IPSHU研究報告シリーズ

研究報告 No. 6

テクスト語彙処理プログラムLEX

松 尾 雅 嗣

広島大学平和科学研究センター

THE INSTITUTE FOR PEACE SCIENCE,
HIROSHIMA UNIVERSITY

Mar. 1982

広島大学平和科学研究センター

〒730 広島市中区東千田町1丁目1番89号

目　　次

I　LEX の特長と概要

LEX（program package for lexical analyses of a text）は，文献，文書等，所謂テクストの単語や行に関する基本的諸表や索引をコンピューターによって作成することを主目的とするテクスト処理のためのソフトウェアである。本稿は，LEX の開発報告であるとともに，記述をできる限り利用者の立場に即して行うことにより，利用の手引たることも併せ意図したものである。

I．1　LEX の特長

LEXはテクストの語彙に関して多面的な処理が可能な多目的プログラムであることを最大の特長とするが，次のような特長をも併せもっている。

a　データの入力，コンピューターへの指示等が簡略化されており，利用がきわめて容易である。

b　大量データも扱えること。即ち，最大 999999 語までのテクストを扱える。

c　テクストの語彙に関して多様な処理が可能であるばかりでなく，同一種の処理についてもシステムが容易したオプションを適宜選択することにより一層多様な結果が得られる。

d　データの置換・選別機能があり，テクストの特定の部分，テクスト中の特定の単語(群)についての処理が可能である。

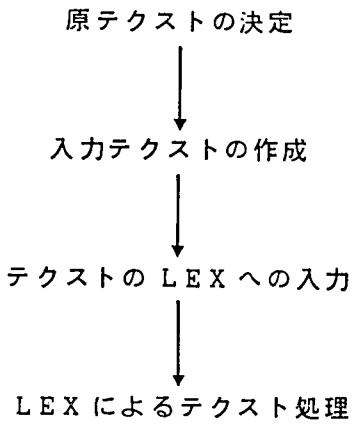
e　処理結果を外部記憶媒体に出力する機能があり，LEX での処理結果を容易に他のプログラムへ渡すことができる。

f　処理の対象となるテクストに関しては，それが文字列データから構成されている(という形に変換できる）という制約しか課せられておらず，ほとんどどのような形のテクストでも処理できる。

I．2　LEX によるテクスト処理の概要

LEXは前述のようにテクスト中の単語や行に関する様々な索引，表などを作成するプログラムであるが，LEX を用いてテクスト中の単語に関する具体的な処理結果，例えば単語の頻度順リスト，を得ることを，以下 LEX によるテクスト処理と称することに

する。LEXを利用してテクストを処理するためには，幾つかの準備段階が必要である。準備段階から，LEXによるテクスト処理に至る流れは，図式的には次のようになる。



利用者はまず，単行本，論文，文書，メモ，テープ等に記録された，分析対象となるテクスト(群)を選定しなければならない。選定されたテクスト(群)あるいは現実のテクストの一部を**原テクスト**と呼ぶ。

次の段階では，この原テクストを，LEXが処理できる形に加工・変換しなければならない。このステップ，即ち**入力テクスト**の作成，には，次のような作業が含まれる。

a　カード，磁気ディスク，磁気テープ等，コンピューターで処理できる記憶媒体上に原テクストを移しかえる作業，即ち，原テクストの機械可読化作業。

b　原テクストの構造をLEXで処理可能な形式に変換する作業。例えば，行，頁，章等のテクスト構成要素の境界を入力テクスト中で表示し，必要に応じて識別のための数値や名称を与える作業。

c　作成された，あるいは作成中の入力テクストの点検・修正作業。

次の段階では，このようにして作成された入力テクストがLEXシステムに入力される。LEXは入力テクストを読み込んで，後のテクスト処理に便利な形に加工されたテクスト（**LEXテクスト・ファイル**）を磁気ディスク，または磁気テープ上に作成する。またこのとき，利用者が指定すれば，入力テクストをほぼそのままの形で記憶した**LEX行ファイル**も作成される。テクストがLEXに入力され，LEXテクスト・ファイルが

作成されれば，これ以後LEXによるテクスト処理は自由にしかも何度でも行うことができる。LEXはLEXテクスト・ファイルと必要に応じてLEX行ファイルを読み込んでテクスト処理を行うから，これ以後は，データである入力テクストをLEXに与える必要はなくなる。

LEXでは，テクストの入力も含め，すべての処理が，**LEX文**と称される指示を与えることによって行われる。テクスト処理の種類，様々な機能やオプションの選択もすべてこのLEX文によって行われる。利用者は，目的に応じたLEX文を与えて処理結果を得るが，LEX文は一回の処理（正確にはひとつのジョッブ・ステップ）にひとつしか与えることができない。しかし，まったく同一の処理であれ，異なった処理であれ，同一のテクストを何度処理してもよい。

ごく簡単なLEX文の例を幾つか挙げてみよう。今，テクスト中の見出語* の頻度付きリストが得たいのであれば，LEXでは出力される見出語の配列に関し，幾つかの形式が可能である。

＊本稿では単語に関して厳密な区別が必要なときには，延べ語数（異なり語数），即ちtokenを問題にするのであれば**単語**，種類，即ちtypeを問題にするのであれば**見出語**，を用いる。

問題のリストを得るためには，次の3つのLEX文のうち一回のジョッブ・ステップ* につきひとつを選択して与えればよい。ふたつ以上が必要であれば，異なったジョッブ・ステップで与えればよい。

＊コンピューターが行う一回分の仕事（ジョッブ）は，少なくともひとつの仕事の区切り（ジョッブ・ステップ）から成る。

＜例 1. 1＞
```
FREQ
```

＜例 1. 2＞
```
ALPHA
```

＜例 1. 3＞
```
BACK
```

出力される頻度付きの見出語リストの配列は，例1.1であれば頻度上昇順，例1.2であれば見出語のアルファベット（アイウエオ）順，例1.3であれば見出語の綴字逆順(見出語の末尾からのアルファベット順）となる。見出語の頻度下降順のリストが必要であれば，例1.1に下降順のオプションを指定して，例1.4のようにすればよい。

```
＜例 1.4＞
FREQ
ORDER=D
```

また，個々の見出語ではなく，見出語の出現頻度の度数分布に関心があれば，例1.1で別のオプションを指定して，

```
＜例 1.5＞
FREQ
TYPE=T
```

という LEX 文を与えれば見出語の頻度の度数分布表が出力される。

例1.1～例1.5のうちのいずれを選択することもできるし，またこのうちの2種以上の出力が必要であれば，別個のジョッブ（またはジョッブ・ステップ）として実行すればよい。また同一の処理を別個のジョッブ（またはジョッブ・ステップ）として何度実行してもよい。*

*ただし，この場合には，一回の出力を磁気ディスクに出力し，それを繰返し印刷するか，計算機システムの出力ファイルのオペランドで結果を複数回印刷する指示をしたほうが経済的である。

Ⅰ．3　LEXの主な出力

LEXの主な出力は次の通りである。

a　テクストの印刷

単語の印刷

LEX行の印刷

素データの印刷

b　単語のリスト

頻度順（上昇順または下降順）リスト

アルファベット（アイウエオ）リスト

綴字逆順アルファベット順リスト

c　単語列のリスト

頻度順（上昇順）リスト

アルファベット順リスト

d　LEX行のリスト

頻度順（上昇順）リスト

アルファベット順リスト

e　索　引

単語索引

用例索引（KWOCまたはKWIC）

文字列用例索引

単語列索引

単語列用例索引（KWOC形式）

LEX行索引

f　共出現関係

共出現素データ出力

共出現マトリックス

共出現リスト

g　そ の 他

テクスト構成要素の長さに関する統計

識別値リスト

エントロピーの計算

単語頻度の度数分布表

Ⅱ　LEXの実行

LEXを利用してテクストの処理を行うときには，まずどのような処理をすべきかを，前章に与えた例のような形でLEXに対して指示しなければならない。LEXはこの指示に従ってテクストを処理し，その結果を出力（通常は印刷）する。

本章ではLEXの実行に必要な指示について概説し，またLEX実行時のエラーについても述べる。

Ⅱ．1　LEX文とその構造

LEXによるテクスト処理に際しては，厳密には性質の異なった2種類の指示が必要である。ひとつは，計算機システムそのものに対する指示であり，通常（job control language，略称JCL）と呼ばれる。JCLは，コンピューターを利用するときには常に必要とされるものであって，LEX利用時に固有のものではない。また，JCLは機種，設置機関によって異なる。JCLについては第XV章で扱う。

他のひとつの指示は，LEXシステムに対する指示であり，前述のようにLEX文と呼ばれる。ここでは，以下LEX文の構造と入力方法について述べる。

LEX文は一般に先頭から順に次の5つの要素から構成される。

実行命令
作業領域定義命令
ファイル定義命令
データ置換・選別命令
明細指示

この5つの要素はそれぞれ最終的には**LEXステートメント**と呼ばれる最小単位から構成される。

この5つの要素のうち，**実行命令**は，どのような処理を行うかをLEXシステムに指示するものである。実行命令はLEX文の先頭にひとつだけ与えることができる。前章の例1.1～1.5では，FREQ，ALPHA，BACKがそれぞれ実行命令である。

実行命令を除く他の4つの要素の任意性，必要性あるいは使用可能性はすべて実行命令の如何によって決まる。例えば，作業領域定義命令は，FREQ命令に関しては任意的であるが，PRINT WORD命令のもとでは使用不可能である。

作業領域定義命令は，単語や文字列のソートを行う実行命令，例えばFREQ 命令やALPHA命令，に続けて使用可能な任意的命令で，ソートに用いる作業領域の大きさを指定する。

ファイル定義命令は一般に任意的な命令で，LEXで用いる標準的な入出力ファイル以外のファイルを使用するときに与える命令である。入出力ファイルとファイル定義命令については第Ⅴ章で詳細を述べる。

データ置換・選別命令は，テクスト中の単語(群)を他の単語によって置換したり，テクストの特定の部分や，テクスト中の特定の単語(群)を処理の対象とするために与えることのできる，一般に任意的な命令の総称である。データ置換・選別命令はその機能によって5種類に分けられるが，一回のテクスト処理で，即ちひとつの実行命令のもとで，複数個を任意の組合せで用いることができる。

明細指示は，実行命令で指示したテクスト処理の詳細を規定するためのもので，その種類は実行命令によって異なる。大部分の明細指示には省略時の標準値が容易されており，明細指示が省略されたときには，この標準値に従った処理が行われる。

ここでやや複雑なLEX文の例を挙げておこう。次の例2.1は，テクスト中の“A„で始まる単語(“A„自体も含む)の下降順の頻度順リストを作り，その結果をライン・プリンターや端末ではなく，利用者の指定したファイルに書き出させるためのLEX文である。この例ではさらに作業領域として200Kバイトを与えている。

＜例2.1＞

```
FREQ                    …  ①
WORK=200                …  ②
WRITE△OUTPUT            …  ③
INCLUDE                 …  ④
POSITION=C△△B           …  ⑤
A                       …  ⑥
%END                    …  ⑦
ORDER=D                 …  ⑧
```

この例全体がひとつのLEX文であり，①～⑧のそれぞれがLEX文の最小構成単位であるLEXステートメントである。

①はFREQ命令で，言うまでもなくこれが実行命令である。

②は作業領域定義命令で，ここではソート用の作業領域を最大200 Kバイトとしている。②を省略すれば，作業領域としては最大100 Kバイトを取ったものと仮定される。

③は，ファイル定義命令のひとつのWRITE命令で，実行結果（この場合は見出語の頻度下降順リスト）をライン・プリンターや端末ではなく，利用者の指定した媒体に書き出すことを指示している。この例に限らず，この指示がなければ結果は通常ライン・プリンターか端末に出力される。

④～⑦は，データ置換・選別命令のひとつであるINCLUDE命令である。⑤で完全一致と前方一致が指定され，⑥で文字列“A„が与えられている。④～⑥は日常語で表現すれば，「“A„自体を含め，“A„で始まる単語を選べ」という命令に相等する。⑦はひとつのINCLUDE命令の終りを示す%ENDステートメントである。

⑧はFREQ命令に対するORDER明細指示で，見出語の出力順を頻度の下降順にすることを指示している。この明細指示は任意的で，もし省略されていれば，リストは頻度の上昇順になる。FREQ命令に対しては他に幾つかの明細指示があるが，この例ではすべて省略されている。

Ⅱ．2　LEX文の書式と記号の意味

前述のようにLEX文は実行命令によって異なるので，後に各実行命令ごとにそれに対応するLEX文の書式を与えて詳細を述べる。ここでは後に与えるLEX文の書式一般に関する規則と，書式中に用いられる記号の意味について述べる。

LEX文の最終的構成要素がLEXステートメントと呼ばれることは既に述べたが，本稿で与えるLEX文では，Ⅱ．4節で述べるセミコロン方式の例を除き，すべてひとつの行がひとつのLEXステートメントを表わす。但し，本稿で与えるLEX文のうち，

データ置換・選別命令
～　テクスト
～　データ

～　リスト

とあるときは，この例外で，複数の LEX ステートメントないしは入力テクストを示す。

LEX ステートメントはまた書式に与えられた順に与えなくてはならない。この規則に従わないときの結果は保証されない。

個々の LEX ステートメント中では以下の記号，略称等を用いる。

a　**ゴチック体**の文字は利用者が変更してはならない。これに対して，**ローマン体の英小文字，邦字**の部分には値，名称などを与えなくてはならない。例えば，

NAME＝ユニット名称

とあれば，「ユニット名称」の部分に適当な名称を与えなければならない。なお，このような値，名称については，通常，値の範囲，名称の文字数について制約がある。（本節 f，g 参照）。

b　△は空白（スペース）がひとつ，しかもただひとつだけ必要であることを示す。これに対し，△△は少なくともひとつの空白が必要であることを示す。

c　［　　］で囲まれたステートメント，ないしはステートメント中の要素は任意的である。即ち，必要に応じ，与えても与えなくてもよい。ただし，このことは結果が同じであるという意味ではない。また，任意的なステートメントや，ステートメント中の任意的な要素の選択は常に利用者の任意であるとは限らない。任意性が他のステートメントに依存することもあるからである。

d　｛　　｝で囲まれた要素は，そのうちのひとつ，しかもひとつだけを選択しなければならない。例えば，

MODE＝｛R，M，X｝

という書式のステートメントがあるとき，実際に与えるステートメントは次の 3 つのうちのひとつでなければならない。

```
MODE=R
MODE=M
MODE=X
```

ステートメント中の要素が{　}で囲まれており，この{　}がさらに広い[　]で囲まれているとき，{　}内の要素に——が施されることがある。この下線は，[　]で囲まれた部分が省略されたとき，下線を施された部分が省略時の標準値であることを示す。例えば，ステートメントの書式が，

```
READ[△{SYSIN, LXDATA}]
```

