

```

CI No F17
TxL 65254
Next: Default Format to be Carriage return <R>
Auto.
1 Init|Name |Town |S |
2 | | | |
3 R.T.|Covan |16 |Jeremy Grove |Hampton |17|
4 G.L.|Blake |37 |Osaston Road |Knowle |56|
5 A.J.|Hunt |25 |Latimer Road |Wythall |43|
6 H.A.|Fisher |32 |Florence Road |Henley |28|
7 C.D.|Beard |86 |Valley Road |Redditch |25|
8 | | | |
9 | | | |
10 | | | |
11 | | | |
12 | | | |
13 | | | |
14 | | | |
15 | | | |
16 | | | |
17 | | | |
    
```

Now two further such columns are needed, one in front of the town column and one at the end in front of the \$ column:

```

HI 117
Car 65254
Next: ABCDEFGILMNOPQRSTUVWXYZ:+-\>.'** arrows
Auto.
1 Init|Name |C |D |E |F |Town |G |H |I |
2 | | | | | | | | | |
3 R.T.|Covan |16 |Jeremy Grove |Hampton |17|
4 G.L.|Blake |37 |Osaston Road |Knowle |56|
5 A.J.|Hunt |25 |Latimer Road |Wythall |43|
6 H.A.|Fisher |32 |Florence Road |Henley |28|
7 C.D.|Beard |86 |Valley Road |Redditch |25|
8 | | | | | | | | | |
9 | | | | | | | | | |
    
```

You only want to print a portion of the sheet and so you use the Copy block option. Note that the Mail labels option intercepts the carriage returns that you have set and works out the way the printout should be organised to produce the desired effect - you don't actually get a new

line started wherever a carriage return has been inserted.

```

HI 117
Car 65254
Next: Copy block A3...H7
Auto.
1 Init|Name |A |B |C |D |E |F |Town |G |H |I |
2 | | | | | | | | | | |
3 R.T.|Covan |16 |Jeremy Grove |Hampton |17|
4 G.L.|Blake |37 |Osaston Road |Knowle |56|
5 A.J.|Hunt |25 |Latimer Road |Wythall |43|
6 H.A.|Fisher |32 |Florence Road |Henley |28|
7 C.D.|Beard |86 |Valley Road |Redditch |25|
8 | | | | | | | | | |
9 | | | | | | | | | |
10 | | | | | | | | | |
11 | | | | | | | | | |
12 | | | | | | | | | |
13 | | | | | | | | | |
14 | | | | | | | | | |
15 | | | | | | | | | |
16 | | | | | | | | | |
17 | | | | | | | | | |
    
```

Note that it is ABSOLUTELY ESSENTIAL for you to make sure that the end of the range is on a Carriage Return Format column as with cell H7 above. Omitting this requirement will do no harm but you will not get the correct print formatting. By typing M2 (for Mail labels, 2 across) after the request for destination you will be set up for having mail labels printed two abreast:

```

HI 117
Car 65254
Next: Destination Mail Labels, groups of 2
Auto.
1 Init|Name |A |B |C |D |E |F |Town |G |H |I |
2 | | | | | | | | | | |
3 R.T.|Covan |16 |Jeremy Grove |Hampton |17|
4 G.L.|Blake |37 |Osaston Road |Knowle |56|
5 A.J.|Hunt |25 |Latimer Road |Wythall |43|
6 H.A.|Fisher |32 |Florence Road |Henley |28|
7 C.D.|Beard |86 |Valley Road |Redditch |25|
8 | | | | | | | | | |
9 | | | | | | | | | |
10 | | | | | | | | | |
11 | | | | | | | | | |
12 | | | | | | | | | |
13 | | | | | | | | | |
14 | | | | | | | | | |
15 | | | | | | | | | |
16 | | | | | | | | | |
17 | | | | | | | | | |
    
```

Here is what they will look like. Not very organised, but your next task is to change the width of the columns so that various parts line up under one another:

```
G.L. Blake   A.J. Hunt
37 Osaston Road 25 Latimer Road
Knowle      Wythall

H.A. Fisher  C.D. Beard
32 Florence Road 86 Valley Road
Henley      Redditch

R.T. Cowan
16 Jeremy Grove
Hampton
```

After some width adjustment, here is what you can achieve. In this case the labels are going to be printed three abreast:

| GI | Town        | A | B | C  | D             | E | F        | G | H | I  |
|----|-------------|---|---|----|---------------|---|----------|---|---|----|
| 1  | InitName    |   |   | No | Road          |   | <Town    |   | > | IS |
| 2  | R.T. Cowan  |   |   | 16 | Jeremy Grove  |   | Hampton  |   |   | 17 |
| 3  | G.L. Blake  |   |   | 37 | Osaston Road  |   | Knowle   |   |   | 56 |
| 4  | A.J. Hunt   |   |   | 25 | Latimer Road  |   | Wythall  |   |   | 43 |
| 5  | H.A. Fisher |   |   | 32 | Florence Road |   | Henley   |   |   | 28 |
| 6  | C.D. Beard  |   |   | 86 | Valley Road   |   | Redditch |   |   | 25 |
| 7  |             |   |   |    |               |   |          |   |   |    |
| 8  |             |   |   |    |               |   |          |   |   |    |
| 9  |             |   |   |    |               |   |          |   |   |    |

| GI | Town        | A | B | C  | D             | E | F        | G | H | I  |
|----|-------------|---|---|----|---------------|---|----------|---|---|----|
| 1  | InitName    |   |   | No | Road          |   | <Town    |   | > | IS |
| 2  | R.T. Cowan  |   |   | 16 | Jeremy Grove  |   | Hampton  |   |   | 17 |
| 3  | G.L. Blake  |   |   | 37 | Osaston Road  |   | Knowle   |   |   | 56 |
| 4  | A.J. Hunt   |   |   | 25 | Latimer Road  |   | Wythall  |   |   | 43 |
| 5  | H.A. Fisher |   |   | 32 | Florence Road |   | Henley   |   |   | 28 |
| 6  | C.D. Beard  |   |   | 86 | Valley Road   |   | Redditch |   |   | 25 |
| 7  |             |   |   |    |               |   |          |   |   |    |
| 8  |             |   |   |    |               |   |          |   |   |    |
| 9  |             |   |   |    |               |   |          |   |   |    |
| 10 |             |   |   |    |               |   |          |   |   |    |
| 11 |             |   |   |    |               |   |          |   |   |    |
| 12 |             |   |   |    |               |   |          |   |   |    |
| 13 |             |   |   |    |               |   |          |   |   |    |
| 14 |             |   |   |    |               |   |          |   |   |    |
| 15 |             |   |   |    |               |   |          |   |   |    |
| 16 |             |   |   |    |               |   |          |   |   |    |
| 17 |             |   |   |    |               |   |          |   |   |    |

```
G.L. Blake   A.J. Hunt   H.A. Fisher
37 Osaston Road 25 Latimer Road 32 Florence Road
Knowle      Wythall      Henley

C.D. Beard
86 Valley Road 16 Jeremy Grove
Redditch      Hampton
```

Further adjustment may be needed with extra lines at the end, or changes in the spacing to suit your particular labels - this is achieved most simply by trial and error.

Note that if you had used the simple Print option rather than Mail labels the Carriage Return columns will have produced unpredictable and probably undesirable results.

Recap: We have seen how the Copy command is also used for saving and loading data on the disk drive and also for printing the data.

When saving or loading the choice of filename extension signals to THE CRACKER which type of data it should send or expect.

When printing you may have to make use of the OUT command and a special Carriage Return format in order to get the desired effect.

PARTITIONING THE SCREEN

As your worksheet gets larger you will, at times, want to work somewhere like the bottom right hand corner but still see your line and column headings which are at the top or left of the sheet. You can do this with the partition commands. These allow you to split the screen either horizontally, vertically or both so that you get two, or four smaller displays. The usual term for such split screens is 'windows'.

As well as moving the windows independently you also have the option of locking them so that as you move the cursor in one part, the other will move in a synchronized way. For example, if you scroll the lower window the appropriate headings will scroll past in the top of the screen. Cursor control can be moved between the windows with single key commands.

```

F17
65254
Auto.
AI Init
TXCL ABCDEFGIJLHNPQRSTUVWXYZ!+-\>./<.* arrows
Next: A B C D E F
1 <Init>Name |No |Road |Town |$ |
2 R.T. |Covan |16 |Jeremy Grove |Hampton |17|
3 G.L. |Blake |37 |Osmaston Road |Knowle |56|
4 A.J. |Hunt |25 |Latimer Road |Wythall |43|
5 H.A. |Fisher |32 |Florence Road |Henley |28|
6 C.D. |Beard |86 |Valley Road |Redditch |25|
7
8
9
10
11
12
13
14
15
16
17
    
```

First you are going to see the screen partitioned horizontally. As you type PH, for Partition Horizontally, you will see that a grid is put up on the screen. You must use this to judge where you want the split to take place. The number on the grid that you choose will be the first location of the second window on the screen.

```

F17
65254
Auto.
AI Init
TXCL ABCDEFGIJLHNPQRSTUVWXYZ!+-\>./<.* arrows
Next: A B C D E F
1 <Init>Name |No |Road |Town |$ |
2 R.T. |Covan |16 |Jeremy Grove |Hampton |17|
3 G.L. |Blake |37 |Osmaston Road |Knowle |56|
4 A.J. |Hunt |25 |Latimer Road |Wythall |43|
5 H.A. |Fisher |32 |Florence Road |Henley |28|
6 C.D. |Beard |86 |Valley Road |Redditch |25|
7
8
9
10
11
12
13
14
15
16
17
    
```

```

F17
65254
Auto.
AI Init
TXCL ABCDEFGIJLHNPQRSTUVWXYZ!+-\>./<.* arrows
Next: A B C D E F
1 <Init>Name |No |Road |Town |$ |
2 R.T. |Covan |16 |Jeremy Grove |Hampton |17|
3 G.L. |Blake |37 |Osmaston Road |Knowle |56|
4 A.J. |Hunt |25 |Latimer Road |Wythall |43|
5 H.A. |Fisher |32 |Florence Road |Henley |28|
6 C.D. |Beard |86 |Valley Road |Redditch |25|
7
8
9
10
11
12
13
14
15
16
17
    
```

You should now have two sets of row numbers starting from 1. Don't be confused by seeing the same information in both parts of the screen, it merely means that both windows are set to the same part of the worksheet. You can leave the top view where it is and on a big worksheet move the bottom view far down the sheet.

