

**1999-2000 ACM International Collegiate Programming Contest**  
**Asia Regional Contest / Shanghai**  
**Shanghai Univeristy, Shanghai, China, November 28, 1999**

## Problem H

### How Many Days Are There

Input file: days.in

“Unfortunately, I can not attend the party in the 10th this month,” a friend said to me, “ You know, only Thursday is my holiday. Although I’m longing for the party, I’m sorry that I will not be present then.”

“It doesn’t matter. I hope that you will be present the next time.”

.....

Have you ever encounter the situation like that? No matter how regretful you feel about it, still you could do nothing. So, we desire to find a way to know in a period how manys days of which not only the date but the day of week conform to our demand there are.

In this problem, we assume that we use a special calendar from ancient to modern times. In the calendar, we stipulate as follows:

0. In the calendar, there isn’t year 0, and we use 1 year B.C. instead. And also we assume that 1 year B.C. is a leap year. In this problem, 0 means 1 year B.C. and -1 means 2 years B.C., ..., and so on.
1. In the calendar, Jan. 1st, 1 year A.D. is Monday, the following days are Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday in proper order. And then from Monday to Sunday to keep on recurring.
2. If a certain year is divisible by 4 and not divisible by 100, then it’s a leap year. But if it is divisible by 400, it’s a leap year too.
3. In a year there are 12 months. In January, March, May, July, August, October and December there are 31 days each. In April, June, September, November there are 30 days each. And in February, if the year is a leap year, there are 29 days, otherwise there are 28 days.
4. Your program should deal with the date from 2000000 year B.C. to 2000000 A.D..

**1999-2000 ACM International Collegiate Programming Contest**  
**Asia Regional Contest / Shanghai**  
**Shanghai Univeristy, Shanghai, China, November 28, 1999**

### **Input**

The input file for this problem will contains a series of data sets. Each data set will begin with two integers. The first integer  $m$  ( $0 \leq m \leq 6$ ) is the day of week in demand. If  $m = 0$  represents the day of week is Sunday, and then  $m = 1$  represents Monday,  $m = 2$  represents Tuesday, ...,  $m = 6$  represents Saturday. The second integer  $n$  ( $0 \leq n \leq 31$ ) is the day part of the date in demand. The end of input is signified by the data set with  $m = 0$  and  $n = 0$ . And this data set should not be processed. If the data set doesn't indicate the end of the input, there will be 6 integers  $y_1, m_1, d_1, y_2, m_2, d_2$ , represent the start date and the end date. And you should process between the start date and the end date. Among these integers,  $y_1, m_1, d_1$  are the year part, the month part, the day part of the start date, and  $y_2, m_2, d_2$  are the year part, the month part, the day part of the end date. You may assume that the start date and the end date conform to the calendar. The end date will not be earlier than the start date.

### **Output**

The output for each data set should include a line contains a integer which tells from the start date to the end date ( including the start date and the end date ) how many days conform to our demands. That is, if the day of week of a special date equals to  $m$  and the day part of this date equals to  $n$ , we say the date conform to our demands. You should not print any more whitespaces or blank lines in the output.

### **Sample Input**

```
0 28 1999 11 18 1999 11 28
5 28 1999 11 18 1999 11 28
5 13 1999 1 1 1999 12 31
0 30 1976 6 30 1999 11 28
0 0
```

### **Output for the Sample Input**

```
1
0
1
35
```