となっていれば，次のふたつのステートメントはまったく同じ意味をもつ。

```
READ
READ SYSIN
```

また，書式が，

```
[TYPE={A, B}]
```

であれば，このステートメント全体を省略するのと，

```
TYPE=A
```

というステートメントを与えるのとは，まったく同じ意味をもつ。

e　等号“＝„の直前にも，直後にも空白があってはならない。

f　(　)は，与えるべき数値の取りうる値の範囲を示すか，あるいは添字を付けて表わされた要素の与えうる個数の範囲を示す。例えば，

［ WORK=i ］　　（ $10 \geqq i$ ）

という書式のステートメントでは，iに与える数値は10以上でなくてはならない。また一連のステートメント，とくにリストと名付けられた一連のステートメントの書式が，

$$\begin{bmatrix} \text{単語}_1 \\ \text{単語}_2 \\ \cdots \\ \cdots \\ \text{単語}_n \end{bmatrix} \qquad (\ 1 \leqq n \leqq 100\ )$$

となっているとすれば，ここではひとつ以上，100 以下の単語を，それぞれひとつのステートメントとして与えなければならない。

g　＜　＞は数値を与えるべきステートメント（あるいはステートメント中の要素）が省略されたとき，システムが仮定する標準値を示す。例えば，

［ WORK=i ］　　（ $10 \geqq i$ ）　　＜$i = 100$＞

という書式のステートメントが省略されたとき，システムは $i = 100$ と仮定する。

h　□□□□□で囲まれた部分は省略形を示す。

i　二重丸括弧《　》で囲まれた要素は，そのうちのひとつ以上を任意の順序で選択しなければならない。このとき，要素間には少なくともひとつの空白が必要である。

実行命令によっては任意に選択した順序自体が意味をもつ場合ともたない場合がある。また，選択できる要素の数に制限のある場合もある。

j　混同を避けるため，書式やＬＥＸ文の例では英大文字のＯにはバーを付けて$\bar{\text{O}}$で示す。

Ⅱ．3　註釈ステートメント

本稿で掲げるLEX文に関しては以下特に断らないが，LEX文中には任意の数の**註釈ステートメント**を任意の位置に置くことができる。註釈ステートメントは，FORTRANやPL／Ⅰの註釈行と同様，テクストの処理には何ら関係しないが，他のLEXステートメントとともに印刷されるので，註釈，メモとして利用できる。註釈ステートメントの入力方法は次節で述べる一般のLEXステートメントのそれと同じであるが，そのほか次の規則に従う。

a　＜C＞で始まる。

b　ひとつの註釈ステートメントの長さは，接頭記号＜C＞と空白を含め，80文字（80バイト）以内である。

c　接頭記号に続けて，セミコロンを除くすべての文字，記号を用いることができる。

d　LEX文中のどの位置にあってもよいし，また何個連続していてもよい。しかし，利用者が与えた最後のLEXステートメントより前にあることが望ましい。

e　入力テクスト中で使うことは許されない。

註釈ステートメントを含むLEX文の例を次の例2.2に示す。このLEX文は，頻度10以上の見出語のアルファベット順リストを作成するものである。

```
＜例 2.2＞
ALPHA                                          …  ①
<C>   19 OCT.  1981                            …  ②
<C>   DATA:FINAL DOCUMENT, UNSSDI              …  ③
<C>   END OF COMMENTS                          …  ④
LBOUND=10                                      …  ⑤
```

②～④が註釈ステートメントである。

Ⅱ．4　LEXステートメントの入力

LEXでテクスト処理を行うためには，JCL（ジョッブ制御言語）とともにLEX

文を計算機システムに入力しなければならない。これは，カードやTSS端末を使って行われるが，このときのLEXステートメントの入力形式には次の2通りがある。

カード・イメージ（80バイト）方式

セミコロン方式

カード・イメージ方式では，パンチカード1枚にひとつのLEXステートメントを与える。このとき，LEXステートメント自体は，カードのどのカラムから始まっていてもよいが，ひとつのLEXステートメントが複数のカードにまたがってはならない。パンチカードについての本稿の記述は，この場合も含め，端末の1行にもすべて当てはまる。但し，カード・イメージ方式であれ，セミコロン方式であれ，端末の各行に計算機システムの与える行番号が付されているときには，最終的には**必ず削除しておかなければならない。**

この形式で入力するとき，空白のカードがあっても無視されるだけで，エラーにはならない。

セミコロン方式は，PL/Iのステートメントの区切りと同様のやり方で，セミコロンによってLEXステートメントの終りを示す方式である。この方式によるLEXステートメントの入力は以下の規則に従う。

a　各ステートメントの最初の文字，数字，記号と，先行するセミコロンの間にはいくつ空白があってもよく，また，各ステートメントの最後の文字，数字，記号と後続のセミコロンの間にはいくつ空白があってもよい。

b　セミコロンとセミコロンの間は80文字（80バイト）を越えてはならない。但し，複数個の連続したセミコロンは，ひとつのセミコロンと見なされる。

c　ひとつのLEXステートメントが複数のカード（端末行）にまたがってもよく，また1枚のカード（端末の1行）に複数のLEXステートメントがあってもよい。

d　LEX文の最後のステートメントの後にもセミコロンが必要である。

e　行番号付きの端末を使って入力するときには，行番号を消しておく必要がある。

本章の例2.1をこの方式で入力するならば，例えば次の例2.3のようになる。

```
＜例 2.3＞
FREQ  :  WORK=200 ;
WRITE  OUTPUT:
INCLUDE:POSITION=C△△B:A:%END:
ORDER=D;
```

実際にLEX文を入力するときには，上述のふたつの方式のいずれを用いてもよい。また，両方式を適宜併用することも可能である。

LEXは，与えられたLEXステートメントをひとつ読み込んでは印刷し処理するという形でLEX文を順次処理していく*。与えられたLEXステートメントは，結果の出力に先立ち，入力の方式如何に拘らず，カード・イメージ方式で，しかも第1カラムから与えられたのと同じ形式で印刷される。LEXステートメントに誤りがあれば処理は直ちに中断され，それ以後のLEXステートメントは印刷されないが，入力されたLEX文に誤りがなければ，LEX文の処理が正常に終了したことを示す，

```
***  END  ***
```

というメッセージが最後のLEXステートメントに続けて印刷され，以後LEX文の指示に従ったテクスト処理そのものが行われる。

＊LEXは実際には入力のデータ・ストリームを1文字づつ処理してLEXステートメントをひとつ抽出し，直ちに印刷する。そして次のステップでこのステートメントの構文チェックを行う。

II．5　LEX文のエラーとLEX実行時のエラー

LEX利用時に発生するエラーとしては大別して次の3種類がある。

計算機システムの検出するエラー

LEXシステムの検出するエラー

利用者の意図と異なる結果を生ずるという意味でのエラー

このうち，まず計算機システムの検出するエラーに関して言えば，この種のエラーに対しては計算機システムがエラー・メッセージを発行するので，これによって必要な対策なり修正なりを施せばよい。エラー・メッセージの内容，形式は計算機システムにより異なるので，その解釈なり，講ずべき対策については，プログラム相談員や然るべき担当者に尋ねるのが最善の策である。また，この種のエラーの大部分は JCL に関するものであるから，本稿の XV 章なり当該実行命令に関する注意事項も併せ参照すべきである。

これに対して LEX の検出するエラーには，LEX 文のエラーと，テクスト処理中のエラーがある。いずれの場合にも，LEX は，

*** ERROR DETECTED ***

という無愛想きわまりないエラー・メッセージだけを印刷して直ちに処理を中断する。このとき計算機システムの終了コード（リターン・コード）は正常終了の値（通常はゼロ）を示すので注意を要する。

LEX の検出するエラーはほとんどの場合 LEX 文の構文エラーである。このときには，エラーの検出された LEX ステートメントの直後に上記メッセージが印刷され，その後に LEX ステートメントが与えられていても印刷されない。従ってエラーが検出されたときまず心がけることは，エラー・メッセージの直前のステートメントに，記法，綴字，数値等の誤りがないか，義務的ステートメントや義務的要素の欠落がないかを点検することである。次には，直前のステートメントだけでなく，それ以前のステートメントも同様に点検する。エラーの真の原因はエラーの検出されたステートメントそのものというより，むしろそれ以前のステートメントにあることが少なくないからである。各実行命令の項に与えられた注意事項と説明を参照しながらこのような作業を行えば大部分のエラーの原因は容易に発見できるはずである。またデータ選別命令（後述）を用いた LEX 文では，データ選別の結果がテクストの空集合になっていないかをも調べておく必要がある。

LEX システムがテクスト処理中に検出するエラーはごく稀にしか起らないし，特定の実行命令に限られるのでここでは触れない。

計算機システムもLEXシステムも誤りを検出しない場合でも，利用者の意図した結果が得られないという意味でのエラーがありうる。特に，利用者の目からすれば明らかに誤りであっても，形式的にはLEX文の構文法や書式に叶っているためエラーとは見なされないケースがあることは念頭に置いておく必要がある。

この種のエラーには定まった対策はなく，利用者の与えたLEX文と当該実行命令に対応するLEX文の書式を丹念に比較するほかはない。

Ⅲ　入力テクストの作成とLEXへの入力

Ⅲ．1　テクストの構成要素

前述のようにLEXは入力テクストを再構成してLEXテクスト・ファイルと，必要に応じLEX行ファイルを作成する。LEXだけを利用してテクスト処理を行うのであれば，テクスト・ファイルや行ファイルの物理的属性，レコードの構造，ファイルの構造といった物理的実体的構造面に特に配慮する必要はない。しかしながら，LEXによるテクスト処理時と，その前段である入力テクスト作成時には,以下に述べるようなLEXの想定するテクストの，言わば論理的抽象的構造を理解しておく必要がある。それ故，本節ではテクストの論理的構造について述べ，次節ではこの構造を入力テクストにおいて，どのように具体化するかについて詳述する。

LEXの想定するテクスト，換言すればLEXの処理しうるテクストは，次の図3.1に示す論理的構造を有する。入力テクスト，LEXテクスト・ファイル（そしてLEX行ファイル）はいずれもこのテクスト構造に対応する構造，即ち，このテクスト構造に変換可能な物理的構造をもっている。

図 3.1

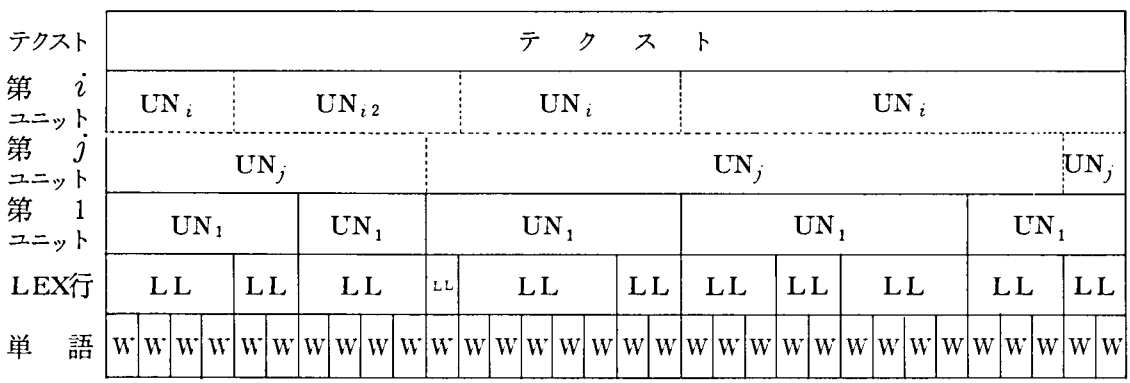


図 3.2

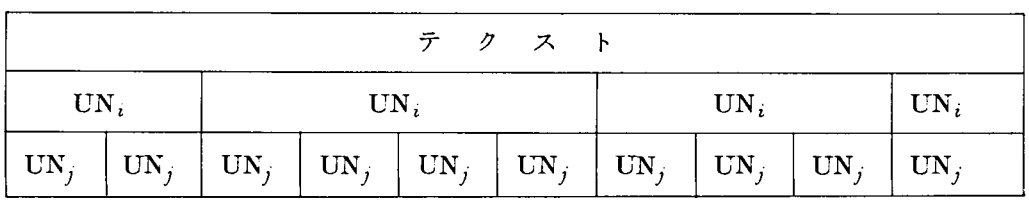


図 3. 3

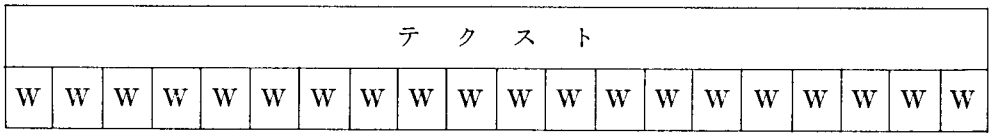
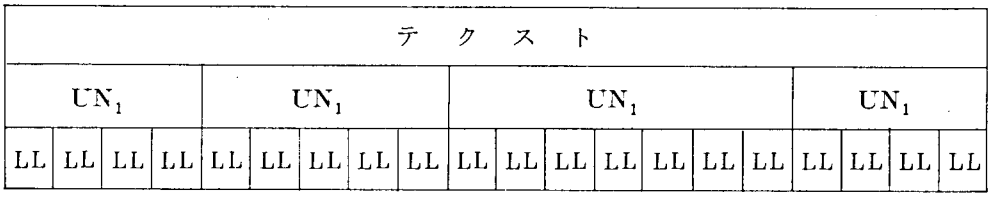


図 3. 4



点線は任意的要素を示す。
UN_x ……第xユニット
LL ……LEX行
W ……単 語

図 3. 1 に示すように，テクストは，**テクスト・ユニット**，**LEX 行**，**単語**という，レベルの異なった 3 種類の構成要素から成る。LEX は必要に応じてこの構成要素を単位としてテクストを処理できる。この意味で，テクスト・ユニット，LEX 行，単語は，LEXにおけるテクスト処理の単位でもある。

厳密に言えば，LEXは単語の構成要素である部分文字列（あるいは簡単に文字列）（ substring ）や，単語を構成要素とする単語列（word string）をも扱うことができる。

図 3. 1 からも明らかなように，LEX におけるテクストの最小単位は**単語**である。詳細は次節に譲るが，LEX で言う「単語」は，日常的な意味，あるいは言語学的な意味における「単語」である必要はなく，（利用者にとって有意味な）文字列あるいは記号列であればよい。

LEX 行は，単語を構成要素とする*テクスト構成要素で，入力テクストと密接な関係を有する。ごく大ざっぱに言えば，テクストを入力するときのカード 1 枚（または端末の 1 行）が LEX 行である。LEX 行と単語の関係は，このカード 1 枚に少なくともひとつの単語がパンチ（あるいは記録）されていると考えればよい。LEX 行はテクストの種類によっては，実質的に意味をもたない場合もある。（例 3. 6 参照）

