

CHAPTER 13

Random!

Heads and tails

When a coin is tossed, it is equally likely that the 'head' or 'tail' will show, at least if the coin is a fair coin. Your computer can simulate the tossing of a coin by using the inbuilt RND function.

RND(1) returns a (pseudo) random number in the range from 0 up to 1 (but not including 1). If you switch on your computer and type the following:

```
FOR I=1 TO 5:PRINT RND(1):NEXT I
```

you will get a sequence of numbers. The sequence which appears depends upon what operating system your computer has and whether it is a BBC micro or an Electron. It is quite likely that one of the following two sequences will appear.

```
A: 0.996322632  
   0.94140622  
   3.85285891E-3  
   1.93786994E-3  
   0.497069362
```

```
B: 0.440400091  
   0.183034912  
   0.279525887  
   0.266425445  
   0.897804189
```

If you switch your computer off and then on again (not recommended) and repeat the instruction

```
FOR I = 1 TO 5 : PRINT RND(1):NEXT I
```

you will get the same sequence of numbers. To overcome this it is usual to start any program involving random numbers with a line like

$Y = \text{RND}(-\text{TIME})$

which has the effect of starting a new sequence of random numbers. In general, if X is negative then $\text{RND}(X)$ starts a new sequence of random numbers. If X is 0 then the resulting number is the same as the last one. If X is greater than 1, $\text{RND}(X)$ generates an integer between (and possibly including) 1 and X .

The following short program simulates the tossing of a fair coin. A list is printed out showing whether a head (H) or tail (T) appears. 100 such letters are printed and a count of the number of heads and tails is displayed.

Listing 13.1

LIST

```
10 REM Heads and tails
20 MODE 1:COLOUR 3:PRINT ' TAB(12);"H
eads and tails"'
30 PRINT "This program simulates the
tossing of a fair coin 100 times."'
40 PRINT TAB(16);"Press Y ";
50 REPEAT:UNTIL GET$="Y"
60 K=0
70 REPEAT
80 Y=RND(-TIME)
90 K=K+1:J=0:COLOUR 2:PRINT ' TAB(1
3);"Run number ";K ':COLOUR 1
100 FOR I=1 TO 100
110 A$="H":IF RND(2)=2 THEN A$="T":J
=J+1

120 PRINT A$;
130 NEXT
140 PRINT '"Number of heads ";100-J;
" tails ";J
150 COLOUR 3:PRINT CHR$(7) ' TAB(10)
;"Another go? Y or N ";
160 REPEAT:G$=GET$:UNTIL G$="Y" OR G$
="N"
170 UNTIL G$="N"
180 CLS:PRINT '"Bye for now.":END
```

RUN

Heads and tails

This program simulates the tossing of a fair coin 100 times.

Press Y

Run number 1

HHHHHTTTTHHTTHHHHHHTTTTTTHHTHTHHHTHHHHHTHH
HTTTTTHTHHHTTTTTHTTTTTHTHTHHHTTHHTTHHH
HTTHHHHTHTTTTTHTTHHTH

Number of heads 53 tails 47

Another go? Y or N

Run number 2

THHTTHHTHHHTTTTTHTHTTHTHTTTTTHTHTHHHTTHTH
HHTHHHHHTTTTTHTHHHTHTTHHHHTHHHTTHTHTHTHT
HTTTTTHTTHHHHTTHHTTHHTH

Number of heads 50 tails 50

Another go? Y or N

Run number 3

TTHTHTTTTTTTTTHTHTTTTTHTHTHHHTTHTTHHTHHHTHH
THTHTHTHHHHHTTTTTHTTHTHHHTTHHHHHHTTTTTTHHH
HHHTHTTTHTHTHHHTTHTHH

Number of heads 49 tails 51

Another go? Y or N

Change line 80 to the following line.

```
80 Y = RND(-1)
```

You should now notice that the same sequence of heads and tails appears every time the program is run. That's why we use

```
80 Y = RND(-TIME)
```

to randomise the sequence of random numbers.

