QUASIC USER MANUAL

VERSION: V3A
MANUAL : 1/7/71

LIST OF CONTENTS

1 INTRODUCTION

THE QUASIC SYSTEM IS DESIGNED TO ENABLE INTERACTIVE DATA-ACQUISITION, ANALYSIS AND RETRIEVAL TO BE CARRIED OUT, MONITORED AND CONTROLLED, THROUGH A COMPACT MINICOMPUTER AND DIGITAL COMMUNICATIONS CONTROLLER. CURRENTLY THE SYSTEM HAS BEEN IMPLEMENTED ON A MODIFIED TSS8 TIME-SHARING COMPUTER WITH 20K WORDS OF 12-BIT CORE, 512K WORDS OF DISC AND TWO TAPE TRANSPORTS. THIS SYSTEM IS BEING OFFERED AS A GENERAL SERVICE FOR CLINICAL AND PSYCHOLOGICAL STUDIES USING A SUITE OF RECORD-KEEPING AND FILM-TERMINAL DATA-ACQUISTION PACKAGES. THE QUASIC SYSTEM IS ALSO BEING IMPLEMENTED ON SMALLER CONFIGURATIONS FOR DIRECT SALE AS IN-HOUSE SYSTEMS.

THE QUASIC SYSTEM IS CONTROLLED THROUGHOUT IN THE QUASIC LANGUAGE, AND ALL FUNCTIONS, INCLUDING SYSTEM-LEVEL EXECUTIVE PROGRAMS ARE SPECIFIED IN QUASIC. THUS THE ENTIRE STRUCTURE OF THE SYSTEM, AS SEEN BY THE USER, MAY BE ESTABLISHED RAPIDLY IN A 'HIGH-LEVEL' LANGUAGE WITHOUT RECOURSE TO ASSEMBLY LANGUAGE. THE SYNTAX OF THE QUASIC LANGUAGE IS MODELLED ON DARTMOUTH COLLEGE BASIC SINCE THIS HAS PROVEN EXCEPTIONALLY SIMPLE IN APPLICATION. HOWEVER, THE IMPLEMENTATION IS THROUGH COMPLETELY TABLE-DRIVEN CALLS ON MODULAR ROUTINES, SO THAT NEW PROCESSORS MAY BE ADDED AT WILL, EG SPECIAL-TERMINAL CONTROLLERS OR DATA-ANALYSERS. THE CURRENT IMPLEMENTATION ON THE TSS8 HAS BEEN DESIGNED FOR INTERACTION THROUGH THE TYPEWRITER AND MICROFILM TERMINALS, AND CONTAINS PARTICULAR FACILITIES FOR NATURAL-LANGUAGE INTERACTION ON THE ONE AND PSYCHOLOGICAL TESTING ON THE OTHER.

APART FROM THESE SPECIAL FACILITIES, THE QUASIC LANUAGE GIVES FULL SYSTEM-PROGRAMMING CAPABILITIES FOR MANIPULATING SYSTEM TIMINGS, FILES AND A VARIETY OF DATA STRUCTURES ON DISC AND TAPE. ONE NOTABLE FEATURE OF THE LANGUAGE IS THAT IS NON-PROCEDURAL AND PROGRAMS CAN MODIFY THEMSELVES, BOTH PRODUCING NEW CODE AND ERAZING OLD (AN IMPORTANT FEATURE ON A SMALL SYSTEM). ARBITRARY CHARACTER STRINGS MAY BE MANIPULATED AS DATA ITEMS THROUGH PATTERN-IDENTIFICATION MACROS, AND HENCE THE SYSTEM MAY BE GIVEN ANY REQUIRED DEGREE OF 'NATURAL-LANGUAGE' INTERACTION. IN PARTICULAR, USING THE MACRO-FACILITIES, INTERPRETERS MAY BE WRITTEN IN QUASIC FOR OTHER LANGUAGES, AND BY COMBINING THIS WITH THE RUN-TIME COMPILATION FACILITIES QUASIC PROGRAMS MAY BE GENERATED BY TRANSLATION FROM OTHER LANGUAGES.

2 THE QUASIC LANGUAGE

QUASIC ON THE TSS8 FOLLOWS THE SYNTAX OF BASIC VERY CLOSELY. THAT IS, A PROGRAM CONSISTS OF A SET OF NUMBERED COMMAND LINES IN WHICH THE LINE NUMBERS CONTROL THE ORDER OF EXECUTION OF THE COMMANDS. THUS :-

```
 10 PRINT 'HELLO'
 45 INPUT X Y
 78 LET Z=X*Y
200 PRINT Z Z*Z
500 STOP
```

IS A LEGAL QUASIC PROGRAM. IF WE TYPE IT AT THE TELEPRINTER AND THEN TYPE 'RUN', THE RESULTS IS :-

```
>RUN    [THE '>' (GO-AHEAD) INDICATES COMPUTER READY FOR INPUT
HELLO
:5 7        [NUMBERS TYPED IN
   35 1225

QUASIC V2K

>
```

HOWEVER, QUASIC HAS MANY FEATURES NOT AVAILABLE IN BASIC, SOME OF WHICH ARE FOR THE CONVENIENCE OF THE PROGRAMMER, BUT OTHERS OF WHICH ARE FUNDAMENTAL TO ITS POWER AS A SYSTEM PROGRAMMING LANGUAGE. AS AN EXAMPLE OF THE FORMER, NO DISTINCTION IS MADE BETWEEN INDIRECT & DIRECT COMMANDS, AND ALL MAY BE EXECUTED INDIRECTLY AS PART OF A STORED PROGRAM, OR DIRECTLY AT THE KEYBOARD. IN ADDITION COMMANDS MAY BE ABBREVIATED TO THE SHORTEST UNAMBIGUOUS STRING, ENABLING THE PROGRAMMER'S INTERACTION WITH THE SYSTEM TO BE AS CONCISE AS POSSIBLE:-

```
>P67
   67
>P56/7
    8
>L A=10
>P A*A
  100
>
```

BUT ALL ABBREVIATIONS IN STORED PROGRAMS ARE RECONSTITUTED IN LISTINGS :-

```
>10L E3=9
>45P9*E3
>567G 10
>LI      [NOTE THAT L ALONE WOULD BE TAKEN AS 'LET'

 10 LET E3=9
 45 PRINT 9*E3
567 GOTO 10
```

AS AN EXAMPLE OF A FACILITY WHICH INCREASES THE POWER OF THE LANGUAGE, MULTIPLE COMMANDS MAY BE INCLUDED IN ONE PROGRAM LINE PROVIDED A COLON IS PLACED BEFORE EACH COMMAND NAME :-

```
>LET A=7 :PRINT A*A
   49
>L A=7:P A*A
   49
>10L I=0
>20P I I*I:L I=I+1:IFI<5:G20
>LI

 10 LET I=0
 20 PRINT I I*I  :LET I=I+1  :IF I<5  :GOTO 20

>RUN
    0     0
    1     1
    2     4
    3     9
    4    16

QUASIC V2K

>
```

IN PARTICULAR ANY COMMAND MAY FOLLOW A CONDITIONAL :-

```
>IF 5>4 :PRINT 6
   6
>LET A=9 :IF A<10 :LET A=7 :PRINT A*A
   49
>
```

THE CONCEPT THAT THE SATIFACTION OF A CONDITIONAL EXPRESSION CAUSES THE REST OF THE PROGRAM LINE TO BE EXECUTED IS EXTENDED SO THAT MANY COMMANDS HAVE IMPLIED CONDITIONALS, THE FLOW OF EXECUTION BEING ALONG THE PROGRAM LINE (ACROSS THE PAGE) IF THEY ARE SATISFIED, AND TO THE NEXT LINE (DOWN THE PAGE) IF THEY ARE NOT. THIS APPLIES PARTICULARLY TO THE STRING MACROS (THE 'PUT' COMMAND) AND FILE-HANDLING COMMANDS TO BE DISCUSSED LATER.

3 ARITHMETIC EXPRESSIONS

ONLY INTEGER ARITHMETIC MODULO 4096 IS PROVIDED IN TSS8 QUASIC SINCE THE SYSTEM IS NOT INTENDED TO BE USED FOR DATA-PROCESSING INVOLVING FLOATING-POINT ARITHMETIC (FILES MAY BE TRANSFERRED OFF-LINE TO OTHER SYSTEM FOR STATISTICAL PROCESSING, OR PROCESSED ON THE DEDICATED PDP8 AT NIGHT). APART FROM THIS, ARITHMETIC EXPRESSIONS ARE EVALUATED IN THE NORMAL MANNER, AND +, -, *, /, ^, !, ., (, 9, MAY BE USED TO FORM EXPRESSIONS, EG -

```
>PRINT (6-4)*(5-(6/3))^2   ((7-5).7)!1+8
   18     11
>
```

THE ARITHMETIC OPERATIONS, IN ORDER OF DECREASING PRECEDENCE, ARE:-

1 EXPRESSIONS WITHIN PAIRED PARENTHESES - THE INNERMOST EXPRESSION BEING EVALUATED FIRST.

2 ^ - RAISE TO A POWER - EG 2^3 = 8

3 * - MULTIPLY - EG 4*5 = 20

4 / - DIVIDE - EG /2 = 4

5 - - SUBTRACT - EG 9-7 = 2

6 + - ADD - EG 6+5 = 11

7 . - LOGICAL AND - EG 3.1 = 1, 7.10 = 0, 71.33 = 31

8 ! - LOGICAL OR - EG 1!1 = 1, 1!0 = 1, 8!1 = 9, 7!1 = 7

9 LEFT TO RIGHT EXECUTION

NOTE THAT MULTIPLICATION HAS A GREATER PRECEDENCE THAN DIVISION SO THAT 3*3/9 = 1, 9/3*3 = 1 (THE GREATER PRECEDENCE OF SUBTRACTION OVER ADDITION HAS NO EFFECT). THE LOGICAL OPERATORS OPERATE ON FULL 12-BIT WORDS & MAY BE USED FOR MASKING AND BIT MANIPULATION.

IF AN EXPECTED DATA ITEM IN A NUMERIC EXPRESSION IS ABSENT, ITS VALUE IS TAKEN TO BE ZERO. THUS -

```
>LET X =
>LET X = :LET Y = 7
>LET X = 7*
>
```

ALL CAUSE X TO BE ASSIGNED THE VALUE ZERO. ONE EFFECT OF THIS IS TO TAKE CARE OF UNARY OPERATORS CORRECTLY (-1 IS TAKEN AS 0-1) AND ANOTHER IS TO MAKE ABSENCE OF A NUMERIC EXPRSSION LEAD TO A ZERO EVALUATION WHICH IS USEFUL IN DEFAULT OPTIONS.

TO ENABLE MULTI-LENGTH CALCULATIONS TO BE CARRIED OUT REASONABLY AND WORD-FIELDS TO BR DECODED EFFECTIVELY, THE SYSTEM VARIABLE QR IS SET TO THE MOST SIGNIFICANT PART OF A PRODUCT, OR THE REMAINDER IN A DIVISION. IT IS NOT AFFECTED BY ADDITION, SUBTRACTION OR LOGICAL OPERATIONS (BUT IS AFFECTED BY ^ SINCE TAKING POWERS IS DONE BY REPEATED MULTIPLICATION). IN THE EVENT OF MULTIPLE OPERATIONS AFFECTING QR IT IS SET BY THE LAST ONE EXECUTED (THIS IS NOT NECESSARILY THE LAST RELEVANT OPERATOR IN A LINE). EG :-

```
>LET A = 8/3 :PRINT QR
     2
>PRINT 13/11   QR
     1     2
>PRINT 3/4   5+6   QR
     0    11     3
>PRINT 50*100   QR
 904      1
>
```

THE SAFEST WAY IN WHICH TO USE QR IS WITH EXPRESSIONS CONTAINING ONLY ONE MULTIPLICATION OR DIVISION AND NO POWERS. NOTE THAT ALTHOUGH QR RETAINS ITS VALUE UNTIL SET UP AGAIN, THERE ARE OTHER COMMANDS WHICH SET IT UP AS A SIDE-EFFECT (INPUT & RETURN) & IT IS SAFEST TO USE ITS VALUE IMMEDIATELY OR TRANSFER IT TO ANOTHER VARIABLE AS SOON AS POSSIBLE.

3.1 SIMPLE VARIABLES

SIMPLE VARIABLE NAMES ARE MORE GENERAL THAN IN BASIC, CONSISTING OF A LETTER OR A LETTER FOLLOWED BY A LETTER OR A DIGIT EG:-

A   X   AE   ZZ   U7   A9   K0   QR   ARE ALL SIMPLE VARIABLE NAMES.

THE SET OF SIMPLE VARIABLE NAMES BEGINNING WITH Q ARE RESERVED AS SYSTEM VARIABLES AND SHOULD NOT BE USED IN THE NORMAL WAY AS THEY MAY BE CHANGED AS SIDE-EFFECTS OF COMMANDS, OR HAVE SOME SIGNIFICANCE IN INTER-PROGRAM COMMUNICATION. CURRENTLY THE FOLLOWING SYSTEM VARIABLES ARE DEFINED:-

Q0 - LOWER 5 BITS DETERMINE WHICH QUASIC SYSTEM PROGRAM IS RUN WHEN QUASIC IS RECALLED BY A MACHINE-CODE PROGRAM.

Q1 - STATUS WORD SET UP BY A QUASIC PROGRAM ON MACHINE-PROGRAM CALLS & NOT CHANGED BY CALLED PROGRAM.

Q2 - REPLY WORD FROM MACHINE PROGRAM.

Q3 - IDENTIFICATION OF ORIGINAL CALLING PROGRAM.

THE FUNCTIONS OF Q0 THROUGH Q3 ARE DESCRIBED IN MORE DETAIL IN THE SECTION ON INTER-PROGRAM COMMUNICATION.

QE - ERROR STATUS WORD SET UP BY PROGRAM OR DATA TRANSFER ERRORS - SEE SECTION ON ERRORS.

QQ - SUBROUTINE START ADDRESS FOR ERROR TRAP - ZERO IF NO TRAPPING.

QI - COUNTER SET UP BY STRING COMPARISONS IN 'IF', 'UNLESS', AND 'PUT' COMMANDS.

QR - RESIDUAL INFORMATION WORD HOLDING UPPER PART OF MULTIPLICATION OPERATIONS, REMAINDER ON DIVISION, SUBROUTINE CALL ADDRESS AFTER RETURNS, AND DELIMITER INDICATION AFTER TTY INPUT.

NOTE THAT THESE SYSTEM VARIABLES ALWAYS EXIST IN A QUASIC PROGRAM AND ARE NOT AFFECTED BY COMMANDS SUCH AS 'CLEAR' AND 'RUN' WHICH WOULD NORMALLY CLEAR THE SIMPLE-VARIABLE TABLE.

3.2 ARRAYS

ARRAYS ARE STORED AS PROGRAM LINES THROUGH A COMMAND OF THE FORM :-

LINE NUMBER 'ARRAY' SIMPLE VARIABLE NAME MAXIMUM ELEMENT

EG -

```
>34 ARRAY E 44
>
N
```

SETS UP AN ARRAY CALLED E IN LINE 34 OF THE PROGRAM WITH ELEMENTS NUMBERED 0 THROUGH 44.