＊ＬＥＸ行ファイルにおいては，Ⅲ．２で後述のように入力テクストがそのままの形で保存・記録されるので，ＬＥＸ行は単語と単語の境界標識のふたつの要素を構成要素とすることになる。

ここで「**構成要素とする**」というのは，上位要素と下位要素が一対一の場合を含め，一対多に対応することを意味する。具体的には，上位要素（この場合 LEX 行）の境界が，常に下位要素（この場合単語）の境界と一致すること，かつ逆は必ずしも真ではないことを意味する。

LEX では単語と LEX 行の他に，利用者が任意に指定できる**テクスト・ユニット**というテクストの構成要素を認める。テクスト・ユニットは後述の **INPUT** 命令によって最大４つまで指定できる。テクスト・ユニットとしては，行，頁，段落，章，文，節，項目，条といった一般的に用いられるテクストの構成単位を指定してもよいし，目的によっては，地の文と会話文の区別，話者や登場人物の区別のための言わばダミーのユニットを指定してもよいし，その他どのような構成単位を指定してもよい。

後者については，例3.3参照。この場合テクスト・ユニットは厳密な意味でのテクスト構成要素というより，むしろ，テクストの任意の部分を識別するための概念に近くなる。

LEX でテクスト・ユニットに課せられる制約は，如何なるテクスト・ユニットもＬＥＸ行と単語を構成要素とするものでなければならないということだけである。図3.1ですべてのテクスト・ユニットの境界が，常に LEX 行の境界，従って単語の境界，と一致していることに注目されたい。LEX 行と単語の関係について述べたのと同様大ざっぱな言い方をすれば，テクスト・ユニットとは，入力されるカード１枚（あるいは端末１行）もしくは，複数枚（複数行）から成るテクスト構成要素であると言える。

複数種のテクスト・ユニットがあるとき，このユニット相互の間には，一方が他方の構成要素であるという関係は必ずしも成立っている必要はない。勿論，図3.2に示すように，一方が他方の構成要素であってもよい。図3.2に示すようなユニット間の階層関係は，すべてのユニット間に存在してもよいし，特定のユニット間にのみ存在してもよいし，またまったく存在しなくてもよい。

LEX によるテクスト処理に際しては，任意のテクスト構成要素を無視して，単純化されたテクスト構造を想定して，テクスト処理を行うことができる。テクストが図3.1のような構造を有するとき，図3.3のようにテクストが単語のみから成るものと想定して処理を行うこともできれば，図3.4の例のように，テクストが第１ユニットとＬＥＸ

行のみから成るものと想定して処理を行うこともできる。前者の例としてはテクスト中の見出語の頻度や単語の長さを求める処理があり，後者の例としては第1ユニット中のLEX行数を求めるといった処理がある。

LEXテクスト・ファイル等においては，上述の3種類のテクスト構成要素の個々の現われ（ token ）を識別する必要がある。LEXではこのため，テクスト構成要素の個々の現われに，識別のための数値や名称を付ける。これを**識別値**と称する。また，これに対応して，入力テクストでは，テクスト・ユニットの境界を原則として明示しなければならない。LEXは明示された境界によってテクスト構成要素に識別値を付ける。

LEXのテクスト構成要素識別値には正確に言えば2種類ある。**利用者指定の識別値**と**システム識別値**である。

システム識別値とは，システムがすべてのテクスト構成要素の個々の現われに付ける1から始まる一連番号である。単語に付けられる一連番号を**単語番号（WSQ）**と呼ぶ，LEX行に付けられる一連番号を**行番号（LSQ）**と呼ぶ。後述のデータ置換・選別命令の処理を除き，テクスト処理時におけるテクスト構成要素の識別はすべてこのシステム識別値によって行われる。しかし，利用者は，単語番号と行番号を除き（テクスト・ユニットに付けられた）システム識別値をLEX文では参照できない。

これに対して，テクスト・ユニットに関しては，システム識別値とは別に，識別の方法，値や名称の与え方を指定することができる。これが利用者指定の識別値である。この識別値は，データ置換・選別命ではキーとして用いられ，索引系の処理では索引項目として印刷することができる。

本稿では，識別値と言うとき，特に断らない限り，利用者が参照できる，単語番号，行番号，および（テクスト・ユニットの）利用者指定の識別値を指すことにしシステム識別値については特に言及しない。

入力テクストでのテクスト構成要素の個々の現われの識別は次のように行われる。単語は単語境界標識によって識別され，LEX行は入力テクストのレコード境界によって識別され，テクスト・ユニットは利用者の与えるユニット境界標識によって識別される。この点についての詳細は次節に述べる。

Ⅲ．2　入力テクスト

Ⅲ．2.ⅰ　入力レコードの構造

入力テクストはレコードの集合である。レコードとは，コンピューターがデータを処理する単位であるが，入力テクストでは，1枚のカードまたは端末の1行である。正確に言えば，LEXへの入力テクストは，固定長80バイトのレコードから成っていなければならない。

入力テクストが固定長80バイトのレコードから成るということは，原テクストから入力テクストを作成するとき，原テクストをレコードに分割しなければならぬことを意味する。しかも，レコードは次のような構造を有する。

個々のレコードは，**データ部**（テクスト本体）と**標識部**のいずれか一方もしくは両方から成る。データ部はテクスト本体であり，単語と**単語境界標識**（word boundary marker 以下WBM）から成る。行ファイルが作成されるときには，このデータ部がLEX行としてそのままの形で保存，記録される。他方，標識部は，前節で述べた**テクスト・ユニット境界標識**（unit boundary marker, 以下UBM）と（利用者指定の）テクスト・ユニットの識別値から成る。また個々の入力レコードにおけるデータ部と標識部の関係としては，図3.5 a－cに示す3つの場合だけが許される。

図3.5　入力テクストの1レコード（80バイト）

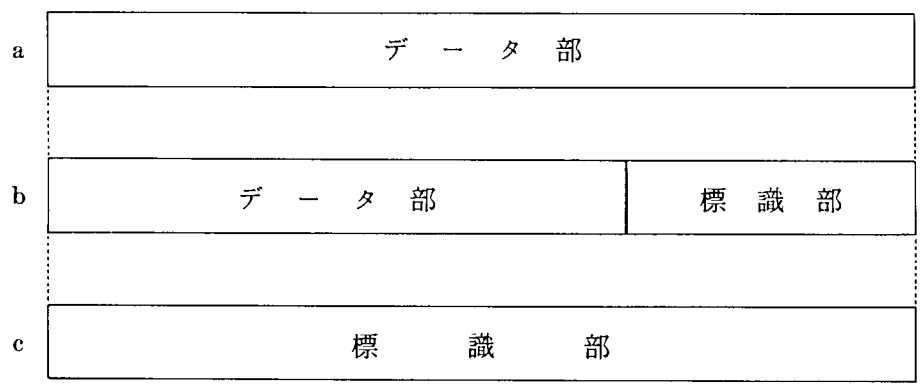


図3.5において，aはレコードがデータ部のみから成る例であり，cは逆にレコードが標識部のみから成る例である。bはレコードがデータ部と標識部から成る例であるが，

レコードがデータ部と標識部をともに含むときには，データ部は常に標識部より前になければならない。仮に標識部の後にデータ部があったとしても，このようなデータ部は無視されるか，標識部の一部として処理される。

Ⅲ．2.ⅱ　データ部

入力レコードのデータ部は前述のように，**単語と単語境界標識**（ＷＢＭ）から成る。ＷＢＭとは次の12種類の記号の少なくとも１種以上から成る，長さ１文字以上，79文字以下の記号列である。

空白（スペース）	
コロン	:
セミコロン	;
左かっこ	(
右かっこ	)
カンマ	,
ピリオド	.
感嘆符	!
疑問符	?
二重引用符	▼▼
縦線（ストローク）	|
下線	_

この12種類の記号は定義からしてＬＥＸで言う単語の構成要素たりえない。また後に述べるユニット境界標識（UBM）として用いられる，＊，＃，％，@の４種の記号も同様にして単語の構成要素としては用いることができない。従って，ＬＥＸで言う単語とは，上記16種の記号を除く，コンピューターで利用可能な任意の文字，数字，記号から成る最大32文字までの記号列であると定義される。この定義からして，ＬＥＸで言う単語は，日常的な意味や言語学的意味における単語である必要はまったくない。

上記16種の記号を単語の構成要素として用いたときには，入力テクストでは適当な記号(列)で代用しておき，テクスト・ファイル作成後REPLACE命令を使って望みの記号に変えるという方法がある。しかし，この方法では，LEX行ファイルは変更できないし，またセミコロンは単語の構成要素たりえない。このような場合には，利用者がテクスト・ファイルや行ファイルの内容を自分で修正するほかはない。

通常の原テクストであれば，データ部には，原テクストをほぼそのままの形でパンチすればよい。識別部については，レコードに十分な余白があればデータ部の後にパンチしてもよいし，次のレコードを識別部のみから成るレコードとしてもよい。

データ部の作成にあたっては，以下の諸点に注意が必要である。

a　単語はふたつのレコードにまたがってはならない。この場合，2語に分割される。

b　データ部には少なくともひとつの単語が含まれていなければならない。WBMのみから成るレコードもしくはデータ部は無視される。同じことであるが，空白のレコードも無視される。ただし，いずれの場合も読み飛ばされるだけであってエラーとはならない。

このことを利用して，空白あるいはWBMのみから成るカード(端末行)を入力テクスト中に適宜挿入し，入力テクストの区切りとして用いることができる。

c　LEX行ファイルが作成されるときには，データ部中のすべてのWBMが意味をもつ。用例索引など，LEX行ファイルを使う処理では，データ部が入力テクストそのままの形で印刷されるからである。

d　上述の16種の記号以外のすべての記号は単語の構成要素と見なされる。例えば，ハイフン(—)，アポストロフィ(▼)，斜線(／)等もすべて単語の一部として処理される。原テクスト中で，このような記号が単語の区切り記号として用いられているときには，他の記号で代用しておく必要がある。

e　端末を使って入力テクストを作成するときには，レコード長が80バイトであることを確認し，かつ，行番号を削除しておかなければならない*。さもないと，行番号自体も単語として扱われる。

*実際には，入力テクストをLEXに入力するときまでに削除されていればよい。

f　ひとつの単語の前後には必ずWBMがなくてはならない。例外は，単語の最初の文字がレコードの第1カラムにあるときと，単語の最後の文字が第80カラムにあるときである。

Ⅲ. 2.ⅲ　識別部

入力レコード中のデータ部がテクスト本体に対応するのに対し，標識部は，テクスト・ユニットの境界を示し，必要に応じてその識別値を与えるものである。ひとつの入力レコード中では，標識部はどこにあってもよいが，前述のように，データ部が標識部より後に存在することは許されない。ひとつのレコードにおける標識部は，ひとつ以上の**ユニット境界標識**（ＵＢＭ）と，それに対応する識別値からなる。入力テクスト中では原則としてテクスト・ユニットの（個々の現われの）境界をＵＢＭによって明示しなければならない。また必要に応じ，識別値を与えなければならない。

ＵＢＭには，テクスト・ユニットに対応して次の４種類がある。

アステリスク	（＊）	… 第１ユニット
シャープ	（＃）	… 第２ユニット
パーセント	（％）	… 第３ユニット
単価記号	（@）	… 第４ユニット

ここで第iユニットとは，ＩＮＰＵＴ命令の，UNIT_i明細指示で存在を宣言されたテクスト・ユニットである。

識別値は前述のようにテクスト・ユニットの個々の現われを識別するために用いられる数値や名称で，索引系の出力では索引項目として印刷され，データ置換・選別命令ではキーとして用いられる。ＬＥＸでは個々のユニットに与えられる識別値はユニークである必要はない。前述のようにユニットは必要に応じシステム識別値によってユニークに識別可能だからである。

これまで，ＵＢＭはユニットの（個々の現われの）境界を示す，という表現を使ってきたが，厳密に言えば，この表現は正確を欠く。厳密には，ＵＢＭは，ユニットの（個々の現われの）開始を示す，と言うべきである。そして，識別部中のＵＢＭは以下の規則に従って与えなければならない。

a　ＵＢＭは，当該ＵＢＭの後にある最初のデータ部から，新たなテクスト・ユニットが始まることを示す。従って，

b　ＵＢＭは当該テクスト・ユニットの前になくてはならない。前述のように，同一レ

コード中でUBMの後にデータ部がくることはないから，UBMは，当該ユニット（の最初の部分）をデータ部に含むレコードより前のレコード中になくてはならない。従って，

c　入力テクストの最初のレコードは，標識部のみから成るレコードである。この例外となるのは，INPUT命令で宣言されたユニットが第1ユニットだけで，しかもその識別モードが後述のRモードである場合と，Ⅲ．2.ⅱbで説明したダミーのレコードが入力テクストの先頭にある場合だけである。

d　入力テクストの最後のレコードは，データ部のみから成るレコードである。ただし，Ⅲ．2.ⅱbのダミー・レコードが入力テクストの末尾にある場合は例外となる。

e　複数のテクスト・ユニットの境界が一致するときは，ひとつのレコードの標識部に複数のUBM（プラス識別値）を与えてもよいし，連続した任意の数のレコードに分割して与えてもよい。いずれの場合にも，UBMを与える順序は任意である。

以上が，UBMを与えるときの一般的規則であるが，実際には，識別値の与え方によっては更に制約が加わることもある。これについては次項で述べる。

Ⅲ．2.ⅳ　識別モードと識別値

識別値の与え方を**識別モード**と称する。LEXでは，次の4種類の識別モードが許される。

Sモード	・・・	整数型
Fモード	・・・	整数型
Rモード	・・・	整数型
Aモード	・・・	文字型

Sモードは，テクスト・ユニットの入力テクストでの出現順に一連番号を付ける識別モードである。初期値は省略時は1であるが，99999以下の任意の整数値を指定できる。Sモードでは，入力テクスト中にユニット境界をUBMを与えて示すだけでよく，識別値を与える必要はまったくない。

Fモードでは，識別値を自由に変えられないSモードと異なり，初期値を必要に応じ何度でも自由に与えることができる。この識別モードでは，入力テクスト中のUBMに続けて整数の識別値を与えると，次に識別値が与えられるまで，その識別値から始まる一連番号が，以後の各ユニットに付けられる。次に識別値が与えられるとそれ以後のユニットには新たな値から始まる一連番号が付けられる。従って，この識別モードでは入力テクスト中で，UBMだけを与える場合と，UBMに続けて新しい初期値たる識別値を与える場合が混在することになる。Fモードは，頁や章が改まるごとに行番号を新しく付け直すときや，原テクストの行，頁等が一部欠落しているときに有効である。

