- **EASY**

- **Max Score: 30 Points**

-

-

-

-

Valid IP Address

Given a string containing only digits, restore it by returning all possible valid IP address combinations.

A valid IP address must be in the form of A.B.C.D, where A, B, C, and D are numbers from 0-255. The numbers cannot be 0 prefixed unless they are 0.

Input Format

The first line of the input contains the string s.

Output Format

The output should contain all possible IP addresses using the string s.

Example 1

Input

```
25525511135
```

Output:

```
255.255.11.135
255.255.111.35
```

Example 2

Input

```
15515511137
```

Output:

```
155.155.11.137
155.155.111.37
```

Constraints

1 <= s.length <= 12

s has only digits

My code

```java
// in java
import java.util.*;

class Solution{
    static void solve(String s, int i, int j, int level, String temp,ArrayList<String>res)
    {
```

```java
        if (i == (j + 1) && level == 5)
        {
                //that is not 5 part of ip address
                res.add(temp.substring(1));
        }
    if(level > 5)
          return;


        // Digits of a number ranging 0-255 can lie only between
        // 0-3
        for (int k = i; k < i + 3 && k <= j; k++)
                {
                  String ad = s.substring(i, k + 1);

                  if ((s.charAt(i) == '0'&& ad.length()>1 ) ||
Integer.valueOf(ad) > 255)
                        return;

                  // Recursively call for another level.
                  solve(s, k + 1, j, level + 1, temp + '.' + ad, res);
            }
}
    public static ArrayList<String> ValidIP(String s){
        //write code here
                int n = s.length();
            ArrayList<String> ans = new ArrayList<>();
              solve(s, 0, n - 1, 1, "", ans);
                return ans;
    }
}

public class Main {
```

```java
public static void main(String[] args) throws Throwable {
    Scanner sc = new Scanner(System.in);
    String s = sc.nextLine();
    Solution obj = new Solution();
    ArrayList<String> ans = obj.ValidIP(s);
    for(String i:ans){
        System.out.println(i);
    }
}
}
```