REFERENCE TO ANY ELEMENT OF THE ARRAY MAY BE MADE BY USING THE TWO-ARGUMENT FUNCTION 'IN' SPECIFYING LINE NUMBER AND ELEMENT NUMBER -

```
IN(NE1,NE2)
```

REFERS TO ELEMENT NE2 OF THE ARRAY IN LINE NUMBER NE1 - EG -

IN(34,6) OR IN(30+4,3*2) BOTH REFER TO THE 6TH ELEMENT OF THE ARRAY STORED IN LINE 34. SINCE THE SIMPLE VARIABLE NAME IN THE ARRAY STATEMENT IS ASSIGNED THE VALUE OF THE LINE NUMBER IN WHICH IT IS STORED BY THE 'RUN' COMMAND, THE SAME ARRAY ELEMENT COULD BE REFERRED TO AS IN(E,6), IE ELEMENTS MAY BE REFERRED TO SYMBOLICALLY. IT WILL BE NOTED THAT ARRAYS ARE AUTOMATICALLY 2-DIMENSIONAL USING THE LINE NUMBER AS ONE DIMENSION IF REQUIRED, AND THAT ARRAYS ARE ALSO DYNAMIC IN THAT ANY PROGRAM LINE CAN BE CREATED OR ERAZED AT RUN-TIME.

3.3 FUNCTIONS

ALL FUNCTIONS IN QUASIC EXCEPT 'IN' TAKE ZERO OR ONE ARGUMENT AND ARE OF THE FORM FN(NE) WHERE FN IS A TWO-LETTER FUNCTION NAME AND NE IS A NUMERIC EXPRESSION.

THE MAIN FUNCTIONS ARE:-

TI() - GIVES THE TIME IN MINUTES FROM MIDNIGHT

DA() - GIVES THE DATE IN TSS8 INTERNAL CODE AS
((YEAR-1964)*12+(MONTH-1))*31+DAY-1

AC(NE) - GIVES THE ACCOUNT NUMBER OF THE JOB NE, WHERE 0<NE<13 IS THE INTERNAL NUMBER OF A JOB (LOGGED-IN USER) ON THE TSS8. THE VALUE IS ZERO IF THERE IS NO ACTIVE JOB NE. IF NE IS ZERO THE VALUE IS THE ACCOUNT NUMBER OF THE CURRENT USER.

KO(NE) - GIVES THE COMMUNICATIONS CHANNEL NUMBER OF THE JOB NE, THE RESULT BEING ZERO IF THERE IS NO ACTIVE JOB NE. IF NE IS ZERO THE VALUE IS THE CHANNEL NUMBER OF THE CURRENT USER.

NL(NE) - GIVES THE LEAST PROGRAM LINE NUMBER GREATER THAN NE, OR ZERO IF THERE IS NO SUCH LINE.

RN() - GIVES A RANDOM NUMBER UNIFORMLY DISTRIBUTED OVER THE RANGE 0 TO 4095. CALLING RN(NE) WITH A NON-ZERO ARGUMENT NE CAUSES THE RANDOM NUMBER GENERATOR TO BE RE-INITIALIZED.

SP() - GIVES THE PROGRAM AREA SPACE LEFT IN WORDS

PT(NE) - IS AN ASSIGNABLE FUNCTION/ARRAY REFERENCING THE VALUES OF POINTERS TO IO DEVICE DATA STRUCTURES. PT(0) GIVES THE CARRIAGE POSITION OF THE TELEPRINTER ACROSS THE PAPER. PT(1) GIVES THE CARRIAGE POSITION DOWN THE PAPER. THESE TWO FUNCTIONS MAY BE USED TO CONTROL PRINTED OUTPUT FORMATTING. PT(2) THROUGH PT(9) ARE FILE POINTERS ASSOCIATED WITH THE DISC AND ARE DESCRIBED IN IN SECTION *9* ON FILE STRUCTURES.

3.4 ASSIGNMENT COMMANDS

VALUES ARE NORMALLY ASSIGNED TO SIMPLE VARIABLES AND ARRAY ELEMENTS BY A COMMAND OF THE FORM -

```
'LET' VAR '=' NUMERIC EXPRESSION
```

OR, MORE GENERALLY, SEVERAL ASSIGNMENTS MAY BE SPECIFIED IN ONE COMMAND -

```
'LET' VAR1 '=' NE1 VAR2 '=' NE2     ETC
```

```
EG. LET A=7
    LET A=2 B1=89*6 E=A^2 IN(C,7)=8 IN(G+3,7^K-7)=1
```

THE ASSIGNMENTS ARE MADE SEQUENTIALLY SO THAT -

```
>LET E=2 X=4/E E=5
```

GIVES E THE FINAL VALUE OF 5 AND X THAT OF 2.

THE IMPLEMENTATION OF QUASIC IS SUCH THAT WHEREVER A NUMBER CAN OCCUR A NUMERIC EXPRESSION MAY REPLACE IT. IN PARTICULAR, THIS ALLOWS TRANSFER OF CONTROL BY 'GOTO' AND 'GOSUB' TO BE MADE THROUGH COMPUTED EXPRESSIONS RATHER JUST TO LINE NUMBERS AS IN BASIC. THIS CAPABILITY GETS OVER ONE OF THE PROBLEMS OF A LINE-NUMBERED SYSTEM IN THAT FORWARD REFERENCES ARE REQUIRED TO ROUTINES WHOSE LINE NUMBERS ARE UNDEFINED AT THE TIME OF PROGRAMMING THE REFERENCE. TO ENABLE FULL ADVANTAGE TO BE TAKEN OF THIS, THERE IS A 'NAME' COMMAND IN QUASIC WHICH CAUSES THE LINE NUMBER OF A PROGRAM LINE COMMENCING WITH A 'NAME' COMMAND TO BE ASSIGNED TO THE SIMPLE VARIABLE SPECIFIED IN THE 'NAME' COMMAND WHEN THE COMMAND 'RUN' IS GIVEN :-

```
>10 GOTO PR
>20 STOP
>30 NAME PR
>40 PRINT 'DONE PR=' PR
>RUN
DONE PR=30

QUASIC V3A

>
```

THE NAME STATEMENT CAN BE FOLLOWED BY OTHERS AND ANY CHARACTERS AFTER THE FIRST TWO IN THE SIMPLE VARIABLE NAME ARE SKIPPED SO THAT IT IS POSSIBLE TO HAVE REASONABLE NAMES :-

```
>10 GOTO END
>20 STOP
>50 NAME END :PRINT 'THE END
>RUN
THE END

QUASIC V3A

>
```

THE 'ARRAY' STATEMENT, AS NOTED PREVIOUSLY, CAUSES AN ASSIGNMENT IN A VERY SIMILAR MANNER TO THE 'NAME' STATEMENT.

4 STRING HANDLING IN QUASIC

THE NORMAL CHARACTER STRING HANDLING EXTENSIONS OF BASIC ARE VERY WEAK AND A FRESH APPROACH IS TAKEN IN QUASIC. SINCE A PROGRAM LINE ITSELF COMPRISES A CHARACTER STRING, A PARTICULAR PSEUDO-COMMAND '$' IS ADDED TO THE LANGUAGE WHICH IS NEVER EXECUTED BUT INDICATES THAT THE REMAINDER OF THE 'PROGRAM' LINE CONTAINS A CHARACTER DATA STRING. THERE ARE NO RESTRICTIONS ON THE CHARACTER SET WHICH MAY BE STORED IN THIS WAY - IT IS THE FULL ISO/ASCII 7-BIT SET. THUS :-

```
>30 $ABCDE++P)**##"$
```

IS A CHARACTER STRING STORED IN LINE 30. IT IS REFERENCED AS A STRING VARIABLE BY THE NAME $30 OR $NE WHERE THE NUMERICAL EXPRESSION NE EVALUATES TO 30. THUS :-

```
>PRINT $30
ABCDE++P)**##"$
>LIST

 30 $ABCDE++P)**##"$

>
```

IN PARTICULAR IT WILL BE NOTED THAT STRING VARIABLES ARE LISTED WITH THE PROGRAM, ENABLING THEIR CONTENTS TO BE SET UP AND EXAMINED AS PART OF NORMAL PROGRAM EDITING. STRING VARIABLES CAN BE ENTERED AT RUN-TIME FROM THE TELETYPE USING THE 'INPUT' STATEMENT, FROM THE DISC BY USING THE 'READ' STATEMENT, AND THEY MAY BE MANIPULATED THROUGH CONTEXTUAL PATTERN-MATCHING USING THE 'PUT' STATEMENT.

4.1 STRING EXPRESSIONS

A STRING EXPRESSION CONSISTS OF A NUMBER OF FIELDS, PERHAPS SEPARATED BY COMMAS, EACH OF WHICH EVALUATES TO A CHARACTER STRING, THE VALUE OF THE EXPRESSION BEING THE CONCATENATION OF THE VALUES OF THE FIELDS. THUS :-

```
>70 $ HI
>80 $GH
>LET N=400
>PRINT 'THIS IS' $70 $80 ' APPROX ' N ' METRES
THIS IS HIGH APPROX  400 METRES
>
```

ILLUSTRATES LITERAL STRINGS, STRING VARIABLES AND NUMERIC STRINGS IN A STRING EXPRESSION.

THE POSSIBLE FIELDS IN A STRING EXPRESSION ARE:-

LITERAL STRING - CHARACTER STRING ENCLOSED IN BALANCED QUOTES, ' OR "
EG '%%' "#" '"' '$4' "9+8" - THE QUOTES MUST BE BALANCED
UNLESS THE LITERAL STRING ENDS A PROGRAM LINE, IN WHICH CASE

THE FINAL ONE MAY BE OMITTED.

STRING VARIABLE - '$'-LINE REFERENCED AS '$' NE - SEE PREVIOUS EXAMPLES.

NUMERIC EXPRESSION - A NUMERIC EXPRESSION IS EVALUATED AND CONVERTED TO A NUMERIC CHARACTER STRING IN THE CURRENT OUTPUT FORMAT. THE FORMAT MAY BE SET BY '@' NE IMMEDIATELY FOLLOWING THE NUMERIC EXPRESSION, WHICH SPECIFIES THAT THERE SHOULD BE AT LEAST NE CHARACTERS OUTPUT, LEADING ZEROS BEING REPLACED BY SPACES EG.

```
>PRINT 123@4
 123
>PRINT 123@5
  123
>PRINT 123@1
123
>PRINT123@
123
>
```

OVERALL FORMAT SPECIFIER - AN '@' SIGN FOLLOWED BY A NUMERIC EXPRESSION WHICH DOES NOT IMMEDIATELY FOLLOW A NUMERIC EXPRESSION SETS UP AN OVERALL OUTPUT FORMAT FOR THE REST OF THE NUMERIC FIELDS IN THE STRING EXPRESSION.

CHARACTER SPECIFICATION - A FIELD OF THE FORM '&C' NE HAS THE ASCII CHARACTER WHOSE 7-BIT BINARY VALUE EVALUATES TO NE AS ITS VALUE. SIMILARLY A FIELD OF THE FORM :-

'&C(' NE1 ',' NE2 ',' NE3 ETC ')'

HAS THE CHARACTER STRING WHOSE 7-BIT VALUES ARE NE1 ETC AS ITS VALUE. THUS &C(10,13) IS CARRIAGE-RETURN LINE-FEED, AND, FOR EXAMPLE :-

```
>PRINT &C48
0
>PRINT &C(192,193,194)
ABC
>
```

PROGRAM LINE - A FIELD OF THE FORM &P NE HAS THE VALUE OF THE PROGRAM LINE NUMBER NE.

NEWLINE - A SEMICOLON HAS THE VALUE CARRIAGE-RETURN LINE-FEED.

COMMA - A COMMA HAS NULL VALUE & MAY BE USED AT WILL TO RESOLVE AMBIGUITY. HOWEVER, IN A 'PRINT' STATEMENT THERE IS AN IMPLIED NEW LINE AT THE END OF A STRING EXPRESSION AND THIS IS SUPPRESSED IF THE FINAL CHARACTER OF THE STRING @ EXPRESSION IS A COMMMA.

4.2 STRING MANIPULATION

THE 'PUT' COMMAND IS UNIQUE TO QUASIC ANF ENABLES CHARACTER

STRINGS TO BE HANDLED AS DATA ITEMS AND CONSTRUCTED THROUGH CONTEXTUAL PATTERN-MATCHING. THE SYNTAX OF 'PUT' IS :-

```
'PUT' SE  [OP  DOLL  MATCH  SE']
```

WHERE SE IS A STRING EXPRESSION, OP IS AN OPERATOR, DOLL IS A '$'-LINE, MATCH IS A SIMPLE STRING (LITERAL STRING OR '$'-LINE) OR A VARIABLE FOR ASSIGNMENT OR A NUMBER-OF-CHARACTERS SPECIFICATION AND SE' IS ANOTHER STRING EXPRESSION. THE FIELDS WITHIN SQUARE BRACKETS MAY BE ITERATED, AND THE FIELDS THEMSELVES ARE TO A LARGE EXTENT OPTIONAL. THE PUT STATEMENT HAS THE EFFECT OF TAKING THE STRING SE, IMMEDIATELY AFTER 'PUT' AND SPLITTING IT UP ACCORDING TO THE OPERATOR AND MATCH, THE PART OF THE STRING BEFORE THE MATCH GOING INTO THE '$'-LINE DOLL, FOLLOWED BY THE REPLACEMENT STRING SE'.

THE OPERATORS IN A 'PUT' STATEMENT ARE:-

.> - CLOSE ANY OUTPUT STRINGS PREVIOUSLY OPENED AND OPEN A NEW OUTPUT STRING SPECIFIED BY THE FOLLOWING '$'-LINE. ANY MATCHES TO BE IMBEDDED.

= - ANY MATCHES TO BE ANCHORED.

< - LEAVE OPEN CURRENT OUTPUT STRING - ANY MATCHES TO BE IMBEDDED.

<> - TEMPORARILY CLOSE THE OUTPUT STRING UNTIL NEXT OPERATOR ANY MATCHES TO BE IMBEDDED.

THE MATCH IS EITHER A LITERAL STRING OR A '$'-LINE OR A SET OF VARIABLES TO BE ASSIGNED NUMERIC VALUES, OR A CHARACTER NUMBER SPECIFICATION OF THE FORM &G NE, MEANING GET THE NEXT NE CHARACTERS. CHARACTERS PASSED OVER IN AN IMBEDDED MATCH, IN FINDING A NUMERIC FIELD TO ASSIGN, OR GOT AS THE NEXT NE CHARACTERS, ARE SENT TO THE OUTPUT STRING IF ONE IS OPEN. WHEN A MATCH IS FOUND ANY IMMEDIATELY FOLLOWING STRING EXPRESSION IS ALSO SENT TO THE OUTPUT STRING.

THE 'PUT' COMMAND HAS A WIDE VARIETY OF APPLICATIONS AND THESE ARE BEST SEEN BY EXAMPLE :-

CONSTRUCTION OF STRING VARIABLES -

```
>PUT 'HELLO' > $1
>PRINT $1
HELLO
>25 $XYZ
>30 $ +++
>PUT $25 $30 > $2
>PRINT $2
XYZ +++
>PUT &C(192,193,194) > $3
>PRINT $3
ABC
>PUT 12+23 @2 > $4
>PRINT $4
```

```
35
>PUT @3, 7+2 5 3, -8+12 4 @7 67@, 89@ > $5
>PRINT $5
  9  5  3  4        76789
>PUT 'UVW' 7*5@ $25 &C 48 > $6
>PRINT $6
UVW35XYZ0

>
```

THESE EXAMPLES HAVE ILLUSTRATED THE USE OF 'PUT' TO ASSIGN AN ARBITRARY STRING EXPRESSION TO A STRING VARIABLE. IT WILL BE NOTED THAT THE EVALUATED STRING EXPRESSION APPEARS IN THE '$'-LINE EXACTLY AS IT WOULD BE PRINTED ON THE TERMINAL THROUGH A 'PRINT' COMMAND.