Of dice and men

If a fair six-sided die is thrown then one of the six numbers 1, 2, 3, 4, 5 and 6 will appear, none more likely than another. Once again, your computer may be used to simulate dice throwing. The next program illustrates the result of rolling a die one hundred and twenty times. A count of each number thrown is printed at the end.

Listing 13.2

LIST

```
10 REM Die rolling
20 MODE 1:COLOUR 3:PRINT ' TAB(14);"Die
ie rolling"'
30 PRINT "This program simulates the
rolling of a fair die 120 times."'
40 PRINT TAB(16);"Press Y ";
50 REPEAT:UNTIL GET$="Y"
60 K=0:DIM A(6)
70 REPEAT
80 Y=RND(-TIME)
90 K=K+1:FOR I=1 TO 6:A(I)=0:NEXT
100 COLOUR 2:PRINT ' TAB(12);"Roll n
umber ";K ':COLOUR 1
110 FOR I=1 TO 120
120 L=RND(6):PRINT ;L;" ";A(L)=A(L)
+1
130 NEXT
140 PRINT ':FOR I=1 TO 6:PRINT "Numbe
r of ";I;"'s = ";A(I):NEXT
150 COLOUR 3:PRINT CHR$(7) ' TAB(10)
;"Another go? Y or N ";
```

Chapter 13 - Random!

```
160 REPEAT:G$=GET$:UNTIL G$="Y" OR G$
="N"
170 UNTIL G$="N"
180 CLS:PRINT '"Bye for now.':END
RUN
```

Die rolling

This program simulates the rolling of a fair die 120 times.

Press Y

Roll number 1

```
6 6 1 3 6 2 3 3 4 5 2 1 1 2 5 5 4 1 5 6
1 4 4 4 3 3 1 3 1 4 4 5 5 6 2 5 6 5 4 5
3 2 3 6 3 3 2 4 5 1 4 3 3 6 4 2 2 4 6 5
2 2 6 2 6 5 6 4 1 6 2 6 5 4 4 3 4 5 4 4
4 4 5 5 4 5 2 2 1 5 6 1 3 5 5 5 6 1 2 3
1 4 2 4 3 3 5 4 2 1 1 4 6 1 1 3 1 5 6 3
```

```
Number of 1's = 18
Number of 2's = 17
Number of 3's = 19
Number of 4's = 25
Number of 5's = 23
Number of 6's = 18
```

Another go? Y or N

Roll number 2

```
4 2 2 4 6 5 1 1 3 2 4 2 2 5 3 2 2 3 6 2
1 6 2 6 5 2 6 4 1 2 5 5 4 2 1 1 4 4 1 3
1 4 1 2 6 1 3 5 1 3 5 2 5 1 6 1 3 2 1 4
4 5 3 4 1 4 3 5 1 6 2 1 5 2 3 2 2 6 1 5
2 5 1 2 1 2 2 1 2 4 1 1 2 5 6 3 5 6 4 5
4 4 6 1 1 6 3 2 1 4 6 1 6 6 3 2 4 6 6 6
```

Essential Maths on the BBC and Electron Computers

Number of 1's = 27
Number of 2's = 27
Number of 3's = 13
Number of 4's = 18
Number of 5's = 16
Number of 6's = 19

Another go? Y or N

Another version of the program appears next. This one includes a display of the face of the die.

