Doing things with words

Most computers can examine the words stored in variables and do various things with them. They can check the contents of a variable and see if it contains a particular word or letter. This is useful for checking the words input by someone using the program. Computers can also rearrange the letters or words in a different order and add them to letters in other variables. Below you can find out how you do these things in BASIC.

1

```
10 A$="I AM STUPID"
20 B$="ONLY FOOLS THINK"
30 C$=B$+" "+A$
RUN
ONLY FOOLS THINK I AM STUPID
```

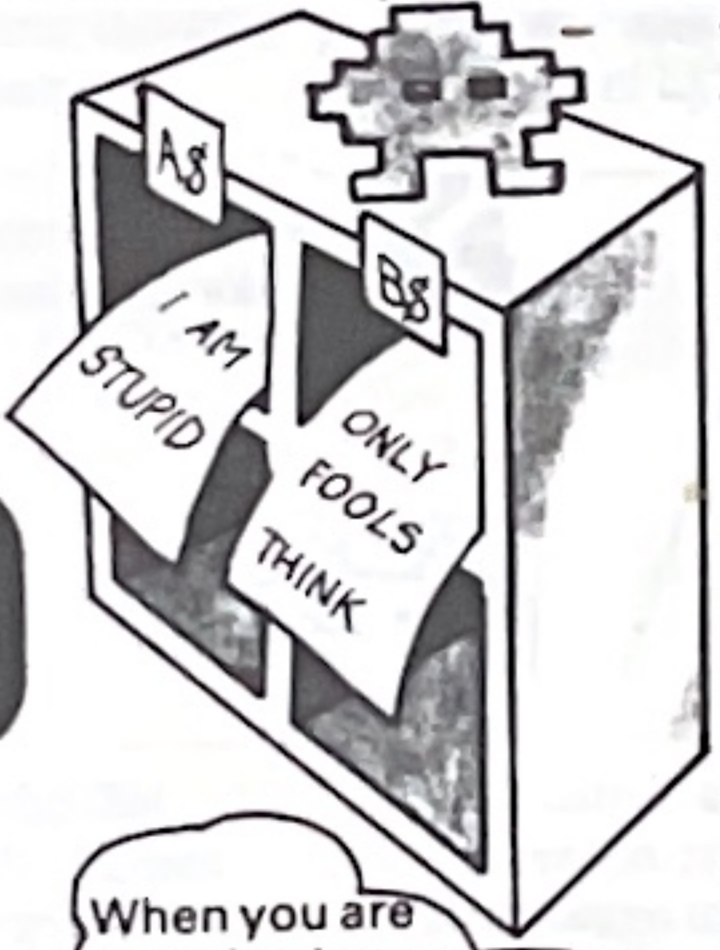
On most computers, but not the ZX81, you can leave out the word LET.

2

```
PRINT LEFT$(B$,4)
ONLY
PRINT LEFT$(B$,4)+" "+A$

ONLY I AM STUPID
```

You can add the contents of two variables like this. You need the space between quotation marks to leave a space between the words.



You can also add parts of variables, like this. LEFT$(B$,4) means take the first four letters from the left of B$.

3

```
PRINT RIGHT$(A$,6)
STUPID
```

To tell the computer to use letters from the right you use RIGHT$ with the name of the string and the numbers of letters you want.

When you are counting letters, count spaces and punctuation too.

4

```
PRINT MID$(B$,6,5)
FOOLS
```

This tells the computer to take the middle letters. The first number tells it where to start and the second tells it how many letters to take.

5

```
10 K$="DING DONG!"
20 PRINT LEN(K$)
RUN
10
```

You can also find out the length of a string – the number of letters, spaces and symbols it contains. To do this you use LEN, short for length.

IF A$="COMPUTER BOOK"
what is LEFT$ (A$,8)?
RIGHT$ (A$,10)?
MID$ (A$,5,8)?

Note for Sinclair users

```
PRINT A$(6 TO 11)
STUPID
PRINT B$(14 TO 16)
INK
```

This means take letters numbers 6 to 11.

The Sinclair computers do not use LEFT$, RIGHT$ and MID$, but you can tell the computer to take any letters you want as shown above.