Rモードは，Sモードの特殊な変形で，しかも第1ユニットに関してのみ指定できる。Rモードでは，LEX行を即ち入力レコードをひとつの第1ユニットと見なし，Sモード同様指定された初期値から始まる一連番号を付ける。従って，他の3つのモードではUBMだけは常に与える必要があるのに対し，この識別モードでは，UBMもまったく与える必要がない。ひとつのテクスト・ユニット，例えば原テクストの1行，が80文字以内で，しかもテクストの構成要素として大きな意味をもつときは，このRモードはきわめて有効である。また，第1テクスト・ユニットだけは必ず宣言しなければならないから，テクスト・ユニットそのものが不要なときには，仮の第1ユニットを宣言して，このモードを指定しておけばよい。

> Rモードのとき，テクスト構成要素としての第1ユニットとLEX行はまったく同じものになるが，第1ユニットでは識別値の初期値を自由に指定できるがLEX行ではできないこと，第1ユニットとLEX行では後のテクスト処理における扱いが異なることなどの違いがある。

S，F，Rの3つの識別モードが数値型の識別値を与えるのに対し，**Aモード**では所謂文字型* の識別値を与える。しかも，他の3つのモードと異なり，Aモードでは，UBMとともに識別値を必ず与えなくてはならない**。

* 数字を文字型として使用できることは言うまでもない。
** 例外については後述する。

以上4つの識別モードのうち，RモードはUBM自体を与える必要がない。また，SモードのUBMについては，識別値を与える必要がないので前項（Ⅲ．2.ⅲ）a～eに述べた以上の制約はない。しかし，識別値を与えるAモードとFモードについては，これに加えて以下のような制約がある。

a　UBMと識別値は同一レコード上になくてはならない。

b　識別値がふたつのレコードにまたがってはならない。

c　識別値はユニークである必要はない。

Fモードの場合，これに加えて次の制約がある。

d　識別値を与えないときには，UBMの後に少なくともひとつの空白が必要である。ただし，UBMがレコード80カラムにあるときはこの例外である。

e　識別値を与えるときには，

ア　識別値は6桁以内の正整数でなくてはならない。

イ　UBMと識別値の間に空白があってはならない。もし，空白があれば識別値は与えられていないものとして処理される。即ち，dの場合と同様に解釈され，「直前のユニットの識別値＋1」が新たなユニットの識別値となる。

ウ　識別値後には少なくともひとつの空白が必要である。ただし，識別値の最後の数字がレコードの80カラムにあるときは，この例外である。

エ　入力テクストの最初のUBMに続けて，初期値となる識別値を与えなくてはならない。

以上の条件のいずれかひとつが満たされないときの結果は保証されない。利用者の意図しない結果を生ずるか，エラーとして処理される。

次にAモードの場合の制約は，

f　識別値は常に与えなくてはならない。

g　識別値は6文字以内でなくてはならないが，空白とUBMとして用いられる4種の記号を除くどのような文字，数字，記号を用いてもよい。

h　識別値はUBMに続けて与えるが，UBMと識別値の間に空白があってはならない。もし空白があれば識別値は与えられていないものと解釈される。

i　識別値の後には少なくともひとつの空白が必要である。ただし，識別値の最後の文字，数字，記号がレコードの80カラムにあれば，この例外となる。

Aモードでは，a～c，f～iの条件が満たされないときでも通常はエラーとして処理されないが，利用者の意図と異なる結果を生ずることは言うまでもない。

Aモードの場合，UBMに続けて少なくともひとつの空白があるときには，上述のfに抵触することになるが，このときシステムは6個の空白を識別値とする。テクストの構造や研究目的の如何によっては，このような空の識別値も利用価値をもつことがある。例えば，地の文には空の識別値を与え，会話や引用には空でない識別値を与えれば，索引では，会話や引用の部分だけに識別値が印刷され，地の文については識別値は空白のままにしておく，といったことも可能である。ただ，このような空の識別値はLEX文では直接には参照できないという不便もある。空の識別値を用いた実例については，後掲の例3.3と例4.6を参照。

Ⅲ. 3　INPUT命令

利用者の作成したLEX入力用テクストは，INPUT命令によってLEXに入力され，LEXテクスト・ファイル（これに加えて利用者の指示があればLEX行ファイル）が作成される。テクスト入力の詳細と実例は後に示すことにして，ここではまずINPUT命令について説明する。INPUT命令の書式は次の通りである。

```
 INPUT
[WORD△LENGTH=i]          (2≦i≦32)    <i=16>
[LINE△FILE={N, Y}]
 UNIT1△△NAME=ユニット名[△△IDMODE={S, F, A, R}][△△START=S1]
[UNIT2△△NAME=ユニット名[△△IDMODE={S, F, A}][△△START=S2]]   (1<Sj≦99999)
[UNIT3△△NAME=ユニット名[△△IDMODE={S, F, A}][△△START=S3]]     <Sj=1>
[UNIT4△△NAME=ユニット名[△△IDMODE={S, F, A}][△△START=S4]]
[%STOPWORD
 単語1
 . . .
 . . .
 単語n                    (1≦n≦100)
 %END]
 READ[△{SYSIN, LXDATA}]
 [入力データ]
```

機　能

利用者の作成した入力テクストを読み込んで，LEXテクスト・ファイルを作成する。

指示があればLEX行ファイルも作成する。テクスト・ファイル，行ファイル作成の媒体は任意である。

WORD LENGTH明細指示

テクスト中の単語の仮想最大長を文字数で指定する。指定は2から32までの正整数で指定する。省略時には，単語の最大長として16（文字）が仮定される。ここで注意しなければならないのは，アポストロフィー，濁点等の記号も1文字として数えられることと，後にREPLACE命令で長い単語を与える可能性がある場合にはその可能性も考慮に入れて，単語長を指定する必要があることである。（後者については，Ⅳ．2，注意事項のg参照。）

省略時解釈も含め，指定された最大長を越える単語がINPUT命令の実行中に検出されると，その単語について，

PNEUMONOULTRAMICROSCOPICSILICOVOLCANOCONIOSIS*TRUNCATED

という形の削除メッセージが印刷され，指定の文字数を越える部分は削除される。

＊硅性肺塵症

LINE FILE明細指示

LEX行ファイルを作成するか否かを指定する。

N：行ファイルを作成しない。

Y：行ファイルを作成する。

省略時には，Nが仮定され，行ファイルは作成されない。行ファイルは，CONCORDANCE, KLIC, KWIC INDEX, LINE LENGTH, LINE LIST, LINE INDEX, PRINT LINE, WORD SET CONCORDANCEの各命令の実行には不可欠であり，このような命令を実行するためにはここでYを指定して行ファイルを作っておく必要がある。

Yを選択したときには，LEX行ファイルが記録されるLXLINEファイルをJCLで定義しておく必要がある。（XV章参照。）

UNIT1～UNIT4明細指示

テクスト・ユニットの存在を宣言し，その名称，識別モード，初期値を与える。以下，UNITi明細指示で宣言されるユニットを第i（テクスト）ユニットと呼ぶ。上掲の書式からも明らかなように，第1ユニットは必ず宣言しなければならない。しかし，テクスト・ユニットがまったく不要なテクストであれば，

UNIT1△△NAME=a△△IDMODE=R

（ただし，aは1～6文字の任意の文字列）

というダミーの明細指示を与えておけばよい。こうしておけば，UBMもまったく与える必要がない。

これに対して，UNIT2～UNIT4明細指示は任意的である。この明細指示を与えるときには，UNITに続く数値の順に与えなくてはならない。しかし，この数値と，ユニットの大小，ユニット間の階層関係の有無（Ⅲ．1参照）とは無関係である。UNITi明細指示を与え，第iテクスト・ユニットの存在を宣言したときには，入力テクスト中で，各ユニットの個々の現われの境界（正確には開始）を，定められたUBMを用いて示さなければならない。例外は，第1ユニットに関して，

UNITI△△NAME・・・△△IDMODE=R

としたときだけである。既に述べたように，第iユニットのUBMは次の通り定められている。

第1ユニット	*
第2ユニット	#
第3ユニット	%
第4ユニット	@

UNIT i 明細指示中には，NAME，IDMODE，STARTの3つのオペランドがある。指定の順序は実際には任意であるが，このうちの2つ以上を与えるときには，間に少なくともひとつの空白がなくてはならない。各オペランドの意味は次の通りである。

NAME

NAME=に続けて，当該テクスト・ユニットの名称を6文字以内で与える。名称は空白，セミコロンおよびUBMとなる4種の記号を除く，どのような文字，数字，記号から成っていてもよい。ここで与えた名称は索引系の出力で，見出しとして用いられる。

UNIT i 明細指示を与えたとき，このオペランドは省略できない。

IDMODE

テクスト・ユニットの識別モードを指定する。このオペランドが省略されたときには，IDMODE=S が仮定される。選択肢のS，F，R，Aはそれぞれ前節で述べたSモード，Fモード，Rモード，Aモードに対応する。Rを指定できるのは，UNIT1明細指示においてのみである。

START

IDMODEオペランドが省略されたときも含め，識別モードがSモードかRモードのときにのみ有効なオペランドで，識別値として与える一連番号の初期値を与える。初期値は1から99999までの正整数で与える。このオペランドが省略されたときは，START=1が仮定される。即ち，初期値は1となる。

Fモードの初期値はこのオペランドでは指定できない。入力テクストの最初のUBMに続けて指定する。（Ⅲ．2. ⅳのeのエ）

%STOPWORD

入力テクスト中に含まれてはいるが，処理にはまった不要な単語（ストップワード）があるときには，まず%STOPWORD ステートメントを与え，これに続くLEXステートメントにストップワードとすべき単語を1語づつ与える。単語は少なくともひとつ，最大100語まで与えることができる。リストが終ったら，リストの終りを示す%ENDステートメントを必ず与えなければならない。

ここで指定された単語はテクスト・ファイルに記録されず，従って，以下の処理での対象とならない。また%STOPWORDステートメント以下で ストップワードを指定したときには，テクスト中の単語につけられる単語番号は，該当の単語がテクスト中に存在しないものとして付けられる。

行ファイルが作成されるとき，行ファイルには入力テクストがそのままの形で保存されるので，ここで指定した単語も削除されることはないし，ここで指定した単語とWBMのみから成るLEX行も削除されることはない。

テクスト中の単語の延べ数が問題になるときや，単語相互の隣接性が問題になる単語列処理の命令を使う可能性のあるときは，ここでストップワードリストを使って単語を削除するより，必要に応じて，EXCLUDE命令を使って不要な単語を削除すべきである。

READ命令

入力テクストの読み込み開始を指示し，かつ入力テクストがどのファイル（あるいはデータセット）上にあるかを指定する。READ命令は，ファイル定義命令のひとつであり，本来であればこの位置ではなく，実行命令の次に位置するが（II. 1参照），INPUT命令の場合は例外である。

READ命令の直後に入力テクストがある場合，即ちINPUT命令のLEX文と，入力テクストが同一のファイル（あるいはデータセット）上にあれば，次のふたつのうちいずれかを与える。

```
READ
READ△SYSIN
```

これに対して，LEX 文と入力テクストが異なったファイル（あるいはデータセット）上にあれば，例えば，JCLとLEX文がカード，入力テクストが磁気ディスク上にあれば，READ命令は，

```
READ  LXDATA
```

としなければならない。

このときには，これに加えて，JCL 中で，入力テクストの存在するファイルである

ＬＸＤＡＴＡという名称のファイルを定義しておかなければならない。（詳細はXV章を参照）。

ＩＮＰＵＴ命令の実行が正常に終了すると，ＩＮＰＵＴ命令のＬＥＸ文と入力されたテクストに関する簡単な情報が，次の例３．１のような形で出力される。

＜例３．１＞

```
*** LEX STATEMENTS ***
INPUT
UNIT1 NAME=LINE START=287
UNIT2 NAME=TALE IDMODE=A
READ
*** END ***

  *** INPUT TEXT INFORMATION ***
    NUMBER OF WORDS=  4963
    NUMBER OF LINES=   640
    UNIT 1
      UNIT NAME=  LINE
      IDENTIFICATION MODE=S
    UNIT 2
      UNIT NAME=  TALE
      IDENTIFICATION MODE=A
```

Ⅲ．４　テクスト入力の実例

本節では，入力テクストとＩＮＰＵＴ命令の具体例を幾つか挙げる。次いで，入力テクストの点検と修正について述べる。

次の例３．２は入力テクストの例であるが，例３．２の各行がひとつの入力レコード（ＬＥＸ行）である。（以下の例でも同様）。

例３．２に示す入力テクストは，実際には，

```
INPUT
UNIT1△△NAME=SENT
UNIT2△△NAME=PARGR
READ
```

というＩＮＰＵＴ命令に続くものである。ＵＮＩＴ１，ＵＮＩＴ２明細指示ともに，ＩＤＭＯＤＥ，ＳＴＡＲＴが省略されているので，第１ユニット，第２ユニットとも１から始まる一連番

＜例 3. 2＞

```
* #
   ATTAINMENT OF THE OBJECTIVE OF SECURITY, WHICH IS AN INSEPARABLE
ELEMENT OF PEACE, HAS ALWAYS BEEN ONE OF THE MOST PROFOUND
ASPIRATIONS OF HUMANITY. *
 STATES HAVE FOR A LONG TIME SOUGHT TO MAINTAIN THEIR SECURITY THROUGH
THE POSSESSION OF ARMS. *
 ADMITTEDLY, THEIR SURVIVAL HAS, IN CERTAIN CASES, EFFECTIVELY DEPENDED
ON WHETHER THEY COULD COUNT ON APPROPRIATE MEANS OF DEFENCE. *
 YET THE ACCUMULATION OF WEAPONS, PARTICULARLY NUCLEAR WEAPONS, TODAY
CONSTITUTES MUCH MORE A THREAT THAN A PROTECTION FOR THE FUTURE OF
MANKIND.
*
 THE TIME THEREFORE COME TO PUT AN END TO THIS STUATION, TO ABANDON
THE USE OF FORCE IN INTERNATIONAL RELATIONS AND TO SEEK SECURITY IN
DISARMAMENT, THAT IS TO SAY, THROUGH A GRADUAL BUT EFFECTIVE PROCESS
BEGINNING WITH A REDUCTION IN THE PRESENT LEVEL OF ARMAMENTS.
*
 THE ENDING OF THE ARMS RACE AND THE ACHIEVEMENT OF REAL DISARMAMENT
ARE TASKS OF PRIMARY IMPORTANCE AND URGENCY.
*
 TO MEET THIS HISTORIC CHALLENGE IS IN THE POLITICAL AND ECONOMIC
INTERESTS OF ALL THE NATIONS AND PEOPLES OF THE WORLD AS WELL AS THE
INTERESTS OF ENSURING THEIR GENUINE SECURITY AND PEACEFUL FUTURE.
* #
    UNLESS ITS AVENUES ARE CLOSED, THE CONTINUED ARMS-RACE MEANS A
GROWING THREAT TO INTERNATIONAL PEACE AND SECURITY AND EVEN TO THE VERY
SURVIVAL OF MANKIND.
*
 THE NUCLEAR AND CONVENTIONAL ARMS BUILD-UP THREATENS TO STALL THE
EFFORTS AIMED AT REACHING THE GOALS OF DEVELOPEMNT, TO BECOME AN
OBSTACLE ON THE RORAD OF ACHIEVING THE NIEO AND TO HINDER THE SOLUTION
OF OTHER VITAL PROBLEMS FACING MANKIND.
* #
   DYNAMIC DEVELOPMENT OF DETENTE, ENCOMPASSING ALL SPHERES OF
INTERNATIONAL RELATIONS IN ALL REGIONS OF THE WORLD, WITH THE
PARTICIPATION OF ALL COUNTRIES,WOULD CREATE CONDITIONS CONDUCIVE TO THE
EFFORTS OF STATES TO END THE ARMS-RACE, WHICH HAS ENGULFED THE WORLD,
THUS REDUCING THE DANGER OF WAR.
*
 PROGRESS ON DETENTE AND PROGRESS ON DISARMAMENT MUTUALLY COMPLEMENT AND
STRENGTHEN EACH OTHER.
*
#
  THE DISARMAMENT-DECADE SOLEMNLY DECLARED IN 1969 BY UN IS COMING TO AN END.
```