Listing 13.3

LIST

```
10 REM Picture die
20 MODE 1:COLOUR 3:PRINT ' TAB(14);"D
ie rolling"'
30 PRINT "This program simulates the
rolling of a fair die."'
40 REM Define characters
50 VDU 23,200,255,255,192,192,192,192
,192,192
60 VDU 23,201,255,255,0,0,0,0,0,0
70 VDU 23,202,255,255,3,3,3,3,3,3
80 VDU 23,203,192,192,192,192,192,192
,192,192
90 VDU 23,204,3,3,3,3,3,3,3,3
100 VDU 23,205,192,192,192,192,192,192
,255,255
110 VDU 23,206,0,0,0,0,0,0,255,255
120 VDU 23,207,3,3,3,3,3,3,255,255
130 VDU 23,208,60,126,255,255,255,255,
126,60
140 K=0:DIM A(6),A$(6),D$(6,6)
150 FOR I=0 TO 6:FOR J=0 TO 6:READ A:A
$(I)=A$(I)+CHR$(A):NEXT:NEXT
160 FOR I=1 TO 6:FOR J=0 TO 6:READ A:D
```

```

$(I, J)=A$(A):NEXT:NEXT
  170 PRINT TAB(16);"Press Y ";
  180 REPEAT:UNTIL GET$="Y"
  190 REPEAT
  200 Y=RND(-TIME)
  210 K=K+1
  220 COLOUR 2:PRINT ' ' TAB(12);"Roll n
umber ";K ':COLOUR 1
  230 L=RND(6):A(L)=A(L)+1
  240 FOR J=0 TO 6:PRINT TAB(16);D$(L, J
):NEXT
  250 PRINT '"Count: ";:FOR I=1 TO 3:P
RINT ;I; "'s = ";A(I);" ";:NEXT
  260 PRINT ' " ";:FOR I=4 TO 6:PR
INT ;I; "'s = ";A(I);" ";:NEXT
  270 COLOUR 3:PRINT CHR$(7) ' ' TAB(10)
;"Another go? Y or N ";
  280 REPEAT:G$=GET$:UNTIL G$="Y" OR G$
="N"
  290 UNTIL G$="N"
  300 CLS:PRINT '"Bye for now.":END
  310 REM Data for design
  320 DATA 200,201,201,201,201,201,202
  330 DATA 203,32,32,32,32,32,204
  340 DATA 203,208,32,32,32,32,204
  350 DATA 203,32,32,208,32,32,204
  360 DATA 203,32,32,32,32,208,204
  370 DATA 203,208,32,32,32,208,204
  380 DATA 205,206,206,206,206,206,207
  390 REM Data for each die face
  400 DATA 0,1,1,3,1,1,6
  410 DATA 0,4,1,1,1,2,6
  420 DATA 0,4,1,3,1,2,6
  430 DATA 0,5,1,1,1,5,6
  440 DATA 0,5,1,3,1,5,6
  450 DATA 0,5,1,5,1,5,6

```

Rolling two dice simultaneously can be simulated just as easily. The

Essential Maths on the BBC and Electron Computers

possible score on each roll is one of the numbers from 2 to 12. As you are no doubt aware, some scores are more likely to occur than others. This fact should become apparent with either of the next two programs. If two dice are rolled several times then the expected proportion of time (or probability) that each score occurs is given in the next table.

Score	Probability
2	1/36
3	2/36
4	3/36
5	4/36
6	5/36
7	6/36
8	5/36
9	4/36
10	3/36
11	2/36
12	1/36

Listing 13.4

LIST

```
10 REM Two dice rolling
20 MODE 1:COLOUR 3:PRINT ' TAB(12);"T
wo dice rolling"'
30 PRINT "This program simulates the
rolling of two fair dice."'
40 K=0:DIM A(12)
50 PRINT TAB(16);"Press Y ";
60 REPEAT:UNTIL GET$="Y"
70 REPEAT
80 Y=RND(-TIME)
90 K=K+1
100 COLOUR 2:PRINT ' ' TAB(12);"Roll n
umber ";K ':COLOUR 1
110 FOR I=1 TO 60
120 L=RND(6):M=RND(6):A(L+M)=A(L+M)+
1
130 PRINT ;"* ";L;" ";M;" *";
140 NEXT
150 PRINT ' "'Score count: "
160 FOR I=2 TO 12
```

```

170 PRINT ;I;"'s = ";A(I);" ";:IF I=
5 OR I=9 THEN PRINT
180 NEXT
190 COLOUR 3:PRINT CHR$(7) ' ' TAB(10)
;"Another go? Y or N ";
200 REPEAT:G$=GET$:UNTIL G$="Y" OR G$
="N"
210 UNTIL G$="N"
220 CLS:PRINT '"Bye for now.':END

```

RUN

Two dice rolling

This program simulates the rolling of
two fair dice.

