Dates

A colleague, who has now left your company for a better-paid job, had just been given a data-processing task which now falls into your hands. The program is intended to perform the following:

1. Read a line of input and determine whether or not it is a valid date between the years 1753 and 3000.
2. Either
   - state that the input date is invalid and why it is invalid

     e.g.,

     ```
     03 JUN 3004 - INVALID: Year out of range.
     ```

   or
   - output a valid date in the following format:

     dd<*space*><*first three chars of month name*><*space*>yyyy

     e.g.,

     ```
     02 Apr 1996
     ```

Valid input:

Input dates can be presented in different formats and in different orders. For our purposes, dates in the following order will be considered valid:

day month year

and acceptable input may be in any of the following formats

day: dd *or* d *or* 0d

month: mm *or* m *or* 0m *or* the first three letters of the month name (all in the same case, or with the first letter upper-case)

year: yy *or* yyyy

separator: - *or* / *or* <space>
Note: only *one* separator type to be used in one date

Example dates following these specifications are:

```
4-6-92
04/06/1992
3 AUG 97
12-Sep-1955
```

Notes:

1. 29^{th} of February is only considered a valid date in leap years.
2. If the year is written with only two digits, the date lies between 1950 and 2049, so 65 means 1965 and 42 means 2042.

Task

Write a computer program that runs according to the specifications above. Each line of input from `stdin` should be processed and appropriate output written to `stdout`. Lines representing valid dates must be output in the format specified above. Invalid lines should be echoed, followed by `- INVALID:` and then a diagnostic for the reason if it can be determined. One possible diagnostic is `UNKNOWN REASON` but, where it's easy to determine the problem, e.g., year out of range as in the example above, then that should be noted. You should test the program thoroughly.

Relates to Objectives

1.1, 1.2, 1.3, 2.2, 2.7, 2.8, 3.4, 3.5, 4.1, 4.5

(1 point, Individual)