CHAPTER 5

Programs for setting up document test collections

The following files are used in the system (and are not necessarily disc resident):

MFP1.S.PROG - a pds of program texts.
MFP1.S.RUN - a pds of phoenix commands to run each program.
MFP1.S.LMOD - a load module library of programs.
MFP1.S.TEXTS - a pds of data files used as defaults by the programs.

To invoke the members of MFP1.S.RUN as a private library, type

```
LIBRARY MFP1.S.LMOD:COM0
```

Commands in the library work from a FROM to a TO file, and the defaults for FROM and TO are %C and %O respectively. TO is set up as /LARGE.
(This is phoenix jargon, phoenix being the command language in use on the IBM 370/165 at Cambridge.)

STORE should be adjusted upwards from the usual default of 120 for online work when substantial sorting or data storage is required by the programs. When the programs do run out of store, the messages give a good indication of how much was really needed.

DECHAR

e.g. DECHAR FROM .VASWANI.SRCE1
DECHAR FROM &S TO &T

This takes the original text of a standard test collection (FROM), and replaces all non-letters in the document abstracts by space. Lower case letters are forced to upper case.

The syntax of the input is validated, in that document numbers must be 1,2,3, ... in turn, documents must be terminated with

/<newline>

and the whole collection with

/<newlines><endoftext>

With the FROM file:

```
1
Compact memories have flexible capacities. A digital data storage
system with capacity up to 24000 bits and random and or sequential access
is described.
   /
2
An electronic analogue computer for solving systems of linear equations.
Mathematical derivation of the operating principle and stability
conditions for a computer consisting of amplifiers.
   /

....

100
```

```
Satellite observations of electrons artificially injected into the
geomagnetic field. The geomagnetically trapped electrons resulting
from the high altitude nuclear detonations of the ARGUS experiment
have been observed on four radiation detectors in satellite explorer.
The measurements for several satellite passes through the ARGUS shells
are described and the significance of the results is summarized.
   /
  /
```

DECHAR produces the TO file:

```
1
COMPACT MEMORIES HAVE FLEXIBLE CAPACITIES  A DIGITAL DATA STORAGE
SYSTEM WITH CAPACITY UP TO       BITS AND RANDOM AND OR SEQUENTIAL ACCESS
IS DESCRIBED
   /
2
AN ELECTRONIC ANALOGUE COMPUTER FOR SOLVING SYSTEMS OF LINEAR EQUATIONS
MATHEMATICAL DERIVATION OF THE OPERATING PRINCIPLE AND STABILITY
CONDITIONS FOR A COMPUTER CONSISTING OF AMPLIFIERS
   /

....

100
SATELLITE OBSERVATIONS OF ELECTRONS ARTIFICIALLY INJECTED INTO THE
GEOMAGNETIC FIELD  THE GEOMAGNETICALLY TRAPPED ELECTRONS RESULTING
FROM THE HIGH ALTITUDE NUCLEAR DETONATIONS OF THE ARGUS EXPERIMENT
HAVE BEEN OBSERVED ON FOUR RADIATION DETECTORS IN SATELLITE EXPLORER
THE MEASUREMENTS FOR SEVERAL SATELLITE PASSES THROUGH THE ARGUS SHELLS
ARE DESCRIBED AND THE SIGNIFICANCE OF THE RESULTS IS SUMMARIZED
   /
/
```

DESTOP

e.g. DESTOP WITH .OWN:STOPLIST
DESTOP FROM &T TO .VAS.A0

This removes from the text of FROM the words supplied in a given stop list (WITH), together with words consisting of only one or two letters. The stop list may contain one or two letter words, but these are redundant. (Here and below words are defined as upper case letter sequences bounded by non-letters.)

With the FROM file:

```
1
COMPACT MEMORIES HAVE FLEXIBLE CAPACITIES  A DIGITAL DATA STORAGE
SYSTEM WITH CAPACITY UP TO       BITS AND RANDOM AND OR SEQUENTIAL ACCESS
IS DESCRIBED
   /
2
AN ELECTRONIC ANALOGUE COMPUTER FOR SOLVING SYSTEMS OF LINEAR EQUATIONS
MATHEMATICAL DERIVATION OF THE OPERATING PRINCIPLE AND STABILITY
CONDITIONS FOR A COMPUTER CONSISTING OF AMPLIFIERS
   /

....

100
SATELLITE OBSERVATIONS OF ELECTRONS ARTIFICIALLY INJECTED INTO THE
GEOMAGNETIC FIELD  THE GEOMAGNETICALLY TRAPPED ELECTRONS RESULTING
FROM THE HIGH ALTITUDE NUCLEAR DETONATIONS OF THE ARGUS EXPERIMENT
HAVE BEEN OBSERVED ON FOUR RADIATION DETECTORS IN SATELLITE EXPLORER
THE MEASUREMENTS FOR SEVERAL SATELLITE PASSES THROUGH THE ARGUS SHELLS
ARE DESCRIBED AND THE SIGNIFICANCE OF THE RESULTS IS SUMMARIZED
   /
/
```

