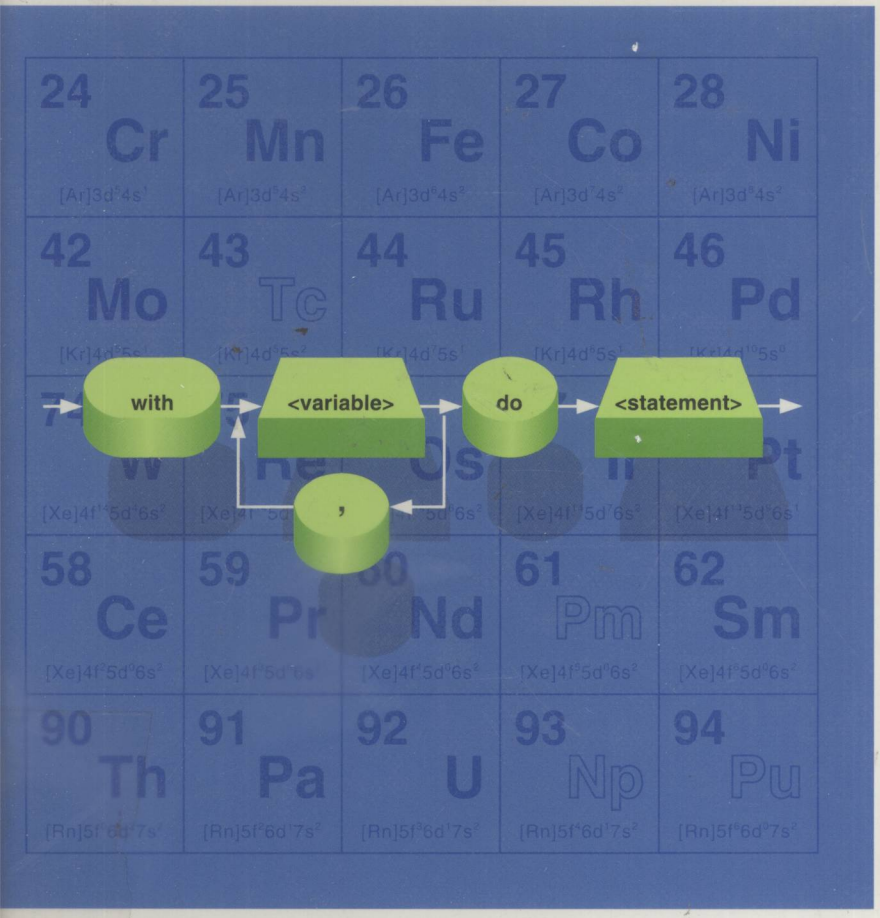
G. Filby, M. Klusmann

Turbo Pascal for Chemists

A problem solving and practical approach



VCH

Gordon Filby, Martin Klusmann

Turbo Pascal for Chemists

A problem solving and practical approach

VCH Weinheim · New York · Basel · Cambridge

Dr. Gordon Filby
Im Zeitvogel 10
D-7500 Karlsruhe 41

Dr. Martin Klusmann
Roonstr. 23
D-7500 Karlsruhe 1

1st edition 1992

Published jointly by
VCH Verlagsgesellschaft mbH, Weinheim (Federal Republic of Germany)
VCH Publishers, Inc. New York, NY (USA)

Editorial Director: Dr. Hans F. Ebel
Production Manager: Dipl.-Wirt.-Ing. (FH) Bernd Riedel

Library of Congress Card No. applied for

British Library Cataloguing-in-Publication Data
Filby, Gordon
Problem solving in chemistry with Turbo Pascal:
A problem solving and practical approach.
I. Title II. Klusmann, Martin
540.285
ISBN 3-527-27830-3

Die Deutsche Bibliothek – CIP-Einheitsaufnahme
Filby, Gordon:
Turbo Pascal for chemists : a problem solving and practical approach / Gordon Filby ; Martin Klusmann. – 1. ed. – Weinheim ; New York ; Basel ; Cambridge : VCH, 1992
ISBN 3-527-27830-3 (Weinheim ...)
ISBN 0-89573-874-0 (New York)
NE: Klusmann, Martin:

Printing: betz-druck gmbh, D-6100 Darmstadt 12
Bookbinding: J. Schäffer, D-6718 Grünstadt

Printed in the Federal Republic of Germany

Turbo Pascal Quick Overview

Simple Statements

Assignment Statement

```
Diff:= N1 - N2;
ARecd:= SameRecdType;
AnArray[Index]  := 8.314;
Town.Name := 'Hameln'
AnArray:= SameArrayType;
```

Compound Statement

```
IF X > 0 THEN
  BEGIN
    WRITELN ('T');
    WRITELN ('T');
    WRITELN ('Tst')
  END
```

Conditional Statements

IF-THEN-ELSE-Statement

```
IF X > 0 THEN
  WRITELN ('Greater than zero')
ELSE
  WRITELN ('Zero or negative');
```

IF-THEN-Statement

```
IF X > 0 THEN
  WRITELN;
```

CASE Statement

```
CASE Letter OF
  Vowel  : WRITELN ('Vowel');
  Consont: WRITELN;('Consonant')
ELSE
  WRITE ('Natl.Character')
END;
```

Looping Statements

WHILE Statement

```
WHILE NrIts < MaxNrIts  DO
  BEGIN
    NrIts := NrIts + 1;
    Calc := Calc + 2.0
  END;
```

REPEAT Statement

```
REPEAT
  Sum := Sum + Next;
  READ (Next)
UNTIL Next <= 0;
```

FOR Statement (basic form)

```
FOR J := 20 to 500 DO
  BEGIN
    Sum := Sum + J;
    WRITELN (Sum)
  END;
```

FOR Statement (DOWNTO form)

```
FOR J := 500 DOWNTO 20 DO
  BEGIN
    Sum := Sum + J;
    WRITELN (Sum)
  END;
```

Input and Output Statements (Examples)

- Standard

```
READ(N1,N2);
READLN(N1,N2);
WRITE('The numbers are ' , N1, N2);
WRITELN('The numbers are ', N1, N2);
```

- File

```
READ(FileVar, Nx);
WRITELN(FileOut, Ny);
```

Sample Declarations

Array

```
TYPE
  NrCalibPoints = 1..5;
  CalibData = ARRAY [NrCalibPoints] of REAL;
VAR
  ExptlData: CalibData;
```

String

```
TYPE
  Str79 = STRING [79];
VAR
  Longstr: Str79;
```

Record

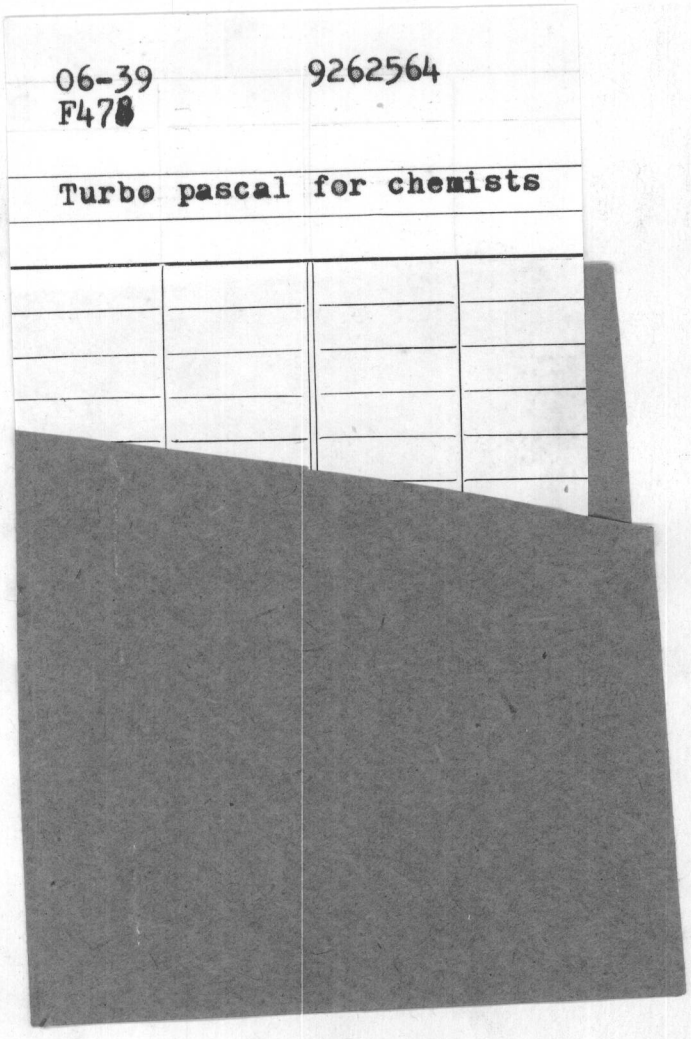
```
TYPE
  Element = RECORD
              ElmtName : STRING [40];
              AtWt     : REAL;
              BoilPt   : -22..300;
            END;
  SomeElements = ARRAY [1..6] of Element;
VAR
  PeriodicGroup: SomeElements;
```

