

TURBO PASCAL[®] PROGRAM LIBRARY

```
procedure MatShow(var Matrix: MatrixType);  
var  
  I, J: integer;  
begin  
  for I := 1 to MatrixSize do  
    begin  
      for J := 1 to MatrixSize do  
        write(Matrix[I,J]:10:2);      (Mod. B1)  
        writeln  
      end;  
      writeln  
    end;  
end;
```

Turbo Pascal Program Library

**Tom Rugg
Phil Feldman**

**Que™ Corporation
Indianapolis**

Turbo Pascal Program Library. Copyright © 1986 by Que Corporation.

All rights reserved. Printed in the United States of America. No part of this book may be used or reproduced in any form or by any means, or stored in a data base or retrieval system, without prior written permission of the publisher except in the case of brief quotations embodied in critical articles and reviews. Making copies of any part of this book for any purpose other than your own personal use is a violation of United States copyright laws. For information, address Que Corporation, 7999 Knue Road, Indianapolis, Indiana 46250.

Library of Congress Catalog No.: LC 86-60590

ISBN 0-88022-244-1

This book is sold *as is*, without warranty of any kind, either express or implied, respecting the contents of this book, including but not limited to implied warranties for the book's quality, performance, merchantability, or fitness for any particular purpose. Neither the authors nor Que Corporation nor its dealers or distributors shall be liable to the purchaser or any other person or entity with respect to any liability, loss, or damage caused or alleged to be caused directly or indirectly by this book.

90 89 88 87

8 7 6 5 4

Interpretation of the printing code: the rightmost double-digit number is the year of the book's printing; the rightmost single-digit number, the number of the book's printing. For example, a printing code of 87-4 shows that the fourth printing of the book occurred in 1987.

Dedication

To Gabby Johnson,
for clearly stating what needed to be said.

About the Authors

Tom Rugg

A native Californian, Mr. Rugg received his B.S. degree in Quantitative Methods from California State University in 1969. His 20 years of experience in computer programming and systems analysis include positions with GTE Data Services, Inc., the U. S. Army (Pentagon), and Jet Propulsion Laboratory. He has written or co-authored many articles in personal computer magazines and has coauthored with Phil Feldman a number of books, including *32 BASIC Programs for the Apple Computer* and *More Than 32 BASIC Programs for the IBM Personal Computer*. Mr. Rugg is president of 32 Plus, Inc., a software development and consulting firm.

Phil Feldman

Mr. Feldman received his B.A. degree in Physics from the University of California in 1968 and did graduate work in Computer Science at U.C.L.A. For 15 years he worked at TRW Systems, Inc., where he was a member of the Technical Staff and Project Manager of Software Development on Aerospace Engineering Projects. He has written or coauthored many articles in engineering technical journals and personal computer magazines, and has coauthored with Tom Rugg a number of books, including *TRS-80 Color Programs* and *Mind Moves*. Mr. Feldman is chairman of 32 Plus, Inc.

Preface



Our design philosophy for the *Turbo Pascal Program Library* is easily stated: "Make it a programming companion. Create a reference for repeated use." Contained in these pages is a comprehensive library of practical Turbo Pascal subprograms (procedures and functions).

At first glance, it may seem odd that our *program* library is composed of *subprograms*, not programs. The reason is that Pascal encourages a modular programming style in which most of a program's detailed tasks are delegated to subprograms. The subprograms are then assimilated into the full program to form one cohesive unit.

This book is a library of subprograms that are meant to be building blocks toward your finished programs. Think of these subprograms as the foundation stones on which your programming construction projects are built. Whether your undertaking is simple or complex, you should find in this book useful components that will save you time and trouble.

Is this book for you? If you program in Turbo Pascal, the answer is yes. By taking advantage of our ready-made routines, you can improve your productivity immediately. The first time you use one of these subprograms, you should be able to justify the cost of the book in the hours you save by not having to develop original code. And if you use any of our complete programs, your savings are even greater.

If you are an intermediate or advanced programmer, think of this book as an instant software reference library. For you, its primary benefits are simplified program development and expediency. If you are a beginner in Pascal, you receive the additional benefit of practical program components you might otherwise be unable to develop yourself.

We assume that you know how to program in Pascal, at least rudimentarily. This book is neither a tutorial on Pascal in general nor a tutorial on Turbo Pascal in particular. (Many excellent books that teach these subjects are available.) At a minimum, you should know how to get Pascal running on your computer, type a program, save it on disk, compile the program, and run it.

Even if you are just beginning to learn Pascal (especially if you are a “learn by doing” person), this book will be a useful companion. We explain many programming tips and techniques throughout the presentation of these subprograms. Additionally, the exposure to well-structured, useful subprograms and full programs will help you understand more than just the rules of Pascal. Pascal encourages a clean, structured, top-down programming style, and we have tried to exemplify it. We hope this book will help you become a more knowledgeable programmer.