号が識別値として与えられる。ここでは，文（SENTence）を第1ユニット，段落（PARGRaph）を第2ユニットとしている。ユニットを宣言したときには，UBMによってユニットの境界（正確には開始）を示さなければならないが，この例は，UBMの位置についての幾つかの例を挙げている。この例でのUBMの位置は，UBMを置きうる位置の様々な例であって，必ずしもこの通りにしなければならぬというものではない。

次の例3. 3はAモードを用いた例である。

＜例 3. 3＞

```
INPUT
<C> FROM "FOOD AND NUTRITION BULLETIN, VOL.3,NO.2,1981"
UNIT1 NAME=SENT
UNIT2 NAME=PAGE
UNIT3  NAME=MODE  IDMODE=A
<C>  CODE: NONE=MAIN BODY OF TEXT
<C>             H=HEADING
<C>             Q=QUOTATION
READ
* # %H
HUNGER AS A PRESSING GLOBAL PROBLEM OF HUMAN  SURVIVAL
* %
HUNGER IS ONE OF THE MOST SERIOUS AND OBVIOUS MANIFESTATIONS OF THE
CRISIS IN THE WORLD. *
WITH THE BREAKDOWN OF THE TRADITIONAL LOCAL FOOD SECURITY MECHANISMS,
HUNGER HAS BECOME ALMOST INSEPARABLE FROM POVERTY. *
IT IS SPREAD BY THE SAME "MARKET MECHANISM" AS COMMODITIES ON THE WORLD
MARKET.
*
UNFAVOURABLE WEATHER, INFLATION, AND CHANGING INVESTMENT PRIORITIES IN
THE RICH, FOOD-SURPLUS-PRODUCING COUNTRIES AFFECT THE CONDITIONS OF
SURVIVAL OF THE POOR IN THE PERIPHERY. *
THE UNITED NATIONS HAS HAD THE ALLEVIATION OF HUNGER ON ITS AGENDA FOR
THE LAST TWENTY YEARS. *
IN THE 1950S AND 1960S MOST OF THE WORK WAS DIRECTED TOWARDS INCREASING
FOOD PRODUCTION, REDUCING THE POPULATION GROWTH RATE, AND PROMOTING THE
THEORY OF THE TRICKLE-DOWN EFFECT OF ECONOMIC GROWTH. *
THE TECHNICAL ASPECTS OF THE PROBLEM OF HUNGER WERE WELL DOCUMENTED AND
THOROUGHLY DISCUSSED IN NUMEROUS BOOKS, REPORTS, AND PUBLICATIONS. *
DURING THE 1970S, HOWEVER, THE VALUE OF THIS APPROACH WAS QUESTIONED BY
```

この例ではＡモードをやや特殊な形で用いている。即ち，第３ユニットを，例の註釈ステートメントからも判るように，「本文」，「見出し」，「引用」の区別をするために用いている。これにより，後のテクスト処理では，見出しや引用部分を除いた処理，あるいは引用部分だけを対象とした処理が可能になる。この例で，入力テクストの３番目のレコードで第３ユニットのＵＢＭ(%)の後に識別値が与えられていないのは，「本文」には識別値とし６個の空白を与えるためである。（あるいは識別値を与えないためと解釈しても差支えない。）（Ⅲ．2.ⅳ最後の註を参照）。

例3. 4は行ファイルを作成していること，Ｆモードを用いていることが特徴である。

この例でＦモードが用いられているのは，第１ユニット（名称はＬＩＮＥ）の番号が途中で再び１から始まるからである。この例では，また，すべてのレコードの同じ位置に第１ユニットのＵＢＭ（＊）があるが，この例のように大部分の第iユニットが１レコードに相当する入力テクストであれば，ＵＢＭを付けない入力テクストをまず作っておき，次のステップで計算機システムのユーティリティを使ってＵＢＭを付け加えるといった

＜例 3. 4＞

```
INPUT
LINE FILE=Y
UNIT1 NAME=LINE   IDMODE=F
UNIT2 NAME=TALE   IDMODE=A
READ
*1 #GENPRO
  WHAN THAT APRILL WITH HIS SHOURES SOOTE                              *
THE DROGHTE OF MARCH HATH PERCED TO THE ROOTE,                         *
AND BATHED EVERY VEYNE IN SWICH LICOUR                                 *
OF WHICH VERTU ENGENDRED IS THE FLOUR;                                 *
WHAN ZEPHIRUS EEK WITH HIS SWEETE BREETH                               *
INSPIRED HATH IN EVERY HOLT AND HEETH                                  *
THE TENDRE CROPPES, AND THE YONGE SONNE                                *
HATH IN THE RAM HIS HALVE COURS YRONNE,                                *
AND SMALE FOWELES MAKEN MELODYE,                                       *
THAT SLEPEN AL THE NYGHT WITH OPEN YE                                  *
(SO PRIKETH HEM NATURE IN HIR CORAGES);                                *
THANNE LONGEN FOLK TO GOON ON PILGRIMAGES,                             *
AND PALMERES FOR TO SEKEN STRAUNGE STRONDES,                           *
TO FERNE HALWES, KOWTHE IN SONDRY LONDES;                              *
AND SPECIALLY FROM EVERY SHIRES ENDE                                   *
OF ENGELOND TO CAUNTERBURY THEY WENDE,                                 *
THE HOOLY BLISFUL MARTIR FOR TO SEKE,                                  *
THAT HEM HATH HOLPEN WHAN THAT THEY WERE SEEKE.                        *
  BIFIL THAT IN THAT SESON ON A DAY,                                   *
IN SOUTHWERK AT THE TABARD AS I LAY                                    *
REDY TO WENDEN ON MY PILGRYMAGE                                        *
TO CAUNTERBURY WITH FUL DEVOUT CORAGE,                                 *
AT NYGHT WAS COME INTO THAT HOSTELRYE                                  *
WEL NYNE AND TWENTY IN A COMPAIGNYE,                                   *
```

方法が便利である。

次の例 3. 5 もＦモードを用いた例であるが，前の例に比べて使い方はかなり複雑である。この例の左側の 5 桁の数字は参照の便のため付けたもので，入力テクストの一部ではない。この数字はテクスト入力時には削除されるべきものである。

原テクストは，Shakespeare の *Hamlet* であるが，第 1 ユニット（ＬＩＮＥ）がＦモードとなっているのは，幕，場が改まるごとに行の番号が 1 から始まることと，180～190，250～260 などのよう割台詞があることによる。なお，第 2 ユニットは「幕一場」，第 3 ユニットは「話者＝登場人物」である。このような例では，入力テクストの作成には多少の手間がかかるが，索引系の出力では誰の台詞か明らかできるし，登場人物の比較分析も可能になる。

次の例 3. 6 は技術移転に関する文献のキーワードを入力テクストとした特殊な例で，この例では，ＬＥＸ行というテクスト構成要素は実質的に何の意味ももたない。

＜例 3.5＞

```
00010 INPUT
00020 LINE FILE=Y
00030 UNIT1 NAME=LINE   IDMODE=F
00040 UNIT2 NAME=ACTSCN   IDMODE=A
00050 UNIT3 NAME=SPEAKR IDMODE=A
00060 READ
00070 *1 #I-1 %BARN
00080 WHO'S THERE ?   * %FRAN
00090 NAY, ANSWER ME.   STAND AND UNFOLD YOURSELF. * %BARN
00100 LONG LIVE THE KING! * %FRAN
00110 BARNARDO? * %BARN
00120 HE. * %FRAN
00130 YOU CAME MOST CAREFULLY UPON YOUR HOUR. * %BARN
00140 'TIS NOW STRUCK TWELVE, GET THEE TO BED, FRANCISCO. *
00150 %FRAN
00160 FOR THIS RELIEF MUCH THANKS, 'TIS BITTER COLD, *
00170 AND I AM SICK AT HEART. * %BARN
00180 HAVE YOU HAD QUIET GUARD? %FRAN
00190                          NOT A MOUSE STIRRING. * %BARN
00200 WELL, GOOD NIGHT: *
00210 IF YOU DO MEET HORATIO AND MARCELLUS, *
00220 THE RIVALS OF MY WATCH, BID THEM MAKE HASTE. *
00230 %FRAN
00240 I THINK I HEAR THEM. STAND HO, WHO IS THERE? * %HOR
00250 FRIENDS TO THIS GROUND. %MAR
00260                        AND LIEGEMEN TO THE DANE. * %FRAN
00270 GIVE YOU GOOD NIGHT. %MAR
00280                     O, FAREWELL HONEST SOLDIER, *
00290 WHO HATH RELIEVED YOU? %FRAN
00300                      BARNARDO HATH MY PLACE;   *
00310 GIVE YOU GOOD NIGHT.      %MAR
00320                       HOLLA, BARNARDO! %BARN
00330                                        SAY   *
00340 WHAT, IS HORATIO THERE? %HOR
00350                             A PIECE OF HIM. * %BARN
00360 WELCOME HORATIO, WELCOME GOOD MARCELLUS. * %HOR
00370 WHAT, HAS THIS THING APPEARED AGAIN TO-NIGHT? * %BARN
00380 I HAVE SEEN NOTHING. * %MAR
00390 HORATIO SAYS 'TIS BUT OUR FANTASY,    *
00400 AND WILL NOT LET BELIEF TAKE HOLD OF HIM *
00410 TOUCHING THIS DREADED SIGHT TWICE SEEN OF US, *
00420 THEREFORE I HAVE ENTREATED HIM ALONG    *
00430 WITH US TO WATCH THE MINUTES OF THIS NIGHT, *
00440 THAT IF AGAIN THIS APPARITION COME, *
00450 HE MAY APPROVE OUR EYES AND SPEAK TO IT. * %HOR
00460 TUSH, TUSH, 'TWILL NOT APPEAR. %BARN
00470                                SIT DOWN AWHILE,
```

以上，入力テクストの実例を幾つか挙げたが，ここで，このような入力テクストの点検と修正について簡単に述べておこう。作成したデータの点検はＬＥＸを利用する場合に限らずコンピューター利用におけるきわめて重要な部分である。ＬＥＸのように言語データを扱うときにはデータ（入力テクスト）が数万行に上ることも珍らしいことではなく，その点検と修正にも入力テクストの作成に劣らぬ時間と労力が必要となる。この

＜例 3.6＞

```
INPUT
UNIT1 NAME=DOCMNT
UNIT2 NAME=ｺｸﾒｲ   IDMODE=A
READ
* #JPN
ｺｳｷﾞﾖｳｼｮｳｹﾝ   ﾃｸﾉﾛｼﾞｰﾄﾗﾝｽﾌｧ   ﾄﾂｷﾖｷﾖｳﾘｮｸｼﾞｮｳﾔｸ   ﾊﾂﾃﾝﾄｼﾞｮｳｺｸ
ﾊﾟﾘｼﾞｮｳﾔｸ *
ｼｮｳﾋｮｳ   ﾃｸﾉﾛｼﾞｰﾄﾗﾝｽﾌｧ   ﾄﾂｷﾖ   ﾄﾂｷﾖｹﾝ   ﾉｳﾊｳ   ﾊﾂﾒｲ *
ｷﾞｼﾞﾕﾂﾄｳﾆｭｳ   ｹｲﾔｸ   ｺｸｻｲｶﾝｹｲ   ﾃｸﾉﾛｼﾞｰﾄﾗﾝｽﾌｧ   ﾄﾂｷﾖ   ﾄﾂｷﾖｹﾝ ﾉｳﾊｳ
ﾊﾂﾒｲ *
ｹｲｴｲｾﾝﾘｬｸ   ｼｼﾞｮｳﾁｮｳｻ   ｼﾞｮｳﾎｳｼｭｳｼｭｳ   ﾃｸﾉﾛｼﾞｰﾄﾗﾝｽﾌｧ   ﾄﾂｷﾖｼﾘｮｳ
ﾋﾝｼﾂﾎｼｮｳ *
ｷﾞｼﾞﾕﾂﾄｳﾆｭｳ   ｺｸｻｲｶｲｷﾞ   ｺｸｻｲﾚﾝｺﾞｳ   ﾃｸﾉﾛｼﾞｰﾄﾗﾝｽﾌｧ   ﾊﾂﾃﾝﾄｼﾞｮｳｺｸ
ﾎﾞｳｴｷ *
ｷｷﾞｮｳ   ｷﾞｼﾞﾕﾂｶｲﾊﾂ   ｹｲｴｲｶﾝﾘ   ｾｲｻﾝｶﾝﾘ   ｾｲｻﾝｷﾞｼﾞﾕﾂ   ﾃｸﾉﾛｼﾞｰﾄﾗﾝｽﾌｧ *
ｲｷｶﾞｲ   ｶﾝｷｮｳﾎｾﾝ   ｷﾞｼﾞﾕﾂｶｲﾊﾂ   ｹｲｻﾞｲｾｲﾁｮｳ   ｻﾝｷﾞｮｳｴｲｾｲ   ｼｮｳｼｹﾞﾝ
ﾃｸﾉﾛｼﾞｰﾄﾗﾝｽﾌｧ *
ｴﾈﾙｷﾞｰｼｹﾞﾝ   ｺｸｻｲｶｲｷﾞ   ｼﾔｶｲｼｭｷﾞ   ﾃｸﾉﾛｼﾞｰﾄﾗﾝｽﾌｧ   ﾐﾗｲｶﾞｸ   ROME-CLUB *
ｶｶﾞｸｾﾝｲｾｲｿﾞｳｷﾞｮｳ   ｷﾞｼﾞﾕﾂｷｮｳﾘｮｸ   ﾃｸﾉﾛｼﾞｰﾄﾗﾝｽﾌｧ *
ｷﾞｼﾞﾕﾂﾄｳﾆｭｳ   ｺｳｷﾞｮｳｼｮｳﾕｳｹﾝ   ﾃｸﾉﾛｼﾞｰﾄﾗﾝｽﾌｧ *
ｷﾞｼﾞﾕﾂｷｮｳﾘｮｸ   ｾｲｻﾝｾｲﾌﾞﾝｾｷ   ﾃｸﾉﾛｼﾞｰﾄﾗﾝｽﾌｧ *
ｵﾝﾗｲﾝｼｽﾃﾑ   ｼﾞｮｳﾎｳｹﾝｻｸｼｽﾃﾑ   ｼﾞｮｳﾎｳｻｰﾋﾞｽ   ﾃｸﾉﾛｼﾞｰﾄﾗﾝｽﾌｧ   ﾃﾞｰﾀﾍﾞｰｽ
ﾉｳﾊｳ   ﾏｰｹﾃｨﾝｸﾞ *
ｷﾞｼﾞﾕﾂｼﾝﾎﾟ   ﾃｸﾉﾛｼﾞｰﾄﾗﾝｽﾌｧ   ﾄﾂｷﾖｾｲﾄﾞ *
ｷｮｳｲｸｺｳｶ   ｷﾞｼﾞﾕﾂｷｮｳｲｸ   ｷﾞｼﾞﾕﾂｼﾔ   ｺｸｻｲｷﾖｳﾘｮｸ   ﾃｸﾉﾛｼﾞｰﾄﾗﾝｽﾌｧ *
ｺｳｷﾞﾖｳｼｮｳﾕｳｹﾝ   ﾃｸﾉﾛｼﾞｰﾄﾗﾝｽﾌｧ   ﾊﾂﾃﾝﾄｼﾞｮｳｺｸ   ﾗｲｾﾝｽ *
ｹｲﾔｸ   ｺｳｷﾞﾖｳｼｮｳﾕｳｹﾝ   ﾃｸﾉﾛｼﾞｰﾄﾗﾝｽﾌｧ   ﾄﾂｷﾖ   ｸﾞﾗｼﾞﾙ   ﾎｳｷ   ﾗｲｾﾝｽ *
ｷﾞｼﾞﾕﾂｶｲﾊﾂ   ﾁｼｷｼｭｳﾔｸｶ   ﾁｭｳｼｮｳｷｷﾞｮｳ ﾃｸﾉﾛｼﾞｰﾄﾗﾝｽﾌｧ *
ｷﾞｼﾞﾕﾂｶｲﾊﾂ   ｿﾋﾞｴﾄ   ﾃｸﾉﾛｼﾞｰﾄﾗﾝｽﾌｧ   COMECON *
ｱﾗﾌﾞｼｭﾁｮｳｺｸﾚﾝﾎﾟｳ   ﾃｸﾉﾛｼﾞｰﾄﾗﾝｽﾌｧ   ﾄﾂｷﾖｹﾝ ﾄﾂｷﾖｾｲﾄﾞ
```