THE EXAMPLES ABOVE USE ONLY THE '>' OPERATOR TO OPEN A SINGLE STRING ON THE RIGHT HAND SIDE AND DO NOT INVOLVE MATCHES OR REPLACEMENTS. IN THE FOLLOWING EXAMPLES THE USE OF FURTHER FACILITIES ON THE RIGHT HAND SIDE TO SPLIT UP THE GENERATED STRING IS ILLUSTRATED:-

LOOK FOR IMBEDDED PATTERN -

```
>PUT '123456789' > $1 '5'   [LITERAL MATCH
>PRINT $1
1234
>50 $EFG
>PUT 'ABCDEFGHIJKL' > $2 $50    ['$' MATCH   - PUT STRING IN $2 UP TO
>PRINT $2                       [WHERE YOU FIND $50
ABCD
>60 $JK
>70 $HIJKL
>PUT $70 > $3 $60  [PUT $70 INTO $3 UPTO $60
>PRINT $3
HI
>PUT 'ABCDEFGHIJKL' > $4 'EFG' > $5  [MULTIPLE ASSIGNMENT - PUT STRING
>PRINT $4; $5                        [INTO $4 UP TO 'EFG' & PUT REST
HIJKL                                [IN $5.
HI
>PUT 'ABCDEFGHIJKL' > $6 $50 > $7 'J' > $8   [MULTIPLE MATCH
>PRINT $6; $7; $8
ABCD
HI
KL
>PUT 'ABCDEFGHIJKL' > $9 '123'  [UNSATISFIED MATCH
>PRINT $9
ABCDEFGHIJKL

>
```

NOTE THAT, IN THE LAST EXAMPLE, AS THE MATCH IS NOT FOUND, ALL THE STRING IS TRANSFERRED. THERE IS ALSO AN IMPLIED CONDITIONAL IN A PUT STATEMENT IN THAT IF ALL THE MATCHES ARE FOUND (IE THE MACRO TEMPLATE IS SATISFIED) THEN EXECUTION OF THE REST OF THE PROGRAM LINE CONTINUES, BUT WHEN A MATCH FAILS CONTROL IS TRANSFERRED TO

THE NEXT PROGRAM LINE, EG -

```
>PUT 'ABCDE' > $1 'CD' :PRINT 'OK ' $1
OK AB
>PUT 'ABCDE' > $2 'JK' :PRINT 'OK'
>LIST

  1 $AB
  2 $ABCDE

>
```

THE IMPLICIT CONDITIONAL AUTOMATICALLY TESTS THAT THE PATTERN-MATCHING MACRO HAS BEEN SATISFIED.

IMMEDIATELY AFTER A MATCH HAS BEEN SPECIFIED A REPLACEMENT STRING EXPRESSION MAY BE SPECIFIED WHICH WILL BE TRANSFERRED INTO THE OUTPUT STRING WHEN THE MATCH IS FOUND - EG -

```
>PUT 'ABCDEF' > $1 'CD' '123' :PRINT $1
AB123EF
>PUT 'ABCDEFGHIJK' > $1 'CD' 2 '++++' > $2 :PRINT $1; $2
AB   2++++
EFGHIJK
>
```

THE '>' OPERATOR CLOSES THE OUTPUT STRING BEFORE FURTHER MATCHES MAY BE MADE - THE '<' OPERATOR LEAVES THE CURRENT OUTPUT STRING OPEN & GOES ON TO A NEW MATCH - EG -

```
>PUT 'ABCDEFGHIJKL' > $1 'CD' '++' < 'IJ' '--'
>PRINT $1
AB++EFGH--KL
>
```

THE FOLLOWING COMMAND REPLACES ALL MULTIPLE SPACES IN $X BY SINGLE SPACES -

```
>100 PUT $X > $X '  ' ' ' < :
>
```

NOTE HOW THE TERMINAL COLON CAUSES THE LOOP TO BE REPEATED UNTIL THERE ARE NO FURTHER MULTIPLE SPACES.

SOMETIMES IT IS NECESSARY TO CLOSE THE OUTPUT ONLY TEMPORARILY & THE '<>' OPERATOR IS USED - EG -

```
>PUT 'ABCDEFG' > $1 'B' '+' <> 'F' '-' <
>PRINT $1
A+-G
>
```

OTHER 'MATCHES' THAN THOSE AGAINST CHARACTER STRINGS MAY BE USED. A NUMERIC FIELD (DEFINED AS SPACES FOLLOWED BY AT LEAST ONE DIGIT

& TERMINATED BY A NON-DIGIT) MAY BE PICKED OUT AND $ASSIGNED TO A SIMPLE VARIABLE OR ARRAY ELEMENT BY PLACING THE VARIABLE NAME IN THE MATCH FIELD - EG -

```
>PUT 'ABCDEF 45 GHIJ' > $1 A >42
>PRINT A
   45
>PRINT $1; $2
ABCDEF
 GHIJ
>
```

THE SEARCH FOR THE NUMBER IS IMBEDDED IF THE PRECEEDING OPERATOR WOULD CAUSE A STRING MATCH TO BE IMBEDDED (>,<,<>) & ANCHORED OTHERWISE.

MULTIPLE VARIABLE NAMES SEPARATED BY SPACES MAY BE USED TO PICK OUT MULTIPLE NUMERIC FIELDS (A NULL STRING MATCH IS REQUIRED TO STOP THE PROCESSOR LOOKING FOR FURTHER NUMERIC ASSIGNMENTS) - EG -

```
>PUT 1 2 3 > A B C
>PRINT A B C
    1    2    3
>PUT XYZ 1 2 3 > $1 A B C '' '+++'
>PRINT $1
XYZ+++
>
```

THE 'MATCH' FIELD, '&G' NE , MAY BE USED TO GET THE NEXT NE CHARACTERS FROM THE INPUT STREAM INTO THE OUTPUT STREAM - EG -

```
>PUT 'ABCDE' > $1 &G2 > $2 :PRINT $1; $2
AB
CDE
>
```

THE 'MATCH' FIELD, &A NE , MAY BE USED TO PUT THE NEXT N CHARACTERS FROM THE INPUT STREAM, AS ASCII VALUES, INTO THE N ELEMENTS OF THE ARRAY IN LINE NE. IF THERE ARE LESS THAN N CHARACTERS, THOSE WHICH ARE THERE ARE PUT IN FOLLOWED BY A ZERO ELEMENT - EG -

```
>10 ARRAY A 2
>PUT 'AB' > &G 10
>PRINT IN(10,0) IN(10,1) IN(10,2)
   65   66     0
>
```

5 TELEPRINTER INPUT/OUTPUT

THE TELEPRINTER (TTY) IS REGARDED AS A STRING-HANDLING DEVICE WHICH PRODUCES A PRINTOUT OF A STRING ON ITS PRINTER, OR GENERATES AN INPUT STRING THROUGH ITS KEYBOARD. ALL INPUT AND OUTPUT TO THE TELEPRINTER IS 7-BIT ASCII AND ON OUTPUT EVEN PARITY IS GENERATED. NUMERIC POINTERS KEEP TRACK OF THE CARRIAGE POSITION FOR TABULATION AND PAGINATION ON OUTPUT, AND CERTAIN KEYS HAVE EDITING AND CONTROL FUNCTIONS ON INPUT. THE INPUT COMMAND GIVES FACILITIES FOR PRINTING A STRING TO REQUEST INPUT, UNDUPLEXING, TIMING OUT, AND USING THE FULL FACILITIES OF THE 'PUT' COMMAND FOR OPERATING ON THE INPUT.

5.1 PRINT A STRING

THE COMMAND -

```
'PRINT'  SE
```

CAUSES THE STRING EXPRESSION SE TO BE EVALUATED AND PRINTED OUT ON THE TELEPRINTER TERMINATED BY A NEWLINE (IE CARRIAGE-RETURN/LINE-FEED - THIS IS INHIBITED IF THE STRING EXPRESSION TERMINATES IN A COMMA), FOR EXAMPLE -

```
>PRINT 2*4 'ABCDE'
   8ABCDE
>PRINT 5@ 'A' ; 'B' , :PRINT 'X'
5A
BX
>
```

NOTE HOW THE SEMICOLON CAUSES A NEWLINE AND THE TERMINAL COMMA PREVENTS THE NEWLINE NORMALLY OUTPUT AFTER THE STRING.

THE 7-BIT CHARACTER WHOSE VALUE IS ZERO CAUSES A NEWLINE WHEN OUTPUT. EACH TIME A CHARACTER OTHER THAN CARRIAGE-RETURN, LINE-FEED OR NEWLINE IS OUTPUT, THE VARIABLE PT(0) IS INCREMENTED BY ONE; IT IS RESET TO ZERO EACH TIME A CARRIAGE-RETURN OR NEWLINE IS OUTPUT. EACH TIME A LINE-FEED OR NEW LINE IS OUTPUT THE VARIABLE PT(1) IS INCREMENTED BY ONE.

5.2 TABULATION

ALTHOUGH THE VARIABLE PT(0) MAY BE USED TO CONTROL HORIZONTAL FORMATTING, A MORE DIRECT COMMAND IS MADE AVAILABLE FOR CONVENIENCE. THE COMMAND -

```
'TAB'  NE  SE
```

CAUSES THE FIRST CHARACTER OF THE STRING EXPRESSION SE TO BE PRINTED OUT UNTIL PT(0) BECOMES EQUAL TO THE VALUE OF THE NUMSRICAL EXPRESSION NE. IF THE VALUE OF PT(0) IS ALREADY GREATER THAN OR EQUAL TO NE THEN

THE COMMAND HAS NO EFFECT. IF THE SE IS NULL (OR ABSENT) IT IS TAKEN TO BE A SPACE. EG -

```
>TAB 7 '+'
+++++++>TAB 10 'ABCD' :PRINT 1
AAAAAAAAAA     1
>TAB 5 '*' :PRINT PT(0)
*****     5
>TAB 6 :TAB 9 '%' :TAB 7 '-' :PRINT PT(0)
      %%%    9
>11 $=+<<
>TAB 7 $11
=======>
```

NOTE THAT THE 'TAB' COMMAND DOES NOT GIVE A TERMINAL NEWLINE.

5.3 INPUT A STRING

THE 'INPUT' COMMAND IS DESIGNED TO ALLOW FULL CONTROL OVER QUESTION AND ANSWER SEQUENCES AT THE TELEPRINTER. THERE ARE MANY OPTIONS CONTROLLING THE INPUT OF THE STRING, AND ALL THE PATTERN-MATCHING AND ASSIGNMENT FACILITIES OF THE 'PUT' COMMAND ARE AVAILABLE TO OPERATE ON THE STRING. THE 'INPUT' COMMAND SPECIFIES A STRING TO BE OUTPUT TO REQUEST KEYBOARD INPUT, SETS UP THE TTY CHARACTER-ECHOING, SPECIFIES THE MAXIMUM TIME TO WAIT FOR INPUT & SPECIFIES WHAT IS TO BE DONE WITH THE INPUT.

THE INPUT FROM THE TTY KEYBOARD IS DEFINED TO TERMINATE WHEN A 'TERMINATION-CHARACTER' IS RECEIVED OR MORE THAN A SPECIFIED TIME HAS ELAPSED SINCE THE LAST KEYBOARD CHARACTER WAS RECEIVED. THE SYSTEM VARIABLE QR IS SET TO INDICATE THE TERMINATION CONDITION. THE TERMINATION CHARACTERS, THEIR OCTAL VALUES, AND THE RETURNED VALUES OF QR ARE -

TERM CHAR.	VALUE	QR
CARRIAGE-RETURN	15	0
ESCAPE	33	1
ETX (CONTROL C)	3	3
TIME OUT	-	2

NOTE THAT THE CARRIAGE-RETURN TERMINATION CHARACTER IS ECHOED BY NEWLINE, WHEREAS THE OTHERS ARE NOT. THIS PROVIDES A CONVENIENT MEANS FOR FORMATTING INPUT (WHICH IS ALSO EFFECTIVE AT PROGRAM-ENTRY TIME SINCE THE SAME THREE TERMINATION CHARACTERS ARE USED).

THE SYNTAX OF THE 'INPUT' COMMAND IS -

```
'INPUT' [ '=' NE1 ] [ '<' ] [ '>' NE2 ] [DOLL MATCH SE' OP]
```

THE LAST OPTION BEING ITERATED. THE OPTIONAL COMMAND FIELDS ARE -

'=' NE1 - CAUSES THE STRING IN $-LINE NE1 TO BE PRINTED OUT WHEN THE INPUT COMMAND IS EXECUTED AND WHEN THE 'EM' KEY (CONTROL Y) IS DEPRESSED DURING INPUT. IF THIS FIELD IS OMITTED A COLON IS PRINTED TO REQUEST INPUT FROM THE TTY.

'<' - TURNS OFF THE CHARACTER-ECHOING DURING INPUT - THE MESSAGES TYPED IN ARE NOT PRINTED BY THE TELEPRINTER AND HENCE THE SECURITY OF MESSAGES CAN BE MAINTAINED AS REQUIRED.

'>' NE2 - CAUSES THE INPUT COMMAND TO TERMINATE IF NO CHARACTER IS RECEIVED FOR NE2 SECONDS EVEN THOUGH AN INPUT TERMINATION CHARACTER HAS NOT BEEN RECEIVED. NOTE THAT THIS FEATURE IS SUBJECT TO SOME PECULIARITIES OF THE TSS8 TIME-SHARING MONITOR. THE TIME NE2 IS MEASURED FROM WHEN THE OUTPUT STRING GENERATED BY THE 'INPUT' COMMAND WAS PLACED IN THE OUTPUT CHARACTER BUFFER - THIS MAY BE SOME SECONDS BEFORE THE STRING IS ACTUALLY PRINTED. ALSO THE TIMER IS RE-STARTED EACH TIME QUASIC FINDS CHARACTERS IN THE INPUT BUFFER - AGAIN THESE MAY BE FOUND SOME SECONDS AFTER THEY HAVE BEEN TYPED. HENCE THE TIME OUT FACILITY IS AN APPROXIMATE ONE USEFUL IN TEACHING SEQUENCES AND VITAL IN COMMUNICATION WITH AUTOMATIC TERMINALS TO AVOID 'DEADLY-EMBRACE' EFFECTS.

ONCE CHARACTERS HAVE BEEN INPUT UP TO A TERMINATOR OR TO A TIME OUT, THEY MAY BE REGARDED AS THE STRING SE ON THE LEFT HAND SIDE OF A 'PUT' COMMAND FOLLOWED BY THE PUT OPERATOR '>' (SEE SECTION ON 'STRING MANIPULATION') & THE REMAINING SYNTAX OF THE 'INPUT' COMMAND IS IDENTICAL TO THAT OF THE 'PUT' COMMAND - IE THE INPUT STRING MAY BE ASSIGNED TO ONE OR MORE $-LINES, MATCHED AGAINST PATTERNS, SET INTO ARRAYS, NUMERICAL PARTS ASSIGNED TO VARIABLES, AND SO ON.

IN THE INPUT MODE SEVERAL TELEPRINTER KEYS HAVE SPRCIAL FUNCTIONS -

BACKSLASH - SERVES TO ERAZE THE LAST CHARACTER INPUT & MAY BE USED REPEATEDLY TO ERAZE SEVERAL CHARACTERS.

EM (CONTROL Y) - SERVES TO ERAZE THE TEXT SO FAR INPUT - IT ECHOES '@' NEWLINE & CAUSES THE INDICATOR STRING TO BE PRINTED AGAIN.

CAN (CONTROL X) - SERVES AS A NON-TERMINATING NEWLINE - IT IS STORED INTERNALLY AS THE 7-BIT CHARACTER WHOSE VALUE IS ZERO.

