Notes:

Chapter 1
USING THE TURBO SYSTEM

This chapter describes the installation and use of the TURBO Pascal system, specifically the built-in editor.

.COM and .CMD files

Files with the extension .COM mark the executable program files in CP/M-80 and PC-DOS / MS-DOS. In CP/M-86 these will instead be marked .CMD. Thus, whenever .COM-files are mentioned in the following, it should be understood as .CMD if your operating system is CP/M-86.

BEFORE USE

Before using the TURBO Pascal you should, for your own protection, make a work-copy of the distribution diskette and store the original safely away. Remember that the User's License allows you to make as many copies as you need **for your own personal use** and for **backup purposes** only. Use a file-copy program to make the copy, and make sure that all files are successfully transferred.

IMPORTANT NOTE !!!

TURBO Pascal provides a number of compiler directives to control special runtime facilities such as index checking, recursion, etc. PLEASE NOTICE that the default settings of these directives will optimize execution speed and minimize code size. Thus, a number of runtime facilities (such as index checking and recursion) are deselected until explicitly selected by the programmer. All compiler directives and their default values are described in Appendix C. (De-selecting recursion applies to CP/M-80 only; in 16-bit versions recursion is always possible.)

Files On The Distribution Disk

The distribution disk contains the following files:

TURBO.COM
The TURBO Pascal program: compiler, editor, and all. When you enter the command `TURBO` on your terminal, this file will load, and TURBO will be up and running.

TURBO.OVR
Overlay file for TURBO.COM (CP/M-80 version only). Needs only be present on the run-time disk if you want to execute .COM files from TURBO.

TURBO.MSG
Text file containing error messages. Needs not be present on your run-time disk if you will accept the system without explanatory compile-time error messages. Errors will in that case just print out an error number, and the manual can be consulted to find the explanation. In any case, as the system will automatically point out the error, you may find it an advantage to use TURBO without these error messages; it not only saves space on the disk, but more importantly, it gives you approx. 1.5 Kbytes extra memory for programs. This message file may be edited if you wish to translate error messages into another language as described in Appendix H.

TINST.COM
Installation program. Just type `TINST` at your terminal, and the program takes you through a completely menu-driven installation procedure. This and the following files need not be present on your run-time disk.

TINST.DTA
Terminal installation data (not present on IBM PC versions).

TINST.MSG
Messages for the installation program. Even this file may be translated into any language desired.

.PAS files
Sample Pascal programs.

GRAPH.P
IBM PC versions only. Contains the **external** declarations necessary to use the extended graphics and turtlegraphics routines contained in GRAPH.BIN. Only necessary on the run-time disk if you want to do turtlegraphics.

GRAPH.BIN
IBM PC versions only. This file contains the extended graphics and turtlegraphics machine language routines. Only necessary on the run-time disk if you want to do extended or turtle graphics.

READ.ME
If present, this file contains the latest corrections or suggestions on the use of the system.

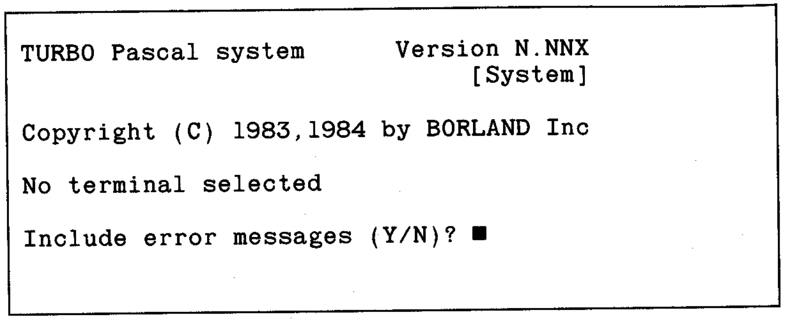
Only *TURBO.COM* **must** be on your run-time disk. A fully operative TURBO Pascal thus requires only **30 K** of disk space (37 K for 16-bit systems). *TURBO.OVR* is required only if you want to be able to execute programs from the TURBO menu. *TURBO.MSG* is needed only if you want on-line compile-time error messages. The *TINST* files are used only for the installation procedure, and the GRAPH files are needed only when you want to do extended graphics or turtlegraphics. The example .PAS files, of course, may be included on the run-time disk if so desired, but they are not necessary.

Starting TURBO Pascal

When you have a copy of the system on your work-disk, enter the command

```
TURBO
```

at your terminal. The system will log on with the following message:

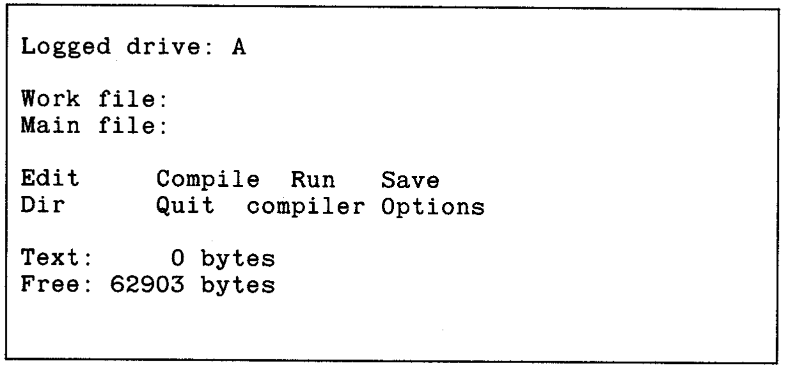


```
TURBO Pascal system      Version N.NNX
                              [System]

Copyright (C) 1983,1984 by BORLAND Inc

No terminal selected

Include error messages (Y/N)? ■
```

Figure 1-1: Log-on Message

N.NNX specifies your release number and *[System]* indicates the operating environment (operating system and CPU), for example `CP/M-86 on IBM PC` . The second-last line tells you which screen is installed. At the moment none - but more about that later.

