

Problem: Letter Combinations of a Phone Number

Student: Ovidiu Mura

```
public class Solution {
    public List<String> letterCombinations(String digits) {
        Map<Integer, String> pad = new HashMap<>();
        pad.put(2, "abc");
        pad.put(3, "def");
        pad.put(4, "ghi");
        pad.put(5, "jkl");
        pad.put(6, "mno");
        pad.put(7, "pqrs");
        pad.put(8, "tuv");
        pad.put(9, "wxyz");
        List<String> combinations = new ArrayList<>();
        if(digits.length() == 0)
            return combinations;
        char[] cs = digits.toCharArray();
        List<String> lettersGrp = new ArrayList();
        for(char c: cs){
            lettersGrp.add(pad.get(Character.getNumericValue(c)));
        }
        char[] a;
        char[] b;
        for (int ii=0; ii<lettersGrp.size(); ii++) {
            for(int jj=ii+1; jj<lettersGrp.size(); jj++) {
                if (lettersGrp.size() == 1) {
                    a = lettersGrp.get(0).toCharArray();
                    b = new char[]{};
                } else {
                    a = lettersGrp.get(ii).toCharArray();
                    b = lettersGrp.get(jj).toCharArray();
                }
                String str = "";
                for (int i = 0; i < a.length; i++) {
                    for (int j = 0; j < b.length; j++) {
                        str = Character.toString(a[i]) + Character.toString(b[j]);
                        combinations.add(str);
                    }
                    if (b.length == 0)
                        combinations.add(Character.toString(a[i]));
                }
            }
        }
        return combinations;
    }
}
```