The next command to investigate is the backslash \. This is used to jump between horizontal window sections.

```

F17
65254
Auto.

A1 Init
Txl ABCDEFGIJLMNOPQRSTUWXYZ!+->/>.(/* arrows
Next:
  A B C D E F
1 <Init>Name No Road Town $
2 R.T. |Covan 16 |Jeremy Grove Hampton 17|
3 G.L. |Blake 17 |Osmaston Road Knowle 56|
4 G.L. |Blake 17 |Osmaston Road Knowle 56|
5 A.J. |Hunt 25 |Latimer Road Wythall 43|
6 H.A. |Fisher 32 |Florence Road Henley 28|
7 C.D. |Beard 86 |Valley Road Redditch 25|
8
9
10
11
12
13
14
15
16
    
```

```

F17
65254
Auto.

A1 Init
Txl ABCDEFGIJLMNOPQRSTUWXYZ!+->/>.(/* arrows
Next:
  A B C D E F
1 <Init>Name No Road Town $
2 R.T. |Covan 16 |Jeremy Grove Hampton 17|
3 G.L. |Blake 17 |Osmaston Road Knowle 56|
4 G.L. |Blake 17 |Osmaston Road Knowle 56|
5 A.J. |Hunt 25 |Latimer Road Wythall 43|
6 H.A. |Fisher 32 |Florence Road Henley 28|
7 C.D. |Beard 86 |Valley Road Redditch 25|
8
9
10
11
12
13
14
15
16
    
```

The effect as you can see is to jump to the same cell but in the other window. Now let us see what happens when we move the cursor in the lower part and then transfer back to the upper part.

```

F17
65254
Auto.

A1 Init
Txl ABCDEFGIJLMNOPQRSTUWXYZ!+->/>.(/* arrows
Next:
  A B C D E F
1 <Init>Name No Road Town $
2 R.T. |Covan 16 |Jeremy Grove Hampton 17|
3 G.L. |Blake 17 |Osmaston Road Knowle 56|
4 G.L. |Blake 17 |Osmaston Road Knowle 56|
5 A.J. |Hunt 25 |Latimer Road Wythall 43|
6 H.A. |Fisher 32 |Florence Road Henley 28|
7 C.D. |Beard 86 |Valley Road Redditch 25|
8
9
10
11
12
13
14
15
16
    
```

Now the transfer back using the same keystroke.

```

F17
65254
Auto.

A1 Init
Txl ABCDEFGIJLMNOPQRSTUWXYZ!+->/>.(/* arrows
Next:
  A B C D E F
1 <Init>Name No Road Town $
2 R.T. |Covan 16 |Jeremy Grove Hampton 17|
3 G.L. |Blake 17 |Osmaston Road Knowle 56|
4 G.L. |Blake 17 |Osmaston Road Knowle 56|
5 A.J. |Hunt 25 |Latimer Road Wythall 43|
6 H.A. |Fisher 32 |Florence Road Henley 28|
7 C.D. |Beard 86 |Valley Road Redditch 25|
8
9
10
11
12
13
14
15
16
    
```

THE CRACKER TUTORIAL IV.  
Partitioning the Screen

Now the swap:

```

F17
65254
Auto.

A7 C.D. ABCDEFGIJLMNOPQRSTUVWXYZ!+-\>.(** arrows
Next:
  A B C D E F
  1 Init|Name |No |Road |Town |S
  2 |R.T.|Covan |16 |Jeremy Grove |Hampton |17
  3 |G.L.|Blake |17 |Osmaaston Road |Knowle |56
  4 |G.L.|Blake |16 |Jeremy Grove |Hampton |17
  1 Init|Name |No |Road |Town |S
  2 |R.T.|Covan |16 |Jeremy Grove |Hampton |17
  3 |G.L.|Blake |17 |Osmaaston Road |Knowle |56
  4 |G.L.|Blake |17 |Osmaaston Road |Knowle |56
  5 |A.J.|Hunt |25 |Latimer Road |Wythall |43
  6 |H.A.|Fisher |32 |Florence Road |Henley |28
  7<C.D.>Beard |86 |Valley Road |Redditch |25
  8
  9
  10
  11
  12
  13
  14
  15
  16

```

The lower cursor has also moved down two places. The screen can also be divided vertically in a very similar way using PV (Partition screen vertically) and specifying the grid number.

```

F17
65254
Auto.

A7 C.D.
Next: Partit screen vertically 20 0-9, <R>
  A B C D E F
  1 Init|Name |No |Road |Town |S
  2 |R.T.|Covan |16 |Jeremy Grove |Hampton |17
  3 |G.L.|Blake |17 |Osmaaston Road |Knowle |56
  4 |G.L.|Blake |16 |Jeremy Grove |Hampton |17
  1 Init|Name |No |Road |Town |S
  2 |R.T.|Covan |16 |Jeremy Grove |Hampton |17
  3 |G.L.|Blake |17 |Osmaaston Road |Knowle |56
  4 |G.L.|Blake |17 |Osmaaston Road |Knowle |56
  5 |A.J.|Hunt |25 |Latimer Road |Wythall |43
  6 |H.A.|Fisher |32 |Florence Road |Henley |28
  7<C.D.>Beard |86 |Valley Road |Redditch |25
  8
  9
  10
  11
  12
  13
  14
  15
  16

```

THE CRACKER TUTORIAL IV.  
Partitioning the Screen

The cursor has not moved in the upper screen because initially the screen parts were not locked together. To lock them use the PSH<R> (Partition Synchronized Horizontal) command sequence:

```

F17
65254
Auto.

A3 R.T.
Next: Partit synchronised
  A B C D E F
  1 Init|Name |No |Road |Town |S
  2 |R.T.|Covan |16 |Jeremy Grove |Hampton |17
  3 |G.L.|Blake |17 |Osmaaston Road |Knowle |56
  4 |G.L.|Blake |16 |Jeremy Grove |Hampton |17
  1 Init|Name |No |Road |Town |S
  2 |R.T.|Covan |16 |Jeremy Grove |Hampton |17
  3 |G.L.|Blake |17 |Osmaaston Road |Knowle |56
  4 |G.L.|Blake |17 |Osmaaston Road |Knowle |56
  5 |A.J.|Hunt |25 |Latimer Road |Wythall |43
  6 |H.A.|Fisher |32 |Florence Road |Henley |28
  7 C.D.>Beard |86 |Valley Road |Redditch |25
  8
  9
  10
  11
  12
  13
  14
  15
  16

```

To test it you can move the cursor down and then swap parts:

```

F17
65254
Auto.

A3 R.T.
Next: Partit synchronised
  A B C D E F
  1 Init|Name |No |Road |Town |S
  2 |R.T.|Covan |16 |Jeremy Grove |Hampton |17
  3 |G.L.|Blake |17 |Osmaaston Road |Knowle |56
  4 |G.L.|Blake |16 |Jeremy Grove |Hampton |17
  1 Init|Name |No |Road |Town |S
  2 |R.T.|Covan |16 |Jeremy Grove |Hampton |17
  3 |G.L.|Blake |17 |Osmaaston Road |Knowle |56
  4 |G.L.|Blake |17 |Osmaaston Road |Knowle |56
  5 |A.J.|Hunt |25 |Latimer Road |Wythall |43
  6 |H.A.|Fisher |32 |Florence Road |Henley |28
  7 C.D.>Beard |86 |Valley Road |Redditch |25
  8
  9
  10
  11
  12
  13
  14
  15
  16

