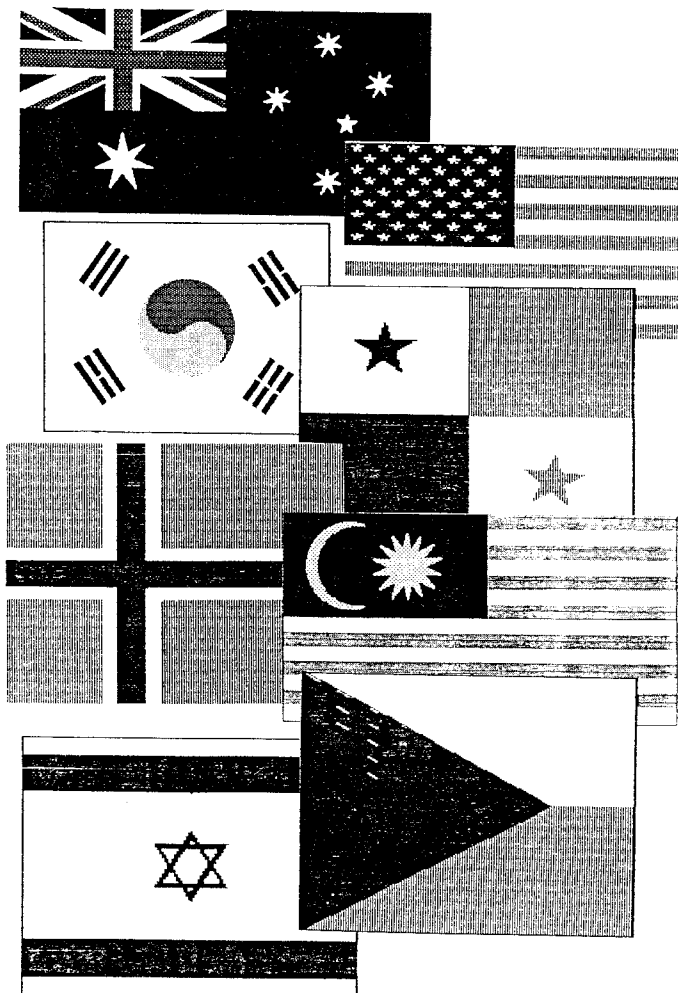


# Turtle Bunting

## A Logo Vexillological Reader

P. J. Carter



Preliminary Edition: Not to be quoted or cited.

# **Turtle Bunting**

## **A Logo Vexillological Reader**

**P. J. Carter**

**Preliminary Edition: for Plympton High School use only.**

© 1990 P. J. Carter

All rights reserved. Except for the programs in this book, which may be used in computers for study purposes, no part of this book may be reproduced in any form by any electronic or mechanical means (including photocopying, recording or information storage and retrieval) without written permission from the author.

Set in Times and Courier fonts. Text prepared with Microsoft Word 4.00A on Macintosh computer, and printed by Hewlett-Packard DeskWriter.

Acornsoft, Apple, Atari, BBC, CGA, Courier, DeskWriter, EGA, Hewlett-Packard, LCSI, Logotron, LogoWriter, Macintosh, MacLogo, Microsoft, MS-DOS, and PS/2 are trademarks registered to you know whom, respectively.

## Contents

Introduction.....	iv
Aboriginal.....	1
Ausflag 88.....	2
Australia.....	3
Bahrain.....	5
How to Program	
Bicentennial.....	6
Pseudocode, the halfway language	
Canada.....	8
Models	
China.....	10
Czechoslovakia.....	11
Earth Flag.....	12
Greece.....	13
Greenland.....	14
Turtle's Measuring Tape	
Guyana.....	15
Israel.....	16
How Long?.....	16
Kuwait.....	17
Malaysia.....	18
Marshall Islands.....	19
Norway.....	20
A Comment	
Panama.....	21
Validation	
St Lucia.....	22
Debugging	
South Korea.....	23
Structure Charts	
United States of America.....	25
W.....	26
Complexity	
Western Samoa.....	27
Yugoslavia.....	28
Final Flaps.....	29
Appendix	
Some Logo Special Words.....	30
Further Reading.....	32

## Introduction

Harold Abelson, one of the 'fathers' of Logo, has stated, 'Programs are for people to read, and only incidentally for machines to execute.'<sup>1</sup> Logo programs are therefore a kind of literature, to be read to gain an insight into the solving of problems. You can perhaps consider a program as a chapter, and a procedure as equivalent to a paragraph.

It follows that programs must be designed to be easy to read, with a clear structure, concise logic and meaningful names.

In the examples that follow, each program is presented in a particular Logo version, usually the one in which it was first written. Most of them are quite short, usually less than a page. There will be some unfamiliar words, colour numbers, screen sizes etc., although many of the primitives are described in an Appendix. In every case, primitive names are in all capitals, names of defined procedures begin with capitals, and variable names are in all lower case, to make them more easily distinguishable.

There is very little explanation: you will have to read, and work out for yourself, what each procedure does, how it fits into the whole program, and how it sends or receives values. What were the problems faced by the programmer(s)? Are the solutions, the procedures, the best solutions? How would you improve on them?

As you read, remember that FD means FORWARD, and should be pronounced as such, and the same is true of other abbreviations like RT, LT, PU, PD and so on. Some procedures in this book have lines that are too long to fit across the page. A line that is indented is continuing from the previous, as in this example:

```
TO Block :length :width
REPEAT :width / 2 [FD :length RT 90 FD 1 RT 90
    FD :length LT 90 FD 1 LT 90]
END
```

In a conventional programming book there would be discussion of the process of analysis, design, coding and validation, with pseudocode, structure charts and that sort of thing. In this book there's very little of that (there's some in boxes) and you will work in reverse: this is programming by example. Read a procedure, think about it, and make sure you understand it before moving to the next.

There are some questions about the flags themselves to keep you busy. At the very least, you should find out what the shapes and colours signify on each flag.

I am indebted to a number of students for several of these programs, and their names appear with their procedures.

Why vexillology? Flags are something humans have been using for centuries. They have bright colours, and the designs are usually made up of simple shapes, rectangles, triangles, stars and the like (look for the 'tool' procedures that draw them). The trick is in fitting them all together logically. Of course, those colours and shapes are symbolic, they have meanings. Logo procedures are like that too.



