SAMPLE PROBLEM

Section 1: Description

Many automated systems represent dates using all numbers; instead of saying 'March 5', they just say '3/5'. This has several problems, one of which is that in many countries they put the day first: that date would be '5/3', and would be written out as '5 March'.

Your sample problem is to read in a five-character string representing a date, and then a single character (**a** or **e**) indicating American or European dating. Based on this input, you should print out the date as a word and a number.

Section 2: Problem Specification

Input

The input, typed from the keyboard, will be in the following format:

1. A single integer, **N**, indicating how many dates will appear.
2. **N** lines with the dates to be read and how they should be interpreted. Each line will have the following fields:
   (a) The date, as a five-character string in the form `NN/NN`. This string will always be exactly 5 characters: 2 digits, a slash, and 2 digits.
   (b) One of the letters **a** or **e**, indicating an American or European style date, respectively.

If the proposed interpretation is impossible because the month would be greater than 12, print 'impossible'. You do *not* have to check that the number of days is legal for the given month; if the input maps to February 45, just print 'February 45'.

If you are using a GUI-based programming environment such as Visual Basic, you may write your program to read from a single text box, or multiple text boxes, or some other obvious mechanism to get the input into the program, and then do the conversion when a button is clicked. You do not have to worry about reading in the number of sample dates.

Output

For each line read, your program should print the input date exactly as it appeared, one of the words 'American' or 'European', as appropriate, and then the date as a word and a number (or a number and a word, if European style). Here are two sample lines to indicate the output format:

```
03/05 - American - March 5
03/05 - European - 3 May
```

Section 3: Sample Input / Output

1. Input:

```
6
03/05 e
03/05 a
09/09 e
09/09 a
12/25 a
17/03 a
```

2. Output

```
03/05 - European - 3 May
03/05 - American - March 5
09/09 - European - 9 September
09/09 - American - September 9
12/25 - American - December 25
17/03 - American - impossible
```

Section 3: Test Data

Run your program on the following data and take a screenshot to show the output is correct:

```
3
04/01 a
03/21 a
03/21 e
```