```

As before a grid will be superimposed:

|    |              |   |      |    |      |    |      |    |      |    |      |    |      |    |      |    |      |    |      |    |          |    |
|----|--------------|---|------|----|------|----|------|----|------|----|------|----|------|----|------|----|------|----|------|----|----------|----|
| A7 | C. D.        |   |      |    |      |    |      |    |      |    |      |    |      |    |      |    |      |    |      |    | F17      |    |
|    | TXLL         | ABCDEFGHIJKLMNORSTUVWXYZ!+->/>./(* arrows |      |    |      |    |      |    |      |    |      |    |      |    |      |    |      |    |      |    | 65254    |    |
|    | Next:        | A   | B    | C  | D    | E  | F    | G  | H    | I  | J    | K  | L    | M  | N    | O  | P    | Q  | R    | S  | Auto.    |    |
| 1  | InitName     | No  | Road | No | Road | No | Road | No | Road | No | Road | No | Road | No | Road | No | Road | No | Road | No | Town     | E  |
| 2  | R.T.  Covan  | 16  | Jere | 16 | Jere | 16 | Jere | 16 | Jere | 16 | Jere | 16 | Jere | 16 | Jere | 16 | Jere | 16 | Jere | 16 | Hampton  | 17 |
| 3  | G.L.  Blake  | 37  | Osma | 37 | Osma | 37 | Osma | 37 | Osma | 37 | Osma | 37 | Osma | 37 | Osma | 37 | Osma | 37 | Osma | 37 | Knowle   | 56 |
| 4  | C.L.  Blake  | 37  | Osma | 37 | Osma | 37 | Osma | 37 | Osma | 37 | Osma | 37 | Osma | 37 | Osma | 37 | Osma | 37 | Osma | 37 | Knowle   | 56 |
| 1  | InitName     | No  | Road | No | Road | No | Road | No | Road | No | Road | No | Road | No | Road | No | Road | No | Road | No | Town     | E  |
| 2  | R.T.  Covan  | 16  | Jere | 16 | Jere | 16 | Jere | 16 | Jere | 16 | Jere | 16 | Jere | 16 | Jere | 16 | Jere | 16 | Jere | 16 | Hampton  | 17 |
| 3  | G.L.  Blake  | 37  | Osma | 37 | Osma | 37 | Osma | 37 | Osma | 37 | Osma | 37 | Osma | 37 | Osma | 37 | Osma | 37 | Osma | 37 | Knowle   | 56 |
| 4  | A.J.  Hunt   | 25  | Lat1 | 25 | Lat1 | 25 | Lat1 | 25 | Lat1 | 25 | Lat1 | 25 | Lat1 | 25 | Lat1 | 25 | Lat1 | 25 | Lat1 | 25 | Mythall  | 43 |
| 5  | H.A.  Fisher | 32  | FloR | 32 | FloR | 32 | FloR | 32 | FloR | 32 | FloR | 32 | FloR | 32 | FloR | 32 | FloR | 32 | FloR | 32 | Henley   | 28 |
| 6  | H.A.  Fisher | 32  | FloR | 32 | FloR | 32 | FloR | 32 | FloR | 32 | FloR | 32 | FloR | 32 | FloR | 32 | FloR | 32 | FloR | 32 | Henley   | 28 |
| 7  | C.D.  Beard  | 86  | Val1 | 86 | Val1 | 86 | Val1 | 86 | Val1 | 86 | Val1 | 86 | Val1 | 86 | Val1 | 86 | Val1 | 86 | Val1 | 86 | Redditch | 25 |
| 8  | C.D.  Beard  | 86  | Val1 | 86 | Val1 | 86 | Val1 | 86 | Val1 | 86 | Val1 | 86 | Val1 | 86 | Val1 | 86 | Val1 | 86 | Val1 | 86 | Redditch | 25 |
| 9  |              |   |      |    |      |    |      |    |      |    |      |    |      |    |      |    |      |    |      |    |          |    |
| 10 |              |   |      |    |      |    |      |    |      |    |      |    |      |    |      |    |      |    |      |    |          |    |
| 11 |              |   |      |    |      |    |      |    |      |    |      |    |      |    |      |    |      |    |      |    |          |    |
| 12 |              |   |      |    |      |    |      |    |      |    |      |    |      |    |      |    |      |    |      |    |          |    |
| 13 |              |   |      |    |      |    |      |    |      |    |      |    |      |    |      |    |      |    |      |    |          |    |
| 14 |              |   |      |    |      |    |      |    |      |    |      |    |      |    |      |    |      |    |      |    |          |    |
| 15 |              |   |      |    |      |    |      |    |      |    |      |    |      |    |      |    |      |    |      |    |          |    |
| 16 |              |   |      |    |      |    |      |    |      |    |      |    |      |    |      |    |      |    |      |    |          |    |

There are now four parts to the screen. To move across the vertical parts you use the forward slash /:

|    |              |   |      |    |      |    |      |    |      |    |      |    |      |    |      |    |      |    |      |    |          |    |  |
|----|--------------|---|------|----|------|----|------|----|------|----|------|----|------|----|------|----|------|----|------|----|----------|----|--|
| A7 | C. D.        |   |      |    |      |    |      |    |      |    |      |    |      |    |      |    |      |    |      |    |          |    |  |
|    | TXLL         | ABCDEFGHIJKLMNORSTUVWXYZ!+->/>./(* arrows |      |    |      |    |      |    |      |    |      |    |      |    |      |    |      |    |      |    |          |    |  |
|    | Next:        | A   | B    | C  | D    | E  | F    | G  | H    | I  | J    | K  | L    | M  | N    | O  | P    | Q  | R    | S  | Auto.    |    |  |
| 1  | InitName     | No  | Road | No | Road | No | Road | No | Road | No | Road | No | Road | No | Road | No | Road | No | Road | No | Town     | E  |  |
| 2  | R.T.  Covan  | 16  | Jere | 16 | Jere | 16 | Jere | 16 | Jere | 16 | Jere | 16 | Jere | 16 | Jere | 16 | Jere | 16 | Jere | 16 | Hampton  | 17 |  |
| 3  | G.L.  Blake  | 37  | Osma | 37 | Osma | 37 | Osma | 37 | Osma | 37 | Osma | 37 | Osma | 37 | Osma | 37 | Osma | 37 | Osma | 37 | Knowle   | 56 |  |
| 4  | C.L.  Blake  | 37  | Osma | 37 | Osma | 37 | Osma | 37 | Osma | 37 | Osma | 37 | Osma | 37 | Osma | 37 | Osma | 37 | Osma | 37 | Knowle   | 56 |  |
| 1  | InitName     | No  | Road | No | Road | No | Road | No | Road | No | Road | No | Road | No | Road | No | Road | No | Road | No | Town     | E  |  |
| 2  | R.T.  Covan  | 16  | Jere | 16 | Jere | 16 | Jere | 16 | Jere | 16 | Jere | 16 | Jere | 16 | Jere | 16 | Jere | 16 | Jere | 16 | Hampton  | 17 |  |
| 3  | G.L.  Blake  | 37  | Osma | 37 | Osma | 37 | Osma | 37 | Osma | 37 | Osma | 37 | Osma | 37 | Osma | 37 | Osma | 37 | Osma | 37 | Knowle   | 56 |  |
| 4  | A.J.  Hunt   | 25  | Lat1 | 25 | Lat1 | 25 | Lat1 | 25 | Lat1 | 25 | Lat1 | 25 | Lat1 | 25 | Lat1 | 25 | Lat1 | 25 | Lat1 | 25 | Mythall  | 43 |  |
| 5  | H.A.  Fisher | 32  | FloR | 32 | FloR | 32 | FloR | 32 | FloR | 32 | FloR | 32 | FloR | 32 | FloR | 32 | FloR | 32 | FloR | 32 | Henley   | 28 |  |
| 6  | H.A.  Fisher | 32  | FloR | 32 | FloR | 32 | FloR | 32 | FloR | 32 | FloR | 32 | FloR | 32 | FloR | 32 | FloR | 32 | FloR | 32 | Henley   | 28 |  |
| 7  | C.D.  Beard  | 86  | Val1 | 86 | Val1 | 86 | Val1 | 86 | Val1 | 86 | Val1 | 86 | Val1 | 86 | Val1 | 86 | Val1 | 86 | Val1 | 86 | Redditch | 25 |  |
| 8  | C.D.  Beard  | 86  | Val1 | 86 | Val1 | 86 | Val1 | 86 | Val1 | 86 | Val1 | 86 | Val1 | 86 | Val1 | 86 | Val1 | 86 | Val1 | 86 | Redditch | 25 |  |
| 9  |              |   |      |    |      |    |      |    |      |    |      |    |      |    |      |    |      |    |      |    |          |    |  |
| 10 |              |   |      |    |      |    |      |    |      |    |      |    |      |    |      |    |      |    |      |    |          |    |  |
| 11 |              |   |      |    |      |    |      |    |      |    |      |    |      |    |      |    |      |    |      |    |          |    |  |
| 12 |              |   |      |    |      |    |      |    |      |    |      |    |      |    |      |    |      |    |      |    |          |    |  |
| 13 |              |   |      |    |      |    |      |    |      |    |      |    |      |    |      |    |      |    |      |    |          |    |  |
| 14 |              |   |      |    |      |    |      |    |      |    |      |    |      |    |      |    |      |    |      |    |          |    |  |
| 15 |              |   |      |    |      |    |      |    |      |    |      |    |      |    |      |    |      |    |      |    |          |    |  |
| 16 |              |   |      |    |      |    |      |    |      |    |      |    |      |    |      |    |      |    |      |    |          |    |  |

DATABASE HANDLING TECHNIQUES

SEARCHING

You may want to find a particular entry within a large worksheet without having to search for it yourself. To help you there is the GET command which goes through the columns and lines starting at the current cursor location looking for any string that you care to enter. (A string is computer jargon for any row of characters that are to be treated as a whole. Any word in this sentence can be looked on as a string.) Note that it does not matter what format the target string is displayed under, only the data as it was entered is searched.

