# Day 26: Nested Logic

**Objective**

Today's challenge puts your understanding of nested conditional statements to the test. You already have the knowledge to complete this challenge, but check out the Tutorial tab for a video on testing.

**Task**

Your local library needs your help! Given the expected and actual return dates for a library book, create a program that calculates the fine (if any). The fee structure is as follows:

1. If the book is returned on or before the expected return date, no fine will be charged (i.e.: $fine = 0$).
2. If the book is returned after the expected return *day* but still within the same calendar month and year as the expected return date, $fine = 15 \text{ Hackos} \times (\text{the number of days late})$.
3. If the book is returned after the expected return *month* but still within the same calendar year as the expected return date, the $fine = 500 \text{ Hackos} \times (\text{the number of months late})$.
4. If the book is returned after the calendar *year* in which it was expected, there is a fixed fine of $10000 \text{ Hackos}$.

**Example**

$d1, m1, y1 = 12312014$ returned date

$d2, m2, y2 = 112015$ due date

The book is returned on time, so no fine is applied.

$d1, m1, y1 = 112015$ returned date

$d2, m2, y2 = 12312014$ due date

The book is returned in the following year, so the fine is a fixed 10000.

**Input Format**

The first line contains $3$ space-separated integers denoting the respective $day$, $month$, and $year$ on which the book was *actually* returned.

The second line contains $3$ space-separated integers denoting the respective $day$, $month$, and $year$ on which the book was *expected* to be returned (due date).

**Constraints**

- $1 \leq D \leq 31$
- $1 \leq M \leq 12$
- $1 \leq Y \leq 3000$
- It is guaranteed that the dates will be valid Gregorian calendar dates.

**Output Format**

Print a single integer denoting the library fine for the book received as input.

**Sample Input**

```
STDIN       Function
-----       --------
9 6 2015    day = 9, month = 6, year = 2015 (date returned)
6 6 2015    day = 6, month = 6, year = 2015 (date due)
```

**Sample Output**

```
45
```

**Explanation**

Given the following return dates:

Returned: $D_1 = 9, M_1 = 6, Y_1 = 2015$

Due: $D_2 = 6, M_2 = 6, Y_2 = 2015$

Because $Y_2 \equiv Y_1$, it is less than a year late.

Because $M_2 \equiv M_1$, it is less than a month late.

Because $D_2 < D_1$, it was returned late (but still within the same month and year).

Per the library's fee structure, we know that our fine will be $15 \text{ Hackos} \times (\# \text{ days late})$. We then print the result of $15 \times (D_1 - D_2) = 15 \times (9 - 6) = 45$ as our output.