ような時間と労力の負担の軽減するために，LEXそのものを使うことができる。

LEX入力用テクストの点検のポイントは，一般的に言ってふたつある。ひとつは原テクストが正しく入力されているかどうかという問題であり，他のひとつは識別値が正しく付けられているかどうかという問題である。前者については，入力テクストをパンチしたカードや端末から入力したデータを直接確かめるという方法もあるが，これは最も労力を要する方法であり，言語データのような大量データについては点検の精度も問題になりかねない。この場合には，作成した入力テクストの一部または全部を印刷して，これを点検する方が遙かに楽である。LEXには後述のようにPRINT RAW DATA命令という入力テクスト印刷のための命令があるので，この命令を使えばよい。

> 素データ印刷のためのユーティリティを使ってもよいが，一般にPRINT RAW DATA命令によるほうが，出力が読みやすく点検が楽である。

入力テクストの点検を行う方法としては次の方法もある。まずINPUT命令を使ってテクストをLEXに入力してしまう。次にFREQ命令か，ALPHA命令を使って出現頻

度1（あるいは大事をとれば頻度n以下）の単語をリスト・アップする。パンチ・ミスのある単語の頻度は経験的にきわめて小さいから，このようなミスの大部分は出力された単語のリスト中に含まれるはずである。誤りのある単語が発見されれば，入力テクスト中での位置は，PRINT WORD命令や他の索引索の命令を使って容易に知ることができる。この方法は単独では危険が大きいが，前述の方法と併用することにより，点検の労を相当に軽減することができる。

次に，識別値が正しく付けられているかどうかの点検であるが，このときにもまずテクストをLEXに入力し，PRINT WORD命令，PRINT LINE命令，VALUE LIST命令，WORD INDEX命令などを使って点検すればよい。

修正が終れば，テクストを再入力することになる* が，実際にはこの後で誤りが発見されることも少なくない。従って，万全を期すためには，上述の点検作業を何回か繰返すことが望ましい。

*同一のテクストを何度入力しても差支えないが，以前に作成されたテクスト・ファイルや行ファイルを削除しておくなどファイルの扱いに対する配慮が必要である。

Ⅳ　データの置換と選別

Ⅳ・1　データの置換・選別命令

テクストの処理において，テクスト全体が対象となることは言うまでもないが，テクストの特定の部分やテクスト中の特定の単語（群）が対象となることも少なくない。LEXではこのような場合に備えて，**データ置換・選別命令**という命令群がある。

データ置換・選別命令とは，テクストの特定の部分や，テクスト中の特定の単語(群)を処理の対象としたり，処理の対象から除外したり，特定の単語群を他の単語によって置換するための補助的命令群の総称である。

データ置換・選別命令は，LEX 文で単独で独立して用いることができず，常に他の実行命令のもとで，しかも定められた位置で用いなければならない。

データ置換・選別命令には大別して次の 5 種類がある。

REPLACE命令———単語（群）の別の単語による置換

SAMPLE命令————単語，LEX 行，テクスト・ユニットを単位とするサンプリング

SELECT命令，REJECT命令———識別値をキーとする単語，LEX 行，テクスト・ユニットの選別

INCLUDE命令，EXCLUDE命令———文字列，単語をキーとする単語（群）の選別

POSITION命令———テクスト構成要素内の単語の位置をキーとする単語の選別

PEPLACE命令を除く，他の命令を総称して**データ選別命令**と呼ぶ。

データ置換・選別命令はすべて任意的である。ただし，実行命令によっては，データ置換・選別命令の一部もしくはすべてが使用不可能なものもある。

データ置換・選別命令は，ひとつの LEX 文中で，即ち，ひとつの実行命令のもとで，任意の順序で，しかも何度でも用いることができる。しかし，同種のデータ置換・選別命令を複数回用いるときを含め，複数個のデータ置換・選別命令を用いるときには，次の点に注意が必要である。

LEXはデータ置換・選別命令を与えられた順にひとつづつ実行する。今，先頭から第i番目のデータ置換・選別命令をD_iとし，このデータ置換・選別命令の実行結果をT_iとする。またT_0はデータ置換・選別命令がひとつも実行されていないテクスト（通常はテクストファイル場合により後述のサブテクスト・ファイル）を表わすものとする。このとき，データ置換・選別命令実行の過程は図式的に次の図4・1のように表わすことができる。

図 4. 1　データ置換・選別命令の実行とテクストの変容

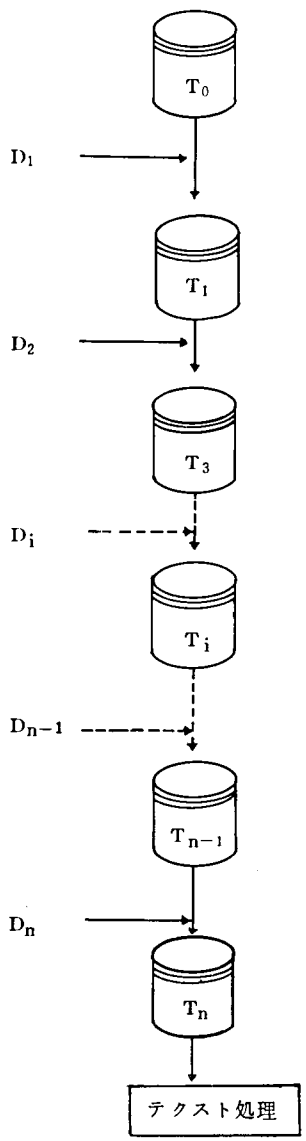


図4.1に示すように，第i番目のデータ置換・選別命令D_iは，それ以前のデータ置換・選別命令の累積的結果であるテクストT_{i-1}に対して実行され，その実行結果はテクストT_iとなる。見方を変えれば，テクストT_{i-1}が，D_iによって，テクストT_{i-1}に変えられると言うこともできるし，D_iはテクストT_{i-1}を入力（ファイル）とし，テクストT_iを出力（ファイル）とすると言うこともできる。また，ひとつの実行命令におけるデータ置換・選別処理の最終結果はテクストT_nであり，実行命令で指示されたテクストの具体的処理の対象となるのはこのテクストT_nである。

T_0を除き，データ置換・選別命令の実行結果であるテクストT_1～T_nはすべて一時的ファイルに格納される。

このようなデータ置換・選別命令の実行形態から，次の2点に関する注意が必要となる。ひとつは，データ置換・選別命令を与える順序そのものは任意であるが，与えられた順序は意味をもつということである。例えば，D_iで，テクストT_{i-1}中の単語Xをすべて単語Yに変え，D_{i+1}でテクストT_i中の単語Xを処理の対象として選ぶことは，無意味であるばかりでなく，この場合実際にエラーとなる。なぜなら，D_{i+1}が処理の対象とするテクストT_iでは，その前のD_iの実行の結果，単語Xは既に単語Yに変えられており，Xなる単語は存在しないからである。

第2の点は，**REPLACE**命令を除く他のデータ置換・選別命令，即ちデータ選別命令に関するものである。個々のデータ選別命令は，テクストの部分集合を定義するものであることは明らかだが，複数個のデータ選別命令の実行結果は，図4.1に示すようなテクスト処理のプロセスからして，個々のデータ選別命令によって定義されるテクストの部分集合の論理積となる。個々のデータ選別命令によるものであれ，このように論理積として定義されるものであれ，テクストの部分集合は空であってはならない。図4.1で言えば，テクストT_iは空であってはならない。空であれば，D_{i+1}もしくはテクスト処理の処理対象がなくなるからである。テクストT_iが空集合となったとき，LEXは直ちにエラー・メッセージを発行して処理を中断する。

図4.1から明らかなように，テクスト・ファイル（正確に言えば後述のサブテクスト・ファイルも含む）自体は，データ置換・選別命令を実行しても，もとの形のままで保存される。そして，次のテクスト処理にも入力時そのままの内容で使用される。これに対して，図4.1のT_iからT_nは，ひとつのテクスト処理の終了後には消去される。テクスト

T_iを何らかの形で保存しておきたいときには，次章で述べるCREATE SUBTEXT命令を用いる。この命令で作られたテクストT_iは，それ以後の処理では，テクストT_0として，テクスト・ファイルの代りに用いることができる。このテクストT_iをサブテクストファイルと称する。

Ⅳ・2 単語の置換 ——— REPLACE命令

書　式

```
 REPLACE
[MODE={S, N}]
[TYPE={P, G}]
[LIST={N, Y}]
 置換リスト
 %END
```

（置換リストはMODE，TYPEの指定により異なるので別項に掲げる）

機　能

置換リストに与えられた文字列，単語，あるいは単語番号（WSQ）に一致する単語を指定された単語によって置換する。文字列，単語あるいは単語番号（WSQ）をキーとして単語を単語によって置換する。置換内容を示すリストも出力できる。

MODE明細指示

置換のキー（手掛り）となる要素の種類を指定する。この明細指示が省略されたときには，Sが仮定される。

S：文字列もしくは単語を与え，これと一致する単語を指定された単語によって置換する。

N：単語番号（WSQ）を与え，この単語番号をもつ単語を指定された単語によって置換する。

単語番号（WSQ）は，システムがすべての単語に入力テクストでの出現順に与える，1から始まる一連番号で，特定の単語の単語番号は，PRINT WORD命令や他の索引系の命令を使って知ることができる。

TYPE明細指示

置換リストの形式を指定する。省略時解釈はPである。

P：置換のキーとなる文字列，単語あるいは単語番号と，置換後の単語を，一対一に対応させ，対として与える形式。個別リスト。

G：キーと置換後の単語を多対一に対応させて与える形式。一括リスト。

LIST明細指示

置換の内容をリストとして印刷するかどうかを指定する。省略時解釈はNである。

N：置換内容を印刷しない。

Y：置換内容を印刷する。

Yを指定したときには，実行命令の処理結果に先立ち，LEX文中に次のようなリストが出力される。ただし，後述の%ELSE明細指示を用いた置換の内容は出力されない。

```
**  REPLACE LIST  **
単語1（識別値）      REPLACED BY 単語2
・・・               REPLACED BY ・・・
・・・               REPLACED BY ・・・
```

このリストでは，単語$_1$に続けて，識別値が与えられるが，その順は，左から，単語

番号，行番号（ＬＳＱ），第１ユニットの識別値，第２ユニットの識別値，……の順である。このリストは，置換が意図通りに行われたことを確認するためのものであるが，置換の記録としての意味ももっている。

置換リスト

置換リストの書式は，ＭＯＤＥ，ＴＹＰＥの指定により４種類に分かれるので，個別的に説明する。

ＭＯＤＥ＝Ｓ，ＴＹＰＥ＝Ｐの場合

単語$_{11}$
単語$_{12}$
［単語$_{21}$
単語$_{22}$
・・ ・
・・ ・
単語$_{n1}$
単語$_{n2}$］　（ $1 \leqq n \leqq 100$ ）
［％ＥＬＳＥ＝単語］

この形式では，単語$_{i1}$が単語$_{i2}$によって置換される。このような単語の対は少なくともひとつ，最大100個まで与えることができる。％ＥＬＳＥ明細指示は，リストにキーとして与えられた単語$_{11}$～単語$_{n1}$のいずれにも一致しない単語を一括してひとつの単語に変えるときに与える。単語$_{11}$～単語$_{ni}$のいずれにも一致しない単語がすべて，％ＥＬＳＥ＝に続けて与えられた単語によって置換される。％ＥＬＳＥ明細指示を省略すれば，キーに一致しない単語は勿論もとのままである。

ＭＯＤＥ＝Ｓ，ＴＹＰＥ＝Ｇの場合

［ＰＯＳＩＴＩＯＮ＝（（Ｃ△△Ｂ△△Ｍ△△Ｅ））］
［文字列$_1$
文字列$_2$

```
[ [ ・  ・・ ]
  [ ・・  ・ ]
  [ 文字列n ]
  %BY
  単　語
[%ELSE＝単語]
```

POSITION明細指示

キーとなる文字列と，単語との一致方式を指定する。省略時にはCが仮定される。この明細指示を与えるときには，4つのキーワードのうちひとつ以上4個までを，任意の順序で与えればよいが，キーワード間には少なくともひとつの空白が必要である。文字列と単語との一致方式を示すキーワードの意味は次の通りである。

