☑ PAT 甲级题目讲解：1008《Elevator》

🧩 题目简介

本题模拟的是一个城市中只有一台电梯，它根据乘客的请求依次停靠多个楼层。电梯从 0 层出发，按照给定的楼层顺序依次访问目标楼层。

电梯移动规则如下：

- 每上升一层楼耗时 **6 秒**；
- 每下降一层楼耗时 **4 秒**；
- 每次到达一个目标楼层，都会 **停留 5 秒**；
- 初始位置固定为 **第 0 层**，最终不需返回。

🧪 样例分析

输入样例：

```
3 2 3 1
```

含义：

- 请求数量：3 次；
- 依次前往楼层：2 → 3 → 1；
- 初始楼层为 0；

模拟过程如下：

1. 从 0 → 2：上升 2 层，耗时 $2 \times 6 = 12$ 秒 + 停留 5 秒；
2. 从 2 → 3：上升 1 层，耗时 6 秒 + 停留 5 秒；
3. 从 3 → 1：下降 2 层，耗时 $2 \times 4 = 8$ 秒 + 停留 5 秒；

总时间为：

$$12 + 5 + 6 + 5 + 8 + 5 = 41$$

输出：

```
41
```

🔍 解题思路

📌 电梯行为模拟 + 简单加法累加

本题是典型的“按规则模拟过程”的题目，只需逐步累加时间开销即可。

🧷 变量说明

变量名	含义
`n`	请求次数（要访问的楼层个数）
`t`	当前请求楼层
`s`	总时间累计值
`p`	当前电梯所在楼层（初始为 0）

☑ Step 1：读入数据与初始化

```
cin >> n;
p = 0; // 初始电梯位置为 0 层
```

☑ Step 2：逐个处理请求楼层

```
while(n--){
    cin >> t;
    if(t > p){
        s += (t - p) * 6; // 上升
    }
    else if(t < p){
        s += (p - t) * 4; // 下降
    }
    s += 5; // 到达后停留 5 秒
    p = t;  // 更新当前位置
}
```

☑ Step 3：输出结果

```
cout << s;
```

☑ 完整代码

```
#include <bits/stdc++.h>
using namespace std;

int n, t, s, p;

int main(){
    cin >> n;
    p = 0; // 初始在 0 层
    while(n--){
        cin >> t;
        if(t > p){
            s += (t - p) * 6;
        }
        else if(t < p){
```

```
            s += (p - t) * 4;
        }
        s += 5; // 每到一层需停留 5 秒
        p = t;  // 更新电梯当前位置
    }
    cout << s;
    return 0;
}
```

常见错误提醒

错误类型	错误表现
忘记初始化楼层位置	忘记设置起始楼层为 0，导致初始偏移错误
停留时间未加	每次到达目标楼层都需 +5 秒
上升/下降耗时搞反	上升应乘 6，下降应乘 4，不能写错
忘记更新当前楼层 `p`	每次处理完时间计算要更新当前楼层 `p` 为本次需到达的 `t`

总结归纳

- 本题为**电梯规则模拟题**，不涉及算法难度，关键是认真理解题意模拟；
- 逐层判断电梯上升或下降，并累加相应时间；
- 注意初始化当前楼层为 0，过程中要逐次更新所在楼层数，最终只需输出总耗时。

复杂度分析

- 时间复杂度：$\mathcal{O}(n)$（一次遍历楼层请求）
- 空间复杂度：$\mathcal{O}(1)$（仅使用常数变量）

思维拓展

- 如果电梯支持同时处理多个请求（非顺序），如何最优调度？
- 若加上“不同乘客等待时间”的优化目标，会变成经典的调度算法问题；
- 本题思想可类比：打印任务队列、线程任务模拟、电梯调度系统设计等场景。