Press Y

Roll number 1

```

* 2 2 ** 1 2 ** 1 1 ** 4 6 ** 2 6 *
* 4 3 ** 3 4 ** 1 6 ** 3 4 ** 2 2 *
* 5 6 ** 5 3 ** 1 6 ** 1 3 ** 1 4 *
* 1 5 ** 6 5 ** 2 6 ** 2 4 ** 2 6 *
* 2 5 ** 3 6 ** 3 4 ** 4 3 ** 6 6 *
* 4 6 ** 5 3 ** 5 3 ** 2 4 ** 5 5 *
* 1 5 ** 5 6 ** 6 3 ** 3 3 ** 1 5 *
* 1 5 ** 3 2 ** 3 1 ** 5 6 ** 2 2 *
* 2 2 ** 1 3 ** 1 5 ** 2 5 ** 2 4 *
* 4 1 ** 5 2 ** 6 5 ** 3 2 ** 2 1 *
* 2 2 ** 6 3 ** 4 4 ** 1 6 ** 2 1 *
* 3 1 ** 6 3 ** 4 6 ** 2 1 ** 4 1 *

```

Score count:

```

2's = 1 3's = 4 4's = 9 5's = 5
6's = 9 7's = 11 8's = 7 9's = 4
10's = 4 11's = 5 12's = 1

```

Essential Maths on the BBC and Electron Computers

Another go? Y or N

Roll number 2

```
* 5 1 ** 4 5 ** 6 2 ** 4 2 ** 3 1 *
* 4 3 ** 3 5 ** 1 5 ** 1 6 ** 4 2 *
* 6 2 ** 3 6 ** 2 1 ** 4 6 ** 3 4 *
* 1 5 ** 2 1 ** 2 3 ** 5 1 ** 4 4 *
* 6 2 ** 2 2 ** 3 3 ** 3 5 ** 5 4 *
* 6 2 ** 3 6 ** 1 1 ** 3 2 ** 4 1 *
* 1 1 ** 6 3 ** 5 2 ** 3 6 ** 6 2 *
* 6 6 ** 6 2 ** 2 2 ** 2 2 ** 5 3 *
* 5 4 ** 2 4 ** 6 2 ** 5 6 ** 6 2 *
* 5 2 ** 6 2 ** 3 4 ** 3 6 ** 6 1 *
* 2 3 ** 3 2 ** 1 4 ** 4 5 ** 5 3 *
* 4 4 ** 6 6 ** 5 1 ** 5 1 ** 5 5 *
```

Score count:

```
2's = 4 3's = 3 4's = 10 5's = 9
6's = 17 7's = 16 8's = 29 9's = 18
10's = 7 11's = 4 12's = 3
```

Another go? Y or N

Listing 13.5

LIST

```
10 REM Two dice rolling with picture
20 MODE 1:COLOUR 3:PRINT ' TAB(12);"T
wo dice rolling"'
30 PRINT "This program simulates the
rolling of two fair dice."'
40 REM Define characters
50 VDU 23,200,255,255,192,192,192,192
,192,192
60 VDU 23,201,255,255,0,0,0,0,0,0
70 VDU 23,202,255,255,3,3,3,3,3,3
80 VDU 23,203,192,192,192,192,192,192
,192,192
```

