Permutations – Leet-Code : 46

CODE :

import java.util.ArrayList;

import java.util.List;

public class Permutations{

    public static void main(String[] args) {

        int[] nums = {1, 2, 3}; // Example input

        Permutations permutation = new Permutations();

        List<List<Integer>> result = permutation.permute(nums);

        System.out.println(result);

    }

    public List<List<Integer>> permute(int[] nums) {

        List<List<Integer>> finalList = new ArrayList<>();

        List<Integer> list = new ArrayList<>();

        helper(0, list, finalList, nums);

        return finalList;

    }

    public void helper(int index, List<Integer> list, List<List<Integer>> finalList, int[] nums) {

        if (index == nums.length) {

            for (int i = 0; i < nums.length; i++) {

                list.add(nums[i]);

            }

            finalList.add(new ArrayList<>(list));

            list.clear();

            return;

        }

        for (int i = index; i < nums.length; i++) {

            swap(i, index, nums);

            helper(index + 1, list, finalList, nums);

            swap(i, index, nums);

        }

    }

    public void swap(int i, int j, int[] arr) {

        int temp = arr[i];

        arr[i] = arr[j];

        arr[j] = temp;

    }

}

Output : 

Letter Combination of Phone Numbers

CODE:

import java.util.ArrayList;

import java.util.List;

public class LetterComb {

    static String keys[] = {"", ".", "abc", "def", "ghi", "jkl", "mno", "pqrs", "tuv", "wxyz"};

    public static List<String> letterCombinations(String digits) {

        // Termination case

        if(digits.length() == 0) {

            List<String> list = new ArrayList<>();

            list.add("");

            return list;

        }

        // Get one digit at a time

        // if digit is "23", then firstchar will store '2'

        char firstChar = digits.charAt(0);

        // convert firstchar into int

        int index = firstChar - '0';

        // currentKey will store element at index i.e., "abc"

        String currentKey = keys[index];

        String remainingString = digits.substring(1);

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

        for(int i = 0; i < currentKey.length(); i++) {

            List<String> res = letterCombinations(remainingString);

            for(String s : res) {

                ans.add(currentKey.charAt(i) + s);

            }

        }

        return ans;

    }

    public static void main(String[] args) {

        String digits = "23";

        List<String> op = letterCombinations(digits);

        System.out.println(op);

    }

}

Output :

A screen shot of a computer

Description automatically generated with low confidence