If you enter a **Y** in response to the error message question, the error message file will be read into memory (if it is on the disk), briefly displaying the message `Loading TURBO.MSG`. You may instead answer **N** and save about 1.5 Kbytes of memory. Then the TURBO main menu will appear:



```
Logged drive: A

Work file:
Main file:

Edit     Compile  Run    Save
Dir      Quit  compiler Options

Text:     0 bytes
Free: 62903 bytes
```

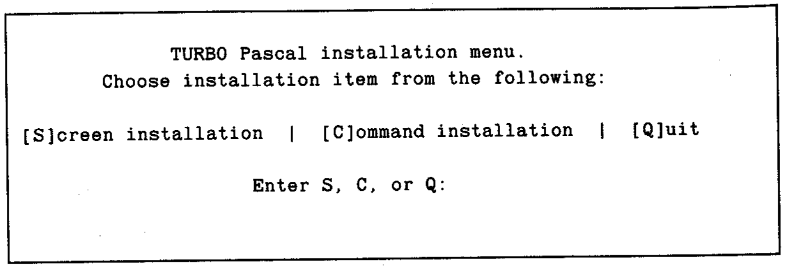
Figure 1-2: Main Menu

The menu shows you the commands available, each of which will be described in following sections. Each command is executed by entering the associated capital letter (highlighted after terminal installation if your terminal has that feature). Don't press <RETURN>; the command executes immediately. The values above for Logged drive and memory use are for the sake of example only; the values shown will be the actual values for your computer.

IBM PC users who are satisfied with the 'Default display mode' can use TURBO as it comes and may skip the following and go to page 14. If you're an non-IBM PC user, you may use TURBO without installation if you don't plan to use the built-in editor - but assuming that you do, type **Q** now to leave TURBO for a minute to perform the installation.

Installation

Type TINST to start the installation program. All *TINST* files and the *TURBO.COM* file must be on the logged drive. This menu will appear:



```
              TURBO Pascal installation menu.
         Choose installation item from the following:

[S]creen installation  |  [C]ommand installation  |  [Q]uit

                    Enter S, C, or Q:
```

Figure 1-3: Installation Main Menu

IBM PC Screen Installation

When you hit **S** to perform Screen installation, a menu will appear which lets you select the screen mode you want the TURBO environment to use (see Appendix L for details). When you have made your choice, the main menu re-appears, and you may now continue with the **Command** installation described on pages 350 pp, or you may terminate the installation at this point by entering **Q** for Quit.

Non-IBM PC Screen Installation

Now hit **S** to select Screen installation. A menu containing the names of the mostly used terminals will appear, and you may choose the one that suits you by entering the appropriate number. If your terminal is not on the menu, nor compatible with any of these (note that a lot of terminals are compatible with ADM-3A), then you must perform the installation yourself. This is quite straightforward, but you will need to consult the manual that came with your terminal to answer the questions asked by the installation menu. See Appendix L for details.

When you have chosen a terminal, you are asked if you want to modify it before installation. This can be used if you have for example an ADM-3A compatible terminal with some additional features. Choose the ADM-3A and add the required commands to activate the special features. If you answer **Yes**, you will be taken through a series of questions as described in Appendix L.

Normally, you will answer **No** to this question, which means that you are satisfied with the pre-defined terminal installation. Now you will be asked the operating frequency of your microprocessor. Enter the appropriate value (2, 4, 6 or 8, most probably 4).

After that, the main menu re-appears, and you may now continue with the **Command** installation described in the next section or you may terminate the installation at this point by entering **Q** for Quit.

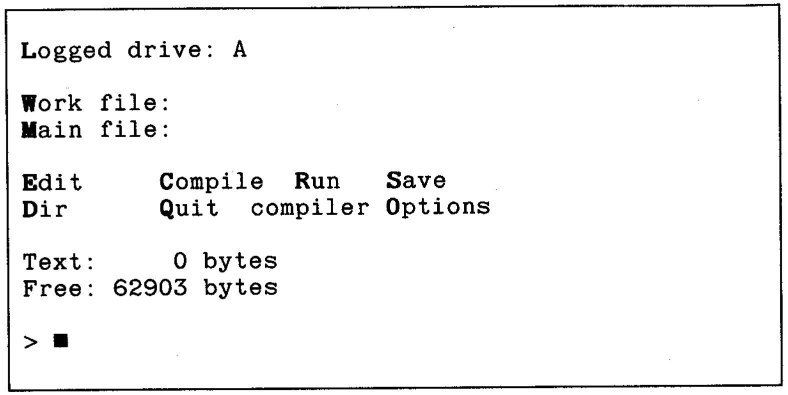
Installation of Editing Commands

The built-in editor responds to a number of commands which are used to move the cursor around on the screen, delete and insert text, move text etc. Each of these functions may be activated by either a primary or a secondary command. The secondary commands are installed by Borland and comply with the 'standard' set by *WordStar*. The primary commands are un-defined for most systems, and using the installation program, they may easily be defined to suit your taste or your keyboard. IBM PC systems are supplied with the arrows and dedicated function keys installed as primary commands as described in chapter 19.

