

Python-字符串进制转换-存在一种虚拟 IPv4 地址

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

说明:

```
import sys
```

```
a = sys.stdin.readline().strip().split('#')
```

```
if len(a)!=4:
```

```
    print('invalid IP')
```

```
else:
```

```
    nums = ['0','1','2','3','4','5','6','7','8','9']
```

```
    istrue = True
```

```
    if not a[0]:
```

```
        istrue = False
```

```
    for num in a[0]:
```

```
        if num not in nums:
```

```
            istrue =False
```

```
    if istrue:
```

```
        c = int(a[0])
```

```
        if c>128:
```

```
        istrue = False
for i in range(1,4):
    if not a[i]:
        istrue = False
    for num in a[i]:
        if num not in nums:
            istrue =False
    if istrue:
        c = int(a[0])
        if c>255:
            istrue = False
if istrue:
    ip = lambda x:sum([256**j*int(i) for j ,i in enumerate(x[::-1])])
    print(ip(a))
else:
    print('invalid IP')
```