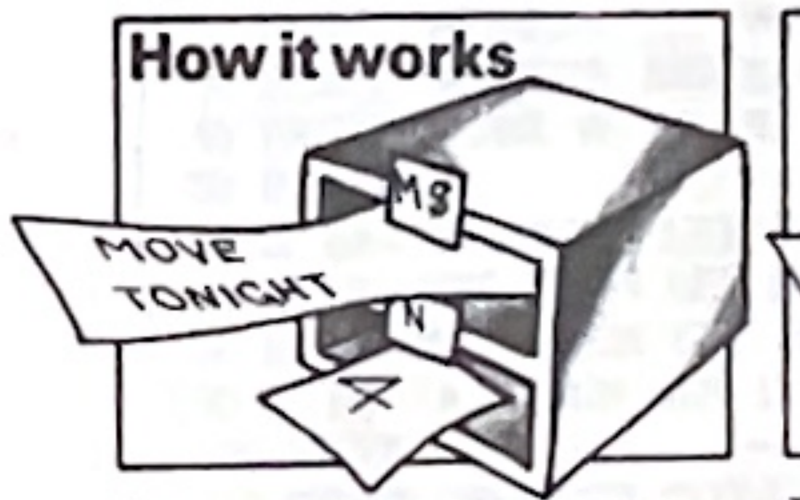
Codemaker program

This program automatically puts words into code. Similar, but much more complex programs are used by intelligence services to write and crack codes.

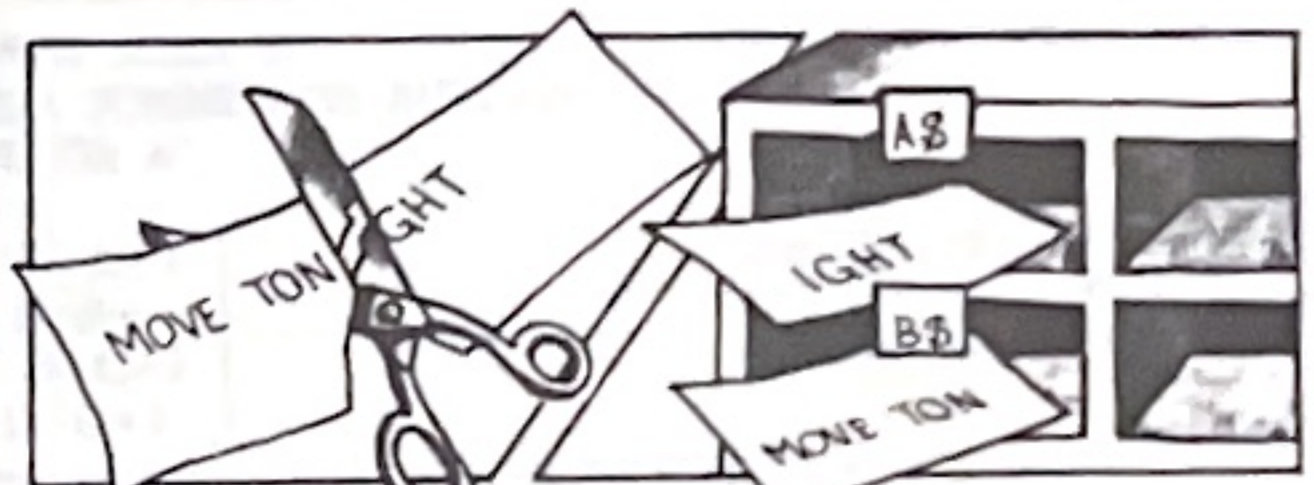
The easiest way to understand this program is to write a secret message on a piece of paper, then work through the lines of the program carrying out the computer's tasks on your message and writing them down.

```
5 LET C$=""   ]
7 LET D$=""   ]-------------------------------- Sets up empty string variables.
10 PRINT "TYPE IN A SHORT MESSAGE"
20 INPUT M$
30 PRINT "NOW TYPE IN A SECRET          ]
   NUMBER BETWEEN 2 AND";LEN(M$)-1     ]--- This means the length of your
                                             message minus 1.
40 INPUT N                                   N (your secret number) letters from
50 LET A$=RIGHT$(M$,N) ]-------------------- the right of M$.
60 LET B$=LEFT$(M$, LEN(M$)-N) ]------------ The length of M$ minus N number of
                                             letters from the left of M$ (i.e. the rest
                                             of the letters).
70 LET M$=A$+B$        ]-------------------- Replaces the letters in M$ with A$+B$.
80 FOR I=1 TO LEN(M$) STEP 2 ]-------------- From 1 to the number of letters in your message
90 LET C$=C$+MID$(M$,I,1)                    going up in twos, i.e. 1, 3, 5, etc. Each time the I
100 NEXT I                                   loop is repeated line 90 takes one letter from
                                             position I of M$ and puts it in C$.
110 FOR J=2 TO LEN(M$) STEP 2 ]------------- From 2 to the number of letters in your
120 LET D$=D$+MID$(M$,J,1)                   message, going up in twos, i.e. 2, 4, 6, etc.
130 NEXT J                                   Works in the same way as the I loop.
140 LET M$=C$+D$       ]-------------------- Replaces the letters in M$ again.
150 PRINT "CODED MESSAGE IS"
160 PRINT M$
170 END
```