and the WITH file:

```
    A, ABOUT, ABOVE, ACROSS, AFTER, AFTERWARDS, AGAIN
    AGAINST, ALL, ALMOST, ALONE, ALONG, ALREADY, ALSO
    ALTHOUGH, ALWAYS, AMONG, AMONGST, AN, AND, ANOTHER
    ANY, ANYHOW, ANYONE, ANYTHING, ANYWHERE, ARE, AROUND
    AS, AT, BE, BECAME, BECAUSE, BECOME, BECOMES
    BECOMING, BEEN, BEFORE, BEFOREHAND, BEHIND, BEING, BELOW
    BESIDE, BESIDES, BETWEEN, BEYOND, BOTH, BUT, BY
    CAN, CANNOT, CO, COULD, DOWN, DURING, EACH
    EG, EITHER, ELSE, ELSEWHERE, ENOUGH, ETC, EVEN
    EVER, EVERY, EVERYONE, EVERYTHING, EVERYWHERE, EXCEPT, FEW
    FIRST, FOR, FORMER, FORMERLY, FROM, FURTHER, HAD
    HAS, HAVE, HE, HENCE, HER, HERE, HEREAFTER
    HEREBY, HEREIN, HEREUPON, HERS, HERSELF, HIM, HIMSELF
    ....
```

DESTOP produces as TO file:

```
    1
    COMPACT MEMORIES FLEXIBLE CAPACITIES  DIGITAL DATA STORAGE
    SYSTEM CAPACITY       BITS RANDOM SEQUENTIAL ACCESS
    DESCRIBED
       /
    2
    ELECTRONIC ANALOGUE COMPUTER SOLVING SYSTEMS LINEAR EQUATIONS
    MATHEMATICAL DERIVATION OPERATING PRINCIPLE STABILITY
    CONDITIONS COMPUTER CONSISTING AMPLIFIERS
       /

    ....

    100
    SATELLITE OBSERVATIONS ELECTRONS ARTIFICIALLY INJECTED
    GEOMAGNETIC FIELD  GEOMAGNETICALLY TRAPPED ELECTRONS RESULTING
    HIGH ALTITUDE NUCLEAR DETONATIONS ARGUS EXPERIMENT
    OBSERVED FOUR RADIATION DETECTORS SATELLITE EXPLORER
    MEASUREMENTS SATELLITE PASSES ARGUS SHELLS
```

```
DESCRIBED SIGNIFICANCE RESULTS SUMMARIZED
  /
 /
```

The default value of WITH is

```
MFP1.S.TEXTS:STOPLIST
```

VOCAB

e.g.
```
VOCAB FROM .VAS.A0 TO .VAS.VOCAB STORE 400
VOCAB TO &VOC
```

This sorts the words in FROM and sends to TO a vocabulary list in the form:

```
ABSORPTION
ACCELERATION
ACCESS
ACCOMPANIED
ACCOUNT
ACCURACY
ACHIEVED
ACOUSTIC
ADDER
ADJUSTABLE
ADVERSELY
AFFECT
AFFECTING
AGREEMENT
AIR
ALLEN
ALTITUDE
AMERICAN
```

```
AMMONIA
AMPLIFICATION
....
```