Your string must be enclosed in single character delineators. Valid delineators that can be used to mark the beginning and end of the string are any characters that are printable but not letters or numerals. The string that you want found will be assumed to have been completed when the second delineator to match the first character after the GET command is found.

If you want to find a second occurrence of the same string then you only need to type a valid delineator twice and the string you last used will be automatically inserted between the characters.

```

A1 Init
TxL /string/
Next: Get /43
      A      B      C      D      E      F
      1 InitName No Road Town IS
      2
      3 R.T.:Cowan 16 Jeremy Grove Hampton 17
      4 G.L.:Blake 37 Osmaston Road Knowle 56
      5 A.J.:Hunt 25 Latimer Road Wythall 43
      6 H.A.:Fisher 32 Florence Road Henley 28
      7 C.D.:Beard 86 Valley Road Redditch 25
      8
      9
      10
      11
      12
      13
      14
      15
      16
      17
  
```

In this example you are searching for the number 43 which is to be found at location F5.

```

F5 (43)
Int ABCDEFGHIJKLMNOPQRSTUVWXYZ+--\>.(!*@zwnj
Next:
      A      B      C      D      E      F
      1 InitName No Road Town IS
      2
      3 R.T.:Cowan 16 Jeremy Grove Hampton 17
      4 G.L.:Blake 37 Osmaston Road Knowle 56
      5 A.J.:Hunt 25 Latimer Road Wythall < 43>
      6 H.A.:Fisher 32 Florence Road Henley 28
      7 C.D.:Beard 86 Valley Road Redditch 25
      8
      9
      10
      11
      12
      13
      14
      15
      16
      17
  
```

The cursor ends up at the location of the string. You should remember that only the actual cell formulae, numeric data and text entries are searched. If for example the last column had been in financial format and you had tried to search for 43.00 you would not find it even though it was displayed. As you can see from the contents line only 43 is actually stored in the memory. In practice this means you can only search for things that can be displayed on the contents line.

The Get command is useful for making long jumps across a complicated sheet. You can insert special text entries as markers which help you to quickly find the right place even if insertions and deletions have been made.

**SORTING THE LINES**

THE CRACKER can selectively sort lines, with either all or part of a column used as the basis of the sort. Text and numeric data can be sorted in either increasing or decreasing form, allowing easy maintenance of address lists and client lists. By using only part of columns in the sorts you carry out many of the activities that you would otherwise use a database management program for.

```
F5 (43)
Int: 65254
Next: Sort lines using range: B3...B7
Auto.

1 Init|Name |No |Road |Town |E |S |F |
2 | | | | | | | | |
3 R.T.|Cowan |16 |Jeremy Grove |Hampton | | |17|
4 C.L.|Blake |37 |Osaston Road |Knowle | | |56|
5 A.J.|Hunt |25 |Latimer Road |Wythall | | |43|
6 H.A.|Fisher |32 |Florence Road |Henley | | |28|
7 C.D.|Beard |86 |Valley Road |Redditch | | |25|
8 | | | | | | | | |
9 | | | | | | | | |
```

In this example the lines are going to be sorted using the name as the basis.

```
F5 (43)
Int: 65254
Next: Increasing or decreasing Inc. Dec
Auto.

1 Init|Name |No |Road |Town |E |S |F |
2 | | | | | | | | |
3 R.T.|Cowan |16 |Jeremy Grove |Hampton | | |17|
4 C.L.|Blake |37 |Osaston Road |Knowle | | |56|
5 A.J.|Hunt |25 |Latimer Road |Wythall | | |43|
6 H.A.|Fisher |32 |Florence Road |Henley | | |28|
7 C.D.|Beard |86 |Valley Road |Redditch | | |25|
8 | | | | | | | | |
9 | | | | | | | | |
10 | | | | | | | | |
11 | | | | | | | | |
12 | | | | | | | | |
13 | | | | | | | | |
14 | | | | | | | | |
15 | | | | | | | | |
16 | | | | | | | | |
17 | | | | | | | | |
```

**THE CRACKER TUTORIAL IV.**  
Database techniques - sorting

```
B3 Beard
TxTL ABCDEFGIJKLMNOPQRSTUVWXYZ!+->.</'* arrows
Next:

1 Init|Name |No |Road |Town |E |S |F |
2 | | | | | | | | |
3 C.D.|Beard |86 |Valley Road |Redditch | | |25|
4 G.L.|Blake |37 |Osaston Road |Knowle | | |56|
5 R.T.|Cowan |16 |Jeremy Grove |Hampton | | |17|
6 H.A.|Fisher |32 |Florence Road |Henley | | |28|
7 A.J.|Hunt |25 |Latimer Road |Wythall | | |43|
8 | | | | | | | | |
9 | | | | | | | | |
10 | | | | | | | | |
11 | | | | | | | | |
12 | | | | | | | | |
13 | | | | | | | | |
14 | | | | | | | | |
15 | | | | | | | | |
16 | | | | | | | | |
17 | | | | | | | | |
```

Column B is now in alphabetical order. Next you will see a numerical sort in descending order. Note that the sort is carried out on the internal value (as displayed on the contents line) of the number and not on the numerals as displayed.

```
B3 Beard
TxTL
Next: Sort lines using range: F3...F7
Auto.

1 Init|Name |No |Road |Town |E |S |F |
2 | | | | | | | | |
3 C.D.|Beard |86 |Valley Road |Redditch | | |25|
4 G.L.|Blake |37 |Osaston Road |Knowle | | |56|
5 R.T.|Cowan |16 |Jeremy Grove |Hampton | | |17|
6 H.A.|Fisher |32 |Florence Road |Henley | | |28|
7 A.J.|Hunt |25 |Latimer Road |Wythall | | |43|
8 | | | | | | | | |
9 | | | | | | | | |
```



```

B3 Beard
TABL
Next: Increasing or decreasing D <R>
      A      B      C      D      E      F
      Init|Name |No |Road |Town |S
1  C.D.|Beard  |86 |Valley Road |Redditch |25
2  G.L.|Blake  |37 |Osmaston Road |Knowle |56
3  R.T.|Cohen  |18 |Jeremy Grove |Hampton |17
4  H.A.|Fisher |32 |Florence Road |Henley |28
5  A.J.|Hunt  |23 |Latimer Road |Wythall |43
6
7
8
9
10
11
12
13
14
15
16
17
    
```

Remember that if you do want to keep address lists you will probably also like to be able to prepare mail labels. See the Copy function for more details on this.  
Note that there are one or two important restrictions you should bear in mind when using the Sort command. Consult the complete command reference for more details.

Recap: There are two functions, a search command and a sort command, that allow you to reproduce some simple database handling effects using THE CRACKER.

```

F3 (56)
Int ABCDEFGIJKLMNOPQRSTUVWXYZ+~>./<.* arrows
Next:
      A      B      C      D      E      F
      Init|Name |No |Road |Town |S
1  G.L.|Blake  |37 |Osmaston Road |Knowle |< 56>
2  A.J.|Hunt  |25 |Latimer Road |Wythall |43
3  H.A.|Fisher |32 |Florence Road |Henley |28
4  C.D.|Beard  |86 |Valley Road |Redditch |25
5  R.T.|Covan  |16 |Jeremy Grove |Hampton |17
6
7
8
9
10
11
12
13
14
15
16
17
    
```

The lines have now been sorted to make the numbers in column F descending.

THE CRACKER TUTORIAL  
V. MORE COMPLEX USE OF THE SHEET

**MORE ADVANCED EXPRESSIONS AND FUNCTIONS**

Here we are going to look at the more advanced Functions available to you for use in your expressions. If you feel that we have already covered all of the inbuilt functions that you can take in during the early stages, then you can skip this part of the manual for now although it is advisable to return to it when you feel more competent.

**THE IF, THEN, ELSE FUNCTIONS**

This is a special function group which is known in computing terms as a conditional branch. This means that a test is made on some data, and if the data passes (ie the test is TRUE) then one function is carried out, whereas if the data fails (ie the test is FALSE) then a different function is applied.

This conditional function is of the form IF("),THEN("),ELSE("). The " stands for an expression. The first expression, following the IF, must be logical. 'Logical' is another piece of jargon that just means it must have an answer of TRUE or FALSE. An example of a logical expression is IF(B3=4) which has a value of TRUE if B3 does equal 4 or FALSE if it does not.

When the test is TRUE the expression associated with the THEN part is executed and, correspondingly, when the test is FALSE the expression associated with the ELSE part is executed.

The full list of special operators you can use to give you an answer of TRUE or FALSE are:

- = equal
- : not equal
- > greater than
- ] greater than or equal (certain keyboards only)
- < less than
- [ less than or equal (certain keyboards only)

You can also use the functions TRUE or FALSE themselves instead of an expression. In other words, you can just type the words 'TRUE' or 'FALSE' in as a cell entry, they do not have arguments so nothing else has to be typed into the entry. These logical functions have numeric values of -1 (TRUE) and 0 (FALSE).