Chapter 13 - Random!

```
90 VDU 23,204,3,3,3,3,3,3,3,3
100 VDU 23,205,192,192,192,192,192,192,192
,255,255
110 VDU 23,206,0,0,0,0,0,0,255,255
120 VDU 23,207,3,3,3,3,3,3,255,255
130 VDU 23,208,60,126,255,255,255,255,
126,60
140 K=0: DIM A(12),A$(6),D$(6,6)
150 FOR I=0 TO 6: FOR J=0 TO 6: READ A:A
$(I)=A$(I)+CHR$(A): NEXT: NEXT
160 FOR I=1 TO 6: FOR J=0 TO 6: READ A:D
$(I,J)=A$(A): NEXT: NEXT
170 PRINT TAB(16); "Press Y ";
180 REPEAT: UNTIL GET$="Y"
190 REPEAT
200 Y=RND(-TIME)
210 K=K+1
220 COLOUR 2: PRINT ' ' TAB(12); "Roll n
umber "; K ': COLOUR 1
230 L=RND(6): M=RND(6): A(L+M)=A(L+M)+
1
240 FOR J=0 TO 6: PRINT TAB(10); D$(L,J
); SPC(6); D$(M,J): NEXT
250 PRINT '"Score count: "
260 FOR I=2 TO 12
270 PRINT ; I; "'s = "; A(I); " "; : IF I=
5 OR I=9 THEN PRINT
280 NEXT
290 COLOUR 3: PRINT CHR$(7) ' ' TAB(10)
; "Another go? Y or N ";
300 REPEAT: G$=GET$: UNTIL G$="Y" OR G$
="N"
310 UNTIL G$="N"
320 CLS: PRINT '"Bye for now.": END
330 REM Data for design
340 DATA 200,201,201,201,201,201,202
350 DATA 203,32,32,32,32,32,204
360 DATA 203,208,32,32,32,32,204
```

Essential Maths on the BBC and Electron Computers

```
370 DATA 203,32,32,208,32,32,204
380 DATA 203,32,32,32,32,208,204
390 DATA 203,208,32,32,32,208,204
400 DATA 205,206,206,206,206,206,207
410 REM Data for each die face
420 DATA 0,1,1,3,1,1,6
430 DATA 0,4,1,1,1,2,6
440 DATA 0,4,1,3,1,2,6
450 DATA 0,5,1,1,1,5,6
460 DATA 0,5,1,3,1,5,6
470 DATA 0,5,1,5,1,5,6
```

Playing cards

A regular pack of playing cards has 52 cards. In a well shuffled pack any one of the 52 cards is likely to appear at the top. The next program, Cards, illustrates how a computer can simulate picking a card from a well shuffled pack of playing cards. On each new selection it is assumed that the previously selected card is replaced and the cards are well shuffled again.

Listing 13.6

LIST

```
10 REM Cards
20 MODE 1:COLOUR 3:PRINT ' TAB(17);"C
ards"'
30 PRINT "This program simulates the
drawing of a card from a well shuffled p
ack." '
40 K=0:DIM A$(12),B$(3),C(3)
50 FOR I=0 TO 12:READ A$(I):NEXT
60 FOR I=0 TO 3:READ B$(I):NEXT
70 FOR I=0 TO 3:READ C(I):NEXT
80 PRINT TAB(16);"Press Y ";
90 REPEAT:UNTIL GET$="Y"
100 REPEAT
110 Y=RND(-TIME)
120 K=K+1
130 COLOUR 2:PRINT ' ' TAB(10);"Select
ion number ";K ':COLOUR 1
```

```
140 L=RND(13)-1:M=RND(4)-1
150 COLOUR C(M):PRINT TAB(15+(L=9));A
$(L);" ";B$(M)
160 COLOUR 3:PRINT CHR$(7) ' ' TAB(10)
;"Another go? Y or N ";
170 REPEAT:G$=GET$:UNTIL G$="Y" OR G$
="N"
180 UNTIL G$="N"
190 CLS:PRINT '"Bye for now.':END
200 REM Data
210 DATA A,2,3,4,5,6,7,8,9,10,J,Q,K
220 DATA Clubs,Diamonds,Hearts,Spades
230 DATA 2,1,1,2
```

RUN

Cards

This program simulates the drawing of a card from a well shuffled pack.