STEM

e.g. STEM FROM &VOC TO &SVOC

STEM is a simple suffix stripping program. The output has the form:

```
ABSORPT                 ABSORPTION
ACCELER                 ACCELERATION
ACCESS                  ACCESS
ACCOMPAN                ACCOMPANIED
ACCOUNT                 ACCOUNT
ACCURAC                 ACCURACY
ACHIEV                  ACHIEVED
ACOUST                  ACOUSTIC
ADDER                   ADDER
ADJUST                  ADJUSTABLE
ADVERS                  ADVERSELY
AFFECT                  AFFECT
AFFECT                  AFFECTING
AGREEMENT               AGREEMENT
AIR                     AIR
ALLEN                   ALLEN
ALTITUD                 ALTITUDE
AMERICAN                AMERICAN
AMMONIA                 AMMONIA
AMPLIF                  AMPLIFICATION
AMPLIF                  AMPLIFIER
AMPLIF                  AMPLIFIERS
```

```
AMPLITUD                AMPLITUDE
ANALOGU                 ANALOGUE
ANALYS                  ANALYSED
ANALYS                  ANALYSER
ANALYS                  ANALYSIS
....
```

Each word in the FROM file is output on a separate line preceded by a character string derived from the word, which is used to bring words together into conflation groups. The character string is printed in a field of width 24, and has a maximum size of 22 characters. (If necessary, characters are chopped out of the middle to bring it down to size.) The TO file should be sorted to bring the words into their conflation groups, e.g.

```
STEM FROM &VOC
SM
```

DICMAT

e.g.
```
DESLASH FROM .VASVOC.TERMS TO &U
DICMAT FROM .VAS.A0 TO .VAS.B0 WITH &U

DICMAT WITH .VTERMS STORE 250
```

The WITH file (the TO output from TERMNOS) consists of every word in the text of the FROM file, arranged in alphabetical order, and followed by an integer. DICMAT replaces each word in the FROM file by its corresponding integer, and sends the result to TO.

Words not in WITH but in FROM are printed out as not being in the dictionary, but are otherwise ignored.

With the FROM file:

```
1
COMPACT MEMORIES FLEXIBLE CAPACITIES  DIGITAL DATA STORAGE
SYSTEM CAPACITY       BITS RANDOM SEQUENTIAL ACCESS
DESCRIBED
   /
2
ELECTRONIC ANALOGUE COMPUTER SOLVING SYSTEMS LINEAR EQUATIONS
MATHEMATICAL DERIVATION OPERATING PRINCIPLE STABILITY
CONDITIONS COMPUTER CONSISTING AMPLIFIERS
   /

....

100
SATELLITE OBSERVATIONS ELECTRONS ARTIFICIALLY INJECTED
GEOMAGNETIC FIELD  GEOMAGNETICALLY TRAPPED ELECTRONS RESULTING
HIGH ALTITUDE NUCLEAR DETONATIONS ARGUS EXPERIMENT
OBSERVED FOUR RADIATION DETECTORS SATELLITE EXPLORER
MEASUREMENTS SATELLITE PASSES ARGUS SHELLS
DESCRIBED SIGNIFICANCE RESULTS SUMMARIZED
   /
/
```

and the WITH file:

```
ABSORPTION     1
ACCELERATION     2
ACCESS     3
ACCOMPANIED     4
ACCOUNT     5
ACCURACY     6
ACHIEVED     7
ACOUSTIC     8
ADDER     9
ADJUSTABLE     10
ADVERSELY     11
```

```
    AFFECT      12
    AFFECTING      12
    AGREEMENT      13
    AIR      14
    ALLEN      15
    ALTITUDE      16
    AMERICAN      17
    AMMONIA      18
    AMPLIFICATION      19
    ....
```

DICMAT produces as TO file:

```
    1
    105 424 268 82  189 161 687
    708 82 63 566 630 3
    174
       /
    2
    220 20 113 664 708 388 230
    415 173 479 538 683
    117 113 128 19
       /

    ....

    100
    613 471 220 37 340
    294 263  294 737 220 600
    317 16 468 180 36 243
    471 280 561 178 613 245
    422 613 500 36 637
    174 646 600 699
       /
    /
```

e.g. BSORT OPT SF
BSORT FROM .VAS.B0 TO .VAS.B1 OPT SN

This takes a file in ab-form (FROM), and adjusts the b's for each a. 'ab-form' means

```
a
b b ... b /
a
b b ... b /
...
a
b b ... b /
/
```

where the a's and b's are integers.

The OPT parameter may contain S, F, B and N.

S causes the b's to be sorted in ascending order.
N causes duplicate b's (i.e. a b equal to the previous b) to be discarded.
F causes a list of b's to be replaced by a single b with frequency count in the output.
B causes the output to be "brief", that is, multiple spaces are reduced to a single space. This is recommended for very large data collections.

The N and F options take effect after sorting. N and F together cause all the frequency counts to be 1.