Please turn to appendix L for a full description of the editor command installation.

The Menu

After installation, you once again activate TURBO Pascal by typing the command `TURBO`. Your screen should now clear and display the menu, this time with the command letters highlighted. If not, check your installation data.



```
Logged drive: A

Work file:
Main file:

Edit     Compile  Run   Save
Dir      Quit   compiler Options

Text:     0 bytes
Free: 62903 bytes

> ■
```

Figure 1-4: Main Menu

By the way, whenever highlighting is mentioned here, it is assuming that your screen has different video attributes to show text in different intensities, inverse, underline or some other way. If not, just disregard any mention of highlighting.

This menu shows you the commands available to you while working with TURBO Pascal. A command is activated by pressing the associated upper case (highlighted) letter. Don't press <RETURN>, the command is executed immediately. The menu may very well disappear from the screen when working with the system; it is easily restored by entering an 'illegal command', i.e. any key that does not activate a command. <RETURN> or <SPACE> will do perfectly.

The following sections describe each command in detail.

Logged Drive Selection

The **L** command is used to change the currently logged drive. When you press **L**, this prompt:

```
New drive: ■
```

invites you to enter a new drive name, that is, a letter from A through P, optionally followed by a colon and terminated with <RETURN>. If you don't want to change the current value, just hit <RETURN>. The **L** command performs a disk-reset, even when you don't change the drive, and should therefore be used whenever you change disks to avoid a fatal disk write error.

The new drive is not immediately shown on the menu, as it is not automatically updated. Hit for example <SPACE> to display a fresh menu which will show the new logged drive.

Work File Selection

The **W** command is used to select a work file which is the file to be used to Edit, Compile, Run, eXecute, and Save. The **W** command will issue this command:

```
Work file name: ■
```

and you may respond with any legal file name: a name of one through eight characters, an optional period, and an optional file type of no more than three characters:

```
FILENAME.TYP
```

If you enter a file name without period and file type, the file type *PAS* is automatically assumed and appended to the name. You may explicitly specify a file name with no file type by entering a period after the name, but omitting the type.

Examples:

`PROGRAM`	becomes `PROGRAM.PAS`
`PROGRAM.`	is not changed
`PROGRAM.FIL`	is not changed

File types .BAK, .CHN, and .COM/.CMD should be avoided, as TURBO uses these names for special purposes.

When the Work file has been specified, the file is read from disk, if present. If the file does not already exist, the message `New File` is issued. If you have edited another file which you have not saved, the message:

```
Workfile X:FILENAME.TYP not saved. Save (Y/N)? ■
```

warns you that you are about to load a new file into memory and overwrite the one you have just worked on. Answer **Y** to save or **N** to skip.

The new work file name will show on the menu the next time it is updated, like when you hit <SPACE>.

Main File Selection

The **M** command may be used to define a main file when working with programs which use the compiler directive $I to include a file. The Main file should be the file which which contains the include directives. You can then define the Work file to be different from the Main file, and thus edit different include files while leaving the name of the Main file unchanged.

When a compilation is started, and the Work file is different from the Main file, the current Work file is automatically saved , and the Main file is loaded into memory. If an error is found during compilation, the file containing the error (whether it is the Main file or an include file) automatically becomes the Work file which may then be edited. When the error has been corrected and compilation is started again, the corrected Work file is automatically saved, and the Main file is re-loaded.

The Main file name is specified as described for the Work file name in the previous section.

Edit Command

The **E** command is used to invoke the built-in editor and edit the file defined as the Work file. If no Work file is specified, you are first asked to specify one. The menu disappears, and the editor is activated. More about the use of the editor on pages 19 pp.

While you may use the TURBO system to compile and run programs without installing a terminal, the use of the editor requires that your terminal be installed. See page 12.

Compile Command

The **C** command is used to activate the compiler. If no Main file is specified, the Work file will be compiled, otherwise the Main file will be compiled. In the latter case, if the Work file has been edited, you will be asked whether or not to save it before the Main file is loaded and compiled. The compilation may be interrupted at any moment by pressing a key.

The compilation may result either in a program residing in memory, in a .COM file, or in a .CHN file. The choice is made on the compiler **O**ptions menu described on pages 190 (PC/MS-DOS systems), 227 (CP/M-86), and 259 (CP/M-80). The default is to have the program residing in memory.

Run Command

The **R** command is used to activate a program residing in memory or, if the **C**-switch on the compiler **O**ptions menu is active, a TURBO object code file (.COM or .CMD file). If a compiled program is already in memory, it will be activated. If not, a compilation will automatically take place as described above.

Save Command

The **S** command is used to save the current Work file on disk. The old version of this file, if any, will be renamed to .BAK, and the new version will be saved.