Our target audience is the largest group of Turbo Pascal users—those with IBM PCs or compatibles. However, those who use Turbo Pascal on other computers (CP/M® computers, Apple®, etc.) should find the subprograms helpful, as will users of other versions of Pascal. You may have to make some changes, though, to enable certain programs to work.

Acknowledgments

The work of a few is always the result of many. We are especially grateful . . .

To Maria Katinos for the figures and research. She never floundered.

To the imponderable Dave Feldman. His contributions are simply not explainable.

To everyone at Que. Their faith and perseverance made this book a reality. Special thanks to Ginny Noble, Pegg Kennedy, and Chris DeVoney.

To Philippe Kahn, president of Borland International. His vision resulted in a watershed software product. Turbo is much more than just a Pascal compiler for microcomputers. It is a well designed programming environment. Its power, ease of use, and fair price have created a standard for future software products. The growing popularity of Turbo Pascal is a testament to this foresight.

And, finally, to Louisa Gillette for her understanding and support. We love her.

Product Development Director

Chris DeVoney

Editorial Director

David F. Noble, Ph.D.

Managing Editor

Gregory Croy

Editor

Virginia Noble, M.L.S.

Technical Editor

Art Sterling

Composed in Garamond and Que Digital

by Que Corporation

Cover designed by

Listenberger Design Associates

Trademark Acknowledgments

Que Corporation has made every effort to supply trademark information about company names, products, and services mentioned in this book. Trademarks indicated below were derived from various sources. Que Corporation cannot attest to the accuracy of this information.

Apple is a registered trademark of Apple Computer, Inc.

CP/M is a registered trademark of Digital Research Inc.

EPSON is a registered trademark of Epson Corporation. MX-80 is a trademark of EPSON America, Inc.

IBM is a registered trademark of International Business Machines Corporation. IBM PC XT and IBM *PCjr* are trademarks of International Business Machines Corporation.

MS-DOS is a registered trademark of Microsoft Corporation.

Turbo Pascal is a registered trademark of Borland International, Inc.

Table of Contents

Preface

Chapter 1	How To Use This Book	1
	The Structure of a Pascal Program.....	1
	The Program Library Concept.....	2
	The Assimilation Process.....	3
	A Disk Offer for Lazy Typists.....	3
	Hardware and Software Configurations.....	3
	Documenting the Subprograms.....	4
	A Quicklook Box.....	4
	Description.....	5
	Input.....	5
	Output.....	5
	Limitations and Error Conditions.....	5
	Sample Usage.....	5
	Subprogram Listing.....	6
	Variables.....	6
	Discussion.....	6
	Modifications.....	6
	Style Conventions.....	6
	How To Get Started.....	6

Subprograms for Input/Output

Chapter 2	Getting Keyboard Input	7
	GetReply Detects a keystroke within a specified range.....	8
	GetKey Detects a keystroke from an eligible list.....	10
	KeyHit Detects whether a key has been pressed and, if so, which one.....	14
	GetNumR Gets a real number (or a special character) typed from the keyboard.....	17
	GetNumI Gets an integer number (or a special character) typed from the keyboard.....	22
	WaitKey Makes a program pause until a key is pressed.....	24
	KeyArr Enters numeric data from the keyboard into an array.....	26

Key2Arr	Enters keyboard data into a two-dimensional array	30
KeyTxt	Enters text data from the keyboard into an array	35
Chapter 3	Creating Graphics Effects	39
TextBox	Draws a “graphics” box while the computer is in text mode	40
Scroll	Scrolls text in any of four directions	45
CursorOn	Turns the cursor on or off while the computer is in text mode	51
GrCursor	Positions the cursor on a graphics screen	54
GraphBox	Draws boxes (with shading) while the computer is in graphics mode	57
DoXAxis	Draws a horizontal x-axis for a subsequent graph	63
DoYAxis	Draws a vertical y-axis for a subsequent graph	68
AnnXAxis	Annotates a horizontal x-axis for a subsequent graph	73
AnnYAxis	Annotates a vertical y-axis for a subsequent graph	80
AxisText	Fills a text array with strings suitable for numeric axes annotations	87
Chapter 4	Creating Screen Displays	93
Tab	Moves the cursor to a specified column	93
TabXY	Moves the cursor to a specified column and row	96
ShowFast	Displays a string of text on the screen at high speed	99
ShowArr	Displays a numeric array’s contents on the screen	102
Show2Arr	Displays a two-dimensional array’s contents on the screen	106
Chapter 5	Using the Printer	113
UnderSc	Prints underscored text	113
BoldFace	Prints boldfaced text	117

StrikeOv	Prints overstruck text	121
NewPage	Causes the printer to eject to the next page	124
PrtSc	Prints a copy of the current screen contents	126