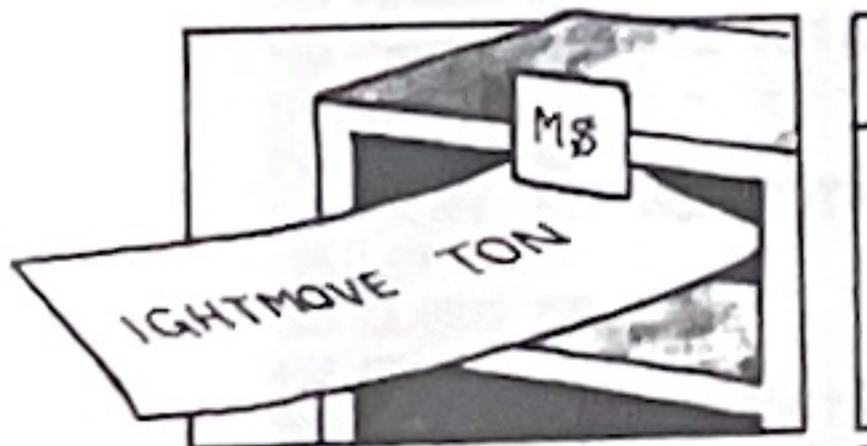
How it works



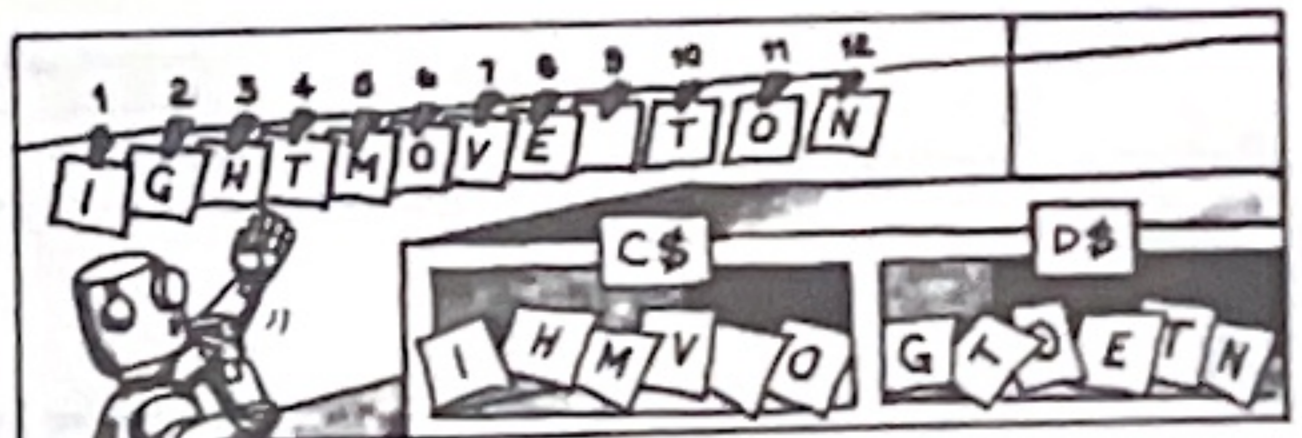
Suppose your message is "Move tonight" and your secret number is 4. These are stored in M$ and N.



In lines 50 and 60 the computer uses your secret number to divide the message. At line 50 it takes four letters from the right of the message and puts them in A$. At line 60 it puts the rest of the letters in B$.



At line 70 it adds A$ and B$. This puts the letters from the end of the message at the front.



Each time the I loop repeats it puts an odd-numbered letter in C$ (e.g. I, H, M, etc.). Each time the J loop repeats it puts an even numbered letter in D$ (e.g. G, T, O, etc.). Then it adds C$ and D$ to make the coded message.