DC3 (CONTROL S) - CAUSES ERROR S IN THE QUASIC PROGRAM & MAY BE USED TO PROVIDE USER INTERUPT CAPABILITY.

SI (CONTROL O) - CAUSES A RETURN TO THE CURRENT SYSTEM MASTER PROGRAM DETERMINED BY THE VALUE OF QS, EG TO QEX.

SO (CONTROL N) - CAUSES A DROP TO MONITOR LEVEL IF THE SYSTEM SWITCH REGISTER HAS BEEN SET NEGATIVE BY A 'QSW' COMMAND, OR QUASIC TO BE RE-STARTED AS ON LOGIN IF THE SWITCH REGISTER IS POSITIVE.

NOTE THAT DC3, SI AND SO ARE EFFECTIVE AT ALL TIMES, NOT ONLY DURING INPUT, AND PROVIDE EFFECTIVE PROGRAM, QUASIC, AND SYSTEM INTERUPTION

FACILITIES RESPECTIVELY.

EXAMPLES IN THE USE OF 'INPUT' :-

```
>1 $HELLO-
>INPUT =1 $0 :PRINT $0
HELLO-HI THEIR@        [THE EM KEY WAS DEPRESSED CAUSING '@'
HELLO-HI THERE
HI THERE
>INPUT =1 < $0 :PRINT $0
HELLO-            [THE INPUT IS NOT ECHOED
HI THERE
*— 10 INPUT =1 :PRINT QR    [NOTE INPUT NEED NOT BE KEPT
>DO 10
HELLO-HI   [TERMINATED BY CARRIAGE RETURN
     0
>DO 10
HELLO-HI    [TERMINATED BY ESC
     1
>DO 10
HELLO-    [WAIT 10 SECONDS
     2
>INPUT $1 'A' > $2 'B' :PRINT $1 ; $2
:123A567B
123
567
>INPUT X Y :PRINT X Y
:45 36
   45   36
>10 INPUT =1 'HI' :PRINT 'GOOD'
>DO 10
123WWW    [WWW TYPED IN
>DO 10
123HI    [HI TYPED IN
GOOD
>
```

IN THIS LAST EXAMPLE THE CONDITIONAL IMPLIED IN THE PATTERN-MATCHING OF THE 'PUT' SYNTAX IS USED TO CHECK THE INPUT WITHOUT ANY ATTEMPT TO STORE IT.

6 CONDITIONALS

QUASIC PROVIDES FACILTIES FOR COMPARING BOTH NUMERICAL EXPRESSIONS AND CHARACTER STRINGS & CONTINUING EXECUTION OF THE CURRENT PROGRAM LINE, OR NOT CONTINUING ITS EXECUTION, ACCORDING TO THE RESULTS OF THE COMPARISON. IN THE STRING COMPARISONS THE SYSTEM VARIABLE QI IS USED AS A COUNTER TO PROVIDE SYMBOL-TABLE FACILITIES.

FOR NUMERIC COMPARISONS THE TWO COMMANDS ARE -

```
'IF' NE1 OP NE2    REST OF LINE

'UNLESS' NE1 OP NE2   REST OF LINE
```

WHERE THE OPERATOR OP IS AN ARITHMETIC RELATION OF THE FORM -

=	EQUALS
>	PROPERLY GREATER THAN
<	PROPERLY LESS THAN
>=	GREATER THAN OR EQUALS
<=	LESS THAN OR EQUALS
<>	NOT EQUAL TO

(IN THE CASE OF THE COMBINED OPERATORS THE ORDER IS IRRELEVANT).

THE RELATIONAL EXPRESSION IS EVALUATED AS IF THE NUMERICAL EXPRESSIONS NE1 AND NE2 WERE BOTH POSITIVE INTEGERS MODULO 4096 (IE SO THAT -1 > 1 IS TRUE SINCE -1 EVALUATES TO 4095). AS THE NAMES IMPLY, IN THE 'IF' COMMAND THE REST OF THE LINE IS EXECUTED ONLY IF THE RELATION IS TRUE, WHEREAS IN THE 'UNLESS' COMMAND IT IS EXECUTED ONLY IF THE RELATION IS FALSE.

NOTE THAT WHEREAS IN BASIC ONLY A TRANSFER OF CONTROL CAN FOLLOW A CONDITIONAL, IN QUASIC ANY COMMAND MAY FOLLOW A CONDITIONAL, SEPARATED FROM IT BY A COLON IN THE NORMAL WAY - EG -

```
>IF 6 > 2 :PRINT 'OK'
OK
>UNLESS 5 = 5 :PRINT 'NOT EQUAL'
>UNLESS 2>3 :PRINT 'TRUE'
TRUE
>
```

THERE ARE THREE POSSIBLE OPERATORS FOR RELATIONS BETWEEN STRINGS, AND TWO POSSIBLE COMMANDS AS BEFORE -

```
'IF' $-LINE OP STRING    REST OF LINE

'UNLESS' $-LINE OP STRING  REST OF LINE
```

NOTE THAT GENERAL STRING-EXPRESSIONS ARE NOT ALLOWED AND THE STRING TO THE LEFT OF THE RELATIONAL OPERATOR MUST BE A $-LINE, WHILST THAT

TO THE RIGHT MAY BE A $-LINE OR A LITERAL (QUOTED) STRING. THE POSSIBLE RELATIONAL OPERATORS ARE -

= EXACT IDENTITY BETWEEN THE TWO STRINGS

> THE RIGHT HAND STRING IS AN INITIAL SUBSTRING OF THE LEFT HAND STRING - IE THE LEFT HAND STRING COMMENCES WITH THE RIGHT HAND STRING - NOTE THIS IS SATISFIED BY EQUALITY.

< THE RIGHT HAND STRING IS IMBEDDED IN THE LEFT HAND STRING - IE THE LEFT HAND STRING CONTAINS THE WHOLE OF THE RIGHT HAND STRING SOMEWHERE WITHIN IT.

THE DIFFERENCE BETWEEN 'IF' AND 'UNLESS' IS THE SAME AS THAT FOR ARITHMETIC COMPARISONS.

ONE USEFUL WAY OF LOOKING AT THE POSSIBLE STRING RELATIONS IS THAT IF -

STRING1 > STRING2 THEN STRING1 = STRING2 STRING3 (CONCATENATION)

AND IF -

STRING1 < STRING2 THEN STRING1 = STRING3 STRING2 STRING4

WHERE EITHER OR BOTH STRING3 AND STRING4 MAY BE NULL.

TO ENABLE SIMPLE SYMBOL TABLE FACILITIES TO BE IMPLEMENTED, THE SYSTEM VARIABLE QI IS SET AS A SIDE EFFECT OF AN IMBEDDED MATCH OPERATION. IT IS INITIALLY SET TO ZERO AND THEN INCREMENTED EACH TIME THE FIRST CHARAC\ER OF THE RIGHT HAND STRING FINDS A MATCH IN THE LEFT HAND STRING (UP TO THE POINT WHERE A TOTAL MATCH, IF ANY, IS FOUND). THUS QI MAY BE USED TO RETURN A COUNT LOCATING THE POSITION OF ONE STRING WITHIN ANOTHER - SEE LATER EXAMPLES.

EXAMPLES OF STRING MATCHES -

```
>1 $ABC
>2 $ABC
>3 $ABCDEF
>4 $CD
>5 $!A !B !C !D !E !F !G !
>IF  1$ = 2$ :PRINT 'OK
OK
>IF  1$ = 'ABC' :PRINT 'OK
OK
>IF  1$ = 'DEF' :PRINT 'OK
>IF  3$ > 1$ :PRINT 'OK
OK
>IF  3$ > 'A' :PRINT 'OK
OK
>IF  3$ > 'B' :PRINT 'OK
>IF  3$ < 4$ :PRINT 'OK
OK
>IF  3$ < 1$ :PRINT 'OK
```

```
OK
>IF 3$ < 'E' :PRINT 'OK
OK
>IF 5$ < '!B!' :PRINT QI
    2
>IF 5$ < '!G!' :PRINT QI
    7
>
```

NOTE HOW IN THE LAST TWO EXAMPLES THE SETTING OF QI IS USED TO INDICATE THE POSITION OF THE LETTER ENCLOSED IN '!'S ON THE RIGHT HAND SIDE IN THE STRING 'TABLE' IN $5. THE FOLLOWING SIMPLE PROGRAM SHOWS HOW THIS FACILITY MAY BE USED MORE GENERALLY -

```
>1 $:PIG:COW:GOOSE:FOX:ELEPHANT:BEAR:
2 $NAME*
>100 INPUT =2 $ :PUT ':' $ ':' > $ :IF $1 < $ : PRINT QI :
>110 PRINT 'ANIMAL NOT KNOWN' :GOTO 100
>RUN

NAME*PIG
    1
NAME*FOX
    4
NAME*BEAR
    6
NAME*HORSE
ANIMAL NOT KNOWN
NAME*ELEPHANT
    5
```

NOTE HOW IN THIS EXAMPLE THE INPUT STRING IS 'BRACKETED' WITH COLONS TO UTILIZE THE QI COUNTING FACILITY OF THE IMBEDDED MATCH, AND HOW THE NON-CONTINUED EXECUTION OF THE LINE IF NO MATCH IS FOUND IS USED TO TEST FOR THE EXISTENCE OF THE INPUT STRING IN THE SYMBOL TABLE. CLEARLY THE TABLE ITSELF NEED NOT BE PRE-STORED BUT

MAY BE GENERATED FROM THE INPUTS THEMSELVES. NORMALLY THE VALUE OF QI WILL NOT BE PRINTED OUT BUT WILL BE USED, FOR EXAMPLE, TO GO TO ROUTINES DEPENDENT ON THE INPUT STRING - EG BY A COMPUTED GOTO -

```
GOTO 500 + 10*QI
```

7 TRANSFER OF CONTROL

QUASIC PROVIDES 6 COMMANDS FOR UNCONDITIONAL, TEMPORARY, OR SUBROUTINE TRANSFER OF CONTROL FROM ONE PART OF A PROGRAM TO ANOTHER, OR FOR TRANSFER BETWEEN KEYBOARD-EDIT AND EXECUTION MODES. ADDITIONAL COMMANDS ARE PROVIDED FOR TRANSFERS BETWEEN QUASIC PROGRAM OVERLAYS OR BETWEEN QUASIC AND MACHINE CODE PROGRAMS (SEE SECTION ON 'CALL' AND 'ENTER' COMMANDS IN SECTIONS *9.5* & *9.6*).

7.1 UNCONDITIONAL TRANSFERS

UNCONDITIONAL TRANSFER OF CONTROL IS EFFECTED BY THE COMMAND -

'GOTO' NE

WHERE NE EVALUATES TO THE LINE NUMBER OF AN EXISTING PROGRAM LINE (ERROR 4 RESULTS IF LINE DOES NOT EXIST). NOTE THAT, UNLIKE BASIC WHERE THE ARGUMENT OF 'GOTO' CAN ONLY BE A NUMBER, IN QUASIC THE ARGUMENT OF 'GOTO' IS AN ARBITRARY EXPRESSION - HENCE COMPUTED TRANSFERS OF CONTROL MAY BE USED AND, BY MEANS OF THE 'NAME' COMMAND (SECTION *3.4*), SYMBOLIC TRANSFERS MAY BE EFFECTED - EG -

```
>100 GOTO A1
>110 STOP
>500 NAME A1 :PRINT 'OK'
>RUN
OK

QUASIC V3A

>100 GOTO FINISH
>110 STOP
>500 NAME FINISH :PRINT 'END
>RUN
END

QUASIC V3A

>
```

NOTE HOW IN THE LAST EXAMPLE ADVANTAGE IS TAKEN OF THE TRUNCATION OF NAMES TO 2 CHARACTERS IN THE 'GOTO' & 'NAME' COMMANDS, ALLOWING MORE MEANINGFUL NAMES TO BE USED FOR CALLED ROUTINES.

7.2 TEMPORARY TRANSFERS

TEMPORARY TRANSFER OF CONTROL FOR ONE LINE ONLY IS EFFECTED BY THE COMMAND -

'DO' NE

WHICH CAUSES A SIMILAR EFFECT TO 'GOTO' NE EXCEPT THAT AFTER EXECUTION OF THE LINE WHOSE NUMBER IS NE CONTROL REVERYTS TO THE LINE FOLLOWING THE 'DO' COMMAND. EG -

```
>100 PRINT 'HI
>110 PRINT 'HO
>200 DO 100
>RUN 200
HI

QUASIC V3A

>
```

IF THE 'DO' HAD BEEN A 'GOTO' THEN HI HO WOULD HAVE BEEN PRINTED REPETETIVELY.

CONTROL DOES NOT RETURN TO THE LINE AFTER THE 'DO' COMMAND IF THE LINE CAUSED TO BE EXECUTED ITSELF CAUSES AN UNCONDITIONAL TRANSFER OF CONTROL - EG -

```
>100 DO 200
>110 PRINT 'HI
>120 STOP
>200 GOTO 300
>300 PRINT 'HO
>RUN
HO
QUASIC V3A

>
```

'DO' COMMANDS MAY BE CASCADED ALLOWING MULTIPLE ENTRY POINTS TO AN EXECUTED COMMAND STRING - EG -

```
>100 PRINT 'HI' :DO 110
>110 PRINT 'HO
>DO 100
HI
HO
>DO 110
HO
>
```

ANY FURTHER COMMANDS FOLLOWING A 'DO' COMMAND IN THE SAME LINE ARE COMPLETELY IGNORED. IF THE LINES EXECUTED SHOULD THEMSELVES CREATE NEW PROGRAM LINES IMMEDIATELY AFTER THE 'DO' COMMAND THEN CONTROL REVERTS TO THESE (IE THE RETURN POINT IS COMPUTED POST-EXECUTION).

7.3 SUBROUTINE TRANSFERS

SUBROUTINE TRANSFER OF CONTROL IS EFFECTED BY THE TWO COMMANDS -

'GOSUB' NE1

'RETURN' NE2

'GOSUB' IS EXACTLY EQUIVALENT IN ITS EFFECT TO 'GOTO' EXCEPT THAT THE LINE NUMBER OF THE LINE CONTAINING THE 'GOSUB' IS PLACED ON A SYSTEM STACK - IE THE CALLING LINE IS REMEMBERED.

'RETURN' IS NORMALLY USED TO TRANSFER CONTROL BACK TO THE CALLING ROUTINE AFTER A 'GOSUB'. ITS EFFECT IS SLIGHTLY MORE COMPLEX IN THAT UNSTACKING WITHOUT RETURN TO THE CALLING ROUTINE HAS TO BE ALLOWED FOR. THE FIRST EFFECT OF A RETURN INSTRUCTION IS TO UNSTACK THE LAST LINE NUMBER PLACED ON THE SYSTEM STACK AND PLACE IT IN THE SYSTEM VARIABLE QR. IF NO FURTHER COMMANDS FOLLOW THE 'RETURN' IN THE SAME PROGRAM LINE THEN A -

'GOTO' QR + NE2

IS EXECUTED EXCEPT THAT IF NO LINE WITH NUMBER QR+NE2 EXISTS THEN A 'GOTO' IS EXECUTED TO THE LINE WITH NEXT HIGHEST NUMBER; ALSO IF NE2 IS ZERO OR NOT PRESENT (VALUE EFFECTIVELY 0) ITS VALUE IS TAKEN TO BE 1 - SO THAT 'RETURN' ALONE DEFAULTS TO THE NORMAL CONVENTION. IF FURTHER COMMANDS DO FOLLOW THE 'RETURN' THEN NO 'GOTO' IS EXECUTED.

