

牛客网-华为机试练习题 18

题目描述

请解析IP地址和对应的掩码，进行分类识别。要求按照A/B/C/D/E类地址归类，不合法的地址和掩码单独归类。

所有的IP地址划分为 A,B,C,D,E五类

A类地址1.0.0.0~126.255.255.255；

B类地址128.0.0.0~191.255.255.255；

C类地址192.0.0.0~223.255.255.255；

D类地址224.0.0.0~239.255.255.255；

E类地址240.0.0.0~255.255.255.255

私网IP范围是：

10.0.0.0~10.255.255.255

172.16.0.0~172.31.255.255

192.168.0.0~192.168.255.255

子网掩码为二进制下前面是连续的1，然后全是0。（例如：255.255.255.32就是一个非法的掩码）

输入描述:

多行字符串。每行一个IP地址和掩码，用~隔开。

输出描述:

统计A、B、C、D、E、错误IP地址或错误掩码、私有IP的个数，之间以空格隔开。

示例1

输入

```
10.70.44.68~255.254.255.0
1.0.0.1~255.0.0.0
192.168.0.2~255.255.255.0
19..0.~255.255.255.0
```

输出

```
1 0 1 0 0 2 1
```

解决代码:

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

public class Main {
```

```

public static void main(String[] args)throws IOException{
    BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
    int a=0,b=0,c=0,d=0,e=0;
    int err=0;
    int pri=0;
    String str;
    String[] ip_mask;
    String[] ip;
    String[] mask;
    int i;
    while((str=br.readLine())!=null){
        ip_mask=str.split("~");
        ip=ip_mask[0].split("\\.");
        mask=ip_mask[1].split("\\.");
        //count error mask
        if(checkMask(ip_mask[1])){//mask correct
            //count ip
            if(checkIp(ip)){
                i=Integer.parseInt(ip[0]);
                if(i>=1&&i<=126){//a
                    a++;
                    if(i==10){
                        pri++;
                    }
                }else if(i>=128&&i<=191){//b
                    b++;
                    if(i==172&&Integer.parseInt(ip[1])>=16&&Integer.parseInt(ip[1])
<=31){
                        pri++;
                    }
                }else if(i>=192&&i<=223){//c
                    c++;
                    if(i==192&&Integer.parseInt(ip[1])==168){
                        pri++;
                    }
                }else if(i>=224&&i<=239){//d
                    d++;
                }else if(i>=240&&i<=255){//e
                    e++;
                }
            }else{
                err++;
            }
        }else{
            err++;
        }
    }
    //output
    System.out.println(a+" "+b+" "+c+" "+d+" "+e+" "+err+" "+pri);
}

public static boolean checkMask(String mask)
{
    //check mask
    String[] mask_arr=mask.split("\\.");
    if(mask_arr[0].equals("255"))
    {
        if(mask_arr[1].equals("255"))
        {
            if(mask_arr[2].equals("255"))

```

```

        {
            if(
mask_arr[3].equals("254") || mask_arr[3].equals("252") || mask_arr[3].equals("248") ||
mask_arr[3].equals("240") || mask_arr[3].equals("224") || mask_arr[3].equals("192") ||
mask_arr[3].equals("128") || mask_arr[3].equals("0"))
                return true;
            else
                return false;
        }
        else
if(mask_arr[2].equals("254") || mask_arr[2].equals("252") || mask_arr[2].equals("248") ||
mask_arr[2].equals("240") || mask_arr[2].equals("224") || mask_arr[2].equals("192") ||
mask_arr[2].equals("128") || mask_arr[2].equals("0"))
        {
            if(mask_arr[3].equals("0"))
                return true;
            else
                return false;
        }
        else
            return false;
    }
    else
if(mask_arr[1].equals("254") || mask_arr[1].equals("252") || mask_arr[1].equals("248") ||
mask_arr[1].equals("240") || mask_arr[1].equals("224") || mask_arr[1].equals("192") ||
mask_arr[1].equals("128") || mask_arr[1].equals("0"))
        {
            if(mask_arr[2].equals("0") && mask_arr[3].equals("0"))
                return true;
            else
                return false;
        }
        else
        {
            return false;
        }
    }
    else
if(mask_arr[0].equals("254") || mask_arr[0].equals("252") || mask_arr[0].equals("248") ||
mask_arr[0].equals("240") || mask_arr[0].equals("224") || mask_arr[0].equals("192") ||
mask_arr[0].equals("128") || mask_arr[0].equals("0"))
        {
            if(mask_arr[1].equals("0") && mask_arr[2].equals("0") && mask_arr[3].equals("0"))
                return true;
            else
                return false;
        }
        else
        {
            return false;
        }
    }
}

```

```
static boolean checkIp(String[] ip){  
    if(ip.length==4&&!ip[0].equals("")&&!ip[1].equals("")&&!ip[2].equals("")&&!ip[3].equals(""))  
    {  
        return true;  
    }  
    return false;  
}  
}
```

总结:

- BufferedReader br=new BufferedReader(new InputStreamReader(System.in));是读取数据的另外一种形式;