Directory Command

The **D** command gives you a directory listing and information about remaining space on the logged drive. When hitting **D**, you are prompted thus:

```
Dir mask: ■
```

You may enter a drive designator or a drive designator followed by a file name or a mask containing the usual wildcards `*` and `?`. Or you may just hit <RETURN> to get a full directory listing.

Quit Command

The **Q**uit command is used to leave the TURBO system. If the Work file has been edited since it was loaded, you are asked whether you want to save it before quitting.

compiler Options

The **O** command selects a menu on which you may view and change some default values of the compiler. It also provides a helpful function to find run-time errors in programs compiled into object code files.

As these options vary between implementations, further discussion is deferred to chapters 20, 21, and 22.

The TURBO Editor

The built-in editor is a full-screen editor specifically designed for the creation of program source text. If you are familiar with MicroPro's *WordStar*, you need but little instruction in the use of the TURBO editor, as all editor commands are exactly like the ones you know from *WordStar*. There are a few minor differences, and the TURBO editor has a few extensions; these are discussed on page 34. You may install your own commands 'on top' of the *WordStar* commands, as described on page 13; and IBM PC systems come with arrows and dedicated function keys already installed. The *WordStar* commmands, however, may still be used.

Using the TURBO editor is simple as can be: when you have defined a Work file and hit **E**, the menu disappears, and the editor is activated. If the Work file exists on the logged drive, it is loaded and the first page of text is displayed. If it is a new file, the screen is blank apart from the *status line* at the top.

You leave the editor and return to the menu by pressing **Ctrl-K-D**; more about that later.

Text is entered on the keyboard just as if you were using a typewriter. To terminate a line, press the <RETURN> key (or CR or ENTER or whatever it is called on your keyboard). When you have entered enough lines to fill the screen, the top line will scroll off the screen, but don't worry, it is not lost, and you may page back and forth in your text with the editing commands described later.

Let us first take a look at the meaning of the *status line* at the top of the screen.

The Status Line

The top line on the screen is the status line containing the following information:

```
Line n    Col n    Insert    Indent    X:FILENAME.TYP
```

Figure 1-5: Editor Status Line

Line n
Shows the number of the line containing the cursor counted from the start of the file.

Col n
Shows the number of the column containing the cursor counted from the left of the line.

Insert
Indicates that characters entered on the keyboard will be inserted at the cursor position. Existing text to the right of the cursor will move to the right as you write new text. Using the *insert mode on/off* command (**Ctrl-V** by default) will instead display the text **Overwrite**. Text entered on the keyboard will then overwrite characters under the cursor instead of inserting them.

Indent
Indicates that auto-indent is in effect. It may be switched off by the *auto-indent on/off* command (**Ctrl-Q-I** by default).

X:FILENAME.TYP
The drive, name, and type of the file being edited.

Editing Commands

As mentioned before, you use the editor almost as a typewriter, but as this is a computerized text editor it offers you a number of editing facilities which make text manipulation, and in this case specifically program writing, much easier than on paper.

The TURBO editor accepts a total of 45 editing commands to move the cursor around, page through the text, find and replace text strings, etc, etc. These commands can be grouped into the following four categories:

Cursor movement commands,
Insert and delete commands,
Block commands, and
Miscellaneous commands

Each of these groups contain logically related commands which will be described separately in following sections. The following table provides an overview of the commands available:

CURSOR MOVEMENT COMMANDS:
- Character left
- Character right
- Word left
- Word right
- Line up
- Line down
- Scroll up
- Scroll down
- Page up
- Page down
- To top of screen
- To top of file
- To top of file
- To end of file
- To left on line
- To right on line
- To beginning of block
- To end of block
- To last cursor position

INSERT & DELETE COMMANDS:
- Insert mode on/off
- Insert line
- Delete line
- Delete to end of line
- Delete right word
- Delete character under cursor
- Delete left character

BLOCK COMMANDS:
- Mark block begin
- Mark block end
- Mark single word
- Copy block
- Move block
- Delete block
- Read block from disk
- Write block to disk
- Hide/display block

MISC. EDITING COMMANDS:
- End edit
- Tab
- Auto tab on/off
- Restore line
- Find
- Find & replace
- Repeat last find
- Control character prefix

Table 1-1: Editing Command Overview

In a case like this, the best way of learning is by doing; so start TURBO, specify one of the demo Pascal programs as your **Work** file, and enter **E** to start Editing. Then use the commands as you read on.

Hang on, even if you find it a bit hard in the beginning. It is not just by chance we have chosen to make the TURBO editor *WordStar* compatible - the logic of these commands, once learned, quickly become so much a part of you that the editor virtually turns into an extension of your mind. Take it from one who has written megabytes worth of text with that editor.

Each of the following descriptions consists of a heading defining the command, followed by the default keystrokes used to activate the command, with room in between to note which keys to use on your terminal, if you use other keys. If you have arrow keys and dedicated word processing keys (insert, delete, etc.), it might be convenient to use these. Please refer to pages 13 pp for installation details.

The following descriptions of the commands assume the use of the default *WordStar* compatible keystrokes.

A Note on Control Characters

All commands are issued using control characters. A control character is a special character generated by your keyboard when you hold down the <CONTROL> (or <CTRL>) key on your keyboard and press any key from A through Z (well, even the [, \,], ^, and _ keys generate control characters for that matter).