*P. J. Carter*

What does *vexillology* mean? What about words like *hoist*, *fly*, *canton* and *saltire*?

---

<sup>1</sup> 'Computation: An Introduction to Engineering Design' in *Logo in Australia: 10 Years On*, 1985

## Aboriginal

It's perhaps fitting that this is the first flag in the book. A simple, bold design, it was first used in the early 1970s, and has become a readily recognised symbol of the aboriginal community's struggle for rights and recognition.

Here are two versions, Atari and Macintosh:

```
To Aboriginal
SetField
Earth
Border
Sun
END
```

```
TO SetField
CS HT FS
SETBG 0
SETPC 0 26
SETPC 1 13
END
```

```
TO Earth
RT 90 BK 158
SETPN 0
Block 319 95
END
```

```
TO Border
SETPOS [-157 -95] SETH 0
REPEAT 2 [FD 190 RT 90 FD 317 RT 90]
END
```

```
TO Sun
PU HOME PD
SETPN 1
REPEAT 720 [FD 60 BK 60 RT 0.5]
END
```

```
TO Aboriginal
CG HT
SETPPATTERN 0
RECT [-70 0 140 50] "TRUE
SETPPATTERN 22
RECT [-70 -50 140 50] "TRUE
SETPPATTERN 19
PENPAINT
OVAL [-30 -30 60 60] "TRUE
END
```

?

Which set of procedures makes more sense to you? Why? What makes procedures meaningful to human readers?

## Ausflag 88

There have been many attempts to replace the Australian flag. Some people think it is time we abandoned the Union Jack, along with our other ties with Britain. In 1987 there was a competition for a new flag, won by one Wayne Stokes, who used the traditional red, white and blue, rather than the green and gold favoured by some. This is his flag, in Macintosh Logo:

```
TO AusFlag88
  SetField
  RedStripe
  WhiteStripe
  Crux
END
```

```
TO SetField
  SETWPOS "Graphics [30 40]
  SETWSIZE "Graphics [432 216]
  CG NOWRAP
  SETPPATTERN 0 PD
  RECT [-216 -60 432 168] "TRUE
END
```

```
TO WhiteStripe
  SETPPATTERN 19
  RECT [-216 -84 432 24] "TRUE
END
```

```
TO Crux
  SETPPATTERN 0
  SETPWIDITH 2
  PU SETPOS [-130 -25] SETH 11 PE
  Star 1 154.3 20
  PU SETPOS [-173 24] SETH 11 PE
  Star 1 154.3 20
  PU SETPOS [-130 73] SETH 11 PE
  Star 1 154.3 20
  PU SETPOS [-87 30] SETH 11 PE
  Star 1 154.3 20
  PU SETPOS [-105 5] SETH 90 PE
  Star 1 144 12
END
```

```
TO RedStripe
  SETPPATTERN 3
  RECT [-216 -108 432 24] "TRUE
END
```

```
TO Star :size :angle :max
  IF :size > :max [STOP]
  FD :size RT :angle
  Star :size + 1 :angle :max
END
```



Why call it Crux and not Stars?  
Design a new flag for Australia.

## Australia

There are some interesting problems in drawing the Australian flag, especially in the Union Jack. Take particular note of the structure of this program. For another way of drawing the Southern Cross, see the flag of Western Samoa. Yet another way is to have a procedure, like CommonwealthStar for each star, Alpha, Beta, etc. This one's in Macintosh Logo, although it was originally written in Apple Logo, and I've tried to keep the proportions as accurate as possible.

```
TO Australia
  SetField
  UnionJack
  CommonwealthStar
  Crux
END
```

```
TO SetField
  SETWPOS "Graphics [30 40]
  SETWSIZE "Graphics [432 216]
  CG NOWRAP
  SETPPATTERN 0 PD
  RECT [-216 -108 432 216] "TRUE
END
```

```
TO UnionJack
  StAndrew
  StPatrick
  StGeorge
  Trim
END
```

```
TO StAndrew
  PE SETPWIDTH 22
  SETPOS [-216 108]
  PU SETPOS [-216 0]
  PE SETPOS [0 108]
END
```

```
TO StPatrick
  SETPWIDTH 7 SETPPATTERN 1
  PU SETPOS [0 4] PD SETPOS [-108 58] SETPOS [15 122]
  PU SETPOS [-216 -3] PD SETPOS [-108 51] SETPOS [-216 104]
END
```

```
TO StGeorge
  SETPWIDTH 36 PU SETPPATTERN 0
  SETPOS [-108 0] PE SETPOS [-108 108]
  PU SETPOS [-216 54] PE SETPOS [0 54]
  SETPWIDTH 20 PU SETPPATTERN 1
  SETPOS [-108 0] PD SETPOS [-108 108]
  PU SETPOS [-216 54] PD SETPOS [0 54]
END
```



```

TO Trim
  SETPPATTERN 0
  PU SETPOS [-216 -10] PD SETPOS [10 -10]
  SETPOS [10 108]
  END

TO CommonwealthStar
  SETPWIDTH 2
  PU SETPOS [-108 -54] PE SETH 11
  Star 1 154.3 60
  END

TO Crux
  PU SETPOS [108 -72] SETH 11 PE ; Alpha
  Star 1 154.3 30
  PU SETPOS [54 13] SETH 11 PE ; Beta
  Star 1 154.3 30
  PU SETPOS [108 72] SETH 11 PE ; Gamma
  Star 1 154.3 30
  PU SETPOS [156 28] SETH 11 PE ; Delta
  Star 1 154.3 30
  PU SETPOS [130 -9] SETH 20 PE ; Epsilon
  Star 1 144 22
  END

TO Star :size :angle :max
  IF :size > :max [STOP]
  FD :size RT :angle
  Star :size + 1 :angle :max
  END

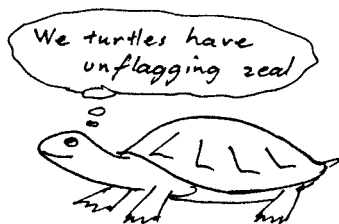
```

?

Like all flags, the Australian flag has an official specification, describing its colours, proportions and other details. What are those proportions? Who designed the flag? When?

How does that Star procedure work? Why not use REPEAT? What are the rules for choosing between REPEAT and recursion? How would you write a Star procedure for a Logo with FILL, or for the BBC?