So if the FROM file is:

```
1
105 424 268 82  189 161 687
708 82 63 566 630 3
174
   /
2
220 20 113 664 708 388 230
415 173 479 538 683
117 113 128 19
   /

....

100
613 471 220 37 340
294 263  294 737 220 600
317 16 468 180 36 243
471 280 561 178 613 245
422 613 500 36 637
174 646 600 699
   /
/
```

OPT=S produces the output:

```
   1      3      63      82      82     105     161     174     189     268
        566     630     687     708 /

   2     19      20     113     113     117     128     173     220     230
        415     479     538     664     683     708 /


....

 100     16      36      36      37     174     178     180     220     220
        245     263     280     294     294     317     340     422     468
        471     500     561     600     600     613     613     613     637
```

```
          699    737 /

 /
```

OPT=SF produces the output:

```
      1      3       1      63       1      82       2     105       1     161       1
           174       1     189       1     268       1     424       1     566       1
           630       1     687       1     708       1 /

      2     19       1      20       1     113       2     117       1     128       1
           173       1     220       1     230       1     388       1     415       1
           479       1     538       1     664       1     683       1     708       1 /


    ....

    100    16       1      36       2      37       1     174       1     178       1
           180       1     220       2     243       1     245       1     263       1
           280       1     294       2     317       1     340       1     422       1
           468       1     471       2     500       1     561       1     600       2
           613       3     637       1     646       1     699       1     737       1 /

 /
```

and OPT=SN produces the output:

```
      1      3     63     82    105    161    174    189    268    424    566
           630    687    708 /

      2     19     20    113    117    128    173    220    230    388    415
           479    538    664    683    708 /


    ....

    100    16     36     37    174    178    180    220    243    245    263
```

```
        280    294    317    340    422    468    471    500    561    600
        613    637    646    699    737 /
/
```

RANK

e.g. RANK FROM .VAS.B1 TO .VAS.RANK
RANK STORE 250

This takes a file in ab-form (FROM) and sorts the different b's into descending order of frequency, producing on TO the list of b's.

TO has the typical form:

```
752   720   283    19    21    95   263   174   220   293
600   173   243   422   296   471   473   488   586   642
645   404   497   567   663   127   179   216   356   426
460   550   597   708    55    87   194   197   388   398
427   562   591   683   734   771    93   106   117   161
175   212   311   613   649   759    29    78   118   218
238   294   412   453   484   507   525   539   575   666
686   688   726   746    37   192   195   203   217   222
230   244   267   275   285   289   317   403   451   479
499   519   528   538   541   571   595   602   607   619
....
```

e.g. BMAP FROM .VAS.BO TO .VAS.CO WITH .VAS.RANK
BMAP TO &U WITH &RANK STORE 200

This takes a file in ab-form (FROM) and a separate ranking list of the b's (WITH) and produces on TO a mapped version of FROM in which each b is replaced by f(b), where b occurs as the f(b)-th number in the WITH file.

The purpose of this operation is to renumber the b's so that 1 is the most common term, 2 the next most common, and so on.

If FROM has the form:

```
1
105 424 268 82  189 161 687
708 82 63 566 630 3
174
   /
2
220 20 113 664 708 388 230
415 173 479 538 683
117 113 128 19
   /

....

100
613 471 220 37 340
294 263  294 737 220 600
317 16 468 180 36 243
471 280 561 178 613 245
422 613 500 36 637
174 646 600 699
   /
/
```

and WITH the form:

```
    752   720   283    19    21    95   263   174   220   293
    600   173   243   422   296   471   473   488   586   642
    645   404   497   567   663   127   179   216   356   426
    460   550   597   708    55    87   194   197   388   398
    427   562   591   683   734   771    93   106   117   161
    175   212   311   613   649   759    29    78   118   218
    238   294   412   453   484   507   525   539   575   666
    686   688   726   746    37   192   195   203   217   222
    230   244   267   275   285   289   317   403   451   479
    499   519   528   538   541   571   595   602   607   619
    ....
```

TO has the form:

```
      1   396   574   477   388   222    50   728    34   388   372
          291   694   333     8 /

      2     9   344   116   169    34    39    81   147    12    90
           94    44    49   116   205     4 /

    ....

    100    54    16     9    75   522    62     7    62   319     9
           11    87   341   601   437   353    13    16   137   163
          436    54   465    14    54   278   353   699     8   702
           11   735 /

 /
```

DEFREQ

e.g. DEFREQ FROM &A TO &B

This takes a file in abb-form, i.e. in ab-form but with the b's coming in pairs, and strips out the second of each pair of b's.