The <CONTROL> key works like the <SHIFT> key: if you hold down the <SHIFT> key and press A, you will get a capital A; if you hold down the <CONTROL> key and press A, you will get a Control-A (Ctrl-A for short).

Before You Start: How To Get Out

The command which takes you out of the editor is described on page 30, but you may find it useful to know already now that the **Ctrl-K-D** command exits the editor and returns you to the menu environment. This command does not automatically save the file; that must be done with the **Save** command from the menu.

Basic Movement Commands

The most basic thing to learn about an editor is how to move the cursor around on the screen. The TURBO editor uses a special group of control characters to do that, namely the control characters **A**, **S**, **D**, **F**, **E**, **R**, **X**, and **C**.

Why these? Because they are conveniently located close to the control key, so that your left little finger can rest on that while you use the middle and index fingers to activate the commands. Furthermore, the characters are arranged in such a way on the keyboard as to logically indicate their use. Let's examine the basic movements: cursor up, down, left, and right:

```
   E
 S   D
  X
```

These four characters are placed so that it is logical to assume that **Ctrl-E** moves the cursor up, **Ctrl-X** down, **Ctrl-S** to the left, and **Ctrl-D** to the right. And that is exactly what they do. Try to move the cursor around on the screen with these four commands. If your keyboard has repeating keys, you may just hold down the control key and one of these four keys, and the cursor will move rapidly across the screen.

Now let us look at some extensions of those movements:

```
     E   R
A  S   D   F
    X   C
```

The location of the **Ctrl-R** next to the **Ctrl-E** suggests that **Ctrl-R** moves the cursor up, and so it does, only not one line at a time but a whole page. Similarly, **Ctrl-C** moves the cursor down one page at a time.

Likewise with **Ctrl-A** and **Ctrl-F**: **Ctrl-A** moves to the left like **Ctrl-S**, but a whole word at a time, and **Ctrl-F** moves one word to the right.

The two last basic movement commands do not move the cursor but scrolls the entire screen upwards or downwards in the file:

```
  W   E   R
A   S   D   F
  Z   X   C
```

Ctrl-W scrolls upwards in the file (the lines on the screen move down), and **Ctrl-Z** scrolls downwards in the file (the lines on the screen move up).

Character left **Ctrl-S**

Moves the cursor one character to the left non-destructively, without affecting the character there. < BACKSPACE > may be installed to have the same effect. This command does not work across line breaks; when the cursor reaches the left edge of the screen, it stops.

Character right **Ctrl-D**

Moves the cursor one character to the right non-destructively, without affecting the character there. This command does not work across line breaks, i.e. when the cursor reaches the right end of the screen, the text starts scrolling horizontally until the cursor reaches the extreme right of the line, in column 128, where it stops.

Word left **Ctrl-A**

Moves the cursor to the beginning of the word to the left. A word is defined as a sequence of characters delimited by one of the following characters: |space| < > , ; . ( ) [ ] ^ ' * + − / $. This command works across line breaks.

Word right **Ctrl-F**

Moves the cursor to the beginning of the word to the right. See the definition of a word above. This command works across line breaks.

Line up **Ctrl-E**

Moves the cursor to the line above. If the cursor is on the top line, the screen scrolls down one line.

Line down **Ctrl-X**

Moves the cursor to the line below. If the cursor is on the second-last line, the screen scrolls up one line.

Scroll up **Ctrl-W**

Scrolls 'up' towards the beginning of the file, one line at a time (the entire screen scrolls down). The cursor remains on its line until it reaches the bottom of the screen.

Scroll down **Ctrl-Z**

Scrolls 'down' towards the end of the file, one line at a time (the entire screen scrolls up). The cursor remains on its line until it reaches the top of the screen.

Page up **Ctrl-R**

Moves the cursor one page up with an overlap of one line; the cursor moves one screenful less one line backwards in the text.

Page down **Ctrl-C**

Moves the cursor one page down with an overlap of one line; the cursor moves one screenful less one line forwards in the text.

Extended Movement Commands

The commands discussed above will let you move freely around in your program text, and they are easy to learn and understand. Try to use them all for a while and see how natural they feel.

Once you master them, you will probably sometimes want to move more rapidly. The TURBO editor provides six commands to move rapidly to the extreme ends of lines, to the beginning and end of the text, and to the last cursor position.

These commands require **two** characters to be entered: first a **Ctrl-Q** and then one of the following control characters: **S**, **D**, **E**, **X**, **R**, and **C**. They repeat the pattern from before:

```
  E R
S D
 X C
```

Ctrl-Q-S moves the cursor to the extreme left of the line, and **Ctrl-Q-D** moves it to the extreme right of the line. **Ctrl-Q-E** moves the cursor to the top of the screen, **Ctrl-Q-X** moves it to the bottom of the screen. **Ctrl-Q-R** moves the cursor all the way 'up' to the start of the file, **Ctrl-Q-C** moves it all the way 'down' to the end of the file.

To left on line **Ctrl-Q-S**

Moves the cursor all the way to the left edge of the screen, to column one.

To right on line **Ctrl-Q-D**

Moves the cursor to the end of the line to the position following the last printable character on the line. Trailing blanks are always removed from all lines to preserve space.