Set

```
TYPE
  Colours = (Red, Orange, Yellow, Green, Blue, Indigo, Violet);
  ColoursSet = SET of Colours;
VAR
  VisSpectrum: ColoursSet;
```

Filby, Klusmann

Turbo Pascal for Chemists



Distribution
VCH, P.O. Box 10 11 61, D-6940 Weinheim (Federal Republic of Germany)
Switzerland: VCH, P.O. Box, CH-4020 Basel (Switzerland)
United Kingdom and Ireland: VCH (UK) Ltd., 8 Wellington Court, Cambridge CB1 IHZ (England)
USA and Canada: VCH, Suite 909, 220 East 23rd Street, New York, NY 10010-4606 (USA)

ISBN 3-527-27830-3 (VCH, Weinheim)

ISBN 0-89573-874-0 (VCH, New York)

Preface

Pascal is a computer programming language developed in the pre-microcomputer days of the late 1960's by Prof. Niklaus Wirth of the Eidgenossische Technische Hochschule in Zürich. At its inception the prime purpose of the language was to impart the then relatively new structured programming techniques to ongoing information scientists.

From its first rather humble beginnings as a "mere" pedagogic aid it has risen to become probably the most widely used of all programming languages for the IBM PC. Furthermore, it is believed by many to be the best high level language for computing beginners.

A computer literature search using the Bowkers Books in Print database conducted during the initiation phase of this project (early 1987) showed that there are already sufficient books on Pascal to fill a small library. Of these however, few were addressed to a scientific audience and still fewer, namely none, specifically to chemists. It seemed to be high time to change this situation.

In this book we shall be talking about a special dialect of Pascal for micro-computers, known as Turbo Pascal, first released by Borland International of Scotts Valley, California in October 1983. We began writing this book with the third update of the compiler; shortly before its completion (1991), Version 6.0 had been released. All versions feature a complete integrated programming environment including a full screen editor, a source code error detection system and a fast source code compiler. Additional bonuses include many extensions to standard Pascal and the ability to produce executable programs in the environment. These features have all been greatly enhanced in the succeeding versions. With the addition of object-oriented programming (OOP) extensions to the latest versions, the compiler has now also become a powerful tool for professional programmers.

This book is intended as a simple self-teaching guide for chemists of all ages interested in using Turbo Pascal on an IBM or compatible Personal Computer* to tackle a wide range of general chemical problems. The material of this book has been chosen with this in mind. Thus:

---

* We do not cover versions of Turbo Pascal available for other computers, such as the Apple Macintosh. Similarly, the recently released Turbo Pascal for Windows is not discussed.

- The chemistry is of undergraduate level, with only occasional forays into some contemporary research topics.

- No mathematics beyond high school algebra and calculus is required.

- It is not a completely rigorous treatment of Pascal. Rather it is the presentation of a useful subset intended to introduce interested chemists to the possibilities of this ubiquitous computer language.