THUS THE SUBROUTINE FACILITIES ALLOW FOR MULTIPLE ENTRY-POINT ROUTINES (THROUGH THE STACK) AND FOR MULTIPLE RETURN POINTS EITHER RELATIVE TO THE CALLING ROUTINE OR TO ABSOLUTE ADDRESSES - EG -

```
>100 GOSUB ABCD
>110 PRINT 'FIRST RETURN'
>120 PRINT 'SECOND RETURN'
>130 GOTO 100
>200 NAME ABCD :INPUT $ :IF $='A' :RETURN
>210 IF $ = 'B' :RETURN 15
>220 RETURN :GOTO 500
>500 PRINT 'ABSOLUTE RETURN' :GOTO 100
>RUN
:A
FIRST RETURN
SECOND RETURN
:B
SECOND RETURN
:C
ABSOLUTE RETURN
:
```

NOTE THAT 'GOSUBS' AND 'RETURNS' CAN BE EXECUTED USING THE 'DO' COMMAND, UNDER WHICH CIRCUMSTANCES THE LINE NUMBER STACKED WILL BE THAT OF THE LINE CONTAINING THE 'DO' COMMAND.

7.4 TRANSFERS BETWEEN EDIT & EXECUTION MODES

TO CAUSE EXECUTION OF A QUASIC PROGRAM CURRENTLY BEING EDITED

FROM THE TELEPRINTER KEYBOARD THE COMMAND -

```
'RUN' NE
```

IS USED. ITS FINAL EFFECT IS TO CAUSE A 'GOTO' TO THE FIRST PROGRAM LINE WHOSE NUMBER IS GREATER THAN OR EQUAL TO NE, HOWEVER 'RUN' ALSO HAS VARIOUS INITIALIZATION EFFECTS -

1 THE SUBROUTINE STACK IS RESET TO BE EMPTY

2 ALL SIMPLE VARIABLES EXCEPT THE SYSTEM VARIABLES ARE DESTROYED

3 ALL VARIABLES DECLARED IN 'ARRAY' AND 'NAME' COMMANDS ARE SET UP AND ASSIGNED THE VALUE OF THE LINE NUMBER OF THEIR PROGRAM LINE.

BECAUSE 'RUN' CLEARS THE SIMPLE VARIABLES IT IS ALSO USEFUL AS A STORED COMMAND WHEN ENTERING A NEW OVERLAY. NOTE THAT 'GOTO' MUST BE USED TO RESTART A PROGRAM IF ITS SIMPLE VARIABLES ARE NOT TO BE DESTROYED.

CONTROL MAY BE TRANSFERRED BACK TO KEYBOARD EDIT MODE FROM A QUASIC PROGRAM BY THE COMMAND -

```
'STOP'
```

OR BY THERE BEING NO FURTHER PROGRAM LINES TO EXECUTE (NOTE THAT THE 'END' COMMAND IS UNNECESSARY AND NOT RECOGNIZED IN QUASIC).

CONTROL MAY BE FORCED BACK TO KEYBOARD EDIT MODE BY STRIKING THE DC3 (CONTROL S) KEY ON THE TELEPRINTER. THIS CAUSES AN ERROR C WHEN QUASIC GOES TO EXECUTE THE NEXT LINE OF THE PROGRAM (OR IMMEDIATELY IF AN 'INPUT' STATEMENT IS BEING EXECUTED). HENCE, UNLESS QQ IS SET NON-ZERO (SEE SECTION ON EFFECTS OF ERRORS *12*), CONTROL RETURNS TO EDIT MODE AND THE LINE BEING EXECUTED WHEN DC3 WAS DEPRESSED IS PRINTED OUT.

7.5 TRANSFER TO MONITOR LEVEL

A TRANSFER TO MONITOR LEVEL MAY BE EFFECTED BY THE COMMAND -

```
'K'
```

OR BY DEPRESSION OF THE SO KEY (CONTROL N). SUCH A TRANSFER IS ONLY EFFECTIVE IF THE SWITCH REGISTER HAS BEEN SET NEGATIVE BY A 'QSW' COMMAND, OTHERWISE QUASIC IS RESTSARTED AS IF THE USER HAD JUST LOGGED IN.

LOGOUT MAY BE EFFECTED BY THE COMMAND -

```
'B'
```

WHICH CAUSES THE USER TO BE LOGGED OUT AND NO OTHER EFFECTS (AS OPPOSED TO THE USE OF 'K' AT MONITOR LEVEL WHICH CAUSES THE QUASIC PROGRAM 'QUIT' TO BE RUN).

8 PROGRAM CONSTRUCTION AND EDITING COMMANDS

QUASIC PROVIDES MISCELLANEOUS COMMANDS FOR ANNOTATING, LISTING, DELETING AND EDITING QUASIC PROGRAMS. IT ALSO PROVIDES A DYNAMIC COMPILATION FACILITY THROUGH THE 'CODE' COMMAND.

THE COMMAND -

'REM' STRING

CAUSES CONTROL TO BE PASSED IMMEDIATELY TO THE NEXT LINE OF THE PROGRAM. THE REMAINDER OF THE LINE, STRING, IS NEVER SEEN BY QUASIC AND HENCE MAY BE USED FOR COMMENTS. NOTE THAT COMMENTS MAY ALSO BE INSERTED IN A 'NAME' COMMAND, BUT IN THIS CASE THE STRING IS ACTUALLY EXAMINED FOR FURTHER COMMANDS - EG -

```
>100 REM THE NEXT LINE ALSO CONTAINS COMMENTS
>110 NAME AB A PRINT COMMAND :PRINT 'HI
>RUN
HI

QUASIC V3A

>
```

THE COMMAND -

'LIST' NE

MAY BE USED TO LIST THE CURRENT QUASIC PROGRAM COMMENCING AT THE FIRST LINE WHOSE NUMBER IS GREATER THAN OR EQUAL TO NE. ALL LISTINGS ARE FORMATTED WITH COMMAND NAMES FULLY RECONSTRUCTED.

THE COMMAND -

'CLEAR'

MAY BE USED TO DELETE THE CURRENT PROGRAM COMPLETELY. TO DELETE INDIVIDUAL LINES THE LINE NUMBER ALONE SHOULD BE TYPED (SEE 'CODE' COMMAND FOR INFORMATION ON STORED PROGRAM DELETION OF PROGRAM LINES).

THE COMMAND -

'GARB'

MAY BE USED TO FORCE THE CURRENT PROGRAM TO BE COMPACTED INTO THE SMALLEST POSSIBLE SPACE. THIS OCCURS AUTOMATICALLY WHENEVER THERE IS LESS THAN 128 WORDS OF FREE SPACE WHEN QUASIC COMES TO EXECUTE THE NEXT LINE OF A PROGRAM. HOWEVER, THERE ARE OCCASIONS WHEN A SPACE-CRITICAL PROCESS REQUIRES MORE THAN 128 WORDS OF FREE SPACE AND 'GARB' MAY BE USED TO ENSURE IT IF AVAILABLE; AFTER A 'GARB' COMMAND CONTROL PASSES IMMEDIATELY TO THE NEXT PROGRAM LINE.

8.1 LINE EDITING

THE COMMAND -

```
'X' NE LS1 LS2
```

MAY BE USED TO EDIT THE PROGRAM LINE WHOSE NUMBER IS NE BY REPLACING THE LITERAL STRING (QUOTED STRING) LS1 IN IT BY THE LITERAL STRING LS2. AFTER THE REPLACEMENT THE RESULTING LINE IS PRINTED OUT - EG -

```
>23 P C2 :L Y=9:G79
>X 23 '9' '900'
  23 PRINT C2 :LET Y=900 :GOTO 79
>X 23 'P' 'IF C
  23 IF C=3 :PRINT C2 :LET Y=900 :GOTO 79
>X 23 "C=3" "$='F'"
  23 IF $='F' :PRINT C2 :LET Y=900 :GOTO 79
>X 23 '2' '64
 643 IF $='F' :PRINT C2 :LET Y=900 :GOTO 79
>LIST

  23 IF $='F' :PRINT C2 :LET Y
 643 IF $='F' :PRINT C2 :LET Y=900 :GOTO 79
>
```

NOTE THAT THE TEXT STRING OPERATED UPON BY THE 'X' COMMAND IS EXACTLY THAT WHICH WOULD BE PRINTED OUT BY A 'LIST' COMMAND, AND HENCE BOTH COMMAND NAMES AND LINE NUMBERS CAN BE EDITED. NOTE ALSO THAT WHEN A LINE NUMBER IS CHANGED THE ORIGINAL VERSION IS NOT DESTROYED. NOTE ALSO THE USE OF EITHER TYPE OF MATCHED QUOTE TO ALLOW QUOTES TO BE USED IN THE BODY OF THE LITERAL STRING, AND THAT THE TERMINAL QUOTE MAY BE OMITTED.

THE 'X' COMMAND IS IN FACT A VARIANT OF THE 'PUT' COMMAND AND THE FULL PATTERN-MATCHING/REPLACEMENT CAPABILITY OF THE PUT COMMAND IS AVAILABLE. HOWEVER, ANY FORMS OTHER THAN THE LITERAL STRINGS ILLUSTRATED WILL RARELY BE USED.

8.2 DYNAMIC COMPILATION

THE COMMAND -

```
'CODE' SE
```

CAUSES THE STRING EXPRESSION SE TO BE EVALUATED AND TREATED AS IF HAD BEEN TYPED IN AT THE TELEPRINTER DURING EDIT MODE. IF SE EVALUATES TO A LINE NUMBER ONLY THE RELEVANT LINE WILL BE DELETED. HENCE THE 'CODE' COMMAND MAY BE USED TO DYNAMICALLY CREATE AND DESTROY PROGRAM LINES UNDER PROGRAM CONTROL. IN PARTICULAR THIS MEANS THAT ARRAYS MAY BE DYNAMICALLY CREATED AND DESTROYED AS REQUIRED. THIS SELF-MODIFYING CAPABILITY IS PARTIULARLY IMPORTANT ON A SMALL MACHINE AS IT ALLOWS MAXIMUM USE OF THE SPACE AVAIABLE - FOR EXAMPLE, INITIALIZING PROGRAM LINES CAN BE OVER-WRITTEN AS ARRAY WORKSPACE

LATER IN A PROGRAM.

THE EFFECT OF ANY CODE COMMAND IS BEST EVALUATED BY CONSIDERING THE EFFECT OF PRINTING THE STRING EXPRESSION USING A 'PRINT' COMMAND, AND THEN CONSIDERING THE EFFECT OF TYPING IN THE PRINTED EXPRESSION DURING KEYBOARD EDIT MODE - EG -

```
>CODE 'PRINT 6'
   6
>PRINT 6
   6
>CODE '10 ARRAY C 4'
>12 ARRAY D 6
>LIST
  10 ARRAY C 4
  12 ARRAY D 6
>LET X=100 :CODE X 'ARRAY' X
>LIST 100

 100 ARRAY 100

>LET B=4 :CODE 200 'LET N' B@ '= X*' B@
LIST 200

200 LET N4= X*4

>CODE 10*20
>LIST 200

>
```

9 FILE SYSTEM

DISC AND MAGNETIC TAPE ARE BOTH AVAILABLE TO HOLD DATA FILES AND PROGRAM OVERLAYS. THE QUASIC DISC SYSTEM USES THREE DISTINCT FILE DATA STRUCTURES -

1 PROGRAM FILES - COMPILED PROGRAMS MAY BE SAVED AND CALLED DYNAMICALLY AS OVERLAYS.

2 QUASIC LINE FILES - THE INDIVIDUAL LINES OF A QUASIC PROGRAM MAY BE STORED & RETRIEVED - SINCE STRINGS, ARRAYS & EXECUTABLE LINES ARE ALL

PART OF THE PROGRAM THIS MAY BE USED TO BUILD & OPERATE ON MIXED NUMERIC & CHARACTER-STRING FILES.

3 BINARY FILES - ARBITRARY FILE STRUCTURES MAY BE CREATED & RETRIEVED IN RAW FORM

ANY FILE HAS ASSOCIATED WITH IT THREE PARAMETERS :-

1 FILENAME - A ONE TO SIX CHARACTER NAME CONSISTING OF ALPHANUMERICS (IN FACT ANY PRINTING CHARACTERS EXCEPT ^[]_ BUT OTHER SYSTEM PROGRAMS ALLOW ONLY ALPHANUMERICS & IT IS BEST TO RESTRICT TO THESE)

2 ACCOUNT NUMBER - THE ACCOUNT UNDER WHICH THE FILE WAS CREATED - A NUMBER IN THE RANGE 1 THROUGH 4094 - THE NAME & ACCOUNT NUMBER UNIQUELY IDENTIFY A FILE

3 PROTECTION STATUS - A NUMBER IN THE RANGE 0 THROUGH 31 WHICH IDENTIFIES THE MODE OF ACCESS POSSIBLE TO A FILE - ITS FUNCTION IS EXPLAINED IN DETAIL UNDER THE COMMAND 'QPR', 'SET PROTECTION STATUS'

WHENEVER A FILE IS IN USE IT IS NECESSARY FOR IT TO BE 'OPEN' ON ONE OF FOUR DATA CHANNELS NUMBERED 1 THROUGH 4 (THESE CORRESPOND TO MONITOR CHANNELS 0 THROUGH 3). THE 'OPENING' MAY BE AUTOMATIC WITH SOME COMMANDS BUT MAY NEED TO BE EXPLICIT WITH OTHERS.

ASSOCIATED WITH EACH FILE DATA CHANNELS IS A PAIR OF WORDS FORMING A 'POINTER' TO THE FILE WHICH DETERMINES WHERE IN THE FILE 'READS & WRITES' OCCUR. THESE POINTER ARE AUTOMATICALLY SET UP & UPDATED BY SOME FILE OPERATIONS, BUT MAY ALSO BE REFERENCED EXPLICITLY THROUGH THE FUNCTION PT(NE), WHERE NE IS A NUMERIC EXPRESSION WITH A VALUE IN THE RANGE 0 THROUGH 9 :-

THE FIRST TWO PONTERS REFER TO THE TYPEWRITER CHANNEL -

PT(0) POSITION OF TYPEWRITER CARRIAGE ACROSS PAPER
PT(1) POSITION OF TYPEWRITER CARRIAGE DOWN PAPER - FOR PAGINATING

PT(2) LOWER POINTER CHANNEL 1

PT(3) UPPER POINTER CHANNEL 1

PT(4) LOWER POINTER CHANNEL 2
PT(5) UPPER POINTER CHANNEL 2

PT(6) LOWER POINTER CHANNEL 3
PT(7) UPPER POINTER CHANNEL 3

PT(8) LOWER POINTER CHANNEL 4
PT(9) UPPER POINTER CHANNEL 4

THE ACTUAL VALUE OF A FILE POINTER IS -

4096 * UPPER POINTER + LOWER POINTER.

THE 'FUNCTION' PT(NE) MAY BE ASSIGNED VALUES IN A 'LET' COMMAND, ENABLING FILE POINTERS TO BE SET AS REQUIRED, EG -

```
>LET PT(4)=76 PT(5)=0
```

9.1 FILE OPERATIONS

THE COMMAND -

'CREATE' NE SE

MAY BE USED TO CREATE A FILE NAMED SE AND OPEN IT ON CHANNEL NUMBER NE. IF NE IS ZERO OR ABSENT THE FILE IS OPENED ON CHANNEL 1. IF A FILE WITH NAME SE ALREADY EXISTS IT IS DELETED AND A NEW FILE CREATED. THE LENGTH OF A FILE ON CREATION IS 256 WORDS (1 DISC SEGMENT). WHEN THE FILE IS OPENED ON A CHANNEL THE ASSOCIATED FILE POINTERS FOR THAT CHANNEL ARE AUTOMATICALLY SET TO ZERO.