Why use names like StAndrew instead of WhiteCross (or WhiteSaltire)? What's in a name?



## Bahrain

This one was written by two Year 10 students, Paula Beaumont and Karla Jones. It's in Apple Logo //.

```

TO Bahrain
CS FS HT
SETBG 4 SETPC 1
PU SETPOS [-140 120] PD
RT 90
REPEAT 2 [FD 278 RT 90 FD 239 RT 90]
ZigZag
Colour
END

TO ZigZag
FD 100 LT 210
REPEAT 8 [FD 30 LT 120 FD 30 RT 120]
END

TO Colour
PU SETPOS [-130 0]
PD FILL
END

```



There's another flag something like this one. Write the procedures for it.

#### How to Program:

Programmers use the terms 'top-down' and 'bottom-up' to describe how programs are planned. A flag designer who starts with sketches could be said to be taking a top-down approach, while one who starts by browsing through a fabric catalogue is taking the bottom-up route.

Top-down in programming goes something like this: to draw the Australian flag we take a blue field and draw the Union Jack, the Southern Cross and the Commonwealth Star. How do we draw the Union Jack? Draw St Andrew's Saltire, St Patrick's and then St George's Cross... In other words, we start with an overall view and gradually divide the problem into sub-problems.

A bottom-up programmer would begin by playing about with rectangle and star procedures.

Most of the programs in this book were designed with a 'middle-out' technique, a sort of combination of top-down and bottom-up. It all depends on the problem in hand, the procedures already available to you, and your own experience and style. According to all the good programming books, top-down is the 'best' way to go. In Logo, you can work however you feel, and change your ideas as you go along.

## Bicentennial

Australia's bicentennial in 1988 had its own flag, although the Tasmanians still wonder where they fit in. This is a Macintosh version.

```
TO Bicentennial
West
East
Trim1
Trim2
PD Words
END
```

```
TO West
PU SETPOS [-98 -31] SETH 138 PD
Block 17.5 148 113 67
Jump 22.5 SETPPATTERN 22
Block 8 148 113 67
Jump 20 SETPPATTERN 0
Block 17.5 148 113 67
END
```

```
TO Block :width :length :angle1 :angle2
FILLSH [REPEAT 2 [FD :width LT :angle1 FD :length
    LT :angle2]]
END
```

```
TO Jump :dist
PU FD :dist PD
END
```

```
TO East
PU SETPOS [-25 -16] SETH 130 PD
Block 38 77.5 105 75
Jump 42.5
Block 19 138 105 75
Jump 24 SETPPATTERN 22
Block 9 140 105 75
Jump 21 SETPPATTERN 0
Block 19 140 105 75
END
```

```
TO Trim1
PU SETPOS [-57.5 54] SETH 90 PE
Block 65 50 90 90
END
```

```
TO Trim2
PU SETPOS [67.5 80] SETH 162 PE
FILLSH [FD 95 LT 137 FD 57.5 SETPOS [67.5 80]]
END
```

```

TO Words
SETWRITE "Graphics
SETFONT "Geneva
SETSTYLE [32 12]
SETCURSOR [100 -60]
PRINT CHAR 169
SETSTYLE [32 20]
SETCURSOR [-40 -100]
PRINT "Australia
SETCURSOR [-50 -125]
SETSTYLE [32 18]
PRINT [1788\1988]
SETWRITE "Text
END

```



Design your own flag for a special occasion.

**Pseudocode, the halfway language:**

One of the tools used by professional programmers is *pseudocode*, sometimes called structured English. When a program is being designed, everything that has to be done is first thought through in the programmer's head then written in pseudocode. When that is checked, the actual code is written. Here's a sample:

*to draw the Canadian flag  
 clear the screen, turn it white and hide the Turtle  
 draw the red rectangles on the two sides  
 draw the maple leaf...*

You might try pseudocode in reverse by 'translating' some of these flag procedures from Logo.

If you find that pseudocode helps to clarify your ideas that's fine, use it. Otherwise, think in Logo.

## Canada

Canada once had a flag with the Union Jack in the canton. The present flag was adopted amid controversy, but is a distinctive example of good vexillology. Logotron this time.

```
TO Canada
SetField
Red
MapleLeaf
END
```

```
TO SetField
WINDOW HT
VDU [19 0 1 0 0 0]
SETPC 0 SETBG 1
.SETNIB 85
END
```

```
TO Red
PU SETPOS [-640 -415] Safe PD
Block 830 320 Safe
PU SETPOS [640 415]
SETH 180 Safe PD
Block 830 320 Safe
END
```

```
TO Block :length :width
REPEAT 2 [FD :length RT 90 FD :width RT 90]
END
```

```
TO Safe
REPEAT 3 [FD 0]
END
```

```
TO MapleLeaf
RightSide
LeftSide
Stem
END
```

```
TO RightSide
Point 128 272 194 36 280
Point 160 168 175 36 200
Point 216 104 192 60 130
Point 232 -8 220 56 260
Point 144 -248 287 44 160
PU SETPOS [-45 -180] SETH 0 PD
Block 250 90
END
```

```

TO LeftSide
Point 0 320 150 60 200
Point -128 272 131 36 280
Point -160 168 148 36 200
Point -216 104 108 60 130
Point -232 -8 84 56 260
Point -144 -248 30 44 160
END

TO Stem
Point 0 -80 177 6 240
END

TO Point :x :y :heading :angle :length
PU SETPOS SE :x :y SETH :heading Safe PD
FD :length BK :length
RT :angle FD :length Safe
END

```

?

What other flags have leaves on them? What about trees or flowers, or birds or animals? Are there any flags with Turtles on them?

#### Models:

What do you think a model is? Most people would think of a scale representation of something; a model aircraft, car, building or whatever. There are other sorts of models. We can have mathematical models of all sorts of processes: preparing a budget, for instance, is developing a model of the finances of an organisation. The tool for that kind of model would probably be a spreadsheet. Our flags can be thought of models of the actual flags.

Another sort of model is a mental model. Our minds are constantly constructing and reconstructing our understanding of the world around us, in fact the process of learning is the process of building mental models.

What happens when something we see or hear contradicts our mental models? What happens when the procedure we've just written turns out not to do what we wanted? Is programming a model of learning? Do mistakes matter?