Once a cell has been assigned a logical function it can then be referenced as the expression for the IF function of a separate entry. For example, you could set B3 to TRUE or FALSE and then use the conditional in the form IF(B3),THEN("),ELSE("). It should follow from the above explanation that what this means is IF (B3 reads TRUE) THEN ("), ELSE IF (B3 reads FALSE) (").

In place of the normal expressions after THEN and ELSE you can use the special function ERROR. When the Error function is encountered during a calculation then processing is stopped and a message is put up on the prompt line, which can be treated as a normal error message. No harm can be done using this function, but it is a useful method to check for genuine errors, to bring macros to an end, or to warn the user that a certain unwanted result has occurred, for example if profits drop below a certain figure.

You can now try an example which includes some of these functions and features. Clear the worksheet and type IC15<R><R> and IL<R><R> to set up your work area. Now type .TRUE<R> into A1:

```

A1      Enter number or expression      A2
Gen    >> TRUE                          65519
Next:  A                                Auto.
1<    >
2     |

```

Note that A1 takes on the value -1. Type D to move to A2 and type IF(A1),THEN(5),ELSE(ERROR)<R> which means if A1 is TRUE then give A2 the value 5 otherwise indicate an error:

```

A2      Enter number or expression      A2
Gen    >> IF(A1),THEN(5),ELSE(ERROR)    65504
Next:  A                                Auto.
1      -1
2<    >

```

```

A2      (IF(A1),THEN(5),ELSE(ERROR))    A2
Gen    ABCDEFGIJKLMNOPQRSTUVWXYZ!+~\>/>.'* atrows 65467
Next:  A                                Auto.
1      -1
2<    5

```

Because A1 was TRUE, A2 has been set to 5. Now change A1 to see the effect on A2. Type U.FALSE<R>:

```

A1      (TRUE)                          A2
Gen    >> FALSE                          65467
Next:  A                                Auto.
1<    -1
2     5

```

TABLE HANDLING FUNCTIONS

Several functions are available to let you extract values from a specified list of cells. They are used in the same way that we would look up and read values from a table or list.

As a first example you are going to see the LOOKUP function. This function when given a value searches a list for a match to this value and then returns the entry from the adjacent row or column. Consider it as being the same as looking up a value in a printed table where you look for a value in the first column to get your answer from the second, such as logarithmic tables.

A typical example of the use of this function may be finding a commission percentage given sales income. These rates tend to jump from band to band.

| B16 (LOOKUP(B12..B5...R10))                  |                                 | C18 65009 |                |
|--|---------------------------------|-----------|----------------|
| OF In ABCDEFIJLMNOPQRSTUVWXYZ+<->.<!* arrows |                                 | Auto.     |                |
| Next:  | A                               | B         | C              |
|  | COMMISSION CALCULATION          | Sales     | Commission (%) |
| 1  |                                 |           |                |
| 2  |                                 |           |                |
| 3  |                                 |           |                |
| 4  |                                 | 0.00      | 0.00           |
| 5  |                                 | 1,000.00  | 2.50           |
| 6  |                                 | 2,000.00  | 5.00           |
| 7  |                                 | 4,000.00  | 7.50           |
| 8  |                                 | 10,000.00 | 10.00          |
| 9  |                                 | 20,000.00 | 20.00          |
| 10   |                                 |           |                |
| 11   | Sales achieved                  | 15,000.00 |                |
| 12   |                                 |           |                |
| 13   |                                 |           |                |
| 14   |                                 |           |                |
| 15   |                                 |           |                |
| 16   | Commission one value per band < | 10.00 >   | 1,500.00       |
| 17   | Commission on a sliding scale   | 15.00     | 2,250.00       |
| 18   |                                 |           |                |

In this case the salesperson brought in \$15000 worth of business and so he managed to get into the band between \$10000 and \$20000 for which he gets 10% commission.

The form of the function is LOOKUP(value,list), the result returned by the function is taken from the adjacent list.

There is a similar function which you can use in the same way called INTERP which will interpolate a value from a list. It differs from LOOKUP in that the function tries to work out (interpret) the desired answer even if it is present in the list.

| A1 (TRUE)                  | A2 65466 |
|----------------------------|----------|
| Gen ERROR called from <A2> | Auto.    |
| >>                         |          |
| A                          | 0        |
| 1                          | 0        |
| 2                          | 1        |

A1 now takes the value 0 for FALSE and because of the automatic calculation feature the error message has already come up saying where the error was detected. As a precaution you cannot continue passed the error message until you have pressed the [ESC] key, the current cell is also changed to the cell originating the error so that it can be corrected.

```

B17 (INTERP(B12,B5...B10))
OF In ABCDEFGIJLMNOPQRSTUVWXYZ:++>.<.<.* arrows
Next:
  
```

| A                      | B                               | C               |
|------------------------|---------------------------------|-----------------|
| COMMISSION CALCULATION | Sales                           | Commission (\$) |
| 1                      |                                 |                 |
| 2                      |                                 |                 |
| 3                      |                                 |                 |
| 4                      |                                 |                 |
| 5                      | 0.00                            | 0.00            |
| 6                      | 1,000.00                        | 2.50            |
| 7                      | 2,000.00                        | 5.00            |
| 8                      | 4,000.00                        | 7.50            |
| 9                      | 10,000.00                       | 10.00           |
| 10                     | 20,000.00                       | 20.00           |
| 11                     |                                 |                 |
| 12                     | Sales achieved                  |                 |
| 13                     |                                 |                 |
| 14                     |                                 |                 |
| 15                     |                                 | Amount paid     |
| 16                     | Commission one value per band   | 1,500.00        |
| 17                     | Commission on a sliding scale < | 15.00 >         |
| 18                     |                                 |                 |

```

C18
65009
Auto.
  
```

Here the salesperson has been told that the commission will be calculated on a sliding scale based on the sales and commission table. As \$15000 worth was sold this is midway between \$15000 and \$20000 and so he can expect a commission midway between 10% and 20%. The INTERP function does this calculation for you and in this case comes up with the answer 15%.

The CHOOSE function will look at a list and return the value of the cell in the position in the list given by the first argument. The form of this argument is CHOOSE(value,list). The value will be rounded to the nearest whole number if it isn't one already.

```

A8 (CHOOSE(4,A1...A6))
Gen ABCDEFGIJLMNOPQRSTUVWXYZ:++>.<.<.* arrows
Next:
  
```

| A  |       |
|----|-------|
| 1  | 0     |
| 2  | 1000  |
| 3  | 2000  |
| 4  | 4000  |
| 5  | 10000 |
| 6  | 20000 |
| 7  |       |
| 8< | 4000> |
| 9  |       |
| 10 |       |

```

A10
65402
Auto.
  
```

In this example the CHOOSE function at A8 has looked through A1...A6 to find the 4th cell and returned the value held in it, in this case 4000.

NPV stands for net present value and is a discounted cash flow function that calculates the effect of a discount rate on a set of cash flow figures. The form of the function is NPV(value,list) where 'value' is the discount rate in percent and the 'list' is a list of cells that contain cash flows. If you do not intend to use THE CRACKER for financial calculations then there is no need to bother following the next example.

```

UNT (NPV(B4,B2...F2))
B5 ABCDEFGIJLMNOPQRSTUVWXYZ:++>.<.<.* arrows
Next:
  
```

| A    | B              | C          | D        | E        | F        |          |
|------|----------------|------------|----------|----------|----------|----------|
| YEAR | 1984           | 1985       | 1986     | 1987     | 1988     |          |
| 1    |                |            |          |          |          |          |
| 2    | CASH FLOW      | 1,000.00   | 1,200.00 | 1,500.00 | 2,000.00 | 1,000.00 |
| 3    |                |            |          |          |          |          |
| 4    | DISCOUNT RATE  | 15.00      |          |          |          |          |
| 5    | PRESENT VALUE< | 4,403.90 > |          |          |          |          |

```

F5
65337
Auto.
  
```

In this example you can assume that in 1983 some money is to be invested and the figures above represent the expected yearly returns on that investment. To find out how the investment will perform, the yearly values each need to be converted to 'present values' and summed. In this instance they are all converted to 1983 values.

The first return in 1984 will be calculated as  $1000/(1+dr/100)$ . The 1000 is effectively worth less because of the one year taken to get it. The next year 1200 is obtained but this is worth less still because it is discounted once for 1984 and again in 1985 so its present value is calculated as  $1200/(1+dr/100)/(1+dr/100)$  and so on. The value of the return in n years is  $return/((1+dr/100)^n)$ .

The 'internal rate of return' is the discount rate necessary to make the present value equal to the initial investment. It can be found by trial and error, changing the value of discount rate until you get the right answer. An example of how they can be automated and the internal rate of return can be calculated by THE CRACKER is given on the distribution disk and is called IRR.

If you want to do a numerical integration then you would probably use Simpson's rule. You can use the function SIMPRULE to do this directly.

```

B8 (SIMPRULE(PI/8,B1...B5))
Gen ABCDEFGIJLMNOPQRSTUVWXYZ:++>.<.<.* arrows
Next:
  
```

| A | B                   |                 |
|---|---------------------|-----------------|
| 1 | SIMR(0)             | 0               |
| 2 | SIMR(PI/8)          | 0.36268342365   |
| 3 | SIMR(PI/4)          | 0.7576781187    |
| 4 | SIMR(3*PI/8)        | 0.9287932511    |
| 5 | SIMR(PI/2)          | 1               |
| 6 |                     |                 |
| 7 | INTEGRAL OF SIMR(X) |                 |
| 8 | BETWEEN 0 AND PI/2< | 1.000134584974> |

```

B5
65289
Auto.
  