So with the FROM file:

```
      1      3      1     63      1     82      2    105      1    161      1
           174      1    189      1    268      1    424      1    566      1
           630      1    687      1    708      1 /

      2     19      1     20      1    113      2    117      1    128      1
           173      1    220      1    230      1    388      1    415      1
           479      1    538      1    664      1    683      1    708      1 /

    ....

    100    16      1     36      2     37      1    174      1    178      1
           180      1    220      2    243      1    245      1    263      1
           280      1    294      2    317      1    340      1    422      1
           468      1    471      2    500      1    561      1    600      2
           613      3    637      1    646      1    699      1    737      1 /

 /
```

DEFREQ produces as TO file:

```
      1      3     63     82    105    161    174    189    268    424    566
           630    687    708 /

      2     19     20    113    117    128    173    220    230    388    415
           479    538    664    683    708 /
```

```
....
100    16    36    37   174   178   180   220   243   245   263
      280   294   317   340   422   468   471   500   561   600
      613   637   646   699   737 /
/
```

INVERT

e.g. INVERT FROM .VAS.CO TO .VAS.XO STORE 350
INVERT STORE 400 OPT F200T300

This takes a file in ab-form (FROM) and produces an inverted file in ab-form (TO) in which each a,b pair of FROM corresponds to a b,a pair in TO.

So with the FROM file:

```
1    396   574   477   388   222    50   728    34   388   372
     291   694   333     8 /

2      9   344   116   169    34    39    81   147    12    90
      94    44    49   116   205     4 /

....
100   54    16     9    75   522    62     7    62   319     9
      11    87   341   601   437   353    13    16   137   163
     436    54   465    14    54   278   353   699     8   702
      11   735 /
/
```

INVERT gives a TO file:

```
      1      5      6      7      9     11     18     20     29     35     38
            46     47     50     51     61     68     72     89     93     99 /

      2      8     11     13     16     18     23     24     28     33     42
            45     53     55     56     60     63     63     63     65     67
            94 /

    ....

    783     84     84 /

 /
```

A limitation of INVERT is that it can only cope with one storeful of material at a time. To get over this, the range of values of b may be restricted by settings in the OPT parameter. If the OPT string is FxTy (F and T stand for 'from' and 'to'), the TO file only contains those inverted b's for which

    x <= b < y.

If y is set, the TO file will not contain the terminating solidus character, which means that a valid inverted file can be set up by concatenations.

```
    e.g.  DELETE &INV
          CURRENT .VAS.CO
          INVERT TO &INV/MOD STORE 400 OPT T300
          INVERT TO &INV/MOD STORE 400 OPT F300T600
          INVERT TO &INV/MOD STORE 400 OPT F600
```

INVERT2

e.g. INVERT2 FROM .VAS.CO TO .VAS.XO STORE 350

This has the same spec as INVERT, but does not have the limitation of only one storeful at a time, and so is to be preferred for large data collections. It uses the IBM sort-merge utility, and involves a number of passes over the data. INVERT2 has no OPT parameter.

IMAP

e.g. IMAP FROM &A TO &B WITH &RANK

This is like BMAP (q.v.) but is intended for use with output AND of TERMNOS. Each number b at the beginning of the line in FROM is mapped to f(b), where b occurs as the f(b)-th number in the WITH file.

The numbers in TO are right justified in a field of width 6, e.g.

```
   331 ABSORPTION
   332 ACCELERATION
   333 ACCESS
   334 ACCOMPANIED
   335 ACCOUNT
   181 ACCURACY
   182 ACHIEVED
   336 ACOUSTIC
   337 ADDER
   338 ADJUSTABLE
   339 ADVERSELY
   183 AFFECT
   184 AGREEMENT
   340 AIR
```

```
   185 ALLEN
   341 ALTITUDE
   342 AMERICAN
   343 AMMONIA
     4 AMPLIFICATION
   344 ANALOGUE
....
```

SM

e.g. SM FROM &B OPT 1,6

This runs the IBM sort-merge utility and sorts the records of FROM to the file TO, ordering them by the EBCDIC collating order of the characters in columns m to n inclusive, where m,n is the value of the OPT parameter. (By default m=1 and n=79.)