Chapter 6 Manipulating Disk Files 129

DiskSp	Determines how much disk space is available	130
VerDisk	Turns the DOS disk verification switch on or off	133
ShowDir	Displays a directory or subdirectory on the screen	135
NextFile	Finds the name of the next file in a disk directory	140
FileInfo	Obtains information about a specified disk file	145
FileDel	Deletes a disk file	151
SaveArr	Saves an array's contents onto a disk file	154
Save2Arr	Saves a two-dimensional array's contents onto a disk file	158
SaveTxt	Saves a string array's contents onto a text disk file	162
LoadArr	Loads an array from data saved in a disk file	166
Load2Arr	Loads a two-dimensional array from data saved in a disk file	170
LoadTxt	Loads a text (string) array from data saved in a text disk file	175

Chapter 7 Making Music and Other Sounds 181

PlayInit	Initializes global variables for playing music	181
PlayTune	Plays music, using a tune definition language	187
SoundEff	Plays various sound effects through the computer's speaker	196
Beep	Sounds beeping noises from the speaker	202

ParseKW	Detects a keyword at a specific position in a string	283
ParseRN	Detects a real number at a specific position in a string	288
ParseIN	Detects an integer number at a specific position in a string	292

Chapter 11 Converting 297

IntToHex	Converts an integer number into a four-character hex string	297
HexToInt	Converts a hex string of one to four characters into an integer number	300
ConvUp	Converts a string to uppercase letters.....	304
ConvLow	Converts a string to lowercase letters.....	306

Subprograms That Manipulate Data (Numerical)

Chapter 12 Mathematics and Engineering 309

Power	Performs exponentiation of real numbers	310
HyperLib	Calculates the hyperbolic functions sinh, cosh, and tanh	313
CrossPr	Calculates the cross product of two vectors.....	315
DotPr	Calculates the dot product of two vectors.....	318
MatShow	Displays the contents of a matrix.....	321
MatAdd	Adds two matrices	324
MatSub	Subtracts one matrix from another	327
MatMult	Multiplies two matrices	330
MatTrans	Transposes a matrix	334
MatInv	Inverts a matrix	336
Determ	Evaluates the determinant of a matrix.....	341
Integral	Evaluates a definite integral of a given function.....	345
Root	Calculates real roots of a given function	351
Deriv	Calculates the derivative of a given function.....	357
DiffEqn	Solves a specified differential equation	361

Chapter 13 Statistics and Probability 367

Stats	Calculates statistics for a set of discrete values	368
NumPerm	Calculates the number of permutations	372
NumComb	Calculates the number of combinations	375
BinomDis	Computes binomial distribution probabilities	379
NormDis	Evaluates frequencies and probabilities of a normal distribution function	383
RndNorm	Generates random numbers from a normal distribution	389
LinReg	Performs linear regression on a set of data pairs	393
GetChiSq	Calculates the chi-square statistic	398
ChiProb	Calculates the probability of achieving particular values of chi-square	402

Chapter 14 Business and Finance 409

FutGrow	Projects future values based on a growth rate	410
GrowRate	Calculates the average growth rate of time-series data	413
InvestFV	Calculates the final value of an investment	417
InvestP	Calculates the principal of an investment	420
InvestIR	Calculates the interest rate of an investment	422
InvestNM	Calculates the term of an investment	425
LoanPay	Calculates the regular payment for a loan	428
LoanPrin	Calculates the principal for a loan	431
LoanTNP	Calculates the total number of payments required to pay back a loan	433
AnnuFV	Calculates the final value of an annuity	436
AnnuDep	Calculates the regular deposit for an annuity	439
AnnuTND	Calculates the total number of deposits for an annuity	442
WDrawP	Calculates the principal for an investment with withdrawals	445
WDrawWD	Calculates the maximum periodic withdrawal from an investment	447
WDrawTNW	Calculates the total number of withdrawals until an investment is depleted	450

Subprograms of Useful Utilities

Chapter 15	Examining and Altering the Computer Environment	455
GetSysID	Determines which type of IBM PC is in use	455
GetEquip	Determines the computer's equipment configuration	458
VidMode	Determines information about the current video mode	462
VideoChg	Changes between monochrome and color/graphics video screens or determines which screen is in use	465
Chapter 16	Handling Errors	473
IOError	Performs error checking after an I/O operation.....	473
RunError	Intercepts run-time errors and displays informative messages before aborting	477
Chapter 17	Manipulating Times and Dates	483
GetTime	Retrieves the current time of day in numeric form	483
ShowTime	Displays the current time of day on the screen	486
SetTime	Sets the computer's internal clock to a specified time	488
TimeDiff	Calculates the number of seconds between two times	490
GetDate	Retrieves the current date in numeric form	493
ShowDate	Displays the current date on the screen.....	496
SetDate	Sets the computer's internal date.....	498
Julian	Calculates the Julian day number for a given year, month, and day	500
JulToYMD	Converts a Julian day number into year-month-day form.....	504
DayWeek	Determines the day of the week for a given date	506