

# Solution 8.6

## T1 时间

给出一个时间，求出在 24 小时制下的 3 小时 30 分后的时间为多少。直接模拟即可，注意两天间时间的变化。

具体可以把时间都换算成分钟，然后对一天的分钟数取模，然后再换算成标准时间。

【核心代码】

```
scanf("%d:%d", &h, &m);
m = h * 60 + m;
m += 3 * 60 + 30;
m %= 24 * 60;
h = m / 60, m = m % 60;
printf("%02d:%02d\n", h, m);
```

## T2 玩具

我们考虑将旋转操作分为两类：

第一类是在边上旋转，如样例中的第一次旋转；

第二类是在角上旋转，如样例中的第二次旋转。

然后分别讨论两类旋转的次数，求和输出即可。

【核心代码】

```
ll a, m, b, n;
while (cin >> a >> m >> b >> n) {
    ll x = a * (n * b) / __gcd(a, n * b);
    ll c1 = x / a;
    ll y = a * b / __gcd(a, b);
    ll c2 = x / y * (y / b - 1);
    cout << c1 + c2 << '\n';
}
```

## T3 巨炮的重生

### 一句话题意

有一个环，每次可以

- 合并相邻两数  $A, B$ ，得到的新数  $A + B$
- 合并相邻三数  $A, B, C$ ，得到的新数  $A \times C - B$

求最终剩下的数的最大值。

- $T \leq 50$
- $n \leq 35$
- $Ans \leq 2^{63} - 1$

### 算法一（前三个点）：手玩

$n = 1$

.....

$n = 2$

A+B 问题!

$n = 3$

$Ans = \max(A + B + C, A \times B - C, A \times C - B, B \times C - A)$

$n = 4 / n = 5$

把所有情况都枚举出来就好了！反正一共才几十种。

### 算法二（前五个点）：暴力

DFS。

枚举合并的位置进行操作。

环长为  $n$  的时候有  $O(n)$  种合并的可能，所以复杂度是  $O(n!)$  级别。

### 算法三（所有点）：标算

区间 DP。

这么明显的“相邻”“合并”还想不到区间 DP 吗

先破坏成链，把环复制一份，然后直接在  $2n$  的序列上做区间 DP。

我们看到了减法，还看到了有正有负..... 所以要维护最大和最小值了。

设  $f[l][r]$  = 将  $[l, r]$  合并为一个数的最大值， $g[l][r]$  = 将  $[l, r]$  合并为一个数的最小值

$f$  的转移有 3 种：

$f[l][k] + f[k+1][r]$  (合并两个)

$f[l][i] \times f[j][r] - g[i+1][j-1]$

$g[l][i] \times g[j][r] - g[i+1][j-1]$  (合并三个，最大  $\times$  最大 / 最小  $\times$  最小。注意减最小值才是最大值)

$g$  的转移有 5 种：最大  $\times$  最大，最小  $\times$  最小，最大  $\times$  最小，最小  $\times$  最大都有可能成为最小值。减的时候要减最大值。

本题数据比较水， $g$  只用最大  $\times$  最小和最小  $\times$  最大转移也可以做。

最终答案就是  $\max(f[1][n], f[2][n+1], \dots, f[n][2 * n - 1])$ 。

时间复杂度： $O(n^4)$

## T4 取石子游戏

结论：当且仅当石子堆数为 1 堆，且石子数量为偶数时，陈老师有必胜策略，否则一定是藤藤获胜。

证明：藤藤的策略为：若场上存在偶数堆石子，那么将这一堆取走奇数个，使其变成奇数堆石子；否则，取走一整堆石子（因为此时每一堆石子数量均为奇数）。由此可见，无论陈老师如何操作，场上的偶数石子堆的数量一直减少，因而无法使得 藤藤 无法操作。