So with OPT=1,6 and FROM in the form:

```
   331 ABSORPTION
   332 ACCELERATION
   333 ACCESS
   334 ACCOMPANIED
   335 ACCOUNT
   181 ACCURACY
   182 ACHIEVED
   336 ACOUSTIC
   337 ADDER
   338 ADJUSTABLE
   339 ADVERSELY
   183 AFFECT
   184 AGREEMENT
   340 AIR
```

```
      185 ALLEN
      341 ALTITUDE
      342 AMERICAN
      343 AMMONIA
        4 AMPLIFICATION
      344 ANALOGUE
    ....
```

TO would have the form:

```
        1 USE
        2 THEORETICAL
        3 FREQUENCIES
        4 AMPLIFICATION
        5 ANALYSED
        6 CIRCUIT
        7 FIELD
        8 DESCRIBED
        9 ELECTRON
       10 GENERAL
       11 RESULT
       12 DERIVATION
       13 EXPERIMENT
       14 MEASURED
       15 GIVEN
       16 OBSERVATION
       17 OBTAIN
       18 OSCILLATION
       19 RELATED
       20 SHOWN
    ....
```

Before sorting, FROM is converted to a file with FB format and LRECL=80. This means that the records to be sorted must not exceed 80 characters in length. The output from sort-merge is then filed to TO, so that TO does not have to be FB.

e.g. TERMNOS FROM &S TO &T
TERMNOS TO &T AND &U

The FROM file should be the sorted output from STEM. The TO file consists of the words of vocabulary followed by an integer which gives a count of the conflation class. The AND file, if present, consists of the first word out of each conflation class preceded by its conflation class number. By default AND goes to %DUMMY.

So with the FROM file in the form:

ABSORPT	ABSORPTION
ACCELER	ACCELERATION
ACCESS	ACCESS
ACCOMPAN	ACCOMPANIED
ACCOUNT	ACCOUNT
ACCURAC	ACCURACY
ACHIEV	ACHIEVED
ACOUST	ACOUSTIC
ADDER	ADDER
ADJUST	ADJUSTABLE
ADVERS	ADVERSELY
AFFECT	AFFECT
AFFECT	AFFECTING
AGREEMENT	AGREEMENT
AIR	AIR
ALLEN	ALLEN
ALTITUD	ALTITUDE
AMERICAN	AMERICAN
AMMONIA	AMMONIA
AMPLIF	AMPLIFICATION
AMPLIF	AMPLIFIER

AMPLIF	AMPLIFIERS
AMPLITUD	AMPLITUDE
ANALOGU	ANALOGUE
ANALYS	ANALYSED
ANALYS	ANALYSER
ANALYS	ANALYSIS
....	

TO would have the form:

ABSORPTION 1
ACCELERATION 2
ACCESS 3
ACCOMPANIED 4
ACCOUNT 5
ACCURACY 6
ACHIEVED 7
ACOUSTIC 8
ADDER 9
ADJUSTABLE 10
ADVERSELY 11
AFFECT 12
AFFECTING 12
AGREEMENT 13
AIR 14
ALLEN 15
ALTITUDE 16
AMERICAN 17
AMMONIA 18
AMPLIFICATION 19
AMPLIFIER 19
AMPLIFIERS 19
AMPLITUDE 20
ANALOGUE 21
ANALYSED 22
ANALYSER 22
ANALYSIS 22

....

and AND would have the form:

```
1 ABSORPTION
2 ACCELERATION
3 ACCESS
4 ACCOMPANIED
5 ACCOUNT
6 ACCURACY
7 ACHIEVED
8 ACOUSTIC
9 ADDER
10 ADJUSTABLE
11 ADVERSELY
12 AFFECT
13 AGREEMENT
14 AIR
15 ALLEN
16 ALTITUDE
17 AMERICAN
18 AMMONIA
19 AMPLIFICATION
20 AMPLITUDE
21 ANALOGUE
22 ANALYSED
....
```