## China

Why, you might ask, didn't I simply SETBG 25 for this one? The answer is that I wrote it in early June 1989, and wanted a black border. It's in Atari Logo.

```
TO China
Setup
Field
Stars
END
```

```
TO Setup
CS HT FD
SETBG 0
SETPC 0 25
SETPC 1 15
END
```

```
TO Field
SETPN 0
SETPOS [158 95] SETH 90
Block 310 180
END
```

```
TO Block :length :width
REPEAT :width / 2 [FD :length RT 90 FD 1 RT 90
  FD :length LT 90 FD 1 LT 90]
END
```

```
TO Stars
SETPN 1
PU SETPOS [-112 42] SETH 90 PD
Star 1 50
PU SETPOS [-45 78] SETH 90 PD
Star 1 15
PU SETPOS [-35 53] SETH 90 PD
Star 1 15
PU SETPOS [-35 22] SETH 90 PD
Star 1 15
PU SETPOS [-45 0] SETH 90 PD
Star 1 15
END
```

```
TO Star :size :max
IF :size > :max [STOP]
FD :size RT 144
Star :size + 1 :max
END
```

?

What do the stars represent?

The black border, and the comment about it and the date, could be read as a political statement. Flags are an integral part of politics and nationalism. Why is that? Is that a good thing?





## Earth Flag

The Earth Flag is the flag of an ecology group, and the discs represent the Sun, Earth and Moon. It's in Logotron Logo.

```
TO EarthFlag
Field
Border
Sun
Earth
Moon
END
```

```
TO Field
SETMODE 5
WINDOW HT
VDU [19 1 4 0 0 0]
END
```

```
TO Border
PU SETPOS [-640 -380] PD
REPEAT 2 [FD 760 RT 90 FD 1250 RT 90]
END
```

```
TO Sun
PU SETPOS [-1250 0]
SETH 40 PD
SETPC 2
Safe .SETNIB 85
REPEAT 100 [FD 1150 BK 1150 RT 1]
END
```

```
TO Earth
Safe PU HOME Safe
SETPC 1 PD
REPEAT 181 [FD 300 BK 300 RT 2]
END
```

```
TO Moon
Safe PU SETPOS [400 -300] Safe PD
SETPC 3
REPEAT 90 [FD 80 BK 80 RT 90]
END
```

```
TO Safe
REPEAT 3 [FD 0]
END
```



Design a flag to promote ecological values.

## Greece

You need to be able to draw only one shape for the Greek flag, rectangles. Logotron for this rendition.

```

TO Greece
  SetField
  Stripes
  Cross
END
      TO SetField
        SETMODE 5
        WINDOW HT
        VDU [19 1 4 0 0 0]
        SETBG :white
        SETPC :blue
      END

  TO Stripes
    PU SETPOS [-640 380] SETH 180 PD
    .SETNIB 85
    REPEAT 7 [Block 85 1280 PU FD 170 PD]
  END

  TO Cross
    PU SETPOS [-640 380] PD
    Block 422 422
    PU SETPOS [-640 211] PD
    SETPC :white
    Block 85 422
    PU SETPOS [-471 380] PD
    Block 425 85
  END

  TO Block :width :length
    Safe
    REPEAT 2 [FD :width RT 90 FD :length RT 90]
    Safe
  END
      TO Safe
        REPEAT 3 [FD 0]
      END

  MAKE "blue 1
  MAKE "white 3

```

?

Who were the first people to really think about, and formalise, geometry?  
 What is the MAKE "blue 1 business? Why not simply SETPC 1?  
 If you're using LogoWriter you'll need a procedure like this...

```

TO Globals
  MAKE "blue 1
  MAKE "white 3
END

```

...with a call to it first thing in SetField.

## Greenland

Although Greenland has been ruled by Denmark, its flag is totally different from those of the other Scandinavian countries. (See Norway on a later page.) This is Logotron Logo for the BBC.

```
To Greenland
SetField
PU SETPOS [-645 0] RT 90
Safe .SETNIB 85 PD
Block 1300 390
PU SETPOS [-200 0] SETH 270
HalfDisk
SETPC 3
HalfDisk
END
```

```
TO SetField
SETMODE 5
WINDOW HT
SETBG 3
SETPC 1
END
```

```
TO Block :length :width
REPEAT 2 [FD :length RT 90 FD :width RT 90]
Safe
END
```

```
TO HalfDisk
REPEAT 91 [FD 260 BK 260 RT 2]
END
```

```
TO Safe
REPEAT 3 [FD 0]
END
```



Why is this flag so different from Denmark's? What do the people of the Arctic call themselves?

#### Turtle's Measuring Tape:

How big is the Logo screen you're using? (Not in centimetres, but Turtle steps.) You may find the answer in the reference manual, but if you can't what will you do?

Type `SHOW POS`. The result is the coordinates of the Turtle's position. Put Turtle on the edge of the screen and you're on the way to finding its size.

## Guyana

Flags don't have to be complicated to be effective. Simple shapes and bold, significant, colours make for a good flag. The procedures are in LogoWriter for the PC (EGA), and are an example of 'brute force' programming, in that there are four very similar procedures. I could have written the program using a triangle procedure with four inputs, three for the vertices and one for the colour. As it was, I simply used copy and paste and changed the numbers. Colour number 8 is not really the black it should be, but dark grey, because colour 0, allegedly black, is transparent.

```
TO Guyana
CG HT SETBG 7
White
Gold
Black
Red
END
```

```
TO White
SETC 1
SETPOS [-159 95]
SETPOS [160 0]
SETPOS [-159 -95]
SETPOS [-159 95]
PU SETPOS [-150 0] PD FILL
END
```

```
TO Gold
SETC 14
SETPOS [-159 90]
SETPOS [140 0]
SETPOS [-159 -90]
SETPOS [-159 90]
PU SETPOS [-150 0] PD FILL
END
```

```
TO Black
SETC 8
SETPOS [-159 95]
SETPOS [-15 0]
SETPOS [-159 -95]
SETPOS [-159 95]
PU SETPOS [-150 0] PD FILL
END
```

```
TO Red
SETC 12
SETPOS [-159 90]
SETPOS [-25 0]
SETPOS [-159 -90]
SETPOS [-159 90]
PU SETPOS [-150 0] PD FILL
END
```

?

Rewrite the program to use a Triangle procedure with inputs for the pen colour and vertices.

