

响应报文时间

题目描述：

IGMP 协议中， 有一个字段称作最大响应时间（Max Response Time）， HOST 收到查询报文， 解析出 MaxResponseTime 字段后， 需要在  $(0, \text{MaxResponseTime}](s)$  时间内选取随机时间回应一个响应报文,如果在随机时间内收到一个新的查询报文,则会根据两者时间的大小,选取小的一方刷新回应时间。

最大响应时间有如下计算方式：

当  $\text{Max Resp Code} < 128, \text{Max Resp Time} = \text{Max Resp Code};$

当  $\text{Max Resp Code} \geq 128,$

0 1 2 3 4 5 6 7

+--+--+--+--+--+

| 1 | exp | mant |

+--+--+--+--+--+

$\text{Max Resp Time} = (\text{mant} | 0 \times 10) \ll (\text{exp} + 3);$

注： exp 最大响应时间的高 5~7 位； mant 为最大响应时间的低 4 位。

其中 接收到的 MaxRespCode 最大值为 255， 以上出现所有字段均为无符号数。

现在我们认为 HOST 收到查询报文时， 选取的随机时间必定为最大值。 现给出 HOST 收到查询报文个数 C， HOST 收到该报文的时间 T， 以及查询报文的最大响应时间字段值 M， 请计算出 HOST 发送响应报文的时间。

输入： 第一行为查询报文个数 C， 后续每行分别为 HOST 收到报文时间 T， 及最大响应字段 M， 以空格分割。

输出： HOST 发送响应报文的时间

输入描述：

第一行为查询报文个数 C， 后续每行分别为 HOST 收到报文时间 T， 及最大响应时间 M， 以空格分割。

输出描述：

HOST 发送响应报文的时间。

补充说明：

用例确定只会发送一个响应报文， 不存在计时结束后依然收到查询报文的情况。

示例 1

输入：

3

0 20

1 10

8 20

输出：

11

说明：

收到 3 个报文，

第 0 秒收到第 1 个报文，响应时间为 20 秒，则要到  $0+20=20$  秒响应；

第 1 秒收到第 2 个报文，响应时间为 10；则要到  $1+10=11$  秒响应，与第上面的报文的响应时间比较获得响应时间最小为 11 秒；

第 8 秒收到第 3 个报文，响应时间为 20 秒，则要到  $8+20=28$  秒响应；与第上面的报文的响应时间比较获得响应时间最小为 11 秒；

最终得到最小响应报文时间为 11 秒

示例 2

输入：

2

0 255

200 60

输出：

260

说明：

收到 2 个报文，

第 0 秒收到第 1 个报文，响应时间为 255 秒，则要到  $(15 | 0 \times 10) \ll (7 + 3) = 31744$  秒响应；（ $\text{mant} = 15$ ， $\text{exp} = 7$ ）

第 200 秒收到第 2 个报文，响应时间为 60；则要到  $200+60=260$  秒响应，与第上面的报文的响应时间比较获得响应时间最小为 260 秒；

最终得到最小响应报文时间为 260 秒

```
def transfer_num(m):
```

```
    exp = (m & 0x70) >> 4
```

```
    mant = (m & 0x0F)
```

```
    return (mant | 0x10) << (exp + 3)
```

```
def get_time(t_time_arr):
```

```
    min_time = 100000
```

```

n = len(t_time_arr)

for t_m_time in t_time_arr:
    if t_m_time[1] < 128:
        min_time = min(min_time, t_m_time[0]+t_m_time[1])
    else:
        m = t_m_time[1]
        min_time = min(min_time, t_m_time[0] + transfer_num(m))

return min_time

c_num = int(input())
t_time_arr = []

for i in range(c_num):
    t_time_arr.append(list(map(int,input().split())))

print(get_time(t_time_arr))

```