To top of screen **Ctrl-Q-E**

Moves the cursor to the top of the screen.

To bottom of screen **Ctrl-Q-X**

Moves the cursor to the bottom of the screen.

To top of file **Ctrl-Q-R**

Moves to the first character of the text.

To end of file **Ctrl-Q-C**

Moves to the last character of the text.

Finally the **Ctrl-Q** prefix with a **B**, **K**, or **P** control character allows you to jump far within the file:

To beginning of block **Ctrl-Q-B**

Moves the cursor to the the position of the *block begin* marker set with **Ctrl-K-B** (hence the Q-**B**). The command works even if the block is not displayed (see *hide/display block* later), or the *block end* marker is not set.

To end of block **Ctrl-Q-K**

Moves the cursor to the position of the *block end* marker set with **Ctrl-K-K** (hence the Q-**K**). The command works even if the block is not displayed (see *hide/display block* later), or the *block begin* marker is not set.

To last cursor position **Ctrl-Q-P**

Moves to the last **P**osition of the cursor. This command is particularly useful to move back to the last position after a **S**ave operation or after a find or find/replace operation.

Insert and Delete Commands

These commands let you insert and delete characters, words, and lines. They can be divided into three groups: one command which controls the text entry mode (insert or overwrite), a number of simple commands, and one extended command.

Notice that the TURBO editor provides a 'regret' facility which lets you 'undo' changes *as long as you have not left the line.* This command (**Ctrl-Q-L**) is described on page 31.

Insert mode on/off **Ctrl-V**

When you enter text, you may choose between two entry modes: *Insert* and *Overwrite*. Insert mode is the default value when the editor is invoked, and it lets you insert new text into an existing text. The existing text to the right of the cursor simply moves to the right while you enter the new text.

Overwrite mode may be chosen if you wish to replace old text with new text. Characters entered then replace existing characters under the cursor.

You switch between these modes with the *insert mode on/off* command **Ctrl-V**, and the current mode is displayed in the status line at the top of the screen.

Delete left character <DEL>

Moves one character to the left and deletes the character there. Any characters to the right of the cursor move one position to the left. The <BACKSPACE> key which normally backspaces non-destructively like **Ctrl-S** may be installed to perform this function if it is more conveniently located on your keyboard, or if your keyboard lacks a <DELETE> key (sometimes labeled <DEL>, <RUBOUT>, or <RUB>). This command works across line breaks, and can be used to remove line breaks.

Delete character under cursor **Ctrl-G**

Deletes the character under the cursor and moves any characters to the right of the cursor one position to the left. This command does not work across line breaks.

Delete right word **Ctrl-T**

Deletes the word to the right of the cursor. A word is defined as a sequence of characters delimited by one of the following characters: lspacel < > , ; . ( ) [ ] ^ ' * + − / $. This command works across line breaks, and may be used to remove line breaks.

Insert line **Ctrl-N**

Inserts a line break at the cursor position. The cursor does not move.

Delete line **Ctrl-Y**

Deletes the line containing the cursor and moves any lines below one line up. The cursor moves to the left edge of the screen. No provision exists to restore a deleted line, so take care!

Delete to end of line **Ctrl-Q-Y**

Deletes all text from the cursor position to the end of the line.

Block Commands

All block commands are extended commands (two characters each in the standard command definition), and you may ignore them at first if you feel a bit dazzled at this point. Later on, when you feel the need to move, delete, or copy whole chunks of text, you should return to this section.

For the persevering, we'll go on and discuss the use of *blocks*.

A block of text is simply any amount of text, from a single character to several pages of text. A block is marked by placing a *Begin block* marker at the first character and an *End block* marker at the last character of the desired portion of the text. Thus marked, the block may be copied, moved, deleted, and written to a file. A command is available to read an external file into the text as a block, and a special command conveniently marks a single word as a block.

Mark block begin **Ctrl-K-B**

This command marks the beginning of a block. The marker itself is not visible on the screen, and the block only becomes visibly marked when the *End block* marker is set, and then only if the screen is installed to show some sort of highlighting. But even if the block is not visibly marked, it is internally marked and may be manipulated.

Mark block end **Ctrl-K-K**

This command marks the end of a block. As above, the marker itself is not visible on the screen, and the block only becomes visibly marked when the *Begin block* marker is also set.

Mark single word **Ctrl-K-T**

This command marks a single word as a block, and thus replaces the *Begin block* - *End block* sequence which is a bit clumsy when marking just one word. If the cursor is placed within a word, then this word will be marked; if not then the word to the left of the cursor will be marked. A word is defined as a sequence of characters delimited by one of the following characters: |space| < > , ; . ( ) [ ] ^ ' * + − / $.

Hide/display block **Ctrl-K-H**

This command causes the visual marking of a block (dim text) to be alternately switched off and on. Block manipulation commands (copy, move, delete, and write to a file) work only when the block is displayed. Block related cursor movements (jump to beginning/end of block) work whether the block is hidden or displayed.

Copy block **Ctrl-K-C**

This command places a copy of a previously marked block starting at the cursor position. The original block is left unchanged, and the markers are placed around the new copy of the block. If no block is marked, the command performs no operation, and no error message is issued.