## Israel

This set of procedures was written by another team of Year 7 students, Tim, Russell and Greg. The star caused them numerous problems: perhaps you can come up with a better solution. It's in Apple Logo.

```
TO Israel
  CS HT FULLSCREEN
  SETBG 1 SETPC 5
  PU FD 103 RT 90 PD
  Block
  RT 90 PU FD 138 LT 90 PD
  Block
  Star
  END
```

```
TO Block
  REPEAT 17 [FD 280 RT 90 FD 1 RT 90 FD 280 LT 90 FD 1 LT 90]
  END
```

```
TO Star
  PU SETPOS [-30 20] SETH 90 PD
  Triangle 60
  PU SETPOS [30 -20] SETH 270 PD
  Triangle 60
  END
```

```
TO Triangle :size
  IF :size < 50 [STOP]
  REPEAT 3 [FD :size RT 120]
  RT 45 FD .5 LT 45
  Triangle :size - 1
  END
```

?

There really ought to be another procedure in this program, one to set up the screen, and you might like to use SETPOS to put Turtle in position for the blue stripes.

### How Long?

How long should a procedure be? As long as it needs to be and no longer.

That doesn't seem a satisfactory answer, so let's try again. A procedure should do only one task; in the case of these programs, a procedure should draw only one part of the flag. When you try to cram too much into a procedure it becomes too hard to understand, particularly if you wrote it last week and it's not working properly.

One of the keys to successful programming is the ability to divide a problem into sub-problems ('mind-sized bits' as they're sometimes called), each of which can be solved with one or two small procedures.

## Kuwait

Like many flags of Arab nations, this one has red, green and black. This is a LogoWriter version for the Apple. The Black procedure needs three FILLS because it crosses lines drawn in other colours.

```
TO Kuwait
HT SETBG 1
Green
Red
Black
END
```

```
TO Green
SETC 2
PU SETPOS [-140 90] SETH 90 PD
Rectangle 280 60
END
```

```
TO Red
SETC 4
PU SETPOS [139 -89] SETH 270 PD
Rectangle 280 60
END
```

```
TO Rectangle :length :width
REPEAT 2 [FD :length RT 90
          FD :width RT 90]
PU RT 45 FD 10 PD FILL
END
```

```
TO Black
SETC 0
PU SETPOS [-140 -89] PD
SETPOS [-140 90]
SETPOS [-80 30]
SETPOS [-80 -30]
SETPOS [-140 -89]
PU SETPOS [-130 70] PD FILL
PU SETPOS [-130 0] PD FILL
PU SETPOS [-130 -70] PD FILL
END
```

?

Although this flag uses the pan-Arab colours, they're interpreted differently. Why is this? Can you write a program to draw all the similar Arab flags?

## Malaysia

As in the United States flag, the red and white stripes represent the states of the federation. The crescent is a common symbol on the flags of Moslem countries. This is Acornsoft Logo for the BBC.

TO Malaysia

SetColours

Stripes

Canton

END

TO SetColours

DRAW SETMODE 5

WINDOW HT

PAL 0 4

SETBG :white

END

TO Stripes

PU SETPOS [-630 410] SETH 180 PD

Safe SETNIB 80 SETPC :red

REPEAT 7 [Block 59 1260 Safe PU FD 119 Safe PD]

END

TO Block :width :length

REPEAT 2 [FD :width LT 90 FD :length LT 90]

END

TO Canton

PU SETPOS [0 -62] SETH 0 Safe PD

SETPC :blue

Block 470 630

Moon

Star

END

TO Moon

PU SETPOS [-400 174] Safe 20

SETPC :gold SETH 20

REPEAT 110 [FD 170 BK 170 LT 2]

PU SETPOS [-370 174] Safe SETH 10 PD

SETPC :blue

REPEAT 100 [FD 160 BK 160 LT 2]

END

TO Star

PU SETPOS [-300 174] SETH 18 Safe PD

SETPC :gold

REPEAT 14 [FD 150 RT 154.29 FD 150 RT 115.7 FD 67 RT 115.7]

END

TO Safe

REPEAT 3 [FD 0]

END

MAKE "blue 0

MAKE "red 1

MAKE "gold 2

MAKE "white 3

?

Why are there fourteen stripes, and fourteen points on the star, when there are only thirteen states in Malaysia?

## Marshall Islands

This striking flag was designed, not by a vexillologist, but by Emlain Kabua, the wife of the country's president. This is LogoWriter for the PC (with EGA graphics), and the Star procedure seems a bit more complicated than it needs to be, but this is because FILL doesn't work very well on complex shapes. The Orange and White procedures are very similar. Type one, then use the editor's copy and paste to create the other.

```
TO MarshallIs
CG HT SETBG 5
Orange
White
Star 25
END
```

```
TO Orange
SETC 6
PU SETPOS [-159 -60] PD
SETPOS [160 85]
SETPOS [160 50]
SETPOS [-159 -70]
SETPOS [-159 -60]
PU SETPOS [0 -5] PD FILL
END
```

```
TO White
SETC 1
PU SETPOS [-159 -70] PD
SETPOS [160 50]
SETPOS [160 15]
SETPOS [-159 -80]
SETPOS [-159 -70]
PU SETPOS [0 -12] PD FILL
END
```

```
TO Star :size
PU SETPOS [-100 50] SETH 4 PD
REPEAT 4 [Point :size * 3 / 2
          REPEAT 5 [Point :size]]
PU BK 10 PD FILL
END
```

```
TO Point :size
FD :size LT 5
REPEAT 2 [BK :size FD :size RT 5]
RT 165 FD :size LT 155
END
```

?

Why are stars such common symbols on flags? Why does the one on this flag have so many points?



## Norway

The Norwegian flag follows the pattern of the other Scandinavian flags in having an offset cross. Once you have this flag, you can draw the others by simply changing colours, although their proportions vary slightly. This is another flag by Tim, Russell and Greg, in Apple Logo. If they were writing it today, I'd expect to see it structured differently, with cross procedures: that main procedure is far too long.

```

TO Norway
CS HT FULLSCREEN
SETBG 4 SETPC 1
FD 24 RT 90
Block 280 48
PU BK 60 LT 90 PD
Block 240 48
SETPOS [-48 12]
RT 90 SETPC 5
Block 280 24
LT 90
Block 240 24
END

TO Block :length :width
REPEAT :width / 2 [FD :length RT 90 FD 1 RT 90
  FD :length LT 90 FD 1 LT 90]
END

```

?

