☑ PAT 乙级题目讲解：1014《福尔摩斯的约会》

🧩 题目简介

福尔摩斯收到一张奇怪的字条，写着乱码一般的四行字符。作为一名侦探，他很快发现这些乱码其实是一次约会的时间信息。

我们的任务是根据字符间的相同规则，解码出正确的星期、小时和分钟。规则如下：

星期：前两个字符串中第一个相同的**大写字母**，第几个字母就表示星期几；

小时：前两个字符串中第一个相同的大写字母**后**出现的第一对**相同字符**，用 `'0'` ~ `'9'` 表示 0 ~ 9 点；`'A'` ~ `'N'` 表示 10 ~ 23 点；

分钟： 后面两字符串第 1 对相同的**英文字母**出现的位置下标（从 0 开始计）。

🩹 样例分析

输入示例：

```
3485djDkxh4hhGE
2984akDfkkkkggEdsb
s&hgsfdk
d&Hyscvnm
```

分析过程：

1. 前两个字符串中第一个相同的大写字母是 `'D'`，表示星期四（THU）；
2. 前两个字符串中紧接着第二个相同的字符是 `'E'`，表示 14 点；
3. 后面两字符串第一对相同的英文字母是 `'s'`，在索引 4（从 0 开始计），表示第 4 分钟。

输出结果为：

```
THU 14:04
```

🔍 解题思路

🪄 变量说明

变量名	含义
a, b	前两行字符串，用于确定星期和小时
c, d	后两行字符串，用于确定分钟
day	表示星期几（1~7），通过字符 'A'~'G' 映射
h	小时
m	分钟
w	存储星期字符串的数组，下标对应 day 值

☑ Step 1：读入输入字符串

使用 `cin` 读取 4 行字符串，分别赋值给变量 `a, b, c, d`。

```
cin >> a >> b >> c >> d;
```

☑ Step 2：匹配星期几（前两行字符串第一个相同的大写字母）

从左往右遍历 `a` 和 `b`，找到第一对相同的大写英文字母（范围 `'A'`~`'G'`），转换为星期几。

```
for(i = 0; i < a.size(); i++){
    if(a[i] == b[i] && (a[i] >= 'A' && a[i] <= 'G')){
        day = a[i] - 'A' + 1;
        break;
    }
}
```

☑ Step 3：匹配小时（前两行字符串从上一步的下一个字符开始）

从上一次匹配的位置继续找第二对相同的字符，要求字符为：

- `'0'` ~ `'9'` 表示 0 ~ 9 点；
- `'A'` ~ `'N'` 表示 10 ~ 23 点。

```
i++; // 从下一个位置开始找
for(; i < a.size(); i++){
    if(a[i] == b[i] && (a[i] >= '0' && a[i] <= '9' || a[i] >= 'A' && a[i] <=
'N')){
        if(a[i] >= '0' && a[i] <= '9') h = a[i] - '0';
        else h = a[i] - 'A' + 10;
        break;
    }
}
```

☑ Step 4：匹配分钟（后两行字符串中第一个相同英文字母）

遍历字符串 `c` 和 `d`，找出第一个匹配的英文字母字符，其位置索引即为分钟。

```
for(int i = 0; i < c.size(); i++){
    if(c[i] == d[i] && isalpha(c[i])){
        m = i;
        break;
    }
}
```

☑ Step 5：格式化输出结果

使用星期数组 `w[day]` 输出星期，小时和分钟使用 `%02d` 保证补零格式。

```
cout << w[day] << " ";
printf("%02d:%02d", h, m);
```

☑ 完整代码

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

string a, b, c, d;
int day, h, m;
string w[10] = {"", "MON", "TUE", "WED", "THU", "FRI", "SAT", "SUN"};

int main(){
    cin >> a >> b >> c >> d;

    // 1. 匹配星期几
    int i;
    for(i = 0; i < a.size(); i++){
        if(a[i] == b[i] && (a[i] >= 'A' && a[i] <= 'G')){
            day = a[i] - 'A' + 1;
            break;
        }
    }

    // 2. 匹配小时
    i++;
    for(; i < a.size(); i++){
        if(a[i] == b[i] && (a[i] >= '0' && a[i] <= '9' || a[i] >= 'A' && a[i] <=
'N')){
            if(a[i] >= '0' && a[i] <= '9') h = a[i] - '0';
            else h = a[i] - 'A' + 10;
            break;
        }
    }

    // 3. 匹配分钟
    for(int i = 0; i < c.size(); i++){
        if(c[i] == d[i] && isalpha(c[i])){
            m = i;
            break;
        }
    }
```

```
    // 4. 输出结果
    cout << w[day] << " ";
    printf("%02d:%02d", h, m);

    return 0;
}
```

常见错误提醒

错误类型	说明
星期字母范围错误	仅限 `'A'`~`'G'`，不是所有大写字母都合法
小时字符范围误判	`'0'`~`'9'` 与 `'A'`~`'N'` 才合法
遍历位置未正确推进	匹配第 2 个字符前需 `i++` 移动位置
输出格式错误	小时/分钟需补零，使用 `%02d`

总结归纳

- 本题考查字符串逐位比较与格式化输出；
- 理解每组字符的映射含义是关键；
- 索引管理与字符范围判断尤为重要；
- **时间复杂度**：$O(n)$
- **空间复杂度**：$O(1)$

思维拓展

- 属于典型的“字符串逐位匹配 + 条件判断解码”题型；
- 使用 `w[day]` 进行【映射解码】是常见解题技巧；
- 与身份证号码校验、协议解析类问题具有结构类似，可作为同类题训练模板；