C：完全一致
B：前方一致，ただし完全一致は除く
M：中間一致，ただし完全一致，前方一致，後方一致はすべて除く
E：後方一致，ただし完全一致は除く

今，文字列 'IN' があり，単語として次の4語があるものとする。

IN，INDIA，SIN，KING

このとき与えられた文字列 "IN" と上記4つの一致方式のそれぞれにより一致する単語は次のようになる。

C （完全一致） [IN]
B （前方一致） [IN] DIA
M （中間一致） K [IN] G
E （後方一致） S [IN]

また，ＰＯＳＩＴＩＯＮ明細指示が，

Ｐ$\bar{O}$ＳＩＴＩ$\bar{O}$Ｎ＝Ｂ△△Ｃ

であれば，このとき文字列“ＩＮ”と一致する単語は，ＩＮとＩＮＤＩＡである。

この形式では，**文字列**は少なくとも１個，最大100個まで与えることができる。リストに与えられた文字列のうちのひとつと，ＰＯＳＩＴＩＯＮ明細指示で指定された方式により一致する単語が，％ＢＹステートメントの次のステートメントに与えられる単語によって置換される。一致方式として完全一致だけが指定されていれば，リストの文字列は実際には単語にほかならない。

％ＢＹステートメント

キー（となる文字列）のリストの終りを示し，次のステートメントに置換後の単語が与えられることを示す。このステートメントは省略してはならないし，また次のステートメントで新しい単語を与えなくてはならない。

％ＥＬＳＥ明細指示の意味は前項で述べた通りである。

ＭＯＤＥ＝Ｎ，ＴＹＰＥ＝Ｐの場合

単語番号$_{11}$［△△Ｔ$\bar{O}$△△単語番号$_{12}$］
単語$_1$
［単語番号$_{21}$［△△Ｔ$\bar{O}$△△単語番号$_{22}$］
単語$_2$
・・　・
・・　・
単語番号$_{n1}$［△△Ｔ$\bar{O}$△△単語番号$_{n2}$］
単語$_n$］　（$1 \leqq n \leqq 100$）
［％ＥＬＳＥ＝単語］

本項と次項はいずれもMODE＝Nの場合である。MODE＝Nのときは，キーとして，文字列の代りに単語番号を与えるが，ひとつのステートメントにおける単語番号の与え方には次のふたつの方法がある。

単独型：ひとつのステートメントにひとつの単語番号を与える。

TO型：ひとつのステートメントに，単語番号$_1$ TO 単語番号$_2$という形式で与える。このときこのステートメントは，単語番号$_1$以上，単語番号$_2$以下のすべての単語番号を表わすことになる。従って，単語番号$_1$は単語番号$_2$より小さいか，等しくなくてはならない。等しいときは，単独型と同じことになり，エラーにはならないが，TO型で与える意味がない。

また，この型では，TOの前後にそれぞれ少なくともひとつの空白が必要である。

MODE＝Nのときには，ステートメントの順序と単語番号$_{i1}$の大小とは無関係であり，単独型，TO型がどのような順で混在していてもよい。

単独型はTO型の特殊な場合であり，TO型で

単語番号$_{i1}$＝単語番号$_{i2}$

となる場合に他ならない。従って，MODE＝N，TYPE＝Pのときには，上掲の書式からも明らかなように，単語番号$_{i1}$以上，単語番号$_{i2}$以下のすべての単語が，直後のステートメントに与えられた単語，即ち単語$_i$によって置換される。

任意の単語の単語番号xについて，単語番号$_{i1} \leqq x \leqq$単語番号$_{i2}$，単語番号$j_1 \leqq x \leqq$単語番号j_2がともに成り立つときには，前に与えられたステートメントが優先される。

このような単語番号と単語の対は少なくとも1個，最大100個まで与えることができる。

%ELSE明細指示の意味は前項までの記述に準ずる。

MODE＝N，TYPE＝Gの場合

単語番号$_{11}$［△△TO△△単語番号$_{12}$］

```
[単語番号21 [ △△ TO△△単語番号22 ]
 ・・ ・
 ・・ ・
 単語番号n1 [ △△ TO△△単語番号n2 ]]  (1 ≦ n ≦ 100)
 %BY
 単 語
[%ELSE=単語 ]
```

この形式でも，単語番号の与え方，その数については前項で述べたことが当てはまるが，この形式では，リストに与えられた単語番号の任意のひとつについて，単語番号$_{i1}$以上，単語番号$_{i2}$以下となる単語が，%BYステートメントの直後のステートメントに与えられる単語によって置換される。

%BYステートメント，%ELSE明細指示の意味，用法はこれまで述べた通りである

%END明細指示は，ひとつのREPLACE命令の終りを示すものであり，必ず与えなくてはならない。

注意事項

a　REPLACE命令の置換リストにキーとして与えられた文字列，単語あるいは単語番号のすべてについて，テクスト中に該当するものがなく，単語の置換が一度も行われない場合も，エラーとはならない。置換の有無とその内容は，LIST明細指示でYを選択することにより確認できる。

b　複数のREPLACE命令あるいはデータ置換・選別命令を用いるときには，任意のREPLACE命令の後では，単語が既に指定された形に変わっていることを忘れてはならない。（Ⅳ・1も参照）

c　キーとして与えるべき文字列，単語あるいは単語番号の数が100を越えるときには，複数のREPLACE命令を用いる。このときの順序は任意であるが，前項の注意を念頭に置く必要がある。

d　MODE=Sを指定した単独あるいは複数のREPLACE命令による置換を実行した

場合，MODE=Nを指定した単独もしくは複数のREPLACE命令によって常に同じ結果を得ることができる。しかし，逆は一般には成立たない。

e　LEX行ファイルが作成されているとき，REPLACE命令を実行すると，前掲図4.1のテクストT_i中の単語と行ファイル中の単語のくいちがいが生ずる。例えばREPLACE命令で，'MAKES,''MADE,''MAKING'といった単語をすべて'MAKE'に変えたとしても，LEX行ファイル中ではこれらの単語はもとの形のままである。CONCORDANCE命令のように，行ファイルを使うテクスト処理では，このことは一応念頭に置く必要がある。

勿論，このくいちがいを利用して，見出語としては'MAKE'に標準化しておき，この見出語のもとに，用例中では'MAKES'等のテクスト中での原形を示すといった利用法もある。

f　%BY明細指示の次のステートメントに与える単語と，%ELSE明細指示に与える単語を含め，置換リストに与える文字列や単語はセミコロンを除くどのような文字，数字，記号から成っていてもよい。ただし，

ア　文字列，単語の最初が空白であってはならない，最後の部分の空白は無視される。

イ　文字列，単語が%BY，%END，%ELSEで始まっていてはならない。

従って，次のような文字列，あるいは単語も許される。

```
* EXAMPLE  1
PEACE  RESEARCH
##??!!
```

g　リストに与える文字列と単語については，%BYの次のステートメントに与える単語と%ELSE明細指示で与える単語も含め，INPUT命令のWORD LENGTH明細指示で与えた仮想最大長を越えてはならない。また，MERGE WORD命令実行後であれば，実質最大長を越えてはならない。いずれの場合も，文字列や単語の後の空白は長さに関係しない。

使用例

a　'A'で始まるすべての単語を，'A－WORD'という単語に変える。単語'A'

も同様に変える。

```
＜例 4.1＞
REPLACE
MODE=S
TYPE=G
LIST=Y
POSITION=B△△C
A
%BY
A−WORD
%END
```

この例では，MODE=Sは必ずしも必要ではない（Sは標準値）。また，置換内容のリストが不要であれば，LIST明細指示も不要となる。POSITION明細指示でBのみを選択すれば，単語“A”は置換されない。

b　BED，INDEED，DEEDの3語を除き，EDで終るすべての単語をVERBに変える。（Ⅳ．5使用例b参照）

```
＜例 4.2＞
REPLACE                    ・・・①
BED                        ・・・②
BED*                       ・・・③
INDEED                     ・・・④
INDEED*                    ・・・⑤
DEED                       ・・・⑥
DEED*                      ・・・⑦
%END                       ・・・⑧
REPLACE                    ・・・⑨
TYPE=G                     ・・・⑩
POSITION=E                 ・・・⑪
```

```
ED                    ・・・⑫
%BY                   ・・・⑬
VERB                  ・・・⑭
%END                  ・・・⑮
REPLACE               ・・・⑯
BED*                  ・・・⑰
BED                   ・・・⑱
DEED*                 ・・・⑲
DEED                  ・・・⑳
INDEED*               ・・・㉑
INDEED                ・・・㉒
%END                  ・・・㉓
```

この例では，最初のREPLACE命令（①～⑧）でBED，INDEED，DEED，をそれぞれ別の形に変えている。これは2番目のREPLACE命令（⑨～⑮）で,これらの単語が置換されるのを防ぐためである。③，⑤，⑦で与えた単語は例示のためであって，3番目のREPLACE命令（⑯～㉓）で正しく復原できるものであれば，実際にはどのような形でもよい。2番目のREPLACE命令でEDで終る単語をすべてVERBに変える。BED等は最初のREPLACE命令によりEDでは終らない形に変えてあるので，VERBに変えられることはない。3番目のREPLACE命令はBED等をもとの形に戻すためのものである。

BED等の出現頻度があまり大きくなければ，この例のようにMODE=S だけで行うより，MODE=Nとして単語番号を使うREPLACE命令を混じえたほうが便利な場合もある。

BED等の一部はVERBに，残りはそのままの形で残したいといった場合も，同様に文字列をキーとする方法と単語番号をキーとする方法を併用することになる。

c　単語番号1から5，13，101から123，879の単語を“EXAMPLE”に変え,他の語はすべて“OTHERS”に変える。

```
＜例 4.3＞
REPLACE
MODE=N
TYPE=G
101 △△ TO △△ 123
13
1△△TO△△5
879
%BY
EXAMPLE
%ELSE=OTHERS
%END
```

Ⅳ・3　サンプリング――SAMPLE命令

書　式

```
  SAMPLE
  KEY={WSQ, LSQ, UN1, UN2, UN3, UN4}
[INTERVAL=i ]     (2 ≧ i ≧ 100) ＜i=10＞
[FIRST=j ]     (1 ≧ j ≧ i)
```

機　能

テクスト構成要素を単位とする等間隔サンプリングを行う。

KEY明細指示

サンプリングの単位となるテクスト構成要素を指定する。この明細指示は省略できない。また，存在しないテクスト・ユニットを表わすキーワードを指定してはならない。

LEXでは，テクスト構成要素を表わすWSQ等のキーワードは常に次の意味で用いられる。

WSQ：単語（または単語番号）

LSQ：LEX行（またはLEX行番号）

UN1：第1ユニット（またはその識別値）

UN2：第2ユニット（またはその識別値）

UN3：第3ユニット（またはその識別値）

UN4：第4ユニット（またはその識別値）

INTERVAL明細指示

サンプリングの間隔を，2以上，100以下の整数で指定する。省略時は$i=10$が仮定される。省略時も含め，INTERVAL=iとすると，テクスト構成要素i個につき1個がサンプルとして抽出される。従って抽出されたテクスト構成要素の間隔そのものは，$(i-1)$となる。

FIRST明細指示

サンプリングの初期値，即ち，KEY明細指示で指定したテクスト構成要素の何番目のものから始めるかを正整数で指定する。与える数値は1以上で，INTERVAL明細指示で与えた値（i）以下でなくてはならない。この明細指示が省略されたときには，システムが，$1 \leqq j \leqq i$の範囲で，初期値jをランダムに定める。従って，FIRST明細指示が省略されたときには，同一のSAMPLE命令を実行しても同じ結果が得られるとは限らない。

サンプリングの方法

今，KEY明細指示で，テクスト構成要素TU_xが指定され，省略時も含め，INTERVAL=i，FIRST=jとする。この時サンプルとして抽出されるのは，テクスト（図4.1におけるT_i）での出現順に，

$j+i\times N$

（但し，Nは整数で，$N \geqq 0$）

番目のテクスト構成要素である。

使 用 例

a　テクスト中の単語数が多いので，1／10に減らして，見出語の頻度分布の傾向を見たい。このために必要なSAMPLE命令は次の例4.4のようになる。

```
＜例 4. 4＞
SAMPLE
KEY=WSQ
INTERVAL=10
```

この例では，標準値10を指定しているから，INTERVAL明細指示はなくてもよい。

Ⅳ・4　識別値を用いたテクスト構成要素の選別 ——SELECT命令とREJECT命令

書　　式

```
{ SELECT, REJECT }
KEY={WSQ, LSQ, UN1, UN2, UN3, UN4}
識別値11 [ △△ TO △△識別値12 ]
[識別値21 [ △△ TO △△識別値22 ]]
・・・
・・・
[識別値n1 [ △△ TO △△識別値n2 ]]    ( 1 ≦ n ≦ 100 )
%END
```

機　　能

与えられた識別値をキーとして，指定されたテクスト構成要素の選別を行う。

SELECT命令は，識別値リストに与えられた識別値をもつテクスト構成要素を選ぶ

REJECT命令は，識別値リストに与えられた識別値をもつテクスト構成要素を除外する。

KEY明細指示

選別の単位となるテクスト構成要素を指定する。この明細指示は省略できないし，また存在しないテクスト・ユニットを指定してはならない。テクスト構成要素を表わすWSQ等のキーワードの意味については前節SAMPLE命令のKEY明細指示を参照。

識別値リスト

ここで与える識別値は，単語番号，LEX行番号，あるいはテクスト・ユニットの識別値であるが，KEY明細指示で指定したテクスト構成要素に対応する識別値でなくてはならない。即ち，KEY明細指示で，WSQを指定したならば単語番号，LSQを指定したならばLEX行番号，UNiを指定したならば第iユニットの識別値，を与えなければならない。これに違反しても必ずしもエラーとはならないが無意味である。

ひとつのステートメントにおける識別値の与え方には**単独型**とTO型があり，意味，用法とも，REPLACE命令のMODE=Nの場合（Ⅳ・2）と同じである。ただし，識別モードがAモードのテクスト・ユニット単位で選別を行うことをKEY明細指示で指定したとき，$\overline{\text{TO}}$型を使うことは，ごく稀な場合を除き無意味である。

単独型であれ，TO型であれ，識別値$_{i1}$は少なくとも1個，最大100個まで与えることができる。

SELECT命令では，リストの中の少なくともひとつ識別値について，識別値$_{i1}$に一致するか，識別値$_{i1}$以上，識別値$_{i2}$以下の識別値をもったテクスト構成要素が選ばれる。

REJECT命令はこの逆である。

%END明細指示

ひとつのSELECT命令，もしくはREJECTの終りを示す。省略してはならない。

使 用 例

a　第2ユニットの識別値が“USA”か“USSR”であり，しかも第3ユニット（S

モード）の値が，1977，1978，1979，1980であるデータだけを選ぶ。

```
＜例 4.5＞
SELECT
KEY=UN2
USA
USSR
%END
SELECT
KEY=UN3
1977 △△ TO △△ 1980
%END
```