This flag was the first to fly at a very remote part of the globe. The leader of the second group to reach the place was, understandably, very disappointed. What were the circumstances? Why do people, at least in English speaking countries, remember the second but not the first?

Write a set of procedures to draw all the Scandinavian flags, with a menu to allow people to choose the flags they want to see. Use Cross procedures.

**A Comment:**

Most programming books suggest that you put comments into your code to make programs easier to understand. My view on that is that if you need lots of comments then the code is poorly written. You should be able to read the code itself and understand it; remember, 'Programs are written for people to read...'

Nevertheless, there are times when a comment can be helpful, and there is one program in this book where I used comments. If your Logo has a comment primitive it's probably ; . If it doesn't, you can write one:

```

TO ; :comment
END

```

Remember that comments with this procedure must follow the usual rules for Logo inputs, and must have a " or be in [ ].

LogoWriter makes comments easy; any line that can't be interpreted as part of a procedure is a comment. Just be careful not to begin a comment with the word to.

## Panama

Tim, Russell and Greg again, and Apple Logo once more.

```

TO Panama
CS HT SETBG 1
FULLSCREEN SETPC 4
Block
HOME LT 180 SETPC 5
Block
PU SETPOS [-70 60] PD
SETH 90
Star 2
PU SETPOS [70 -60] PD
SETH 90 SETPC 4
Star 2
END

```

```

TO Block
REPEAT 70 [FD 119 RT 90 FD 1 RT 90 FD 119 LT 90 FD 1 LT 90]
END

```

```

TO Star :size
IF :size > 50 [STOP]
FD :size RT 144
Star :size + 1
END

```

?

Where would you expect to see this flag in Australia? Why?

#### Validation:

It's easy enough to see if these flag procedures are correct; they either draw the flag or they don't. What about other programs? How do we prove them correct? This is a real issue with business, industrial and military programs. As a case in point, consider the control system in the Airbus A-320. Unlike other commercial aircraft, there is no mechanical connection between the pilot's hand controller and the control surfaces. Everything goes through the aircraft's 5 (with 2 backup) computers. What if the programs have bugs?

Come to that, how do we 'prove' anything in science? How do we know that the Sun is the centre of the solar system and not orbiting the Earth? (It would still look the same.) Why is it that heliocentric (Sun centred) replaced geocentric (Earth centred) views of the solar system, Newton replaced impetus theory, Einstein replaced Newton, evolution replaced creation, and so on?

Is human knowledge advancing in top-down or bottom-up fashion? What about your own personal knowledge? What makes you change your view of the world? What happens when you make a mistake?

What has all this to do with programming in Logo?

## St Lucia

St Lucia was discovered by Columbus, and was a British colony. Atari Logo for this one, and because there is no FILL, an interesting means of filling triangular areas.

```
TO StLucia
  Field
  White
  Black
  Gold
  CT PR [The flag of St Lucia.]
END
```

```
TO Field
  CS HT SS
  SETBG 74
END
```

```
TO White
  SETPC 1 7 SETPN 1
  PU SETPOS [-60 -45] PD
  Triangle [-60 -45] [60 -45] 0 90 -45
END
```

```
TO Black
  SETPC 2 0 SETPN 2
  PU SETPOS [-50 -45] PD
  Triangle [-50 -45] [50 -45] 0 70 -45
END
```

```
TO Gold
  SETPN 0
  PU SETPOS [-60 -45] PP
  Triangle [-60 -45] [60 -45] 0 -10 -45
END
```

```
TO Triangle :base1 :base2 :vertexX :vertexY :baseY
  IF :vertexY = :baseY [STOP]
  SETPOS SE :vertexX :vertexY
  SETPOS :base2 SETPOS :base1
  Triangle :base1 :base2 :vertexX :vertexY - 1 :baseY
END
```

?

How many other flags use triangles as part of their design?

Debugging:

What happens if the procedure you thought was correct turns out not to be right? Do you give up, cry about it, swear at it, or what? What you need to do is check the output (part of the picture perhaps) with what it should have done, and compare it, line by line, with the procedure, until you find the wrong turn or whatever it was. Programming errors are often called 'bugs', and the process of fixing them, 'debugging.' You can actually learn more through debugging than by writing procedures that are right first time.

## South Korea

In the centre of the South Korean flag is the symbol of Yin and Yang, representing the opposites in the universe. Macintosh Logo once more, although it doesn't use ARC or FILLSH because the procedures were translated from Apple Logo //. That's obviously not making the best use of MacLogo, but it will make it easier to rewrite for something else.

```
TO SouthKorea
CG HT SETPPATTERN 0 SETPWIDTH 2
Yin.Yang
Trigrams
Border
END
```

```
TO Yin.Yang
SETH 315 SETPWIDTH 2
SETPPATTERN 2
HalfCircle 50
SETPPATTERN 22
HalfCircle 50
PU FD 25 SETPPATTERN 2
RT 180 PD
HalfCircle 24
PU BK 50 PE
HalfCircle 24
RT 180
SETPPATTERN 22 PD
HalfCircle 24
END
```

```
TO HalfCircle :size
REPEAT 181 [FD :size BK :size RT 1]
END
```

```
TO Trigrams
SETPPATTERN 0
Trigram1
Trigram2
Trigram3
Trigram4
END
```

```
TO Trigram1
Position [-80 50] 35
REPEAT 3 [LongBlock]
END
```

```
TO Trigram2
Position [80 50] 145
ShortBlocks
LongBlock
ShortBlocks
END
```

```
TO Trigram3
Position [80 -50] 215
REPEAT 3 [ShortBlocks]
END
```

```
TO Trigram4
Position [-80 -50] 325
LongBlock
ShortBlocks
LongBlock
END
```

```

TO Position :pos :head
PU SETPOS :pos SETH :head + 90
BK 25 LT 90 BK 20 PD
END

```

```

TO LongBlock
PD Block 50 5
PU RT 90 FD 7 LT 90
END

```

```

TO ShortBlocks
PD Block 22.5 5 PU
FD 27.5 LT 90 FD 5 RT 90
PD Block 22.5 5
PU BK 27.5 RT 90 FD 7 LT 90
END

```

```

TO Block :length :width
REPEAT :width [FD :length RT 90 FD 0.5 RT 90
FD :length LT 90 FD 0.5 LT 90]
END

```

```

TO Border
PU SETPOS [-150 110] SETH 90 PD
REPEAT 2 [FD 300 RT 90 FD 220 RT 90]
END

```

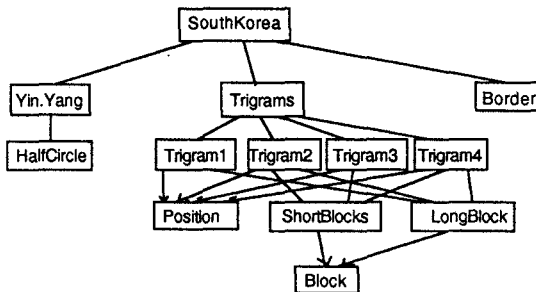
?