IN COMMON WITH THE MAJORITY OF THE FILE COMMANDS IF THE 'CREATE' COMMAND IS SUCCESSFULLY EXECUTED QUASIC GOES ON TO EXECUTE THE REST OF THE LINE, BUT IF EXECUTION IS NOT POSSIBLE QUASIC GOES DIRECT TO THE NEXT LINE. IN BOTH CASES AN ERROR CODE IS PLACED IN SYSTEM VARIABLE QE, ZERO IF SUCCESSFUL - NON-ZERO OTHERWISE (SEE SECTION *12*). HENCE THE FOLLOWING TYPE OF SEQUENCE TRAPS FILE ERRORS -

```
>100 CREATE 'ABCD' :GOTO 120
>110 PRINT 'FILE ERROR' :STOP
>
```

THE COMMAND -

'OPEN' NE1 NE2 SE

OPENS THE FILE SE ON ACCOUNT NUMBER NE2 ON CHANNEL NUMBER NE1. IF NE1 IS ZERO THE CHANNEL IS TAKEN TO BE 2, AND IF NE2 IS ZERO THE ACCOUNT IS TAKEN TO BE THE CURRENT USER ACCOUNT. NOTE THAT IF THE ACCOUNT IS SPECIFIED EXPLICITLY THEN THE CHANNEL NUMBER MUST ALSO BE SPECIFIED EXPLICITLY. AS USUAL, IF THE OPEN COMMAND CANNOT BE EXECUTED CONTROL IS TRANSFERRED IMMEDIATELY TO THE NEXT PROGRAM LINE. WHEN THE FILE

IS OPENED ON A CHANNEL THE FILE POINTERS ASSOCIATED WITH THAT CHANNEL ARE AUTOMATICALLY SET TO ZERO.

THE COMMAND -

```
'CLOSE' NE1 NE2 NE3   ETC
```

MAY BE USED TO CLOSE THE FILES ON CHANNELS NE1, NE2, NE3 ETC. IF THE FIRST NUMERICAL EXPRESSION IN THE CLOSE COMMAND EVALUATES TO ZERO THEN THE FILES ON CHANNELS 1 & 2 ARE CLOSED - EG BY 'CLOSE' ALONE.

THE COMMAND -

```
'DELETE' NE
```

MAY BE USED T_ DELETE THE FILE OPEN ON CHANNEL NE. IF NE IS ZERO THE CHANNEL IS TAKEN TO BE 4. AS USUAL IS THE 'DELETE' COMMAND CANNOT BE EXECUTED CONTROL IS TRANSFERRED IMMEDIATELY TO THE NEXT PROGRAM LINE.

THE COMMAND -

```
'RENAME' NE SE
```

MAY BE USED TO RENAME THE FILE OPEN ON CHANNEL NE WITH THE NAME GIVEN BY SE. IF NE IS ZERO THE CHANNEL IS TAKEN TO BE 4. AS USUAL IF THE 'RENAME' COMMAND CANNOT BE EXECUTED CONTROL IS TRANSFERRED IMMEDIATELY TO THE NEXT PROGRAM LINE.

9.2 SETTING FILE PROTECT STATUS

THE COMMAND -

```
QPR NE
```

MAY BE USED TO ASSIGN A PROTECTION STATUS TO THE FILE OPEN ON CHANNEL X - WHERE :

```
NE = 32*(X-1) + 16*W0 + 8*W1 + 4*R1 + 2*W2 + R2
```

WHERE W0, W1, W2, AND R1 & R2, ARE BINARY VARIABLES WITH VALUES 0 OR 1 EXPRESSING THE WRITE OR READ PROTECT STATUS, RESPECTIVELY, OF THE FILE.

ON THE TSS8 THE USER ACCOUNTS ARE DIVIDED INTO CONSECUTIVE BLOCKS OF 64 ACCOUNTS FOR THE PURPOSES OF FILE PROTECTION & IT IS POSSIBLE FOR USERS WITHIN A BLOCK TO SHARE FILES AMONGST THEMSELVES BUT EXCLUDE USERS OUTSIDE THE BLOCK - THE BLOCK NUMBER OF A USER IS HIS ACCOUNT NUMBER DIVIDED BY 64. THE MEANING OF THE VARIOUS PROTECT STATUSES IS THEN -

W0	=	1	WRITE PROTECT AGAINST OWNER
W1	=	1	WRITE POTECT AGAINST USERS IN SAME BLOCK
W2	=	1	WRITE PROTECT AGAINST USERS IN OTHER BLOCKS
R1	=	1	READ PROTECT AGAINST USERS IN SAME BLOCK

R2 = 1 READ PROTECT AGAINST USERS IN OTHER BLOCKS

IF A FILE IS WRITE PROTECTED AGAINST A USER IT IS ALSO DELETE & RENAME PROTECTED AGAINST HIM. THE PROTECT STATUS OF A FILE WHEN IT IS CREATED IS SET AUTOMATICALLY TO 10 - ALL MAY READ BUT ONLY OWNER MAY WRITE.

9.3 OBTAINING FILE STATUS INFORMATION

THE COMMAND -

'QFI' NE

MAY BE USED TO DEPOSIT INFORMATION ABOUT THE FILE OPEN ON CHANNEL X IN THE ARRAY IN LINE NE. THE ARRAY MUST HAVE AT LEAST 7 ELEMENTS (AN ERROR MESSAGE WILL BE GIVEN OTHERWISE) AND THE ZEROTH ELEMENT OF THE ARRAY MUST BE SET TO X-1. AFTER EXECUTION OF THE COMMAND THE FIRST 7 WORDS OF THE ARRAY ARE SET TO -

WORD 0 - CHANNEL NUMBER (X) MINUS 1

WORD 1 - ACCOUNT NUMBER OF FILE OWNER - ZERO IF NO FILE OPEN ON CHANNNEL

WORD 2 - FIRST TWO CHARACTERS OF FILE NAME
WORD 3 - NEXT TWO CHARACTERS OF FILE NAME
WORD 4 - FINAL TWO CHARACTERS OF FILE NAME

WORD 5 - FILE PROTECT STATUS

WORD 6 - NUMBERS OF SEGMENTS IN FILE (BLOCKS OF 256 WORDS)

THE CHARACTERS FORMING THE FILE NAME ARE PACKED TWO PER WORD BY SUBTRACTING 32 FROM THEIR ASCII VALUES (EG SPACE = 0) AND MULTIPLYING VALUE OF FIRST CHARACTER BY 64.

9.4 DATA FILE OPERATIONS

DATA FILES ARE USED IN QUASIC TO FILE AND RETRIEVE DATA STRUCTURES EITHER IN THE FORM OF PROGRAM LINES (THUS INCLUDING ARRAYS AND STRINGS) OR IN THE FORM OF THE CONTENTS OF ARRAYS (ALLOWING ARBITRARY WORD-ORGANIZED STRUCTURES). IN ALL CASES THE INITIAL ADDRESS OF THE DATA BLOCK IN THE FILE IS THE VALUE OF THE APPROPRIATE PAIR OF FILE POINTERS (SECTION *9.1*), AND THESE POINTERS ARE UPDATED AUTOMATICALLY SO THAT SEQUENTIAL ACCESSES TO FILES MAY BE PERFORMED WITHOUT EXPLICIT MODIFICATION OF THE FILE POINTERS.

THE COMMAND -

'WRITE' NE1 NE2

WRITES OUT THE CONTENTS OF PROGRAM LINE NE1 ONTO THE FILE OPEN ON CHANNEL NE2. IF NE2 IS ZERO OR ABSENT THE CHANNEL IS TAKEN TO BE 1. AS USUAL IF THE 'WRITE' COMMAND IS NOT SUCCESSFUL CONTROL IS

TRANSFERRED IMMEDIATELY TO THE NEXT PROGRAM LINE. THE FILE POINTERS ASSOCIATED WITH THE WRITE CHANNEL ARE AUTOMATICALLY UPDATED AND THE LENGTH OF THE FILE IS INCREASED IF NECESSARY BY THE THE MINIMUM AMOUNT THAT WILL ACCOMODATE THE WRITE.

THE COMMAND -

'READ' NE1 NE2

READS IN FROM CHANNEL NE2 OF THE DISC A DATA-STRUCTURE WHICH IS TAKEN TO BE A PROGRAM LINE & LABELS THIS AS PROGRAM LINE NE1. IF NE2 IS ZERO OR ABSENT THE CHANNEL IS TAKEN TO BE 2. AS USUAL IF THE 'READ' IS NOT SUCCESFUL CONTROL IS TRANSFERRED IMMEDIATELY TO THE NEXT PROGRAM LINE. NOTE THAT READING OVER THE END OF A FILE IS NOT TAKEN AS UNSUCCESFUL.

THERE ARE DANGERS INHERENT IN THE 'READ' COMMAND IN THAT IT OPERATES ESSENTIALLY ON THE ASSUMPTION THAT THE DATA STRUCTURE READ IN HAS BEEN CREATED BY A 'WRITE' COMMAND - AN ATTEMPT TO READ AN ARBITRARY DATA STRUCTURE IN THIS MODE MAY WELL LEAD TO THE QUASIC PROGRAM BEING CORRUPTED. THE DATA-STRUCTURES ON THE DISC ARE IDENTICAL TO THOSE IN CORE (WITH ESCAPE CHARACTERS FOR COMMAND NAMES) EXCEPT THAT THE INITIAL 2-WORD HEADER OF LINE NUMBER AND LINK TO NEXT PROGRAM LINE IS OMITTED. THE MOST IMPORTANT APPLICATION OF THIS MODE OF ACCESSING THE DISC IS TO ALLOW ARBITRARY LENGTH CHARACTER-RECORDS TO BE MANIPULATED AS FILE STRUCTURES BY WRITING OUT $-LINES.

A SECOND MODE OF ACCESSING DATA ON THE DISC IS PROVIDED THAT IS ESSENTIALLY SAFE AND CANNOT CAUSE PROGRAM CORRUPTION UNDER ANY CONDITIONS. IN THIS MODE THE DATA PART ONLY OF AN ARRAY IS WRITTEN TO OR READ FROM THE DISC; '&' IS USED AS AN ESCAPE CHARACTER TO INDICATE THIS MODE. THE COMMAND -

'WRITE' '&' NE1 NE2

WRITES OUT THE DATA PART OF THE ARRAY IN LINE NE1 INTO THE FILE OPEN ON CHANNEL NE2. IF NE2 IS ZERO OR ABSENT THE CHANNEL IS TAKEN TO BE 1. AS USUAL IF THE WRITE IS NOT SUCCESFUL CONTROL IS TRANSFERRED IMMEDIATELY TO THE NEXT PROGRAM LINE. THE LENGTH OF THE BLOCK OF WORDS WRITTEN TO THE DISC IS THE NUMBER OF ELEMENTS IN THE ARRAY AND THE ASSOCIATED FILE POINTERS ARE UPDATED BY THIS AMOUNT. THE FILE IS EXTENDED IF NECESSARY BY THE MINIMUM AMOUNT TO ALLOW THE WRITE TO OCCUR.

THE COMMAND -

'READ' '&' NE1 NE2

READS INTO THE DATA PART OF THE ARRAY IN LINE NE1 FROM THE FILE OPEN ON CHANNEL NE2. IF NE2 IS ZERO OR ABSENT THE CHANNEL IS TAKEN TO BE 2. AS USUAL IF THE READ IS NOT SUCCESFUL CONTROL IS TRANSERRED IMMEDIATELY TO THE NEXT PROGRAM LINE. THE LENGTH OF THE BLOCK READ FROM THE DISC IS THE NUMBER OF ELEMENTS IN THE ARRAY AND THE ASSOCIATED FILE POINTERS ARE UPDATED BY THIS AMOUNT.

9.5 QUASIC PROGRAM FILES

COMPLETE QUASIC PROGRAMS MAY BE SAVED AS FILES AND CALLED & STARTED DYNAMICALLY AS OVERLAYS. MULTIPLE OVERLAYS MAY BE KEPT IN A SINGLE FILE SO AS TO AVOID THE DIRECTORY-SEARCH OVERHEAD OF OPENING EACH OVERLAY FILE WHEN REQUIRED. THERE ARE TWO MODES OF OPERATION - ONE FOR OPENING OR CREATING NAMED PROGRAM FILES & THE OTHER FOR USING ALREADY OPEN PROGRAM FILES. PROGRAM FILES ARE ALWAYS OPENED ON CHANNEL 3.

THE COMMAND -

```
'SAVE' SE
```

CREATES A NEW FILE NAMED SE (DELETING ANY PREVIOUS FILES OF THAT NAME), OPENS IT ON CHANNEL 3 AND DUMPS THE CURRENT QUASIC PROGRAM AS THE FIRST SIX SEGMENTS OF THE FILE (1536 WORDS). ONLY THE PROGRAM LINES ARE DUMPED NOT THE VALUES OF THE SIMPLE VARIABLES.

THE COMMAND -

```
'SAVE' '&' NE
```

DUMPS THE CURRENT QUASIC PROGRAM AS THE NE'TH OVERLAY (STARTING AT SEGMENT NE*6 AND BEING 6 SEGMENTS LONG) OF THE PROGRAM FILE OPEN ON CHANNEL 3. THE PROGRAM FILE IS EXTENDED AS NECESSARY TO ACCOMODATE THE PROGRAM. ANY OTHER OVERLAYS PRESENT IN OTHER PARTS OF THE FILE ARE PRESERVED SO THAT INDIVIDUAL OVERLAYS OF A PROGRAM CAN BE UPDATED WITH THIS COMMAND.

NOTE THAT LOADING A PROGRAM OPENS IT ON CHANNEL 3 AND HENCE CLOSES ANY OTHER FILE OPEN ON THAT CHANNEL. HENCE TO DUMP A SET OF NAMED PROGRAMS AS OVERLAYS WITHIN ONE PROGRAM THE FOLLOWING SEQUENCE IS NECESSARY -

```
>CALL  'NAME1
>SAVE  'NAME
>CALL  'NAME2
>OPEN  3 'NAME
>SAVE  &1
>CALL  'NAME3
>OPEN  3 'NAME
>SAVE  &2
>
```

AND SO ON.

THE USUAL FILE ERROR TRAP IS PROVIDED FOR BOTH FORMS OF THE 'SAVE' COMMAND SO THAT CONTROL IS PASSED IMMEDIATELY TO THE NEXT PROGRAM LINE IF THE 'SAVE' IS UNSUCESSFUL.

THE COMMAND -

```
'CALL'  NE1 NE2  SE
```

OPENS THE FILE NAMED SE UNDER ACCOUNT NE2 ON CHANNEL 3, LOADS THE FIRST 6 SEGMENTS (OVERLAY 0) OF THE FILE AS THE CURRENT QUASIC PROGRAM, AND RUNS IT FROM THE FIRST LINE WHOSE NUMBER IS GREATER THAN OR EQUAL TO NE1. IF NE1 IS ZERO OR ABSENT THEN CONTROL IS TRANSFERRED TO KEYBOARD EDIT MODE. IF NE2 IS ZERO OR ABSENT THE PROGRAM IS LOOKED FOR UNDER THE CURRENT ACCOUNT. WHATEVER THE VALUE OF NE2, IF THE PROGRAM IS NOT FOUND IT IS LOOKED FOR UNDER THE QUASIC LIBRARY ACCOUNT (ACCOUNT 3). THE NORMAL INITIALIZATIONS PERFORMED BY 'RUN' ARE ALSO PERFORMED BY 'CALL' EXCEPT THAT THE SIMPLE VARIABLES ARE NOT DESTROYED ENABLING THEM TO BE USED FOR COMMUNICATION BETWEEN OVERLAYS. IF THE 'CALL' COMMAND CANNOT BE EXECUTED (EG PROGRAM FILE DOES NOT EXIST, THEN CONTROL IS TRANSFERRED TO THE NEXT LINE OF THE CALLING PROGRAM ENABLING CALL ERRORS TO BE TRAPPED.

