TABLE OF CONTENTS

PREFACE
FOREWORD
0. A VERY INFORMAL INTRODUCTION

	x.1. FUNDAMENTALS	*x*.2. PROCEDURES AND NAMES	*x*.3. OPERATIONS	*x*.4. STRUCTURES	*x*.5. MULTIPLE VALUES	*x*.6. UNIONS	*x*.7. DISTINCTIVE FEATURES
1.*y*. BASIC CONCEPTS	1.1. FUNDAMENTALS 1.1.1. Objects 1.1.2. Identifiers 1.1.2.1. Variable declarations 1.1.2.2. Assignation, collateral elaboration 1.1.3. Phrases, serial and collateral elaboration 1.1.4. Routines 1.1.5. Defining and applied occurrences 1.1.6. Coercion	1.2. NAMES AND DECLARERS 1.2.1. Ascription and assignation 1.2.2. Identity declarations 1.2.2.1. Constants 1.2.2.2. Equivalences 1.2.2.3. Local generators 1.2.2.4. Variables and names 1.2.2.5. Casts 1.2.3. The metanation MODE 1.2.3.1. **proc** modes 1.2.3.2. The supply of the actual parameters 1.2.4. Summary	1.3. SYMBOLS, MODES AND OPERATORS 1.3.1. Representations 1.3.2. Symbols, bold words and comments 1.3.3. Other declarations 1.3.3.1. Mode declarations 1.3.3.2. Operation declarations 1.3.3.3. Priority declarations	1.4. STOWED VALUES, STRUCTURES 1.4.0. STOWED values 1.4.1. Enumeration by tagging 1.4.1.1. Structured constants 1.4.1.2. Names of structures 1.4.1.3. Creation of new structures 1.4.2. Different objects in one box 1.4.3. Chaining 1.4.4. Pandora's boxes	1.5. STOWED VALUES, MULTIPLES 1.5.1. Multiple values and descriptors 1.5.2. Indexing 1.5.2.1. Indexers 1.5.2.2. Subscripting 1.5.2.3. Trimming 1.5.3. Identifier declarations for multiples 1.5.4. Slices 1.5.5. Interrogations	1.6. UNIONS 1.6.1. United modes 1.6.1.1. United constants 1.6.1.2. Equivalence of unions 1.6.1.3. Local united generation 1.6.2. Assignations and conformity clauses	1.7. DISTINCTIVE FEATURES 1.7.1. The **long** and **short** modes 1.7.2. Identity relations
2.*y*. DECLARATIONS	2.1. PRIMITIVE DECLARATIONS 2.1.1. Primitives 2.1.2. Variable declarations 2.1.3. Sample declarations	2.2. IDENTITY DECLARATIONS 2.2.1. Identity declarations 2.2.2. Another look at variable declarations 2.2.3. Initialised variable declarations	2.3. MODE DECLARATIONS	2.4. STRUCTURE DECLARATIONS 2.4.1. **struct** declarers 2.4.2. **struct** declarations 2.4.3. Well formed modes 2.4.4. The mode **compl**	2.5. MULTIPLE DECLARATIONS 2.5.1. Row declarers 2.5.2. Row declarations 2.5.2.1. Fixed and flexible names 2.5.2.2. Actual 'row of' declarers 2.5.2.3. Summary 2.5.3. The mode **string**	2.6. UNION DECLARATIONS 2.6.1. **union** declarers 2.6.2. **union** declarations	2.7. BITS, BYTES, LONGS AND SHORTS 2.7.1. **bits** and **bytes** 2.7.2. **long** and **short** modes 2.7.3. **heap** declarations
3.*y*. CLAUSES	3.1. SERIAL CLAUSES 3.1.1. The declarations 3.1.2. The statements 3.1.3. The yield 3.1.4. Completers 3.1.5. Delimiters	3.2 CLOSED CLAUSES 3.2.1. Ranges and reaches 3.2.2. Scopes of names 3.2.3. Identification 3.2.4. ENCLOSED clauses 3.2.4.1. Closed clauses 3.2.4.2. Conditional clauses 3.2.4.3. Case clauses	3.3. BOLD WORDS 3.3.1. Identification of mode indications	3.4. STRUCTURE DISPLAYS	3.5. ROW DISPLAYS AND LOOPS 3.5.1. Row displays 3.5.2. Loop clauses	3.6. CONFORMITY CLAUSES	3.7. COLLATERALITY 3.7.1. Collateral clauses 3.7.2. Parallel clauses
4.*y*. ROUTINES	4.1. PROCEDURES AND OPERATORS 4.1.1. Standard prelude routines	4.2. PROCEDURE DECLARATIONS 4.2.1. **proc** declarers 4.2.2. Routines 4.2.2.1. Routine texts 4.2.2.2. Calling 4.2.2.3. Recursion 4.2.3. Scopes of routines	4.3. OPERATION DECLARATIONS 4.3.1. Priority declarations 4.3.2. Operation declarations 4.3.3. Identification of operators	4.4. **skip**	4.5. ROW-OF PARAMETERS	4.6. **skip**	4.7. JUMPS 4.7.1. Simple jumps 4.7.2. Procedured jumps
	x.1. FUNDAMENTALS	*x*.2. PROCEDURES AND NAMES	*x*.3. OPERATIONS	*x*.4. STRUCTURES	*x*.5. MULTIPLE VALUES	*x*.6. UNIONS	*x*.7. DISTINCTIVE FEATURES