In common with learning a foreign language, learning a programming language becomes easier the more it is used. Thus our goal is a hands-on approach in which the reader is encouraged to run exercise programs on the diskette provided. Suggestions for Practice sections then require him to make a number of especially chosen modifications as aids to understanding the underlying Pascal feature. This method has the consequence that sometimes, in order to provide interesting examples of some substance as early as possible in the text, Pascal features have been presented somewhat out of the familiar order of treatment. Whenever this is the case readers are referred to the location in the text where the main discussion of that feature is presented.

In addition to exercises on the diskette, a number of more general exercises are included. As an aid to their solution they have been placed under the heading of the most similar diskette example.

Although we attempted at all times to keep the example material within the chemical sphere, it was sometimes neither avoidable nor desirable to avoid the "Hello World" and "address book" type programs so beloved of computer book authors. To those offended or bored, we apologise.

Acknowledgements

A number of people shared in our efforts to bring this book to fruition. We are especially grateful to the following:

Mr. K. Kanz (author of Pascal Chemie Programme), Mr. G. Warnke, Drs. W.H. Kalus and D.Bothomley for their comments on the manuscript.

Dr. H.F. Ebel and Mr. Zühlke of the staff of our publisher, VCH.

Mrs. E. Dahlke for struggling with many versions of the manuscript on a Xerox workstation.

Dr. J. Schilling of Juniata College, Huntingdon, USA for his countless suggestions on all aspects of this book.

Thanks go also to Ms. Nan Borreson of Borland International for providing an early release of Turbo Pascal Version 6.

WGF especially wishes to thank the management of the Kernforschungszentrum, Karlsruhe (Prof. Dr. W. Klose; Dr. F. Horsch) for so readily granting him permission to take part in this project and for the use of the electronic publishing equipment used for the final formatting of the book.

A Note to the Reader

The level of computer programming knowledge among chemists being as heterogeneous as it is, we consider it wise to point out how we believe this book might best be used.

Chapter 1 will only be required by those readers with little or no experience of using a computer. Chapter 2 contains a brief review of the essentials of the Pascal language and therefore probably contains useful knowledge for readers at all levels. Programming beginners may need to linger a while on this chapter. Depending on their familiarity with control structures (IF THEN, WHILE etc.) readers can then either carry straight on to the data structure chapters (Chapters 3-5) or jump to Chapter 7 to gain the necessary familiarity with them. The remainder of the book can then be read in numerical order.

Typographic Conventions

For reasons of space not all programs discussed in the text can be reproduced in full. Complete running programs are indicated in the text as Diskette Program \Chapx\ProgName.Pas. All running diskette programs reproduced in the text will be printed as follows:

```
WRITELN(ProgHeader); WRITELN;WRITELN;
WRITELN('Using: ',TempC:6:2,' deg C ');
WRITELN('Activation Energy ',ActivEner:10:4) ;
WRITELN('Pre-Exp.Factor ', PreExp);
```

Program fragments are presented as follows:

```
WRITELN(ProgHeader); WRITELN;WRITELN;
WRITELN('Using: ',TempC:6:2,' deg C ');
```

References to a Line in the main text or the Suggestions for Practice sections, for example:

On line 21 change 35 to 36 and rerun the program. Explain the strange result. Hint: C is defined as BYTE.

are to the line number of the program *when loaded into your Turbo Pascal editor*.

Differences between Version 3 and following versions of Turbo Pascal will be indicated as follows:

Version 4..6 *The text USES CRT belongs to the special syntax of Turbo Pascal 4..6 and will be discussed in a later chapter. It informs the compiler that the UNIT CRT will be used in the program. Version 3.x users will have to remove or comment out this text before attempting to run the program.*

The example programs on the diskette are set to run, in the majority of cases, with Versions 4..6 of the compiler. Version 3 users will need to attend to the program commenting to run the programs in their compiler. Usually this will only entail removing the statement USES CRT.

Some Assumptions about the Reader

In order to keep this book to a tolerable length, we have had to make a number of assumptions about the computer literacy of our audience.