Illustrative example - setting up the VASWANI test collection

Setting up the collection itself was done as follows. The numbers in brackets indicate the track sizes of the created output files. The store size is 120K unless otherwise indicated.

```
DECHAR FROM .TEXT.N6 TO .TEXT.NORMED (290)                          58 secs

DESTOP FROM .TEXT.NORMED TO .TEXT.STOPPED (229)                     25 secs
  479163 words read
  99115 stopwords removed
  104358 short words removed

VOCAB FROM .TEXT.STOPPED TO .VOCAB (13)                  2 min 19 secs, 400K
  No of records read 73524
  No of tokens 275690
  No of types 11712

STEM FROM .VOCAB                                                     9 secs
  11712 words read
SM                                                                   9 secs, 150K
TERMNOS TO .TERMNOS (18) AND .TEMP.TERMDICT (11)                     4 secs
  11712 different words read
  7491 different terms read

DICMAT FROM .TEXT.STOPPED WITH .TERMNOS TO .TEMPDOCS.AB (143)
                                                         1 min 42 secs, 300K
275690 words read
  440 (machine) words of unused workspace

BSORT FROM .TEMPDOCS.AB OPT SN                           1 min  2 secs
    11429 units read
RANK TO .TERMRANK (5)                                               23 secs, 200K
  Max term = 7491
  Max frequency = 2511
```

```
  11429 docs read
  7491 terms output

IMAP FROM .TEMP.TERMDICT WITH .TERMRANK                          4 secs, 200K
  7491 lines read
SM TO .TERMDICT (11)                                             6 secs, 200K

BMAP FROM .TEMPDOCS.AB WITH .TERMRANK TO .DOCS.AB (197)         48 secs, 130K
  7491 terms read from WITH file
  Maximum term was 7491
  11429 docs read

BSORT FROM .DOCS.AB TO .DOCS.ABS (109) OPT BS                    57 secs
  11429 units read

BSORT FROM .DOCS.ABS TO .DOCS.ABF (141) OPT BF                   42 secs
  11429 units read

DEFREQ FROM .DOCS.ABF TO .DOCS.ABN (165)                         40 secs
  11429 docs read

INVERT2 FROM .DOCS.AB TO .TERMS.ABS (137)                3 min 24 secs, 400K

BSORT FROM .TERMS.ABS TO .TERMS.ABF (163) OPT BF                 53 secs
  7491 units read

DEFREQ FROM .TERMS.ABF TO .TERMS.ABN (163)                       51 secs
  7491 docs read
```

Setting up the queries was done as follows. (The resources consumed here were small.)

```
DESTOP FROM .QUERIES.TEXT
  1086 words read
  146 stopwords removed
  208 short words removed
```

```
DICMAT WITH .TERMNOS                                                    300K
  FONT not in dict
  TRANSISTORISED not in dict
  SEND not in dict
  SEND not in dict
  DISCS not in dict
  INTERESTED not in dict
  NUMERIC not in dict
  OPTIMISING not in dict
  PRETREATMENT not in dict
  WISH not in dict
  WISH not in dict
  TRANSISTORISED not in dict
  RESTING not in dict
  732 words read
  440 (machine) words of unused workspace

(.QUERIES.TEXT was then edited so that

    DISCS        ->  DISC*
    INTERESTED   ->  INTERESTE**
    NUMERIC      ->  NUMERICAL
    RESTING      ->  REST***

the words DISC, INTEREST, NUMERICAL and REST being present in .TERMNOS, and
DICMAT was rerun.)

DICMAT WITH .TERMNOS                                                    300K
  FONT not in dict
  TRANSISTORISED not in dict
  SEND not in dict
  SEND not in dict
  OPTIMISING not in dict
  PRETREATMENT not in dict
  WISH not in dict
  WISH not in dict
  TRANSISTORISED not in dict
```

```
  732 words read
  440 (machine) words of unused workspace

BMAP WITH .TERMRANK                                              130K
  7491 terms read from WITH file
  Maximum term was 7491
  100 docs read

BSORT TO .QUERIES.AB OPT SN
  100 units read
```

Appendix A. Other useful programs.

SCALE

e.g. SCALE TO &B AND &SCALE

This takes a file in ab-form (FROM) and produces on TO a version of the FROM file with the a's forming a simple ascending sequence 1,2,3, ... The AND file (which must be present) contains a simple list of the original a's. This file can be used for mapping b's with the BMAP program, e.g.

```
SCALE FROM .TERMS.A1 TO .TERMS.B1 AND &SCALE
BMAP FROM .DOCS.A1 TO .DOCS.B1 WITH &SCALE STORE 190
BMAP FROM .QS.A1 TO .QS.B1 WITH &SCALE STORE 190
```

RELS

e.g. RELS FROM .A TO .B

This puts the text form of a test collection (the output of DECHAR) into 3-tuple relation form for input to the CODD database. The columns are:

DOCNO WORDNO WORD

where WORD is the WORDNOth word in the document with number DOCNO.