Why does the Position procedure use BK and LEFT as well as SETPOS?

### Structure Charts:

Another tool used by programmers is the structure chart. In these diagrams procedures are represented by rectangles and the connections show which procedure calls which. Inputs and outputs are shown by arrows.

Here's a structure chart of SouthKorea; it's a bit of a tangle in one area but note how it shows what fits where:



## United States of America

It seems that no collection of flags is complete without this one. For this PC LogoWriter (EGA) version, I've used a redefined Turtle to STAMP the stars. I'm not sure whether measuring distances to one place of decimals is necessary.

```
TO UnitedStates
CG HT SETBG 12
Stripes
Canton
END
```

```
TO Stripes
SETC 1
PU SETPOS [160 80] SETH 180
REPEAT 3 [PD Block 14.6 190 FD 29.2]
REPEAT 3 [PD Block 14.6 320 FD 29.2]
END
```

```
TO Block :width :length
REPEAT 2 [FD :width RT 90 FD :length RT 90]
PU RT 45 FD 5 PD FILL
PU BK 5 LT 45
END
```

```
TO Canton
SETC 5
PU SETPOS [-159 -7] SETH 0 PD
Block 102 128
Stars
END
```

```
TO Stars
SETC 1 SETSH 1
PU SETPOS [-150 85] SETH 90
ROW6
REPEAT 4 [Row5 Row6]
END
```

```
TO Row5
FD 10.8
REPEAT 5 [PD STAMP PU FD 21.6]
BK 118.8
SETY YCOR - 10
END
```

```
TO Row6
REPEAT 6 [PD STAMP PU FD 21.6]
BK 129.6
SETY YCOR - 10
END
```

?

When was this flag last changed? Why?

How will you draw the stars if you can't STAMP a redefined Turtle? The Star procedure we've used elsewhere won't be much help here.

## W

This isn't a national or state flag at all, but one of the set of 40 flags used for signalling at sea. The flags may be used singly (W means 'I require medical assistance') or in groups. You might try others in the set, and if you're ambitious, try a program that allows you to key in three letter codes and then displays the complete signal on the screen. For the codes, I'd suggest *Brown's Signalling*, from a yachting chandler. The procedures are in Apple Logo //.

```
TO W
CS HT SETBG 5
White
Red
END
```

```
TO White
SETPC 1
PU SETPOS [-109 -70] PD
Block 140 218
END
```

```
TO Red
SETPC 4
PU SETPOS [-79 -40] PD
Block 80 158
END
```

```
TO Block :width :length
REPEAT 2 [FD :width RT 90 FD :length RT 90]
PU RT 45 FD 5 PD FILL
BK 5 LT 45
END
```

?

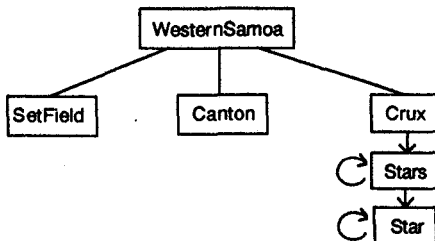
How does this method of signalling compare with modern methods of communication? What are its advantages and disadvantages? How often is it still used? What was Nelson's famous signal?

Complexity:

William of Occam (Ockham, if you prefer) wrote: 'Entia non sunt multiplicanda praeter necessitatem.' What does that mean in English? What's the point of it when applied to programming?

## Western Samoa

Here's a structure diagram for this program, and the part circular arrows show the recursion. This happens to be the only program in this book to use any list processing. It's in Apple Logo //.



```
TO WesternSamoa
```

```
SetField
```

```
Canton
```

```
Crux
```

```
END
```

```
TO SetField
```

```
CS HT
```

```
SPLITScreens
```

```
SETBG 4
```

```
END
```

```
TO Canton
```

```
PU FD 20
```

```
SETPC 5 PD
```

```
REPEAT 2 FD 100 LT 90 FD 139 LT 90]
```

```
LT 45 PU FD 5 PD FILL
```

```
END
```

```
TO Crux
```

```
SETPC 1
```

```
Stars [[[-70 95] 15] [[-45 70] 15] [[-70 45] 15]  
      [[-95 70] 15] [[-50 50] 8]]
```

```
END
```

```
TO Stars :starData
```

```
IF EMPTY? :starData [STOP]
```

```
PU SETPOS FIRST FIRST :starData
```

```
SETH 18 PD
```

```
Star 1 LAST FIRST :starData
```

```
Stars BUTFIRST :starData
```

```
END
```

```
TO Star :size :max
```

```
IF :size > :max [STOP]
```

```
FD :size RT 144
```

```
Star :size + 1 :max
```

```
END
```

?

What's a *list* in Logo? What is list processing about? (The answer's worth several books.)



## Yugoslavia

This one's Atari Logo, and shows one way of drawing the gold outline around the star. You might think of other ways to suit your Logo:

```

TO Yugoslavia
  SetField
  Blocks
  GoldStar
  RedStar
  END

  TO SetField
    CS HT FS
    SETBG :white
    SETPC 0 :blue
    SETPC 1 :red
    SETPC 2 :yellow
  END

  TO Blocks
    PU SETPOS [-158 95] SETH 90 PD
    SETPN 0
    Block 320 63
    PU SETPOS [161 -93] SETH 270 PD
    SETPN 1
    Block 320 63
  END

  TO Block :length :width
    REPEAT :width / 2 [FD :length RT 90 FD 1 RT 90
      FD :length LT 90 FD 1 LT 90]
  END

  TO GoldStar
    SETPN 2
    PU HOME SETH 90 PD
    Star 1 140
  END

  TO RedStar
    SETPN 1
    PU HOME SETH 90 PD
    Star 1 120
  END

  TO Star :size :max
    IF :size > :max [STOP]
    FD :size RT 144
    Star :size + 1 :max
  END

  MAKE "white 7
  MAKE "yellow 12
  MAKE "red 25
  MAKE "blue 64

```

?