```

In this example 5 values of SINR(X) have been calculated at intervals of PI/8. The SIMPRULE function has been used to obtain an approximate value of the integral, whose exact value is 1. Note that you can use upper or lower case characters in your expressions. The form of the function is SIMPRULE(step,range), where the range must have an odd number of values.

#### A NOTE ON LISTS

In most functions a list can be specified using a range such as B1..B5. You can however have blank entries in your range and the function will still be worked out correctly. This feature allows you to set up a template worksheet and enter your particular data later. It will also cater for the situation where the number of items will be variable.

#### AN INTRODUCTION TO 'COMMAND' FUNCTIONS

There are certain functions that we are now going to come across that are fundamentally different to those we have seen so far. The principle difference is that they act on a cell or cells other than the one in which they have been entered, and can be looked on more like commands than like the mathematical functions encountered so far.

These command functions are useful in that they can be used to automate some actions normally entered using direct commands. For example, they can be used to create a worksheet that performs an entire set of calculations as soon as it has been loaded and the automatic recalculation started. Some of the demo files on the distribution disc use this feature. They are also useful for creating subroutines that perform a range of actions or calculations automatically. We will see later how such a technique can be used to create and fill a table of values.

Examples of functions that behave rather like commands are SET and INIT, which assign a value to a distant cell rather than the one they are in, or INC and DEC which can alter the value held in a distant cell.

There are also some command-like functions that may read a value from a distant cell but do not act on anything in particular. An example would be one of the Graphics functions we will see later such as MAINTITLE(crd), which reads the text to be displayed in a graph title from a distant cell, but is unable to display it because of different format types.

Because the command functions act on a cell or cells other than the one in which they have been entered, and because that distant cell may sometimes contain text data, it is sometimes unclear what value the actual function cell itself will display. In fact sometimes you will find they display the same value as held in the distant cell (if they can) and sometimes they will just display a zero.

To avoid confusing the display of the sheet you may wish to place these command functions in a seldom seen area of the sheet, it doesn't matter where they go - THE CRACKER will always find them. Remember that because these are FUNCTIONS with arguments they must always be entered into a cell that has a numeric format.

**MULTIPLE FUNCTION LINES AND DIVIDING COMMAS**

When you are entering functions into a cell it is valid to insert a dividing comma between them. The effect of this dividing comma is to cause an effective restart to the entry as if what follows was the beginning of the line.

The value that will be displayed in the cell will be that of the expression after the last dividing comma.

For example start with a blank sheet and enter 10 into A1 and this expression into B1:

2\*A1,A1

|       |  |     |       |
|-------|--|-----|-------|
| B1    | (2*A1,A1)                                | B8  | 65488 |
| Gen   | ABCDEFGHIJLMNOPQRSTUVWXYZ!+->.(/* arrows |     | Auto. |
| Next: |  |     |       |
| 1     | A  | 10< | 10>   |
| 2     |  |     |       |
| 3     |  |     |       |
| 4     |  |     |       |
| 5     |  |     |       |
| 6     |  |     |       |
| 7     |  |     |       |
| 8     |  |     |       |

The first calculation, 2\*A1, is performed but is effectively forgotten - all that appears in B1 is a copy of A1. This feature so far looks like a waste of time and memory but consider its application to the 'command' functions described above.

Try entering this expression into B1

SET(A1,10),SET(A2,20),SET(A3,30),SET(A4,40),A1

Because the 'command' functions all act on DISTANT cells their effect is performed on the sheet even though cell B1 finishes by only displaying the value obtained after the last comma. None of the earlier commands are wasted. They are also useful with the BLEEP function.

|       |  |     |       |
|-------|--|-----|-------|
| B1    | (SET(A1,10),SET(A2,20),SET(A3,30),SET(A4,40),A1) | B8  | 63406 |
| Gen   | ABCDEFGHIJLMNOPQRSTUVWXYZ!+->.(/* arrows         |     | Auto. |
| Next: |  |     |       |
| 1     | A  | 10< | 10>   |
| 2     |  |     |       |
| 3     |  |     |       |
| 4     |  |     |       |
| 5     |  |     |       |
| 6     |  |     |       |
| 7     |  |     |       |
| 8     |  |     |       |

It should be obvious that only these 'command' functions have a useful effect if they are followed by a comma.

Not all command-like functions can be treated in this way and some of the Graphics functions in particular may give unpredictable results if followed by a comma.

THE CRACKER TUTORIAL V  
Date and Time functions

**SECOND**  
The second

**DATE**  
Returns the date in the form of a single number, for example 312.1985  
- being the 3rd December 1985.

**ZEROTIME**  
Resets the elapsed time counter. Probably best included in an IF, THEN, ELSE entry.

**TIMELAPSE**  
Returns the elapsed time since the ZEROTIME function was last operative. This is in seconds.

**DELAY(n)**  
Does nothing until n seconds have elapsed. 'n' may be a cell reference or a constant value. For practical reasons make it a cell reference with a value 0 until you actually want to run your application.

**I/O FUNCTIONS**

These functions can be used to control, or exchange information with, external peripherals. Specific details for their use will depend on the hardware configuration you have. See the manuals for your computer and/or peripherals for details of the port numbers and values to use.

**IN(part)**  
Reads an 8 bit port given by the number or cell reference 'port'.

**OUT(part,value)**  
Outputs a 'value' given by a number or cell reference to the 'port' given by a number or cell reference.

Complex control of external machinery can be achieved by using DO-WHILE loops, the Date - Time functions and the I/O commands.

THE CRACKER TUTORIAL V  
Date and Time functions

**DATE AND TIME FUNCTIONS**

The DATE and TIME functions have two immediate applications. In financial calculations, such as tax returns or yearly balance sheets, it is often the case that some measure of the time of year has to be taken to determine the required results. By entering values for the time of year automatic adjustments can be made from within the program.

Secondly, there are some functions that are able to keep track of the passage of time since the program was first entered. These are particularly useful when combined with the Input Output control of peripherals when it becomes possible to program THE CRACKER to control equipment such as central heating where the desired output depends on the time of day and year.

**DATEAFTER(date,days)**  
Gives the date that in the number of days specified from the input date. The date must be in the form of a single number, for example 312.1985 - being the 3rd December 1985. Be careful to put months 1 to 9 as 01 to 09.

**DAYSAPART(date1,date2)**  
Gives the number of days between any two specified dates.

**DAYOFWK(date)**  
Returns the day of week as a number. Saturday has a value 0, Sunday 1, Monday 2 etc.

**DAYOFYR(date)**  
Returns the number of days between January 1st and the present day.

The rest of these functions are only available on CP/M PLUS (3) systems. Note for your clock and calendar to be correct you must set them before starting THE CRACKER. Use the utility supplied on your disc called DATE, just type DATE SET at the DOS prompt and follow the screen prompts.

**YEAR**  
Gives the current year.

**MONTH**  
The month.

**DAY**  
The day

**HOURL**  
The hour

**MINUTE**  
The minute

PRODUCING GRAPHS AND CHARTS

The graphics addition to THE CRACKER has been designed to allow you to generate varying types of graphs from the minimum of input by the user. There are special graphics functions, detailed below, to enable you to specify the data and the format of the graph - this information can be stored anywhere in the spreadsheet.

Note that where 'crd' is specified in the functions below you must put it in the form shown. If you don't, the correct values may not be properly passed to the plotting section of the program. You can't use numerical or expression equivalents except where stated.

Here are the functions:

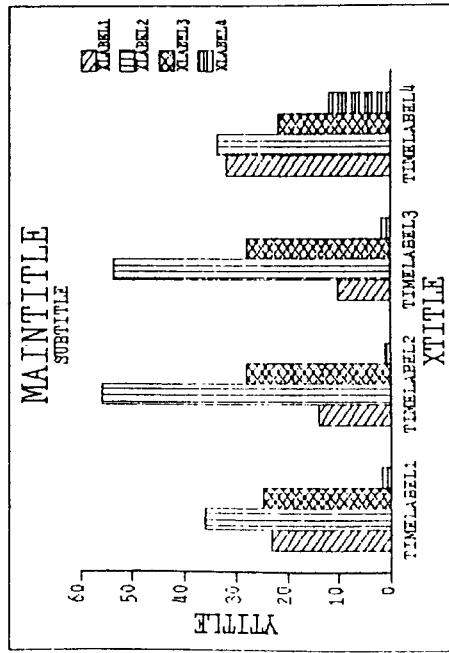
TYPEPLOT(crd)

The type of graph or chart plot you want. Give a value between 1 and 18 in the cell referred to by this function.