Generally speaking we assume the following about the reader:

- That his or her computer is an IBM PC or compatible, fitted with two disk drives or a hard disk.

- That he has a version of Turbo Pascal not lower than Version 3 installed on the hard disk and can load and start it. He should also own the corresponding Borland handbooks.

- That he or she has at least some previous programming experience. Having looked sometime in his or her life at a Pascal program for longer than a couple of minutes would also be a help, but is not essential.

Absolute beginners may be helped by the Glossary and Bibliography provided at the end of the book.

The Preparation of This Book

For those interested in such matters, this book was written entirely on an IBM PC-XT using Microsoft Word 4.0. The text was subsequently copied to a Xerox 6085 Workstation for the final formatting to camera-ready form. Program output was captured from the screen using the program Snapshot contained in the public domain utility package Baker's Dozen.

Diskette Program Directory

The following list shows the location and brief contents of all programs and data files provided on the diskette accompanying this book. All programs carry the conventional .Pas extension unless stated otherwise.

The disk does **not** contain the Turbo Pascal software.

Typographic Note

For "multiple word" file names, like BiMolRct and MFreePth, we maintain the naming convention used for user defined identifiers in the programs. That is, we capitalize the first letter of each "word" in the file name. Please note that this is our convention and not DOS's or Turbo Pascal's. It is merely intended to aid the reader in quickly understanding what is in the file. Both DOS and Turbo Pascal will understand any file name that is spelled correctly, no matter what combination of upper- and lower case letters it contains. There is no need to type in literally BiMolRct.Pas, bimOlRCt or BIMOlRCT.PAS; any combination will do. Of course, the last corresponds exactly to the result of a DOS dir call.

Chapter 2

Anatomy of a Turbo Pascal Program

Arrh: Pre-exponential factor, activation energy and reaction temperature (deg. C) are entered. Program displays rate constant.

Arrh.Err: Another version of the above containing deliberate mistakes.

Chapter 3

Pascal Data Types

Enumex: Shows some simple manipulations of enumerated and subrange types.

IntgrTyp: Illustrates notation and type compatibility of integer types.

LngInt: Demonstrates the new integer data types in Turbo Pascal 4..6.

Vars: Demonstrates Turbo Pascal variable types.

VdWVir: Calculates gas pressure according to van der Waals and second virial coefficient, compares with ideal gas law.

VdWVir.Err: As above, with mistakes.

Chapter 4

Data Structures: Part I- Arrays and Strings

2D-Array: Fills a doubly indexed array with random numbers and displays results.

2D-ArrOp: Illustrates some common manipulations of a two-dimensional array.

ArStrCmp: Demonstrates compatibility of STRING and ARRAY OF CHAR types.

AssgnArr: Shows whole array assignment of compatible array types.

ConnInd: Uses an array of enumerated type in a calculation of the connectivity index of saturated hydrocarbons.

MaxArray: Finds the position and value of the largest element of a one-dimensional array.

MaxArray.Err: As above but containing three different kinds of errors. Hints in the source may help.

Mucalc: Uses simple one-dimensional arrays to calculate ionic strength.

Rd1dim: Reads data into a one-dimensional array.

Rd2dim: Fills a structured two-dimensional array from the keyboard.

SrchArry: Searches for the first occurrence of a given value in an array.

StrDem: Shows the string concatenation operator and the use of a text string and variable of the same name ("Formula" and Formula).

StrOptrs: Uses the string-handling functions to rename a hypothetical data file. Also contains hint for Suggestions for Practice.

TextUtil.Inc: An include file containing a small "toolbox" of text utility subprograms. For example, to centre text, trim leading and trailing blanks etc.

UseTxtUt: A simple demonstration of the Textutil.Inc file.

Chapter 5

Data Structures : Part II - Records and Sets

Addtvty: Uses record structure containing Benson additivity contributions to the standard entropy and enthalpy of alkane bonds. User enters number of bonds and molecular symmetry, program displays standard entropy and enthalpy.