How would you write a program to draw the flags of the republics that form the federation of Yugoslavia?

## Final Flaps

At one stage I toyed with the idea of 26 flags, one for each letter of the alphabet, but realised that I would have to leave out too many very good flags, so I chose this selection. Their styles are different, in both vexillological and programming senses, and that's good. There is no one right way to design flags, write programs, run the country, or do many other things. Look at them all and think about it.

Unusually for a Logo book, there hasn't been much recursion, only the occasional tail recursion in Star procedures and the like. Perhaps a really recursive flag would cause chaos in the flagmakers' workshops, but you could have a try.

It might be argued that by giving you finished programs I'm simply giving away answers, without you having to think. I don't believe that is the case: there seem to be more questions than answers. Perhaps it's a case of: 'Here's the answer, what was the question?'

There are many more flags left in the world for you to try, and the ones you work on will depend as much on the machine you're using as your programming skills. It's up to you now, and should you ever tire of flags, you could try your hand at corporate logos. The Australia Post emblem might be one to start with. Drawing logos with Logo seems quite appropriate.

## Appendix: Some Logo Special Words

Brief descriptions of some of the words in the Logo versions used in this book.

### Apple Logo

The oldest Logo in this book, dating from 1982. There's no FILL. The range of colour is limited: black, white, green, violet, orange and blue. Put green and orange together and you'll wish you hadn't.

### Apple Logo // and LCSI Logo II

Apple Logo // is like its predecessor, but has a FILL command, which will fill outlines of the same colour, ie. to fill with blue the outline must be blue.

LCSI Logo II replaces AL //. Its FILL ignores the outline colour, ie. you can fill a white outline with blue, which can be a nuisance at times. Turtle can be reshaped, and can STAMP its shape.

### Atari Logo

Atari Logo is in a cartridge and runs on the 400, 800, and XL machines. It has four Turtles (which can sense when they run into things) and a range of 128 colours, although only 4 can be shown at once. There's no FILL.

- |       |  |
|-------|--|
| SETPC | Changes the colour of the Turtle's pens. The catch (and it can be a useful trick) is that SETPC can change the colour of things already drawn.                                     |
| SETPN | Changes the pen the Turtle is using. To draw different colours, use SETPC to fill, as it were, the pens with ink, then SETPN to change pens, and therefore the colour being drawn. |

### BBC: Acornsoft and Logotron

The scale on the BBC's screen is about 1:4 compared with other machines, and the number of colours (and the resolution) depends on the display mode in use. There's no FILL, but SETNIB sets the Turtle's pen to draw filled triangles. The catch is that the last three points plotted are the ones filled: be careful.

- |                   |  |
|-------------------|--|
| SETMODE           | Changes the screen mode, ie. its resolution.   |
| PAL, VDU [19 ...] | In AcornSoft and Logotron respectively, these set up the palette, ie. the range of colours that can be used. The number of colours depends on the screen mode.   |
| SETNIB            | Can produce a wide range of effects, the most useful one being to fill triangles (and therefore other shapes). The area filled is within the last three points plotted. The Safe procedure in some programs in this book is to ensure that wrong areas are not filled. |

## LogoWriter

There are versions for the Apple, MS-DOS machines, and the Commodore 64. On the Apple, there are the usual limitations. The colours on an MS-DOS machine will depend on the graphics card, CGA, EGA or PS/2. On a PS/2 machine, 256 colours are allegedly possible. The main advantage of LogoWriter is its very easy environment. Limitations are the lack of WINDOW or NOWRAP.

SETC            Equivalent to SETPC

## Macintosh Logo

Macintosh Logo doesn't use colour; it uses shading patterns instead, and of course everything happens in windows. FILL is not implemented, but rectangles, arcs and other shapes can be filled as they are drawn.

ARC	Draws an arc or wedge shaped section of an oval, centered within the rectangular area specified by the input coordinates. It would be much easier to use if the inputs specified the centre, angles and radius.
FILLSH	Fills the shape specified by the input in the current pen pattern.
OVAL	Draws a circular or oval shape, within the rectangular area specified by the input coordinates.
PENPAINT	Puts the pen in painting mode: it completely covers any lines it passes over.
RECT	Draws a rectangle, whose position and size depend on the input coordinates. The rectangle may or may not be filled. The limitation is that rectangles must be horizontal or vertical.
SETPPATTERN	Sets the pattern which the pen uses for drawing and filling shapes. Equivalent to SETPC in other Logos.

## Further Reading:

### Vexillology:

- Crampton, W. *Collins Eyewitness Guide: Flag* Collins, 1989  
An excellent outline of the history and symbolism of flags.
- Crampton, W. *The Complete Guide to Flags* Kingfisher, 1989  
The history of flags, and descriptions of national and state flags.
- Department of Administrative Services *The Australian National Flag* Australian Government Publishing Service, 1988  
The specifications, history and protocol of the Australian flag.
- Inglefield, E. *Pocket Book of Flags* Kingfisher, 1988  
A concise book of flags.
- Milligan, J. *Brown's Signalling* Brown, Son and Ferguson, 1988  
The latest edition of a book in print since 1933. Code flags and a comprehensive list of signals.
- More, E., Ross, D. and Saunders, M. *Collins Gem Guide: Flags of the World* Collins, 1986  
A small, but comprehensive book.
- Talocci, M. *Guide to the Flags of the World* Sidgwick and Jackson, 1982  
A useful book.

### Logo:

- Abelson, H. and diSessa, A. *Turtle Geometry* MIT Press 1981  
An unusual and challenging mathematics book.
- Clayson, J. *Visual Modeling with Logo* MIT Press  
Logo as art
- Goldenburg, E. and Feurzeig, W. *Exploring Language with Logo* MIT Press  
Grammar as Logo procedures.
- Harvey, B. *Computer Science Logo Style* MIT Press  
In three volumes. Computer Science from the Logo perspective.
- Newell, B. *Turtle Confusion* Curriculum Development Centre, 1988  
A book of challenging puzzles and riddles.