Press Y

Selection number 1

Q Spades

Another go? Y or N

Selection number 2

6 Diamonds

Another go? Y or N

Selection number 3

Q Clubs

Another go? Y or N

Essential Maths on the BBC and Electron Computers

Selection number 4

6 Diamonds

Another go? Y or N

Selection number 5

7 Hearts

Another go? Y or N

Selection number 6

3 Hearts

Another go? Y or N

In the program Cards the computer first selects a number between 0 and 12 which (by adding 1) determines the number appearing on the card. Next, it selects a number between 0 and 3, this determines which of the four suits the card represents. An alternative way would be to select a number between 0 and 51:

$$K = \text{RND}(52) - 1$$

and pick off the suit and number of the card from this number. This is achieved by the following line.

$$L = K \text{ DIV } 4; M = K \text{ MOD } 4$$

The number L now determines the card number while M determines the suit.

The program Cards selects one card from a well shuffled pack. With each new selection the card is replaced and the pack is reshuffled. What if we want to shuffle the pack once and then list the cards as they appear in sequence from top to bottom? A different routine is required to achieve this. We number the cards 0 to 51 and then randomly rearrange these 52 numbers. This rearranging is done systematically; first the first number is exchange randomly with one of the other 51 numbers. Then the second

number is exchanged with one of the remaining 50 numbers. And so on. The program Card Shuffle shows the technique.

Listing 13.7

```

LIST
  10 REM Card shuffle
  20 MODE 1:COLOUR 3:PRINT ' TAB(14);"C
ard shuffle"'
  30 PRINT "This program illustrates th
e shuffling of a pack of cards.'"
  40 K=0:DIM A$(12),B$(3),C(3),D%(51)
  50 FOR I=0 TO 12:READ A$(I):NEXT
  60 FOR I=0 TO 3:READ B$(I):NEXT
  70 FOR I=0 TO 3:READ C(I):NEXT
  80 FOR I=0 TO 51:D%(I)=I:NEXT
  90 PRINT TAB(16);"Press Y ";
100 REPEAT:UNTIL GET$="Y"
110 REPEAT
120 Y=RND(-TIME)
130 K=K+1
140 COLOUR 2:PRINT ' TAB(10);"Shuffl
e number ";K ':COLOUR 1
150 REM Mixing
160 FOR I%=0 TO 50
170 L%=RND(52-I%)+I%-1
180 REM L% satisfies I% <= L <= 51
190 T%=D%(I%):D%(I%)=D%(L%):D%(L%)=T
%
200 NEXT
210 FOR I%=0 TO 51
220 L%=D%(I%) DIV 4:M%=D%(I%) MOD 4:
IF L%<> 9 THEN PRINT " ";
230 COLOUR C(M%):PRINT A$(L%);" ";B$
(M%);" ";
240 NEXT
250 COLOUR 3:PRINT CHR$(7) ' TAB(10)
;"Another go? Y or N ";
260 REPEAT:G$=GET$:UNTIL G$="Y" OR G$

```

Essential Maths on the BBC and Electron Computers

```
= "N"  
270 UNTIL G$="N"  
280 CLS:PRINT "Bye for now.":END  
290 REM Data  
300 DATA A,2,3,4,5,6,7,8,9,10,J,Q,K  
310 DATA Clubs ,D'mond,Hearts,Spades  
320 DATA 2,1,1,2
```

RUN

Card shuffle

This program illustrates the shuffling
of a pack of cards.

