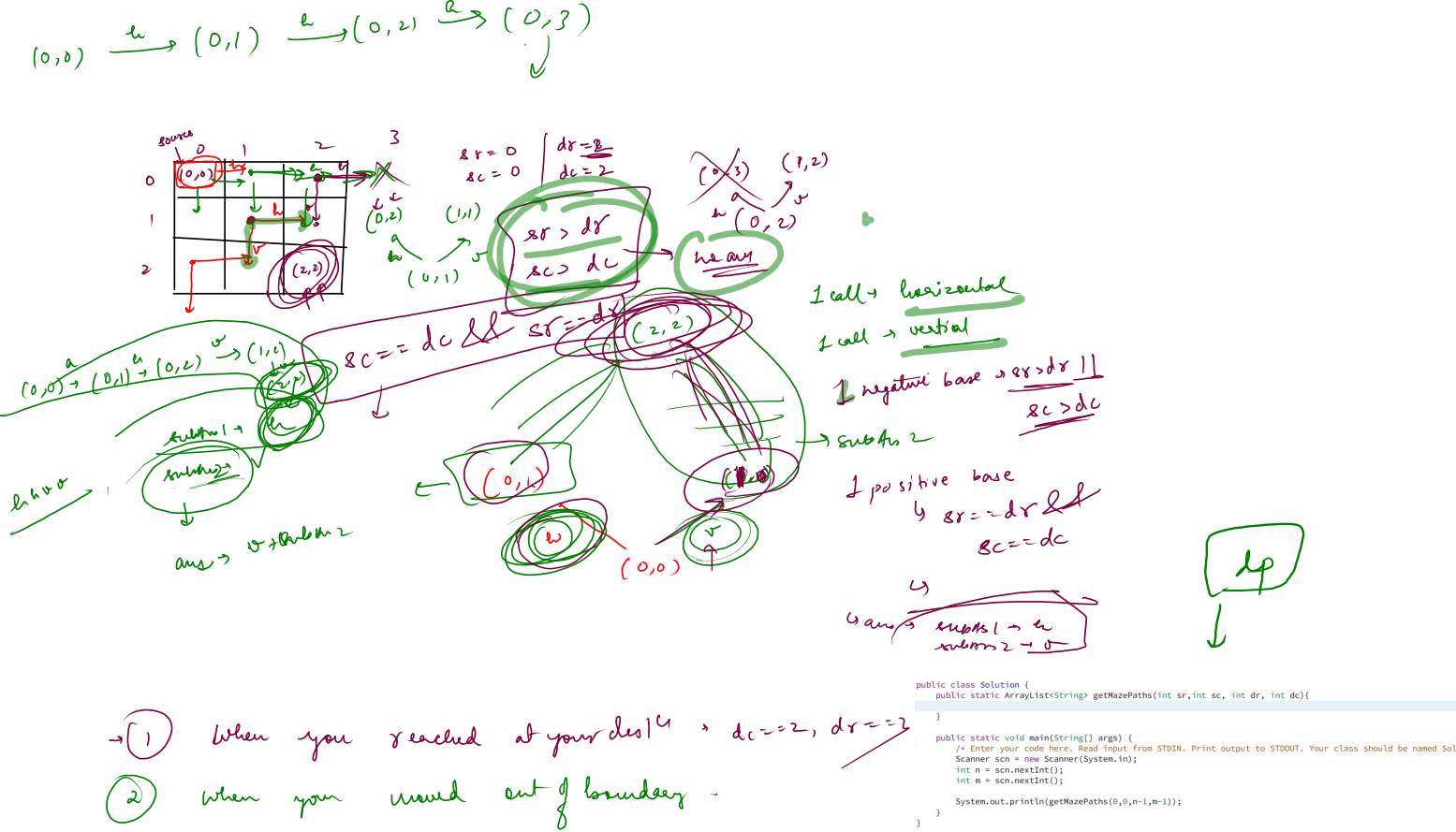








(0,0) , (2,2)

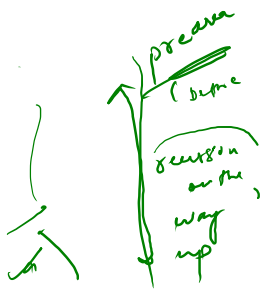


(1) When you reached at your destination, dc == 2, dr == 2

(2) When you moved out of boundary.

```
public class Solution {
    public static ArrayList<String> getMazePaths(int sr,int sc, int dr, int dc){

    }

    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be named Sol
        Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();
        int m = scn.nextInt();

        System.out.println(getMazePaths(0,0,n-1,m-1));
    }
}
```


```java
public class Solution {
    public static ArrayList<String> getMazePaths(int sr,int sc, int dr, int dc){
        if(sc> dc || sr>dr){
            return new ArrayList<>();
        }

        if(sr == dr && sc == dc){
            ArrayList<String> base = new ArrayList<>();
            base.add("");
            return base;
        }

        ArrayList<String> subAns1 = getMazePaths(sr, sc+1, dr, dc);
        ArrayList<String> subAns2 = getMazePaths(sr+1, sc, dr, dc);

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

        for(int i=0;i<subAns1.size();i++){
            ans.add("h" + subAns1.get(i));
        }

        for(int i=0;i<subAns2.size();i++){
            ans.add("v" + subAns2.get(i));
        }

        return ans;

    }

    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be named Soluti
        Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();
        int m = scn.nextInt();

        System.out.println(getMazePaths(0,0,n-1,m-1));
    }
}
```



Recursion on the way.