Move block **Ctrl-K-V**

This command moves a previously marked block from its original position to the cursor position. The block disappears from its original position and the markers remain around the block at its new position. If no block is marked, the command performs no operation, and no error message is issued.

Delete block **Ctrl-K-Y**

This command deletes the previously marked block. No provision exists to restore a deleted block, so be careful!

Read block from disk **Ctrl-K-R**

This command is used to read a file into the current text at the cursor position, exactly as if it was a block that was moved or copied. The block read in is marked as a block. When this command is issued, you are prompted for the name of the file to read. The file specified may be any legal filename. If no file type is specified, .PAS is automatically assumed. A file without type is specified as a name followed by a period.

Write block to disk **Ctrl-K-W**

This command is used to write a previously marked block to a file. The block is left unchanged, and the markers remain in place. When this command is issued, you are prompted for the name of the file to write to. If the file specified already exists, a warning is issued before the existing file is overwritten. If no block is marked, the command performs no operation, and no error message is issued.The file specified may be any legal filename. If no file type is specified, .PAS is automatically assumed. A file without type is specified as a name followed by a period. Avoid the use of file types .BAK, .CHN, and .COM/.CMD, as they are used for special purposes by the TURBO system.

Miscellaneous Editing Commands

This section collects a number of commands which do not logically fall into any of the above categories. They are nonetheless important, especially this first one:

End edit **Ctrl-K-D**

This command ends the edit and returns to the main menu. The editing has been performed entirely in memory, and any associated disk file is not affected. Saving the edited file on disk is done explicitly with the **S**ave command from the main menu or automatically in connection with a compilation or definition of a new Work file.

Tab **TAB/Ctrl-I**

There are no fixed tab positions in the TURBO editor. Instead, tab positions are automatically set to the beginning of each word on the line immediately above the cursor. This provides a very convenient automatic tabbing feature especially useful in program editing where you often want to line up columns of related items, like variable declarations and such. Remember that Pascal allows you to write extremely beautiful source texts - do it, not for the sake of the purists, but more importantly to keep the program easy to understand, especially when you return to make changes after some time.

Auto indent on/off **Ctrl-Q-I**

The auto indent feature provides automatic indenting of successive lines. When active, the indent of the current line is repeated on each following line, that is, when you hit < RETURN > , the cursor does not return to column one but to the starting column of the line you just terminated. When you want to change the indent, use any of the cursor right or left commands to select the new column. When auto indent is active, the message **Indent** is displayed in the status line, and when passive the message is removed. Auto indent is active by default.

Restore line **Ctrl-Q-L**

This command lets you regret changes made to a line *as long as you have not left the line.* The line is simply restored to its original contents regardless of what changes you have made. But only as long as you remain on the line; the moment you leave it, changes are there to stay. For this reason, the *Delete line* (**Ctrl-Y**) command can regrettably only be regretted, not restored. Some days you may find yourself continuously falling asleep on the Ctrl-Y key, with vast consequences. A good long break usually helps.

Find **Ctrl-Q-F**

The Find command lets you search for any string of up to 30 characters. When you enter this command, the status line is cleared, and you are prompted for a search string. Enter the string you are looking for and terminate with < RETURN > . The search string may contain any characters, also control characters. Control characters are entered into the search string with the **Ctrl-P** prefix. Example: enter a **Ctrl-A** by holding down the Control key while pressing first P, then A. You may thus include a line break in a search string by specifying **Ctrl-M Ctrl-J**. Notice that **Ctrl-A** has a special meaning: it matches any character and may be used as a wildcard in search strings.

Search strings may be edited with the *Character Left, Character Right, Word Left,* and *Word Right* commands. *Word Right* recalls the previous search string which may then be edited. The search operation may be aborted with the Abort command (**Ctrl-U**).

When the search string is specified, you are asked for search options. The following options are available:

B Search backwards. Search from the current cursor position towards the *beginning* of the text.

G Global search. Search the entire text, irrespective of the current cursor position.

n n = any number. Find the n'th occurrence of the search string, counted from the current cursor position.

U Ignore upper/lower case. Regard upper and lower case alphabeticals as equal.

W Search for whole words only. Skip matching patterns which are embedded in other words.

Examples:

W search for whole words only. The search string 'term' will only match the word 'term', not for example the word 'terminal'.

BU search backwards and ignore upper/lower case. 'Block' will match both 'blockhead' and 'BLOCKADE', etc.

125 Find the 125th occurrence of the search string.

Terminate the list of options (if any) with <RETURN>, and the search starts. If the text contains a target matching the search string, the cursor is positioned at the end of the target. The search operation may be repeated by the *Repeat last find* command (**Ctrl-L**).

Find and replace **Ctrl-Q-A**

The Find and Replace command lets you search for any string of up to 30 characters and replace it with any other string of up to 30 characters. When you enter this command, the status line is cleared, and you are prompted for a search string. Enter the string you are looking for and terminate with <RETURN>. The search string may contain any characters, also control characters. Control characters are entered into the search string with the **Ctrl-P** prefix. Example: enter a **Ctrl-A** by holding down the Control key while pressing first P, then A. You may thus include a line break in a search string by specifying **Ctrl-M Ctrl-J**. Notice that **Ctrl-A** has a special meaning: it matches any character and may be used as a wildcard in search strings.