THE COMMAND -

'CALL' NE1 '&' NE2

CAUSES THE NE2'ND OVERLAY OF THE FILE OPEN ON CHANNEL 3 TO BE LOADED AS THE CURRENT QUASIC PROGRAM AND THIS TO BE RUN AT THE FIRST LINE WHOSE NUMBER IF NE1 OR GREATER. IF NE1 IS ZERO CONTROL PASSES TO KEYBOARD EDIT MODE - THE SIMPLE VARIABLES ARE NOT DESTROYED BY THE CALL.

9.6 SYSTEM PROGRAM COMMUNICATION

QUASIC PROVIDES FACILITIES FOR QUASIC PROGRAMS TO CALL MACHINE-CODE PROGRAMS, PASSING THEM STATUS INFORMATION, AND FOR QUASIC ITSELF TO BE RECALLED AND A QUASIC PROGRAM AUTOMATICALLY INITIATED.

THE COMMAND -

'ENTER' NE SE

CAUSES THE FIRST 15 SEGMENTS (OR ALL IF LESS THAN 15) OF THE FILE SE ON ACCOUNT NE TO BE LOADED INTO CORE STARTING AT LOCATION 0, AND CONTROL

TO BE PASSED TO THE INSTRUCTION IN LOCATION 0 OF THE LOADED PROGRAM. LOCATIONS 10-17 SHOULD NOT BE PRESET IN THE LOADED PROGRAM & LOCATION 10 IS PRESET TO 1 BY QUASIC TO INDICATE THAT IT IS THE CALLER (THE PROGRAM IS ASSUMED TO HAVE THIS LOCATION PRESET ITSELF TO 0 SO THAT IT WILL HAVE A DEFINITE VALUE WHEN THE PROGRAM IS CALLED BY THE MONITOR). IF THE FILE IS NOT FOUND ON THE PRESCRIBED ACCOUNT IT IS LOOKED FOR ON ACCOUNT 2 (MACHINE-CODE LIBRARY) & IF IT IS NOT FOUND AT ALL CONTROL IS TRANSFERRED TO THE NEXT LINE OF THE QUASIC CALLING PROGRAM. IF NE IS ZERO OR ABSENT THE CURRENT ACCOUNT IS ASSUMED.

THE FOLLOWING LOCATIONS CONTAIN INFORMATION & SHOULD BE PRESERVED IF A RETURN IS TO BE MADE TO QUASIC -

7777 - Q0 - LOWER 6 BITS USED BY QUASIC TO DECIDE QUASIC PROGRAM TO RUN ON RETURN

7775 - Q1 - STATUS INFORMATION ON RETURN

7773 - Q2 - MODE INFORMATION ON ENTRY - REPLY INFORMATION ON RETURN

THESE LOCATIONS MAY BE SET UP BY THE QUASIC CALLING PROGRAM AS SYSTEM VARIABLES, Q0 THROUGH Q2. ONLY Q2 MAY BE MODIFIED BY THE ENTERED MACHINE-CODE PROGRAM.

QUASIC MAY BE RECALLED BY LOADING THE PROGRAM FILE 'Q' FROM ACCOUNT 2, SETTING LOCATION 10 TO 1, RESETTING LOCATIONS 7775 & 7777 IF THEY HAVE BEEN MODIFIED, AND PUTTING ANY REPLY INFORMATION IN 7773. IF CONTROL IS THEN PASSED TO LOCATION 0 QUASIC CALL A PROGRAM DETERMINED BY THE LOWER HALF OF LOCATION 7777 -

```
 0 - QEX
 1 - QUARK
 2 - QUEST
 3 - QUERY
 4 - QUOTE
 5 - QUIRK
 6 - QUILL
 7 - QMS
 8 - QUIT
 9 - QEX
10 - QUIT
```

THESE ARE CALLED BY A COMMAND OF THE FORM 'CALL 1 "QUARK"' WHICH WILL FIRST LOOK FOR QUARK ON THE USER ACCOUNT & THEN ON THE LIBRARY ACCOUNT 3 (QEX, HOWEVER, IS ONLY RETRIEVED FROM ACCOUNT 4).

THUS MACHINE-CODE PROGRAMS ARE READILY LINKED INTO A SUITE OF QUASIC PROGRAMS. IN PARTICULAR THE UTILITY PROGRAMS, ED, COPY, ZAP, LILO & K, MAY BE SLAVED IN THIS WAY.

9.7 MAGNETIC TAPE

THE SYSTEM PROGRAM, C, IS NORMALLY USED TO TRANSFER FILES TO & FROM TAPE AS NAMED ENTITIES - IT PROVIDES FACILITIES FOR COPYING, DELETING, RENAMING, UPDATING & LISTING FILES ON TAPE OR DISK (OR TRANSFERRING BETWEEN TAPE & DISC) WHICH MAY ALSO BE VERIFIED BY READ-AFTER-WRITE AUTOMATICALLY IF REQUIRED. THE PROGRAM, C, MAY BE SLAVED BY A QUASIC PROGRAM AS DESCRIBED IN SECTION *9.6*, OR ENTERED FROM THE QUASIC EXECUTIVE, QEX, AS DESCRIBED IN SECTION *13*.

THERE ARE OCCASIONS, HOWEVER, WHEN MAGTAPE IS TO BE USED AS AN ON-LINE DATA STORAGE/RETRIEVAL MEDIUM, RATHER THAN AS A SEMI-OFF-LINE FILE DUMPING AND ARCHIVING SYSTEM. QUASIC PROVIDES FACILITIES FOR USING TAPES AS BLOCK-STRUCTURED (RATHER THAN FILE-STRUCTURED) ENTITIES CONSISTING OF SOME 1474 CONSECUTIVE DATA BLOCKS EACH CONTAINING 129 WORDS. DATA MAY BE TRANSFERRED TO AND FROM ANY BLOCK, REFERRED TO BY NUMBER, FROM OR TO AN ARRAY OF AT LEAST 129 ELEMENTS. SINCE THE TAPES ARE A GENERALLY AVAILABLE RESOURCE, COMMANDS ARE ALSO PROVIDED TO ASSIGN THEM TO, AND RELEASE THEM FROM, THE CURRENT JOB.

IT CANNOT BE OVER-EMPHASIZED THAT THE MAGNETIC TAPE COMMANDS MUST BE USED WITH GREAT CAUTION. WHEREAS A FILE PROTECT SYSTEM PREVENTS USERS ACCESSING NON-PERMITTED DISC FILES, NO SUCH SYSTEM EXISTS FOR THE TAPES UNLESS THEY ARE ACTUALLY ASSIGNED TO ANOTHER USER OR PHYSICALLY WRITE-LOCKED. HENCE IT IS POSSIBLE FOR ANY QUASIC PROGRAM TO WRITE ONTO A TAPE WHICH IS MOUNTED, READ-ENABLED & NOT ASSIGNED. THE OPERATIONAL PROCEDURES & THE STRUCTURE OF THE TAPE COMMAND ARE DESIGNED TO MINIMIZE THIS RISK, BUT IT IS IMPERATIVE THAT THE QUESTEL OPERATOR IS INFORMED OF ANY PROPOSED USE OF THE TAPES.

THERE ARE 8 LOGICAL TAPE UNITS, NUMBERED 0 THROUGH 7, TAPES MAY BE MOUNTED ON ANY UNITS, READ-ENABLED AND WRITE-ENABLED OR WRITE-DISABLED. THE COMMAND -

```
'QAS' NE
```

ASSIGNS THE TAPE TRANSPORT UNIT WHOSE NUMBER IS NE TO THE CURRENT USER. IF THE ASSIGNMENT IS SUCCESSFUL EXECUTION OF THE PROGRAM LINE, CONTINUES, BUT OTHERWISE CONTROL PASSES DIRECTLY TO THE NEXT LINE WITH AN ERROR CODE IN QE; THE ERROR-CODE IS 4095 IF NE IS OUT OF RANGE, OR THE JOB-NUMBER OF THE HOLDER OF THE UNIT IF IT IS ALREADY ASSIGNED TO ANOTHER UNIT.

THE COMMAND -

```
'QRE' NE
```

RELEASES THE UNIT WHOSE NUMBER IS NE, MAKING IT AVAILABLE TO OTHER USERS. THE UNIT REMAINS ASSIGNED UNTIL THIS COMMAND IS GIVEN OR THE USER LOGS OUT.

THE COMMAND -

```
'QTA' NE1 NE2 NE3 NE4
```

CAUSES THE 129-WORD BLOCK NUMBERED NE2 ON THE TAPE ON UNIT NE3 TO BE READ INTO (NE4 ZERO) OR WRITTEN FROM (NE4 NON-ZERO) THE FIRST 129 WORDS OF THE ARRAY IN LINE NE1. NOTE THAT IF NE4 IS ABSENT THE COMMAND IS A READ - IE A 3-ARGUMENT COMMAND IS A READ FROM TAPE ONLY. IF THE COMMAND IS SUCCESFUL EXECUTION OF THE LINE CONTINUES, OTHERWISE CONTROL IS TRANSFERRED TO THE NEXT LINE WITH AN ERROR CODE IN QE - THIS IS :

```
    2048 + 1024*MT + 512*ET + 256*SE + 128*PA + 64*TI
```

WHERE MT, ET, SE, PA, TI ARE NORMALLY ZERO, BUT HAVE THE VALUE ONE FOR A MARK-TRACK, END-OF-TAPE, SELECTION, PARITY, OR TIMING, ERROR, RESPECTIVELY.

THE TIME TAKEN TO BLOCK TRANSFER 1474 BLOCKS FROM A TAPE IS SOME 11 MINUTES - IE JUST OVER TWO BLOCKS A SECOND. THE MAXIMUM SEEK TIME ON THE TAPE IS UNDER 40 SECONDS.

10 MONITOR COMMUNICATION COMMANDS

THE COMMAND -

'QSY'

CAUSES THE EXECUTION OF QUASIC TO BE SUSPENDED UNTIL IT AGAIN COMES TO THE HEAD OF THE MONITOR'S JOB QUEUE. THIS COMMAND CAN BE USED TO PROVIDE A NON-COMPUTE-BOUND CONDITIONAL WAIT LOOP IN INTER-SYSTEM COMMUNICATION PROGRAMS.

THE COMMAND -

'QWA' NE

CAUSES EXECUTION OF QUASIC TO BE SUSPENDED FOR NE SECONDS. THIS IS PARTICULARLY USEFUL IN SYSTEM MONITOR PROGRAMS & IN 'BATCH' CONTROL PROGRAMS THAT WAKE UP PERIODICALLY TO SEE IF THE SYSTEM IS LIGHTLY LOADED. NOTE THAT ONLY THE SI & SO KEYS ON THE TELEPRINTER HAVE ANY INTERUPT EFFECT WHEN THIS COMMAND IS BEING EXECUTED.

THE COMMAND -

'QNA' SE

HAS THE EFFECT OF INFORMING THE MONITOR THAT THE USER IS RUNNING A PROGRAM CALLED SE ('CALL' & 'ENTER' DO THIS AUTOMATICALLY). SYSTEM PROGRAMS MAY USE THIS FACILITY TO GIVE THE SYSTEM SUPERVIZOR INFORMATION ABOUT THERE ACTIVITIES. THE NAME INFORMATION IS AVAILABLE TO A QUASIC PROGRAM USING THE 'QPK' COMMAND.

THE COMMAND -

'QPK' NE1 NE2 NE3

CAUSES THE ARRAY IN LINE NE1 TO BE FILLED WITH A CORE IMAGE OF THE TSS8 MONITOR COMMENCING AT WORD NE2 OF THE FIELD SPECIFIED BY NE3 - NE3 IS ZERO FOR FIELD ZERO AND 8 FOR FIELD 1. THIS COMMAND MAY BE USED TO PICK UP SYSTEM STATUS INFORMATION, PARTICULARLY THAT RELATING TO OTHER JOBS ON THE SYSTEM.

11 THE MEDELA TERMINAL

THE QUESTEL MEDELA TERMINAL CONSISTS OF A RANDOM-ACCESS MICROFILM DISPLAY, A KEYBOARD WITH TWO OR MORE KEYS & A MILLISECOND STOP-CLOCK, AND A TELEPRINTER. THE ENTIRE TERMINAL SYSTEM IS INTERFACED TO APPEAR TO THE COMPUTER AS IF IT WERE ONLY A TELEPRINTER. HOWEVER, MESSAGES FROM THE COMPUTER, APPROPRIATELY FORMATTED, MAY BE INTERPRETED BY THE TERMINAL AS INSTRUCTIONS TO MOVE THE FILM, AND THE DEPRESSION OF SINGLE KEY ON THE KEYBOARD MAY SEND BACK TO THE COMPUTER A COMPLETE MESSAGE INCLUDING RESPONSE-TIME DATA. QUASIC OFFERS A SPECIAL MACRO COMMAND TO CONTROL THE QUESTEL MEDELA TERMINAL, WHICH COMBINES CONTRO OF THE KEYBOARD, MOVEMENT OF THE FILM, TIMEOUT ON GARBLED MESSAGES, A RESPONSE ACQUISITION (AND ANALYSIS) INTO A SINGLE FORMAT.

THE COMMAND -

```
'SHOW' [ '*' CHAR ]  NE1 [ '>' NE2 ]  $-LINE
```

CAUSES THE FILM IN THE DISPLAY TO BE MOVED BY NE1 FRAMES (NE1 MAY BE POSITIVE OR NEGATIVE), AND THE RESPONSE FROM THE TERMINAL TO BE PLACE IN THE $-LINE. THE FIRST OPTION IS TO SEND A CHARACTER TO CONTROL THE MODE OF THE TERMINAL, PARTICULARLY THE KEYBOARD. THE SECOND OPTION IS TO ALLOW ONLY NE2 SECONDS FROM SENDING THE MESSAGE TO THE MEDELA, OR RECEIVING A PARTIAL REPLY, BEFORE TERMINATING THE COMMAND - THIS ENABLES 'HANG-UPS' WHEN GARBLED MESSAGES OCCUR TO BE AVOIDED.

THE FORMAT OF ALL MESSAGES TO THE DISPLAY & FROM THE KEYBOARD IS -

```
STX   MESSAGE   ETX
```

& ETX IS THE ONLY DELIMITER RECOGNIZED BY THE COMPUTER FOR MEDELA MESSAGES. THE 'PRINT' COMMAND MAY BE USED IN THE NORMAL WAY TO PRINT ON THE TELEPRINTER, BUT STX (CONTROL B) SHOULD NOT BE USED AS PART OF A PRINTED STRING. WHEN THE COMPUTER IS WAITING FOR INPUT FROM THE MEDELA IT WILL ACCEPT INPUT FROM THE TELEPRINTER KEYBOARD BUT WILL NOT REGARD THE MESSAGE AS COMPLETE UNTIL ETX IS RECEIVED (IE ETX RATH THAN CARRIAGE-RETURN MUST BE USED AS A MESSAGE TERMINATOR FOR ABORTIN OR OTHERWISE INTERACTING WITH, AN ONGOING MEDELA TEST). HOWEVER THE 'INPUT' COMMAND MAY BE USED IN THE NORMAL WAY TO OBTAIN INPUT SPECIFICALLY FROM THE TELEPRINTER.