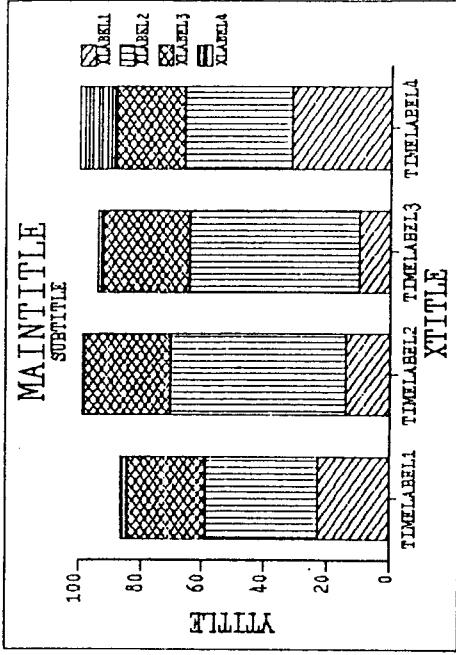
The plot types are detailed below:-

Business (these graphs require timelabels rather than actual Xvalues:

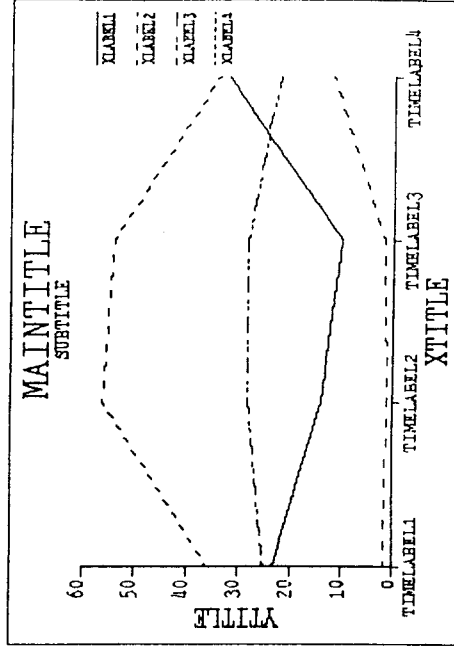
1. Bar chart (histogram) - there may be more than one bar over each timelabel



2. Stacked bar chart - each bar for each timelabel is superimposed over the others so that only the overlap is shown

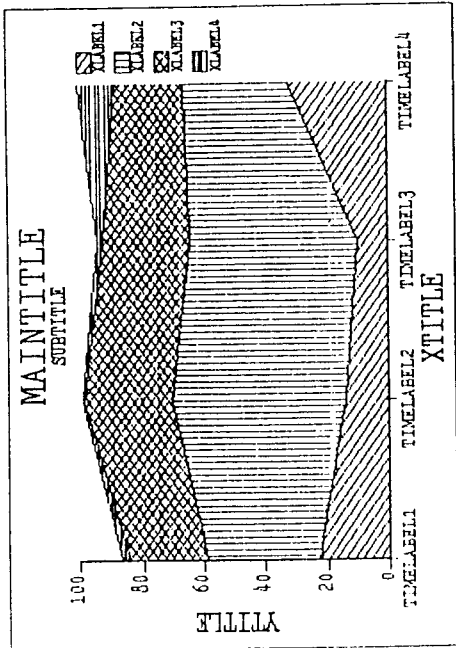


3. Line chart - lines join each point on each of the plot categories

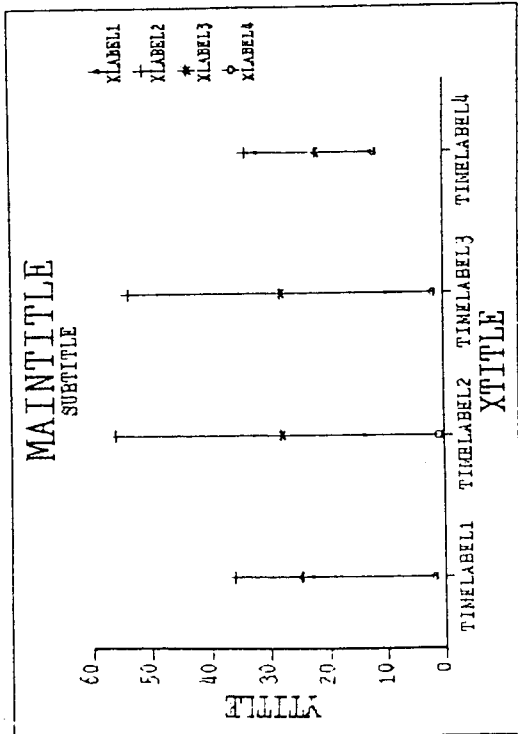




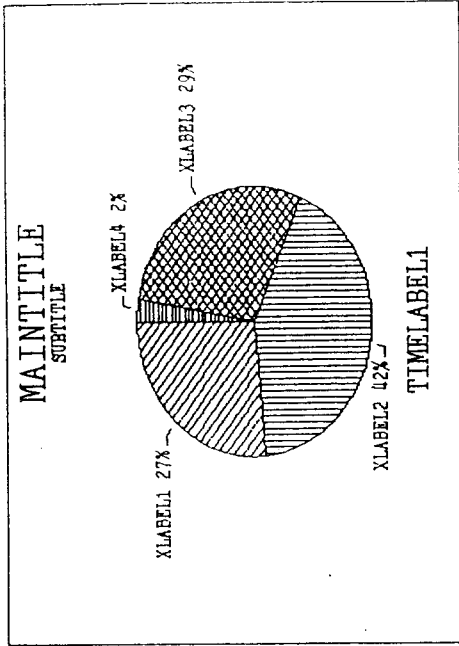
4. Area chart - as above except that each enclosed area is hatched



5. Hi-lo chart - each of the plot categories are shown as a marker above each timelabel. The markers are joined by a vertical line.

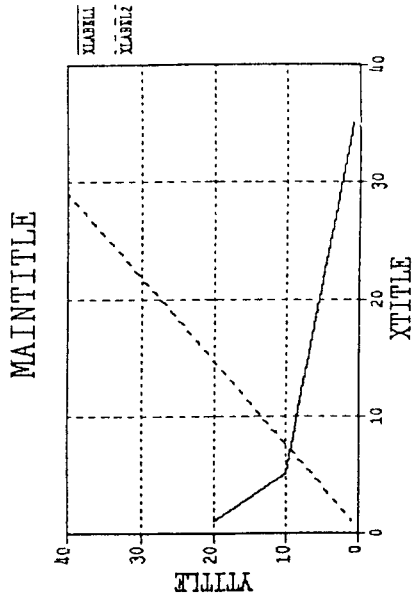


6. Pie chart - the familiar segmented circular chart.



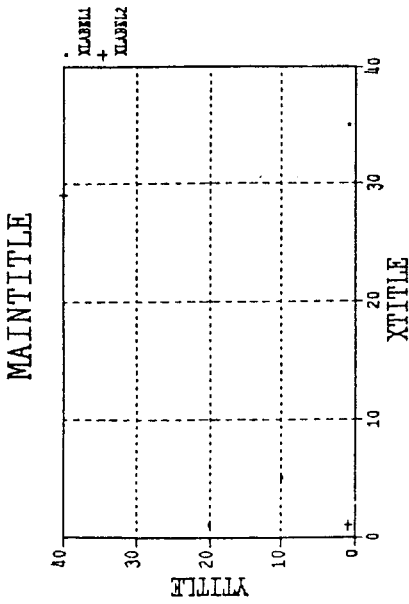
Statistical, engineering and scientific:

7. X:Y line joining points



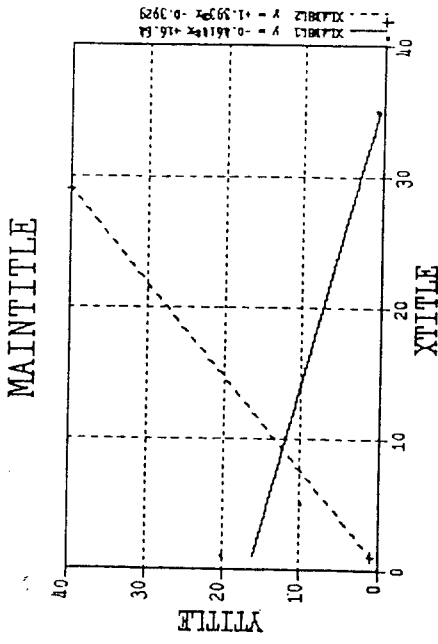
- 8. LogX:Y line joining points
- 9. X:LogY line joining points
- 10. LogX:LogY line joining points

11. X:Y points only



- 12. LogX:Y points only
- 13. X:LogY points only
- 14. LogX:LogY points only

15. X:Y points and best fit line



- 16. LogX:Y points and best fit line
- 17. X:LogY points and best fit line
- 18. LogX:LogY points and best fit line

The best fit lines of plot types 15...18 are obtained by the least squares method.

The other graphics functions are:

`MAINTITLE(crd)`

The cell coordinate is a pointer to a text cell where the main title is to be found.

`SUBTITLE(crd)`

The same idea as for the maintitle applies to the subtitle.

`YTITLE(crd)`

The coordinate pointer is to the title up the Y axis on the left hand side.

`XTITLE(crd)`

Where coordinate refers to where the X axis title is. This is the one at the bottom of the graph.

**YMAXIMUM(crd)**

The **crd** may be a value or a reference to a value that specifies the maximum value to be shown on the X axis. Your choice will be rounded to a suitable nearby value to improve the presentation. (plot types 7...18)

**YMINIMUM(crd)**

Similarly for a minimum value.

See the notes given above for **YMINIMUM**.