Search strings may be edited with the *Character Left, Character Right, Word Left,* and *Word Right* commands. *Word Right* recalls the previous search string which may then be edited. The search operation may be aborted with the Abort command (**Ctrl-U**).

When the search string is specified, you are asked to enter the string to replace the search string. Enter up to 30 characters; control character entry and editing is performed as above, but **Ctrl-A** has no special meaning in the replace string. If you just press <RETURN>, the target will be replaced with nothing, in effect deleted.

Finally you are prompted for options. The search and replace options are:

B Search and replace backwards. Search and replace from the current cursor position towards the *beginning* of the text.

G Global search and replace. Search and replace in the entire text, irrespective of the current cursor position.

n n = any number. Find and replace *n* occurrences of the search string, counted from the current cursor position.

N Replace without asking. Do not stop and ask *Replace (Y/N)* for each occurrences of the search string.

U Ignore upper/lower case. Regard upper and lower case alphabeticals as equal.

W Search and replace whole words only. Skip matching patterns which are embedded in other words.

Examples:

N10 Find the next ten occurrences of the search string and replace without asking.

GW Find and replace whole words in the entire text. Ignore upper/lower case.

Terminate the list of options (if any) with <RETURN>, and the search and replace starts. Depending on the options specified, the string may be found. When found (and if the **N** option is not specified), the cursor is positioned at the end of the target, and you are asked the question: `Replace (Y/N)?` on the prompt line at the top of the screen. You may abort the search and replace operation at this point with the Abort command (**Ctrl-U**). The search and replace operation may be repeated by the *Repeat last find* command (**Ctrl-L**).

Repeat last find **Ctrl-L**

This command repeats the latest *Find* or *Find and replace* operation exactly as if all information had been re-entered.

Control character prefix **Ctrl-P**

The TURBO editor allows you to enter control characters into the file by prefixing the desired control character with a **Ctrl-P**, that is, first press **Ctrl-P**, then press the desired control character. Control characters will appear as low-lighted capital letters on the screen (or inverse, depending on your terminal).

Abort operation **Ctrl-U**

The **Ctrl-U** command lets you abort any command in process whenever it pauses for input, like when Search and Replace asks *Replace Y/N?*, or during entry of a search string or a file name (block Read and Write).

The TURBO editor vs. WordStar

Someone used to *WordStar* will notice that a few TURBO commands work slightly different, and although TURBO contains only a subset of *WordStar's* commands, a number of special features not found in *WordStar* have been added to enhance the editing of program source code. These differences are discussed in the following.

Cursor Movement

The cursor movement controls **Ctrl-S**, **D**, **E**, and **X** move freely around on the screen and do not jump to column one on empty lines. This does not mean that the screen is full of blanks; on the contrary, all trailing blanks are automatically deleted. This way of moving the cursor is especially useful for example when matching indented **begin** - **end** pairs.

Ctrl-S and **Ctrl-D** do not work across line breaks. To move from one line to another you must use **Ctrl-E**, **Ctrl-X**, **Ctrl-A**, or **Ctrl-F**.

Mark Single Word

Ctrl-K-T is used to mark a single word as a block which is more convenient than the two-step process of marking the beginning and the end of the word separately.

End Edit

The **Ctrl-K**-D command ends editing and returns you to the menu. As editing in TURBO is done entirely in memory, this command does not change the file on disk (as it does in WordStar). Updating the disk file must be done explicitly with the **S**ave command from the main menu or automatically in connection with a compilation or definition of a new Work file. TURBO's **Ctrl-K-D** does not resemble WordStar's **Ctrl-K-Q** (quit edit) command either, as the changed text is not abandoned; it is left in memory ready to be **C**ompiled or **S**aved.

Line Restore

The **Ctrl-Q-L** command restores a line to its contents before edit *as long as the cursor has not left the line.*

Tabulator

No fixed tab settings are provided. Instead, the automatic tab feature sets tabs to the start of each word on the line immediately above the cursor.

Auto Indentation

The **Ctrl-Q-I** command switches the auto indent feature on and off.

Notes:

Chapter 2
BASIC LANGUAGE ELEMENTS

Basic Symbols

The basic vocabulary of TURBO Pascal consists of basic symbols divided into letters, digits, and special symbols:

Letters
A to Z, a to z, and _ (underscore)
Digits
0 1 2 3 4 5 6 7 8 9
Special symbols
+ - * / = ^ < > ( ) [ ] { } . , : ; ' # $

No distinction is made between upper and lower case letters. Certain operators and delimiters are formed using two special symbols:

Assignment operator: :=
Relational operators: <> <= >=
Subrange delimiter: ..
Brackets: (. and .) may be used instead of [and]
Comments: (* and *) may be used instead of { and }

Reserved Words

Reserved words are integral parts of TURBO Pascal. They cannot be redefined and must therefore not be used as user defined identifiers.

```
* absolute
  and
  array
  begin
  case
  const
  div
  do
  downto
  else
  end
* external
  file
  forward
  for
  function
  goto
* inline
  if
  in
  label
  mod
  nil
  not
* overlay
  of
  or
  packed
  procedure
  program
  record
  repeat
  set
* shl
* shr
* string
  then
  type
  to
  until
  var
  while
  with
* xor
```