```
750 LET Y = Y/4
760 LET Z = INT(Y)
770 IF Y-Z = 0 THEN GOTO 790
780 RETURN
790 LET X(2) = X(2) + 1
800 RETURN
```

A leap year is defined as one which is wholly divisible by the number 4. If it is a century, it must also be divisible by 400 to qualify as a leap year. To keep it simple, we have not attempted to check the century, only the divisibility by 4.

Line 750 sets Y to the old value of Y (the year) divided by 4. The new Y will be a whole number if the year is exactly divisible by 4. Otherwise it will have a decimal fraction.

Line 760 uses the function INT to find the 'integer' value of Y. Integer means whole number. While having no effect on integers, the INTeger function will round down fractional numbers to the nearest whole number. The number to be rounded down is placed in brackets after INT. Alternatively, a variable name can be put in the brackets. So LET Z = INT(496.25) would set Z to 496.

Line 770 subtracts Z from Y and checks to see if the result is 0. If it is, it means the year is a leap year (as there was no decimal fraction in the new Y). If that is the case, the program branches to line 790 using GOTO. Line 790 adds 1 to the second item in the array (the second item was 28, the number of days in an ordinary February).

If the result of the subtraction in line 770 was not zero, X(2) is left as it is and the subroutine RETURNs to the main program, to line 400.

Line 400 is another REM used just to space out the program to aid readability. The next line that actually does something is 410, where R is the variable holding the number of remaining days. It is set here to the number of days in the month entered minus the day entered. If we had entered, for example, 12, FEBRUARY, 1983, D would be equal to 12 and M would be 2. Therefore X(M) would be the same as X(2) and the second item in the X array is 28 (it would not have had 1 added to it as 1983 is not a leap year). Consequently R will be set to 28 − 12, i.e. 16, the number of days remaining in the current month, February.

Line 420 starts another loop.This one is designed to increment the value of M. Can you see why we say FOR L = 1 TO 11 rather than FOR L = 1 TO 12? If M was 2 because we had entered the month as FEBRUARY, line 430 will increment it to 3. Line 440 then sets R, the number of days remaining, to the old R plus X(M). The latter is now equivalent to X(3) since M has been incremented by 1. The value of X(3) is 31, the number of days in March. Line 440 therefore sets the new value of R to 16 + 31 (16 was the result of subtracting 12 from 28).The next time round the loop, M is incremented to 4 and the number of days in April, X(4), is added to the old value of R. The variable R therefore becomes 16 + 31 + 30.

The last circuit through the loop occurs when L = 11, and X(12)'s value, 25, is added to R.

What happens to the loop if a December date is input, so that M = 12? Because of the discrepancy in the limits, some machines skip the loop entirely, while others execute it once, so that X(13) is added to R. X(13) has been set to 0 to give the correct result.

```
470 IF R = 1 THEN GOTO 500
```

This line simply checks if there is only one day remaining to Christmas so that we get a grammatically correct sentence on the screen. If R is not 1, there must be more than one day remaining, so the PRINT statement in line 480 will be grammatically correct.

So that's all there is to it. The version of BASIC we have used should run on most computers (see the 'Basic Flavours' box) except possibly for the 'leap year' subroutine. BASIC is very inconsistent in the way it uses LET. If lines like IF M$ = "SEPTEMBER" THEN LET M = 9 do not work on your computer, the subroutine can be rewritten like this:

```
560 IF M$ = "JANUARY" THEN GOTO 900
570 IF M$ = "FEBRUARY" THEN GOTO 910
580 IF M$ = "MARCH" THEN GOTO 920
:
900 LET M = 1
905 RETURN
910 LET M 3 2
915 RETURN
920 LET M = 3
925 RETURN 925
(. . . and so on)
```

This solution is more space-consuming and less easy to follow with all its GOTOs and RETURNs. However, it does demonstrate that there are usually several ways of solving every problem.

Basic Flavours

READ DATA

These commands are not available on the ZX81, so delete lines 300, 350 – 370 and 510. Add:

```
10 DIM X(13)
20 FOR K = 1 TO 13
30 PRINT "INPUT ITEM NO.";K
40 INPUT X(K)
50 NEXT K
60 STOP
```

RUN the program and enter the data. Delete lines 10–60 and SAVE the program, which also saves the contents of the array. After LOADing the program in future, use GOTO 100 rather than RUN, thus preserving variables

INPUT

The ZX81 requires an INPUT command for each item, so add 285 PRINT "WHEN PROMPTED", and:

```
310 PRINT "INPUT DAY"
312 INPUT D
```

with similar lines for month and year

REM

On the BBC Micro, Commodore 64 and Vic-20, REM statements at the end of a program line must be preceded by a colon(:)