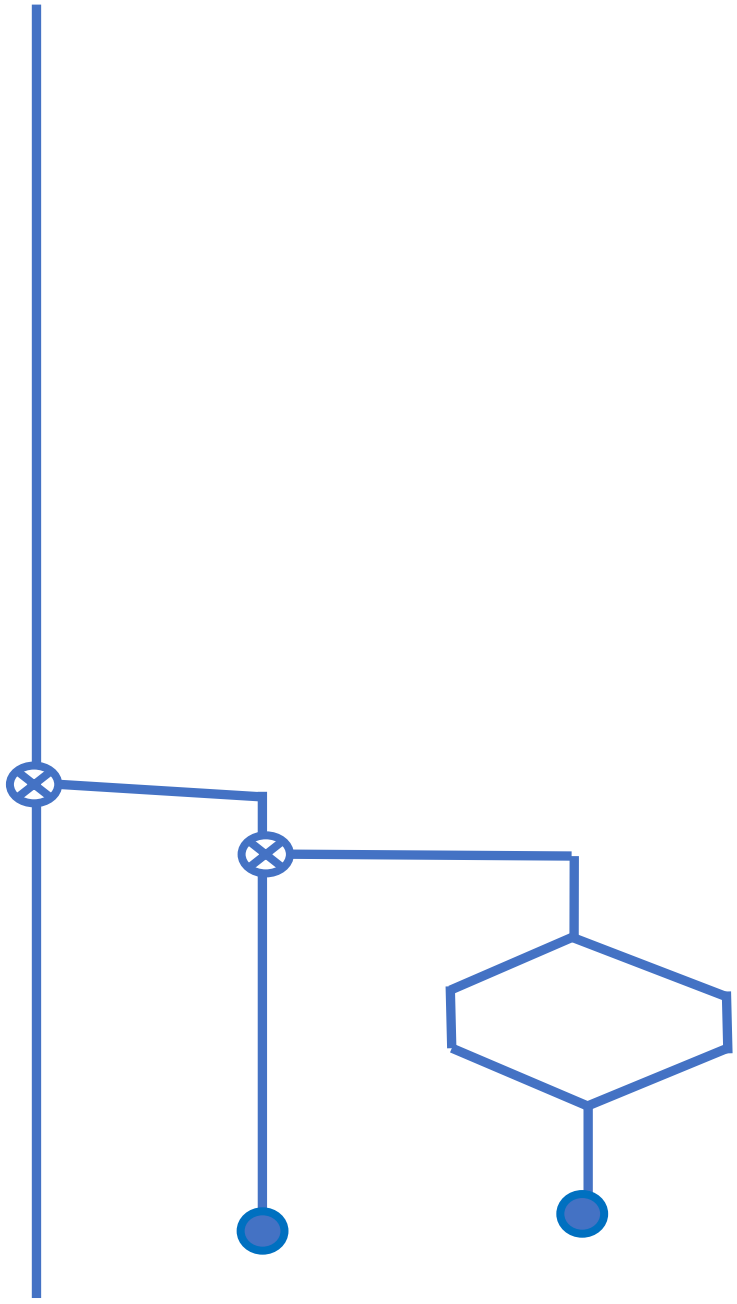

Forth Understands Leap Year

SVFIG

Jan. 22, 2022

Bill Ragsdale

The Challenge

Determine if a year is a leap year.

Extra credit: Accept the year as a Roman numeral. i.e. MMXXII

The Simplest Form

Petremann Marc‘s solution

```
: leap? ( year -- flag )
  year 400 mod 0=
  year 100 mod 0=  invert
  year   4 mod 0=  and or  ;
```

My Approach

1. Inside every small problem there resides a big problem to be solved.

My Approach

1. Inside every small problem there resides a big problem to be solved.
2. Structure as a prototype for a general AI solution.

My Approach

1. Inside every small problem there resides a big problem to be solved.
2. Structure as a prototype for a general AI solution.
3. Each word does 'exactly' one task.

My Approach

1. Inside every small problem there resides a big problem to be solved.
2. Structure as a prototype for a general AI solution.
3. Each word does 'exactly' one task.
4. Report in plain language.

The Process

Accept the year number on the stack

Classify to one of all possible outcomes.

Use a CASE statement to select.

Format for a clear report.

```
:  answer  dup cr .
     LYprocess  selection ;
```

The Classifier Output

If divisible by   4, output 1 (LSB)

If divisible by 100, output 2 (next bit)

If divisible by 400, output 4 (next bit)

‘or’ all the outputs into one, 3 bit result.