5.*y*. UNITS	5.1. SIMPLE UNITS	5.2 BALANCE AND CALL	5.3. **skip**	5.4. UNITS AND STRUCTURES	5.5. UNITS AND MULTIPLES	5.6. UNITS AND UNIONS	5.7. BITS AND PIECES OF GARBAGE
5.*y*.0. Coercion	5.1.0.1. Coercends 5.1.0.2. Coercion 5.1.0.3. Dereferencing 5.1.0.4. Widening	5.2.0.1. ENCLOSED clauses and balancing 5.2.0.2. Deproceduring		5.4.0. Complex widening	5.5.0. Rowing	5.6.0. Uniting	5.7.0.1. Voiding 5.7.0.2. **bits** and **bytes** widening
5.*y*.1. Primaries	5.1.1.1. Denotations 5.1.1.2. Applied identifiers 5.1.1.3. Casts	5.2.1. Procedure calls		5.4.1. Applied identifiers	5.5.1.1. String denotations 5.5.1.2. Applied identifiers 5.5.1.3. Slices		5.7.1.1. **bits** denotations 5.7.1.2. **long** and **short** denotations
5.*y*.2. Secondaries	5.1.2. Secondaries			5.4.2. Selections	5.5.2. Multiple selections		5.7.2.1. **loc** generators 5.7.2.2. **heap** generators
5.*y*.3. Tertiaries	5.1.3. Formulas	5.2.3. **nil**		5.4.3. Formulas with complex operators	5.5.3. Bound interrogations		5.7.3. Order of elaboration of operands
5.*y*.4. Quarternaries	5.1.4.1. Assignations 5.1.4.2. **skip**	5.2.4. Assignations involving names			5.5.4.1. Flexible assignations 5.5.4.2. Assignation to slices 5.5.4.3. Overlapping slices	5.6.4. Assignations of unions of rows	5.7.4. Identity relations
6.*y*. STANDARD PRELUDE	6.1 OPERATORS 6.1.1. Monadic operators 6.1.2. Dyadic operators	6.2. CONSTANTS AND PROCEDURES 6.2.1. Constants 6.2.2. Procedures	6.3. ASSIGNING OPERATORS	6.4. **skip**	6.5. INTERROGATIONS 6.5.1. Dyadic operators 6.5.2. Monadic operators	6.6. **skip**	6.7. LONG OPERATORS 6.7.1. Environment enquiries 6.7.2. Procedures 6.7.3. Operators 6.7.4. **leng** and **shorten** 6.7.5. **up** and **down**
7.*y*. TRANSPUT	7.1 FORMATLESS TRANSPUT 7.1.1. Formatless output 7.1.2. Formatless input	7.2. FILES 7.2.1. Channels, books and files 7.2.2. Environment enquiries 7.2.3. Procedures for opening and closing 7.2.4. Position enquiries 7.2.5. Layout routines	7.3. **skip**	7.4. STRUCTURES AND EVENTS 7.4.1. Straightening of structures 7.4.2. Files 7.4.3. Code conversion 7.4.4. Event routines 7.4.4.1. On logical file end 7.4.4.2. On physical file end 7.4.4.3. On page end 7.4.4.4. On line end 7.4.4.5. On format end 7.4.4.6. On value error 7.4.4.7. On char error	7.5. ROWS AND STRINGS 7.5.1. Straightening of multiple values 7.5.2. Conversion procedures 7.5.3. Conversion environment enquiries	7.6. FORMATTED TRANSPUT 7.6.1. **format** texts 7.6.1.1. Literals 7.6.1.2. Alignments 7.6.1.3. Frames 7.6.1.4. Replicators and collections 7.6.2. **formats** 7.6.3. The formatted transput procedures 7.6.4. Events	7.7. BINARY TRANSPUT 7.7.1. Binary transput procedures 7.7.2. Some restrictions
8.*y*. EXAMPLES	8.1. SIMPLE EXAMPLES	8.2. PROCEDURE EXAMPLES 8.2.1. Easter	8.3. EXAMPLES OF OPERATORS 8.3.1. Parallel plus	8.4. TWO EXAMPLES OF LIBRARY PRELUDES 8.4.1. Operations on vectors in E_3 8.4.1.1. Comments on 8.4.1. 8.4.1.2. An example of the use of **vecs** 8.4.2. Operations on rational operands 8.4.2.1. Comments on the library prelude 8.4.2. 8.4.2.2. Some remarks on the use of rationals	8.5. A LIBRARY PRELUDE FOR VECTOR AND MATRIX OPERATIONS IN E_n 8.5.1. Operations on vectors in E_n 8.5.2. Operations on matrices and vectors 8.5.3. Operations on square matrices	8.6. EXAMPLES OF TRANSPUT 8.6.1. The happy family	8.7. EXAMPLES OF EVERYTHING 8.7.1. Analytic differentiation
APPENDICES	APPENDIX 1 Alternative representations	APPENDIX 2 Sample declarations	APPENDIX 3 Glossary 1. Internal objects and modes 2. External objects 3. Technical terms	APPENDIX 4 The sublanguage	APPENDIX 5 The standard hardware representation	APPENDIX 6 Syntax charts	