The **YMINIMUM** command only works on plot types 7 or above.

Remember that because the graphics commands are entered as **FUNCTIONS** followed by values or coordinates each of the above should be placed into a cell that has been given a numerical format, even though the information they refer to may be text. The text itself of course has to be placed in cells of a text format.

You may be wondering what each of these cells that contain the above functions will actually display on the screen. In the majority of the cases the display will show zero. Some of the functions will display a numeric value if it is felt to be useful. For example the cell that contains the **XLABELS** function will show the number of **Xlabels** defined. The cell that contains **TYPEPLOT** will show the number of the graph type chosen.

There is an example of a business plot spreadsheet on the file **PLOT1.MEM** and one for scientific work on **PLOT2.MEM**. Load these spreadsheets and use the exchange rules/formulae command to the how the functions are used in practice. The **HELP** utility will give you reminders about each of these functions and plot types.

**XLABEL(crd...crd), or XLABEL(crd,crd,crd,etc.)**

This points to the X labels which are the legend box items on the right explaining what the plot lines etc. refer to. **THE CRACKER** also uses this function to discover how many plot lines you want to display.

Unlike the other functions in this list the range can also be a list of individual items each one pointing to one of the X labels. Please make an effort to get the number of X labels correct as the program counts them to see how many lines or groups of items there are. If for example you get it wrong and enter too many **Xlabels** you will be informed there are Y values missing because it is expecting to have to draw further lines.

**TIMELABEL(crd...crd)**

These are the tagged items on the X or bottom axis that show what exactly is being plotted. They are not always time labels but it is very common in financial graphs for example to plot values according to month or year. This option refers to plot types 1...6 only, the other graph types will expect actual numerical values on the lower line.

There are some restrictions on the maximum size of the graph labels. consult the Command Reference for more details.

**YVALUE(n,crd...crd)**

This function points to the actual Y values to be plotted. The 'n' refers the number of the plotting line to which you are referring. There will be one of these functions for each line.

**YMAXIMUM(crd)**

The 'crd' in this case can be a value or a reference to a value that specifies the maximum figure to be shown on the Y axis. Your choice will be rounded to a suitable nearby value to improve the presentation.

**YMINIMUM(crd)**

This is a similar function to the above for a minimum value.

Note that because **THE CRACKER** rounds the value for the minimum to the best nearby value to give an attractive display you may have to experiment and perhaps set the **YMINIMUM** value lower than you first thought in order to get the desired display.

**XVALUE(n,crd...crd)**

This function points to the actual X values to be plotted. The 'n' refers the number of the line to which you are referring. There will be one of these functions for each line. (plot types 7...18 only, use **TIMELABELS** for 1...6)

MAKING A GRAPH IN PRACTICE

Business type

Prepare a sheet with the data you want plotted on it. The actual data values can be up and down columns or across lines, as long as they lie in a range of coordinates. It is not critical how the other items are laid out so just set aside an area to put in the plotting instructions. Each of these instructions is a pointer to where the particular data items you want displayed are to be found.

Start by putting in the TYPEPLOT function and then the pointers to the various titles, although these are optional. These are MAINTITLE, SUBTITLE, YTITLE and XTITLE.

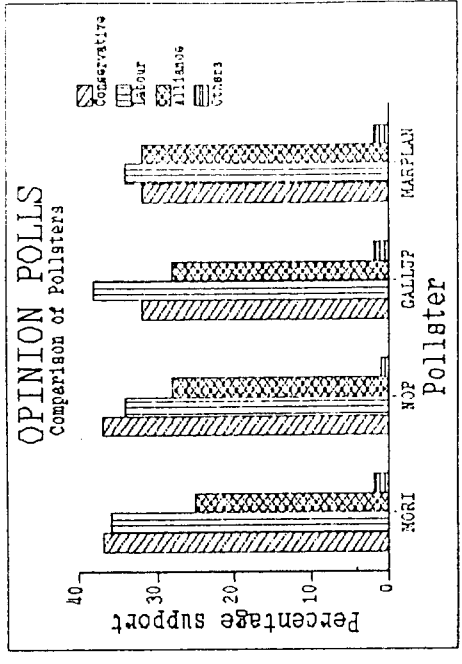
Next show where the TIMELABELS text can be found. Remember these are typically the months or years that go across the bottom of the page, they don't in fact have to relate to time but usually do.

Lastly point to the YVALUES, which are the actual sets of data. There will be one YVALUE entry for each set.

If you particularly wish to specify the maximum or minimum values you want plotted then use the YMAXIMUM and YMINIMUM functions. Normally don't use these functions as the program will work out all the maximums and minimums for you automatically.

```

A6 Gen ABCDEFGHIJKLMNOPQRSTUVWXYZ+->/>.{** arrows E21
Next:                                     52341
                                           Auto.
                                           E
1 (1) A
2 OPINION POLLS
3 Comparison of Pollsters
4 Percentage support
5 Pollster
6<
7 MORI
8 GOP
9 CALLUP
10 MARPLAN
11 <1> (Typeplot(A1))
12 <10> (subtitle(A2))
13 <9> (subtile(A3))
14 <8> (subtile(A4))
15 <7> (subtile(A5))
16 <6> (subtile(A6...E6))
17 <5> (timeLabel(A7...A1))
18 <4> (Yvalue(1,B7...B10))
19 <3> (Yvalue(2,C7...C10))
20 <2> (Yvalue(3,D7...D10))
21 <1> (Yvalue(4,E7...E10))
    
```



Similar instructions apply for plot types 7...18, scientific graphs. In these cases XVALUES replace TIMELABELS.

If any errors are encountered while using the graphics an error message will be given and a return will be made to the spreadsheet. This is an early version of the graphics program and so if you find you can not return to the main program don't worry as your work to date will be found on the file SECURITY.MEM (280 versions only, 16 bit versions are not overlaid).

If you do any EDITING of the plotting instructions or make any changes that do not force a recalculation, you may end up with an error message or values that do not seem to be true. This is most likely when you use direct values in your functions rather than cell coordinates.

An example may be

YMAXIMUM(20000)

and a case where it would not occur would be

YMAXIMUM(B7)

Don't worry about this, just remember the safest way to handle plotting functions that do not contain cell references is to overwrite them rather than edit them.

The instructions for plotting are updated when a recalculation is done so if you have made changes the latest instructions may not have been passed causing an error. If you get such an error message just use the ! FORCE RECALCULATION COMMAND and try the plot again.

THE CRACKER TUTORIAL  
VI. AUTOMATING DATA MANIPULATION

SOME MORE ADVANCED FEATURES OF THE WORKSHEET

USING MACRO COMMAND GROUPS

Often you will want to go through the same set of commands repeatedly. A typical example is the changing of the format of all the cells in one column which can be both time consuming and tedious. To get round this you can use the \*MACRO command. A macro is a computing term that you will come across in many programs, and it is defined as a linked set of commands that can be set in operation by issuing a single instruction. It is really only a shortcut to save you from a lot of unnecessary typing.

THE CRACKER will let you create predefined sets of linked commands, by entering the command letters into a cell as if you were actually typing them in to be acted on immediately, with the exception that an 'e' symbol is inserted in place of a carriage return.

The macro is executed by typing '\*' followed by the coordinate reference to the cell in which it is stored. For example, \*A1 executes the macro stored in cell A1.

The cell that contains the commands should be set to any text format. The maximum size of a macro is limited but you can put your macro in more than one cell by finishing with a reference to the continuation macro. For example, you might finish your A1 macro with a reference \*A2 to force it to continue with the commands entered in A2.

If you want your macro to loop and be carried out repeatedly then finish it with a reference to itself. For example with cell A1 this would mean finishing with \*A1. Don't worry about this causing an endless loop, there are lots of ways to make the macro come to an end.

As an illustration the following example takes a column of numbers in general format and changes them all to one place decimal format.

```

B1 (2.345)
Gen outside worksheet
Next: Up
  A
1 < 2.345>
2 34.56|
3 76.54|
4 123.4|
5 2345.6|
6 7.2345|
7 |
8 |

BB 65421
Auto.
  
```

Start by moving the cursor to cell A1 where the macro is to be entered. Type ',' to get into entry mode and then type `NFID@*A1` which says 'new format one place decimal, carriage return, cursor down and finally do macro A1 again'.

The @ symbol stands for the <R> carriage return.

```

A1 Enter characters
  >> NFID@*A1
  A
1< 2.345|
2 34.56|
3 76.54|
4 123.4|
5 2345.6|
6 7.2345|
7 |
8 |

BB 65421
Auto.
  
```

```

A1 NFID@*A1
IxL ABCDEFGHIJKLMNOPQRSTUVWXYZ|+~\>.(/*@#%&'
Next:
  A
1<NFID@*A1
2 2.345|
3 34.56|
4 76.54|
5 123.4|
6 2345.6|
7 7.2345|
8 |

BB 65411
Auto.
  
```

Next move the cursor to the location you want the macro to start its operations. In this case it is cell B1, and then type `*A1` to start the macro going.

```

B1 (2.345)
Gen 0-9, <R>
Next: *A1
  A
1 NFID@*A1
2 2.345>
3 34.56|
4 76.54|
5 123.4|
6 2345.6|
7 7.2345|
8 |

BB 65411
Auto.
  
```