Press Y

Shuffle number 1

J Hearts	3 Clubs	Q Spades	6 D'mond
4 D'mond	7 D'mond	8 Spades	4 Spades
5 D'mond	2 D'mond	4 Hearts	Q D'mond
2 Spades	5 Spades	3 D'mond	9 Hearts
6 Spades	5 Hearts	9 Clubs	8 D'mond
6 Hearts	10 Clubs	10 Hearts	6 Clubs
Q Clubs	10 D'mond	J Spades	9 D'mond
2 Hearts	7 Spades	K D'mond	3 Spades
K Clubs	K Hearts	J D'mond	A Clubs
4 Clubs	J Clubs	10 Spades	A D'mond
A Spades	7 Clubs	K Spades	5 Clubs
A Hearts	3 Hearts	8 Hearts	8 Clubs
2 Clubs	Q Hearts	9 Spades	7 Hearts

Another go? Y or N

Shuffle number 2

```

4 Hearts   8 Spades   A Spades   9 D'mond
K Spades   6 D'mond    J Clubs    J Hearts
J Spades   3 Spades    4 D'mond   3 D'mond
6 Hearts   Q Spades    A Hearts   K Hearts
9 Clubs    6 Clubs     Q D'mond   2 Hearts
9 Spades   8 Hearts    10 Hearts  2 Clubs
3 Hearts   5 Clubs     7 D'mond   5 D'mond
2 D'mond   Q Clubs     Q Hearts   10 D'mond
K Clubs    4 Spades    5 Hearts   A D'mond
9 Hearts   2 Spades    J D'mond   3 Clubs
A Clubs    K D'mond    8 Clubs    6 Spades
5 Spades   10 Spades   7 Clubs    8 D'mond
4 Clubs    7 Hearts    10 Clubs   7 Spades
    
```

Another go? Y or N

Non-equally likely events

Most of the examples we have looked at so far have the property that any one of the events that can occur is as likely as any one of the others. The next example is different.

A bucket contains 100 coloured buttons. There are 6 red buttons, 54 blue ones and 40 green ones. To simulate the selection of a button from the bucket we use the following lines.

```

X = RND(1)
R$ = "RED"
IF X >= 0.06 THEN R$="BLUE"
IF X >= 0.60 THEN R$="GREEN"
PRINT R$
    
```

The program Buttons simulates the selection of a button from the bucket. 100 selections are made, after each selection the button is replaced.

Listing 13.8

```

LIST
    10 REM Buttons
    20 MODE 1:COLOUR 3:PRINT ' TAB(16);"B
uttons" '
    
```

Essential Maths on the BBC and Electron Computers

```
30 PRINT "This program illustrates pi
cking a      button from a bucket with 6
red, 54 blueand 40 green buttons.'"
40 K=0:DIM A(4)
50 PRINT TAB(16);"Press Y ";
60 REPEAT:UNTIL GET$="Y"
70 MODE 2
80 REPEAT
90 Y=RND(-TIME)
100 K=K+1
110 COLOUR 3:PRINT ' ' TAB(4);"Run num
ber ";K '
120 X=RND(1):R$="  Red":S=1
130 IF X>=0.06 THEN R$="  Blue":S=4
140 IF X>=0.60 THEN R$="  Green":S=2
150 COLOUR S:PRINT TAB(6);R$
160 A(S)=A(S)+1
170 COLOUR 5:PRINT '"Count:"
180 PRINT "  Red ";A(1)'" Blue ";A(4)
'"Green ";A(2)
190 COLOUR 6:PRINT CHR$(7) '"Another
go? Y or N ";
200 REPEAT:G$=GET$:UNTIL G$="Y" OR G$
="N"
210 UNTIL G$="N"
220 MODE 1:PRINT '"Bye for now.":END
```

RUN

Buttons

This program illustrates picking a
button from a bucket with 6 red, 54 blue
and 40 green buttons.

Press Y

Run number 1

Green

Count:
 Red 0
 Blue 0
 Green 1

Another go? Y or N

Run number 2

Blue

Count:
 Red 0
 Blue 1
 Green 1

Another go? Y or N

Run number 3

Green

Count:
 Red 1
 Blue 1
 Green 1

Another go? Y or N

