

题目描述：

存在一种虚拟 IPv4 地址，由 4 小节组成，每节的范围为 0~255，以 # 号间隔，虚拟 IPv4

地址可以转换为一个 32 位的整数，例如：

128#0#255#255，转换为 32 位整数的结果为 2147549183 (0x8000FFFF)

1#0#0#0，转换为 32 位整数的结果为 16777216 (0x01000000)

现以字符串形式给出一个虚拟 IPv4 地址，限制第 1 小节的范围为 1~128，即每一节范围分别为 (1~128)#(0~255)#(0~255)#(0~255)，要求每个 IPv4 地址只能对应到唯一的整数上。如果是非法 IPv4，返回 invalid IP

输入描述：

输入一行，虚拟 IPv4 地址格式字符串

输出描述：

输出以上，按照要求输出整型或者特定字符

补充说明：

输入不能确保是合法的 IPv4 地址，需要对非法 IPv4（空串，含有 IP 地址中不存在的字符，非合法的 # 分十进制，十进制整数不在合法区间内）进行识别，返回特定错误

示例 1

输入：

100#101#1#5

输出：

1684340997

说明：

示例 2

输入:

1#2#3

输出:

invalid IP

```
import java.util.*;
```

```
public class Main {
```

```
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        String ips = in.nextLine();
        if ("".equals(ips) || ips.length() == 0) {
            System.out.println("invalid IP");
            return;
        }
        String[] ip = ips.split("#");
        if (ip.length != 4) {
            System.out.println("invalid IP");
            return;
        }
        for (int i = 0; i < 4; i++) {
            int l = ip[i].length();
            if (l == 0 || l > 3) {
                System.out.println("invalid IP");
                return;
            }
            for (int j = 0; j < l; j++) {
                if (!Character.isDigit(ip[i].charAt(j))) {
                    System.out.println("invalid IP");
                    return;
                }
            }
            if (ip[i].charAt(0) == '0' && l != 1) {
                System.out.println("invalid IP");
                return;
            }
        }
        if (Integer.parseInt(ip[0]) < 1 || Integer.parseInt(ip[0]) > 128
            || Integer.parseInt(ip[1]) > 255
            || Integer.parseInt(ip[2]) > 255
            || Integer.parseInt(ip[3]) > 255) {
            System.out.println("invalid IP");
        } else {
```

```
        System.out.println((Long.parseLong(ip[0]) << 24) + (Long.parseLong(ip[1]) << 16)
            + (Long.parseLong(ip[2]) << 8) + Long.parseLong(ip[3]));
    }
}
```