

## 牛客网-华为机试练习题 39

### 题目描述

子网掩码是用来判断任意两台计算机的IP地址是否属于同一子网络的根据。

子网掩码与IP地址结构相同，是32位二进制数，其中网络号部分全为“1”和主机号部分全为“0”。利用子网掩码可以判断两台主机是否中同一子网中。若两台主机的IP地址分别与它们的子网掩码相“与”后的结果相同，则说明这两台主机在同一子网中。

示例：

I P 地址    192.168.0.1  
子网掩码    255.255.255.0

转化为二进制进行运算：

I P 地址    11010000.10101000.00000000.00000001  
子网掩码    11111111.11111111.11111111.00000000

AND运算

11000000.10101000.00000000.00000000

转化为十进制后为：

192.168.0.0

I P 地址    192.168.0.254  
子网掩码    255.255.255.0

转化为二进制进行运算：

I P 地址    11010000.10101000.00000000.11111110  
子网掩码    11111111.11111111.11111111.00000000

AND运算

11000000.10101000.00000000.00000000

转化为十进制后为：

192.168.0.0

通过以上对两台计算机IP地址与子网掩码的AND运算后，我们可以看到它运算结果是一样的。均为192.168.0.0，所以这二台计算机可视为是同一子网络。

/\*

\* 功能：判断两台计算机IP地址是同一子网络。

\* 输入参数：     String Mask：子网掩码，格式：“255.255.255.0”；

\*                String ip1：计算机1的IP地址，格式：“192.168.0.254”；

\*                String ip2：计算机2的IP地址，格式：“192.168.0.1”；

\*

\\* 返回值：        0：IP1与IP2属于同一子网络；        1：IP地址或子网掩码格式非法；        2：IP1与IP2不属于同一子网络

\*/

```
public int checkNetSegment(String mask, String ip1, String ip2)
{
```

```
    /*在这里实现功能*/
```

```
        return 0;
    }
}
```

## 输入描述:

输入子网掩码、两个ip地址

## 输出描述:

得到计算结果

示例1

输入

255.255.255.0 192.168.224.256 192.168.10.4

输出

1

## 解决代码:

```
import java.util.Arrays;
import java.util.Scanner;

/**
 * 判断两个IP是否属于同一子网
 *
 * @author yuanyao
 */
public class Main {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        while (sc.hasNext()) {
            String[] mask = sc.next().split("\\.");
            String[] ip1 = sc.next().split("\\.");
            String[] ip2 = sc.next().split("\\.");

            for (int i = 0; i < 4; i++) {
                if (Integer.valueOf(mask[i]) < 0 || Integer.valueOf(mask[i]) > 255 ||
                    Integer.valueOf(ip1[i]) < 0
                        || Integer.valueOf(ip1[i]) > 255 || Integer.valueOf(ip2[i]) < 0
                        || Integer.valueOf(ip2[i]) >
255 || mask.length != 4 || ip1.length != 4 || ip2.length != 4) {
                    System.out.println(1); // IP地址或子网掩码格式非法
                    break;
                }
                else if ((Integer.valueOf(mask[i]) & Integer.valueOf(ip1[i])) ==
                    (Integer.valueOf(mask[i])
                        & Integer.valueOf(ip2[i]))) {
                    System.out.println(0); // IP1与IP2属于同一子网
                }
            }
        }
    }
}
```

```
        else {  
            System.out.println(2); // IP1与IP2不属于同一子网络  
            break;  
        }  
    }  
}  
}
```