Keypad Combinations

May 18, 2020

0.1 Keypad Combinations

A keypad on a cellphone has alphabets for all numbers between 2 and 9, as shown in the figure below:

You can make different combinations of alphabets by pressing the numbers.

For example, if you press 23, the following combinations are possible:

`ad, ae, af, bd, be, bf, cd, ce, cf`

Note that because 2 is pressed before 3, the first letter is always an alphabet on the number 2. Likewise, if the user types 32, the order would be

`da, db, dc, ea, eb, ec, fa, fb, fc`

Given an integer `num`, find out all the possible strings that can be made using digits of input `num`. Return these strings in a list. The order of strings in the list does not matter. However, as stated earlier, the order of letters in a particular string matters.

```
In [ ]: def get_characters(num):
            if num == 2:
                return "abc"
            elif num == 3:
                return "def"
            elif num == 4:
                return "ghi"
            elif num == 5:
                return "jkl"
            elif num == 6:
                return "mno"
            elif num == 7:
                return "pqrs"
            elif num == 8:
                return "tuv"
            elif num == 9:
                return "wxyz"
            else:
                return ""


        def keypad(num):

            # TODO: Write your keypad solution here!
```

```
        pass
```

Show Solution

```
In [ ]: def test_keypad(input, expected_output):
            if sorted(keypad(input)) == expected_output:
                print("Yay. We got it right.")
            else:
                print("Oops! That was incorrect.")
```

```
In [ ]: # Base case: list with empty string
        input = 0
        expected_output = [""]
        test_keypad(input, expected_output)
```

```
In [ ]: # Example case
        input = 23
        expected_output = sorted(["ad", "ae", "af", "bd", "be", "bf", "cd", "ce", "cf"])
        test_keypad(input, expected_output)
```

```
In [ ]: # Example case
        input = 32
        expected_output = sorted(["da", "db", "dc", "ea", "eb", "ec", "fa", "fb", "fc"])
        test_keypad(input, expected_output)
```

```
In [ ]: # Example case
        input = 8
        expected_output = sorted(["t", "u", "v"])
        test_keypad(input, expected_output)
```

```
In [ ]: input = 354
        expected_output = sorted(["djg", "ejg", "fjg", "dkg", "ekg", "fkg", "dlg", "elg", "flg",
        test_keypad(input, expected_output)
```