Possible results

0, 1, 3, 7, otherwise an error

The Classifier

If a letter exists, or its weight into the stack value ‘class’.

```
: classifier
\ ( year class divisor weight --- year class )
   >r  2 pick swap  mod  0=  \  year class boolean
   negate                   \  year class 1/0
   r>  *  or ;              \  year new-class
```

Words That Classify

```
: a-class
    create  , ,    does> 2@ classifier  ;


  4 1  a-class    4/classify   \ year/4
100 2  a-class  100/classify   \ year/100
400 4  a-class  400/classify   \ year/400
```

Apply All Classifiers

```
: LYprocess ( year --- class )
    0  4/classify   100/classify  400/classify
       nip ;

Possible results
0,  1,  3,  7,
```

Apply The Selection Template

```
: selection ( year class --- )
   case
     0 of ." and is not a leap year."  endof
     1 of ." and is a leap year."      endof
     3 of ." and is not a leap year."  endof
     7 of ." and is a leap year."      endof
    dup . ." Is an error in classification."
  endcase  ;
```

Demonstration

```
: answer  dup cr .
    LYprocess selection ;


1995 answer 1995 is not a leap year.

1996 answer 1996 is a leap year.

1990 answer 1900 is not a leap year.

1990 answer 2000 is a leap year.

ok
```

Extra Credit

```
Accept years as Roman numerals.


final MCM
final MCMXCV
final MCMXCVI
final MM
final MMXXI
```

The method.

1947 is MCMXLVII M CM XL V II

Process right to left.

Accept a character and accumulate its weight.
If next character to the left is smaller, subtract its weight.
If next character is the original character accumulate its value, with repetition.
Move to the next higher weighted character.
Repeat

Setting Up

```
create scratch 20 allot  \ workspace

0 value offset  \ offset into scratch

0 value result  \ decimal value

: initialize-roman
  scratch c@ to offset 0 to result ;
```

Accept Roman/ASCII Date

```
: get-Roman
   bl word                          \ accept text
   scratch over c@ 1+ cmove \ into scratch
   initialize-roman ;
```

Fetch The Current Character

```
: @character
\ exit: char true or false false
   offset 0=
   if false false  exit then
   scratch offset + c@ true ;
```

For Each Character Apply Its Weight +/-

```
: process-xx ( weight "x" dir --- )
    rot >r
  begin  @character swap  2 pick = and
    while  decrement-offset
           over r@  *  +to result
     repeat
 r> 3drop  ;
```

Weights For Each Letter

```
: process-I    1 ascii I  process-xx ;
: process-V    5 ascii V  process-xx ;
: process-X   10 ascii X  process-xx ;
: process-L   50 ascii L  process-xx ;
: process-C  100 ascii C  process-xx ;
: process-D  500 ascii D  process-xx ;
: process-M 1000 ascii M  process-xx ;
```

The Nineteen Problem

If a letter appears to the left of a higher valued letter, its value is subtracted.

XIX = 19, traditionally

The Nineteen Problem

If a letter appears to the left of a higher valued letter, its value is subtracted.

XIX = 19, traditionally . . . But

XVIIII, XVXIIII, IXX,

all meet that rule.

The Nineteen Problem

If a letter appears to the left of a higher valued letter, its value is subtracted.

XIX = 19, traditionally . . . But

XVIIII, XVXIIII, IXX,

all meet that rule.

There is no official syntax. A Roman legion went by IIXX = 18.

Parsing R>L Letter by Letter

M D C L X V I

```
: RomanProcess  initialize-roman
  1 process-I
  1 process-V    -1 process-I
  1 process-V
  1 process-X    -1 process-I -1 process-V
  1 process-X
  1 process-L    -1 process-I -1 process-V  -1 process-X
  1 process-L
  1 process-C    -1 process-I -1 process-V  -1 process-X
                 -1 process-L
  1 process-C

And so on for D and M  .  .  .

;
```

Subtractive "I"

Top Level Code and Test

```
: final
  get-roman cr cr scratch count type
  RomanProcess ."  is "  result .
  result LYprocess selection   ;

final MCM
final MCMXCV
final MCMXCVI
final MM
final MMXXII
```

The Example

```
MCM is 1900 and is not a leap year.

MCMXCV is 1995 and is not a leap year.

MCMXCVI is 1996 and is a leap year.

MM is 2000 and is a leap year.

MMXXII is 2022 and is not a leap year.
ok
```

Summary

Leap year detection could have been done in three lines.

But 32 lines gave a general, modular solution with possible reuse.

The generalize structure of the two programs made linkage instantaneous.