The secret Swamp County Revolutionary Council subcommittee on The Calendar has finally, after more than 20 years, agreed on the calendar system which will be used after the revolution. Your team's job is to prepare the Council-blessed subroutine to calculate the number of days between any two dates in the new system. Rather than have negative or Before Revolution dates, the old Gregorian calendar will remain in use for pre-revolutionary dates. Since the Revolution has not yet occurred, and the day it does will be the first day of the new year, you will not have to deal with any old dates for now.

The year has 13 months of 28 days, a 1 day month for the end of year holiday, and an extra 1 day leap year month added between the end of year month and the first day of the next year when necessary.

The 28 day months, in order, are: Alligator, Bog, Crayfish, Damp, Eel, Fen, Gumbo, Hurricane, Inundation, Jaguar, Kudzu, Lake, and Marsh. The days of these months are numbered 0 through 27. The day and month are cycled *simultaneously,* similar to the Mayan tzolkin calendar. The first day of the year is 0-Alligator, followed by 1-Bog, 2-Crayfish, ..., 12-Marsh, 13-Alligator, and so on. Since there are 13 of these months, and 13 is prime, all combinations of days and months are used in $13 \cdot 28 = 364$ days.

The end of year month, Newt, has only 1 day: 0.

There are 683 leap years in a 2820 year cycle, somewhat like the modern Persian calendar. Year y will be a leap year if $((683 \cdot y) \bmod 2820) < 683$. Year 0 will be a leap year, as will years 5, 9, 13, 17, and so on. There are 3 or 4 standard years between each leap year. Using this method for leap years will result in an error of 1 day in 2.39 million years. While a Y2.39M committee is being formed to address this problem, just ignore it for now.

The leap year month is Overflow, the day number is in the range 0-682 with the first leap year in a 2820 year cycle being 0-Overflow and the last being 682-Overflow. For example, 0-Overflow-0, 1-Overflow-5, 2-Overflow-9, 0-Overflow-2820 are valid dates, 3-Overflow-2829 is not.

Input

The input consists of a series of lines ended by end-of-file. Each line consists of a pair of dates in the form *day*–*month*–*year*, with *day* being at most 3 decimal digits, *month* being an alphabetic string of at most 10 characters, and *year* being in the range 0-1999999. The dates of a pair will be separated by one or more spaces or tabs. Accept *month* if it is any unique initial substring of a valid month name ignoring case.

Output

For each test case print the date difference between the dates of the pair. The date difference is the number of days that need to be added to the earlier date to reach the later date. The date difference between *today* and *today* is 0, between *today* and *tomorrow* is 1.

If either date of the pair is illegal print the official Revolutionary Council Provisional Error Message: 'eh?'.

Print the output for each test case on a separate line starting in the first column.

Sample Input

```
0-Alli-0 2-Overflow-2829
0-alli-0 3-Over-2829
0-a-0 0-newt-1999999
0-alligator-1997 28-bog-2000
16-jaguar-1054284 16-gum-19973
23-damp-394304 0-newt-590355
43-over-1948798 20-mar-1058036
```

Sample Output

```
1033635
eh?
730484396
eh?
377773883
71606355
325343964
```