HalgnSet: Uses the halogens to demonstrate some set manipulations.

Hplc: An hplc evaluation program. Uses array of records data structure containing fields for substance name, retention time, area and response factor. After initialisation with typed constants, displays analytical results as table.

Pollhlfl: Calculates the half-life of an atmospheric pollutant. Uses initialised fields from an array of records.

RecinRec: Shows manipulation of nested record types.

TstSets: Exercises set operations on a broad basis of set types.

ValidRd: Illustrates how a set can be used to obtain valid character input.

WithExam: Simple use of WITH and period (.) handling of record fields.

Chapter 6

Operators, Expressions and Simple Statements

BiMolRct: Estimates the rate of a bimolecular gas reaction using collision theory.

BiMolRct.Err: As above but won't give the right answer. Why?

BoolIns: A demo program showing use of Booleans expressions and comparison operators.

ByteOps: Simple illustration of byte operators.

Cramer: A precedence exercise. Cramer's rules for the solution of simultaneous equations go wrong several times.

Effus: Vapour pressure determination by the Knudsen method. Because of very small intermediate results, needs to use co-processor emulation and TP4..6's DOUBLE data type.

HarmOsc: Implements the standard equation for the fundamental vibrational frequency of a harmonic oscillator for NO. Uses constant expressions (therefore TP5 and higher only).

MFreePth: Implements the equation for the mean free path of a gas molecule for oxygen.

StefBolt: Calculates the Stefan - Boltzmann constant.

StefBolt.Err: Calculates above wrongly. Examine precedence to correct it.

Chapter 7

Decision Statements and Program Flow Control

C13: Uses the BSA empirical parameters to calculate C13 shifts in hydrocarbon NMR spectra. A nested REPEAT ..UNTIL example.

CoCuSep: Simulates a countercurrent separation of amino acids. Uses the WHILE statement.

Cosi: A REPEAT..UNTIL example. Calculates a cosine using the Taylor series.

DnglElse: Shows how nested IF THEN ELSEs can lead to strange results.

ForFor: Shows nesting of FOR loops.

GasEff: Illustrates a simple FOR loop with integer counter using Graham's law of gaseous effusion.

GasSpd: Demonstrates an enumerated type as control variable in a FOR loop. Calculates velocities of several gaseous substances at desired temperature.

HiRes: A FOR example. Given the lower and upper limits on the number of CHNO atoms expected in a target molecule and a high resolution mass spectral peak, the program searches for fragments fitting it.

HydOrbtl: A REPEAT UNTIL example. Calculates bonding and antibonding orbitals for the hydrogen molecule.

Metas: A nested FOR example. Using mass spectrum of benzene as dummy data, calculates parent-daughter peaks from metastable ion spectrum.

MolCond: Uses an IF THEN construct to calculate the molar conductivity of KCl solutions (see also MolIfIf).

Molifif: The same calculation as MolCond carried out slightly differently.

OffByOne.Err: A Suggestions for Practice program. Incorrect Boolean condition causes off-by-one error.

PHCorr: Performs temperature correction of pH of standard M/20 potassium phthalate solution. Uses a CASE statement and subrange to select appropriate correction equation.

pHSlt: Uses a FOR loop to calculate and display concentration dependence of pH for a dilute aqueous salt solution.

RptVWhil: A Suggestions for Practice program. REPEAT UNTIL and WHILE won't even add 1-5.

WhleExam.Err: A Suggestions for Practice example. An infinite loop using WHILE.

Chapter 8

Procedures and Functions

AtmHyd: Calculates spectral lines of atomic hydrogen. User chooses series name, program displays the first five members.

BchMrk: A modified Savage benchmark program. Illustrates a number of predefined Turbo Pascal functions in a single assignment statement. Also uses Version 5's GETTIME function.

ChrSide: Shows how mixing local and global variables can produce surprising results.

DebyTmp: Uses the Debye equation to calculate the specific heats of several solids. Uses procedure parameter passing to a Simpson's numerical integration subprogram.

Entrop: Calculate entropy change using Simpson integration.