THE MODES OF TERMINAL OPERATION SET UP BY THE OPTIONAL ASTERISK CHARACTER ARE:-

*B - MULTI-CHOICE - MOVE FILM - OPEN SHUTTER - START CLOCK - ACTIVATE 15-BUTTON MULTIPLE-RESPONSE KEYBOARD - CLOSE SHUTTER WHEN 'ENTER' KEY IS DEPRESSED.

*C - AUTO-ADVANCE - MOVE FILM - OPEN SHUTTER - START CLOCK - AFTER PRESET DELAY OF FEW SECONDS (USER OPTION SET UP IN KEYBOARD) CLOSE SHUTTER

*G - 2-CHOICE - MOVE FILM - OPEN SHUTTER - START CLOCK - ACTIVATE 2-BUTTON BINARY-CHOICE KEYBOARD - CLOSE SHUTTER WHEN ONE OF BINARY-CHOICE KEYS IS DEPRESSED.

IF NO * OPTION IS GIVEN THE FILM IS MOVED WITH THE SHUTTER CLOSED AND THE KEYBOARD IS NOT ACTIVATED - IN THESE CIRCUMSTANCES NO MESSACE WILL BE RETURNED BY THE TERMINAL UNLESS THE FILM HITS AN END-STOP (SE LATER), AND A TIMEOUT MUST BE USED TO AVOID A HANG-UP. NOTE THAT UNDE ALL MODES OF KEYBOARD OPERATION THE 'HELP' & 'CANCEL' KEYS ARE ACTIVE - THEY CAUSE SHUTTER TO CLOSE AN@ SEND A COMPLETE, DISTINCTIVE MESSAGE BACK TO THE COMPUTER.

THE 'SHOW' COMMAND IS A VARIANT OF THE 'INPUT' COMMAND AND FURTHER REMARKS CONCERNING TIMEOUT & THE USE OF THE FACILITIES OF THE 'PUT' COMMAND (INSTEAD OF THE $-LINE) ARE MADE IN SECTION *5.3*.

THE MESSAGES RECEIVED FROM THE MEDELA CONSIST GENERALLY OF NON-PRINTING CHARACTERS AND THESE ARE TRANSLATED TO PRINTING CHARACTERS BY THE 'SHOW' COMMAND BEFORE BEING PLACED IN THE $-LINE. THE FINAL FO OF THE MESSAGES RECEIVED IS -

FILM HAS HIT END-STOP -

```
      !=!V!T
```

AFTER AUTO-ADVANCE -

```
      !A!=!V/03000
```

WHERE 03000 IS AN EXAMPLE OF A 5-DIGIT NUMBER GIVING THE TIME THE FRAME WAS SHOWN.

TWO-BUTTON RESPONSE -

```
      !A!=!V*01234
```

INDICATES LEFT-HAND IQVPDEPRESSED, RESPONSE-TIME 01234 MSEC.

```
      !A!=!V,01234
```

INDICATES RIGHT-HAND KEY DEPRESSED, RESPONSE-TIME 01234 MSEC.

15-KEY RESPONSE -

```
      !A!=!V£01234
```

INDICATES 'ENTER' KEY ALONE DEPRESSED AFTER 01234 MSEC. EACH DEPRESSI OF ONE OF THE 15 MULTI-CHOICE KEYS CAUSES A MESSAGE OF THE FORM -

```
'!' CHAR
```

TO BE INSERTED BETWEEN THE 'V' AND THE '£' IN THE MESSAGE, WHERE CHAR IS A SINGLE LETTER FROM THE SET -

L M N O P Q R S T U V W X Y Z

CORRESPONDING TO THE 15 KEYS FROM LEFT TO RIGHT. THUS DEPRESSING EACH KEY IN TURN FROM LEFT TO RIGHT THEN 'ENTER' SENDS THE MESSAGE -

```
        !A!=!V!L!M!N!O!P!Q!R!S!T!U!V!W!X!Y!Z£09876
```

WHERE 09876 MSEC IS THE TIME TAKEN TO DEPRESS THE KEYS UP TO 'ENTER'. DEPRESSING THE FURTHER LEFT-HAND KEY TWICE FOLLOWED BY THE FURTHEST RIGHT-HAND KEY TWICE SENDS THE MESSAGE -

```
        !A!=!V!L!L!Z!Z£04567
```

NOTE THAT THE MESSAGE FROM EACH KEY IS SENT TO THE COMPUTER WHEN THE KEY IS DEPRESSED - THERE IS NO STORAGE IN THE KEYBOARD.

THE RUBOUT KEY INSERTS A SINGLE COLON IN THE MESSAGE - THUS -

```
        !A!=!V!Z!Z!L:!Z!Z!M£08765
```

IS SENT WHEN THE FURTHEST RIGHT HAND KEY IS DEPRESSED TWICE, THEN THE FURTHEST LEFT-HAND KEY, THEN RUBOUT, THEN A CORRECTED MESSAGE, THEN 'ENTER'. THUS 'RUBOUT' IS HANDLED BY THE PROGRAM (USING 'PUT' COMMAND TO STRIP OFF STRING AFTER LAST COLON) RATHER THAN IN THE KEYBOARD.

'CANCEL' KEY -

```
        !A!=!V>
```

INDICATES THAT 'CANCEL' KEY HAS BEEN DEPRESSED.

'HELP' KEY -

```
        !A!=!V<
```

INDICATES THAT 'HELP' KEY HAS BEEN DEPRESSED.

IT WILL HAVE BEEN NOTED THAT THE FILM LOCATION IS NOT SET UP ABSOLUTELY BY THE 'SHOW' COMMAND, BUT THAT RELATIVE MOVEMENTS ARE CAUSED WITH FEEDBACK WHEN THE FILM HITS AN END STOP. THE USUAL PROCEDURE IS TO INITIALIZE THE MEDELA TO AN END STOP WITH THE SHUTTER CLOSED & THEN MOVE TO THE REQUIRED FRAME. FURTHER MOVEMENTS MAY THEN BE RELATIVE, BUT THE FILM SHOULD BE RE-INITIALIZED WHEN A MESSAGE IS GARBLED, TIMEOUT OCCURS, OR AFTER A PROLONGED PERIOD OF RELATIVE MOVEMENT. THE CONTROL ELECTRONICS IS SET UP TO BE INITIALIZE BY A CARRIAGE-RETURN CHARACTER WHICH OCCURS SO FREQUENTLY THAT NO SPECIAL PROCEDURE IS REQUIRED. HOWEVER, IT IS PROBABLY GOOD PRACTICE TO SEND A CARRIAGE-RETURN BEFORE INITIALIZING THE FILM - EG -

```
>100 NAME INIT :PRINT &0141, :SHOW -1000 >10 $
>110 IF $ = '!=!V!T' :LET FP = 0 :RETURN 10
>120 RETURN
```

IS A ROUTINE CALL INIT (OR IN) WHICH SENDS A CARRIAGE-RETURN (141

DECIMAL), ATTEMPTS TO MOVE THE FILM -1000 FRAMES (TO FIND AN END-STOP), GIVING A TIMEOUT OF 10 SECONDS & PUTTING THE RETURN MESSAGE IN $ (IE $0). IF THE RETURN MESSAGE IS 'END-OF-FILM' IT SETS FP, THE FRAME POSITION, TO ZERO & RETURNS 10 ON FROM CALLING LINE, OTHERWISE IT RETURNS IMMEDIATELY AFTER THE CALLING LINE.

A ROUTINE TO MOVE THE FILM TO AN ABSOLUTE LOCATION, AL, ASSUMING THAT FP IS THE CURRENT FRAME POSITION & RE-INITIALIZING IF NECESSARY, IS -

```
>100 NAME MOVE :SHOW *G AL-FP >30 $
>110 PUT $ > '!A!=!V' >$ :LET FP = AL :RETURN
>120 GOSUB INIT
>125PPRINT 'FILM FAULT' :RETURN :GOTO ERROR
>130 GOTO 100
```

NOTE THAT 30 SECONDS IS ALLOWED FOR THE RESPONSE & THAT THE RESIDUAL MESSAGE (, OR * AND RESPONSE-TIME) IS LEFT IN $ FOR LATER ANALYSIS.

12 ERROR TRAPPING

AN ERROR IN COMPILATION OR EXECUTION IN QUASIC CAUSES EXECUTION OF THE CURRENT LINE TO CEASE AND CONTROL TO PASS TO AN ERROR TRAPPING ROUTINE WHICH SETS THE SYSTEM VARIABLE QE TO A VALUE INDICATING THE TYPE OF ERROR, AND THEN TAKES ACTION DEPENDENT ON THE CURRENT VALUE O THE SYSTEM VARIABLE QQ -

IF QQ=0 AN ERROR MESSAGE IS PRINTED OUT & CONTROL TRANSFERRED TO KEYBOARD EDIT MODE

IF QQ>0 AN EFFECTIVE 'GOSUB' IS EXECUTED FROM THE PROGRAM LINE IN WHICH THE ERROR OCURRED TO THE LINE NUMBERED QQ, AND QQ ITSELF IS SET TO 0. THIS MODE ENABLES ERRORS TO BE TRAPPED AND ANALYSED BY QUASIC PROGRAMS.

THE ERROR NUMBER PUT IN QE INDICATES -

0 - KEYBOARD INTERUPT - THE DC3 KEY (CONTROL S) HAS BEEN DEPRESSE THIS ENABLES THE DC3 KEY TO BE USED AS, FOR EXAMPLE, A 'CEASE-ACTIVITY' KEY FOR QUASIC PROGRAMS.

1 - SYNTAX ERROR - GENERALLY A WRONG CHARACTER IN ARITHMETIC EXPRESSION - BRACKETS NOT MATCHED, ETC.

2 - NUMERICAL EXPRESSION OUT OF RANGE - FOR EXAMPLE, ARRAY ELEMENT.

3 - ILLEGAL FUNCTION NAME

4 - SIMPLE VARIABLE NOT YET DEFINED.

5 - PROGRAM LINE REFERRED TO DOES NOT EXIST OR IS WRONG TYPE.

6 - SUBROUTINE STACK OVERFLOW ON 'GOSUB' OR UNDERFLOW ON 'RETURN'

7 - INCORRECT NUMBER OF ARGUMENTS FOR A FUNCTION.

8 - NO SPACE LEFT - DYNAMIC OR PROGRAM WORKSPACE EXCEEDED.

9 - COMPILATION ERROR - GENERALLY BADLY SPELT COMM;£_Z4IQ

THE ERROR MESSAGE PRINTED OUT WHEN QQ IS ZERO IS OF THE FORM -

```
'E' ERROR NUMBER    LINE BEING EXECUTED
CURRENT PROGRAM LINE
```

TWO LINES ARE PRINTED OUT BECAUSE THE CURRENT PROGRAM LINE MAY NOT BE THAT BEING EXECUTED - EG -

```
>100 DO 200
>200 PRINT X Y Z
```

```
>RUN

E4 PRINT X !  Y Z
 100 DO 200

QUASIC V3A

>
```

NOTE THE SHRIEK, '!', PRINTED OUT TO SHOW WHERE EXECUTION OF THE LINE CEASED - THIS GIVES A GUIDE TO THE PART OF THE LINE IN ERROR.

WHEN A NON-ZERO VALUE OF QQ IS USED TO CAUSE PROGRAM-LEVEL ERROR-TRAPPING A 'RETURN' COMMAND FOLLOWED BY ANY OTHER COMMAND MAY BE USED TO OBTAIN THE LINE NUMBER OF THE LINE CONTAINING THE ERROR (SEE SECTION *7.3*). NOTE THAT QQ IS SET TO ZERO, TURNING THE TRAP OFF, WH AN ERROR OCCURS, ENSURING THAT ERRORS IN THE ERROR SUBROUTINE (SUCH A ITS ABSENCE) CANNOT CAUSE A LOCKOUT CONDITION. AN EXAMPLE OF ERROR TRAPPING IS -

```
>10 LEP QQ = 500
[MAIN PROGRAM
>500 LET QQ = 500 :RETURN :PRINT 'ERROR' QE ' IN LINE' QR :GOTO NL(QR
>
```

WHICH PRINTS THE ERROR NUMBER AND THE LINE NUMBER OF THE LINE CAUSING THE ERROR & THEN RETURNS CONTROL TO THE LINE FOLLOWING IT.

13 QUASIC EXECUTIVE - QEX

THE QUASIC EXECUTIVE IS A QUASIC PROGRAM CALLED 'QEX' WHICH IS AUTOMATICALLY RUN WHENEEVER A USER LOGS IN TO THE TSS8. QEX SETS UP VARIOUS DATA ITEMS, SUCH AS THE CURRENT DATE, AND THEN TAKES ACTIO DEPENDENT ON THE USERS ACCOUNT NUMBER - FOR EXAMPLE, SOME USERS HAVE SPECIFIC APPLICATION PROGRAMS WHICH ARE AUTOMATICALLY RUN FOR THEM.

QEX CAN BE RUN FROM QUASIC BY THE COMMAND -

'QEX'

OR, AT ANY TIME, BY DEPRESSING THE SI KEY (CONTROL O) ON THE TELEPRINTER.

TO SYSTEM PROGRAMMERS QEX OFFERS ACCESS TO A NUMBER OF FACILITIES REQUESTED BY TYPING IN A COMMAND TO THE OUTPUT -

*QEX:

THESE ARE -

```
*QEX:Q          - ENTER QUASIC WITH PROGRAM CLEAR

*QEX:Q NAME     - ENTER QUASIC WITH PROGRAM 'NAME' LOADED

*QEX:R NAME     - RUN QUAS C PROGRAM 'NAME' FROM LINE 1

*QEX:T NAME     - TRANSFER TO MACHINE-CODE PROGRAM 'NAME' STARTING AT
                  LOCATION 0

*QEX:E          - TRANSFER TO EDITOR

*QEX:E NAME      - EDIT FILE 'NAME'

*QEX:C          - TRANSFER TO COPY

*QEX:C NAME     - TRANSFER TO COPY WITH COMMANDS IN FILE 'NAME'

*QEX:Z          - TRANSFER TO ZAP

*QEX:L          - TRANSFER TO LILO

*QEX:L NAME     - TRANSFER TO LILO WITH COMMANDS FROM FILE 'NAME'

*QEX:A NAME1 NAME2    - ASSEMBLE MEDELA FILE 'NAME1' INTO 'NAME2'

*QEX:D     - LIST FILES UNDER USER ACCOUNT

*QEX:DI    - LIST FILES UNDER ANOTHER ACCOUNT

*QEX:M     - LIST MEMOS UNDER ACCOUNT 7
```

```
*QEX:M NAME     - PRINT MEMO CALLED 'NAME' ON USER ACCOUNT OR 7

*QEX:P NAME     - GIVE PAGINATED LISTING OF FILE 'NAME' ON USER
                  ACCOUNT OR 7

*QEX:DEL NAME     - DELETE FILE 'NAME'

*QEX:REN NAME1 NAME2     - RENAME FILE 'NAME1' AS 'NAME2'

*QEX:ZERO     - DELETE ALL FILES ON USER ACCOUNT

*QEX:W     - LIST CURRENT USERS OF SYSTEM

*QEX:CTS     - RUN CLINICAL TRIALS SYSTEM

*QEX:MAR     - RUN MARKET RESEARCH SYSTEM

*QEX:MAN     - RUN MANAGEMENT TEST DEMONSTRATION
```

13.1 OTHER RELEVANT MANUALS

TSS8 SYSTEM USERS HANDBOOK

EDITOR MANUAL

ZAP MANUAL - TSS8 ASSEMBLER

LILO MANUAL - TSS8 LINKING LOADER

COPY MANUAL - TSS8 FILE UTILITIES