この例では，ふたつのSELECT命令の順序は任意である。また，第3ユニットの識別値が1980であるもののうち，第4ユニットの識別値が“REAGAN”であるものを除きたいのであれば，この例の後に，

```
REJECT
KEY=UN4
REAGAN
%END
```

とすることもできるが，このREJECT命令を使うと，第3ユニットの識別値が1979で，かつ，第4ユニットの識別値が，“REAGAN”であるものが存在すれば,これも除去されてしまう。このときには，第3ユニットの識別値が1980，第4ユニットの識別値が“REAGAN”となる部分のLEX行番号を指定して，

```
REJECT
KEY=LSQ
989 △△TO△△1011
```

```
・・・
%END
```

といった形のREJECT命令を与えればよい。

b　第1ユニットの識別値が空白と，“SD”と“Q”の3種類しかないとき，識別値が空白である場合について，単語のサンプリングを1/3の割合で行う。

```
<例4.6>
REJECT
KEY=UN1
SD
Q
%END
SAMPLE
KEY=WSQ
INTERVAL=3
```

この例では識別値が空白であるという特殊な条件を扱かっている。識別値が空白のときには，直接に参照できないので，空白でないものをすべて参照することにより間接的に空白のものを参照することになる。この例のREJECT命令により，SELECT命令を使って，識別値が空白である第1ユニットを選ぶのと同じ結果が得られる。

Ⅳ・5　単語の選別 ——INCLUDE命令とEXCLUDE命令

書　式

```
{INCLUDE, EXCLUDE}
[POSITION=((C△△B△△M△△E))]
```

文字列$_1$
[文字列$_2$
・・・
・・・
文字列$_n$]　　($1 \leq n \leq 100$)
%END

機　能

単語の選別を行う。INCLUDE命令は，与えられた文字列と，指定された形式で一致する単語を処理の対象とする。EXCLUDE命令は，与えられた文字列と，指定された形式で一致する単語を処理から除外する。

POSITION明細指示

文字列と単語の一致形式を指定する。この明細指示の意味，用法は，REPLACE命令の，MODE=S，TYPE=Gのときの置換リストにおけるPOSITION明細指示とまったく同じである。（Ⅳ・2参照）。なお，省略時はC（完全一致）が仮定される。

文字列リスト

文字列は少なくとも1個，最大100個まで与えることができる。ここに与えられた文字列のうちのひとつと，POSITION明細指示で指定された形式で一致する単語が選別の対象となる。POSITION=Cであれば，文字列は実際には単語に相当することになる。

%END明細指示

ひとつのINCLUDE命令もしくはEXCLUDE命令の終りを示すステートメントであり，省略することはできない。

注意事項

PRINT LINE，LINE LIST，LINE INDEX，KLICの各命令の場合，一般の場合と次の点で異なる。

a　POSITION明細指示は使用できない。

b　リストに与える文字列の長さは途中の空白も含め最大60文字まで許される。どのような文字，記号，数字から成っていてもよいがセミコロンを含んではならない。また，先頭，末尾の空白は文字数には関係がない。

c　INCLUDE命令では，与えられた文字列のうちのひとつを含むLEX行が処理の対象となり，EXCLUDE命令では処理から除外される。

使 用 例

a　“DEVELOP”という文字列を含むすべての単語を選ぶ。但し，“DEVELOPER”と“DEVELOPERS”という単語は除く。

```
<例 4.7>
INCLUDE
POSITION=C △△ B △△ M △△ E
DEVELOP
%END
EXCLUDE
DEVELOPER
DEVELOPERS
%END
```

EXCLUDE命令では，完全一致だけが問題になっているから，POSITION 明細指示が省略されている。このEXCLUDE命令は，完全一致と前方一致を指定して，

```
EXCLUDE
POSITION=C  B
DEVELOPER
%END
```

としてもよい。しかし，このときには，“DEVELOPER'S”も除外される。

b　“ED”で終るすべての単語を選ぶ。ただし,“BED”“DEED”“INDEED”は除くが,“BED”のうち単語番号589と10124のBEDは除外しない。
（Ⅳ・2使用例bを参照）

```
<例4.8>
REPLACE
TYPE=G
BED
DEED
INDEED
%BY
DUMMY
%END
REPLACE
MODE=N
TYPE=G
589
10124
%BY
BED
%END
INCLUDE
POSITION=E
ED
%END
```

最初のREPLACE命令はBED等を，最後のINCLUDE命令で選ばれない形に変えるためのものであって，この例のように“DUMMY”という形でなくても“ED”で終らなければどのような形でもよい。次のREPLACE命令は，単語番号589と10124の単語を“BED”という形に戻すものである。最後のINCLUDE命令は，POSITION

明細指示でEを指定し，EDと後方一致する単語を選ぶことを指示している。

Ⅳ・6　テクスト構成要素内の位置による単語の選別――POSITION命令

書　式

```
  POSITION
  KEY={ LSQ, UN1, UN2, UN3, UN4 }
[TYPE={ S, E }]
[LENGTH=i]      ( i ≧ 2 )   < i = 100 >
[%FORWARD
 位置11  [ △△TO△△位置12 ]
[位置21  [ △△TO△△位置22 ]]
 ・・・
 ・・・
[位置m1  [ △△TO△△位置m2 ]]          ( 1 ≦ m ≦ 100 )
 %END
[%BACKWARD
 位置11  [ △△TO△△位置12 ]
[位置21  [ △△TO△△位置22 ]]
 ・・・
 ・・・
[位置n1  [ △△TO△△位置n2 ]]          ( 1 ≦ n ≦ 100 )
 %END]
```

機　能

%FORWARD，または%BACKWARDに続けて指定されたテクスト構成要素内の位置にある単語を処理の対象として選ぶか，処理の対象から除外する。選択，除外は，明細指示によって指示する。

KEY明細指示

単語の位置を決定する枠組となるテクスト構成要素を略称で指定する。略称の意味については，Ⅳ・3のSAMPLE命令のKEY明細指示を参照。但し，このPOSITION命令では，WSQ（単語番号）は指定できない。また，この明細指示は省略できない。

この明細指示で指定されたテクスト構成要素内での位置が単語を選別するときのキーとなる。

TYPE明細指示

選択処理か除外処理かを指定する。省略時はSが仮定される。

S：指定されたテクスト構成要素中の，指定された位置にある単語を処理の対象として選ぶ。

E：指定されたテクスト構成要素中の，指定された位置にある単語を処理の対象から除外する。

LENGTH明細指示

KEY明細指示で指定したテクスト構成要素に含まれる仮想最大単語数を指定する 指定は，2以上の正整数で行う。省略時には，LENGTH＝100が仮定される。省略時も含め，ここで指定した単語数を越えるテクスト構成要素があるときにはエラーとなる。心配なときには，UNIT LENGTH命令（後述）によって，当該テクスト構成要素中の単語数を予め確めておくのが望ましい。エラーのときには次のようなメッセージが出力される。

** TOO MANY WORDS **

位置リスト

%FORWARD，または%BACKWARDで始まり，%ENDで終る。この位置リストの書式自体は，SELECT命令，REJECT命令の識別値リストと同じである。ただ，このリストでは，位置$_{ij}$は整数値で与え，KEY明細指示で指定されたテクスト構成要素中での単語の位置を示す。しかも，この位置は，LEXテクスト・ファイル中における位置ではなく，テクストT_i中での相対的位置である。従って，POSITION命令の

実行以前に，何らかの形で単語の選別が行われているときには，テクスト・ファイルT_0における位置と，T_iにおける位置が一致しないことがありうる。

位置リストとしては，%FORWARDで始まり，%ENDで終るリストと，%BACKWARDで始まり，%ENDで終るリストのいずれか一方，または両方を与えることができる。しかし，両方を与える時には書式に示した順に与えなければならないし，また両方を省略することはできない。

%FORWARDで始まるリストには，テクスト構成要素の先頭からの位置を与える。これに対して，%BACKWARDで始まるリストには，テクスト構成要素の末尾からの位置を与える。従って，テクスト構成要素，例えば，第2ユニットの，最初の単語と最後の単語だけを処理の対象としたければ，POSITION命令の位置リストは次のようになる。

```
%FORWARD
1
%END
%BACKWARD
1
%END
```

注意事項

POSITION命令は何度用いても，他のデータ置換・選別命令と併用してもよいが，POSITION命令を含め，単語の選別を伴なうデータ選別命令を実行したあとで実行するときには，テクスト構成要素中の単語の位置はテクスト・ファイルあるいはテクストT_0におけるそれとは異なっている可能性があることを忘れてはならない。例えば，

```
EXCLUDE
AND
%END
POSITION
```

```
KEY=UN1
%FORWARD
1
%END
```

というようなLEX文があるとき，POSITION命令実行前のEXCLUDE命令で，第1ユニットの先頭にある“AND”も削除される。従って，“AND”が先頭にあった第1ユニット内では，本来“AND”の次に位置していた単語が先頭に来ることになる。

使 用 例

a　第1ユニットの最後の単語を選ぶ

```
<例 4. 9 >
POSITION
KEY=UN1
LENGTH=20
%BACKWARD
1
%END
```

テクストが韻文で，第1ユニットが詩行であれば，このPOSITION命令は，脚韻語のみを処理対象として選択することを意味する。

b　第2ユニットの最初の10語と最後の10語を除いた単語を選び，かつ，“AND”“THAT”“THE”は処理から除く。

```
<例 4. 10 >
POSITION
KEY=UN1
TYPE=E
LENGTH= 200
```

```
%FORWARD
1 △△ TO△△ 10
%END
%BACKWARD
1 △△ TO △△ 10
%END
EXCLUDE
AND
THAT
THE
%END
```

この例で，POSITION命令とEXCLUDE命令の順序を入替えると，注意事項で述べたような問題が生ずる。

V　テクストの恒久的変容

V．1　LEXで作成されるファイル

ファイルはレコード（データ）の集合である。入力テクストがLEXに入力されるとLEXテクスト・ファイルが作成されることは既に述べたが，このLEXテクスト・ファイルもまたひとつのファイルであり，LEXでは常に**LEXSYS**という**ファイル名称**によって参照される。LEXではテクスト・ファイルを様々に変容して利用することができる。前章で述べたデータの置換と選別も，このひとつの例と考えることもできる。しかし，データ置換・選別命令の効果は一回のテクスト処理の間だけ持続するものであり，この意味で一時的である。これに対して，本章では，テクストの恒久的変容の機能，即ち，LEXテクスト・ファイルから新しいファイルを作る機能を扱かう。

LEXで作成され，かつLEXで処理可能なファイルには，内容に従えば，次の5種類がある。

LEXテクスト・ファイル
LEX行ファイル
サブテクストファイル
見出語ファイル
単語アルファベット順ファイル

この5種類のファイルがどのようにして作成されるか，その関係を示したのが図5.1である。**LEXテクスト・ファイル**と**LEX行ファイル**は，INPUT命令によって作成される。**サブテクスト・ファイル**はテクスト・ファイルから後述のCREATE SUBTEXT命令によって作成される。これに加えて，任意のサブテクスト・ファイルから，ほとんど無限に新しいサブテクスト・ファイルを作成することができる。（図4.1に示した，データ置換・選別命令によるテクスト変容のプロセスと同じプロセスと考えてよい。）**見出語ファイル，単語アルファベット順ファイル**は，それぞれCREATE KEYWORD命令，CREATE ALPHA命令により，テクスト・ファイルあるいはサブテクスト・ファイルから作成される。テクスト・ファイル，サブテクスト・ファイルがほとんどすべての実行命令によって処理できるのに対し，見出語ファイル，単語アルファベット順

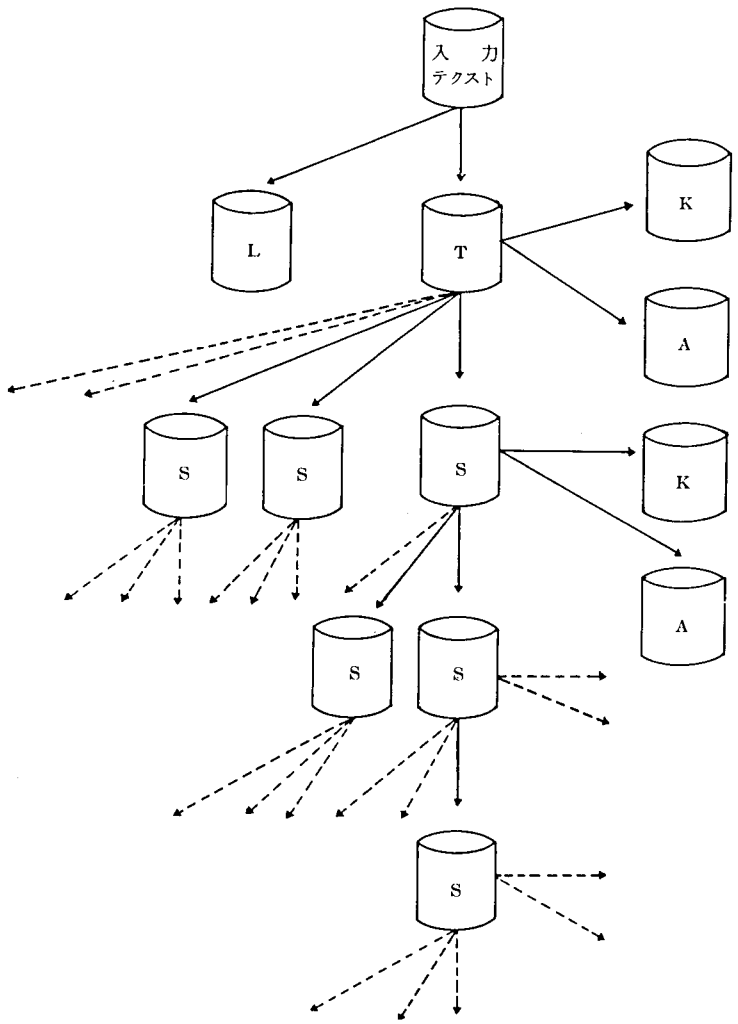


図 5.1　LEX によるファイルの作成

T：LEXテクスト・ファイル
L：LEX行ファイル
S：サブテクスト・ファイル
K：見出語ファイル
A：単語アルファベット順ファイル
⟶と ⇢ は新しいサブテクスト・ファイル，見出語ファイル，または単語アルファベット順ファイルが作成されうることを示す。
ファイルの媒体は通常磁気ディスクまたは磁気テープ

ファイルを処理できる命令は特定の実行命令に限られる。また，テクスト・ファイル，サブテクスト・ファイルからは新しいサブテクスト・ファイルを作ることができるのに対し，見出語ファイル，単語アルファベット順ファイル，そしてLEX行ファイルから

は新しいファイルを作ることができない。

次にファイルの構造という観点からすると，LEX 行ファイルを除く，他の 4種のファイルの最初の 1 レコードにはテクストに関する様々な情報が記録されている。その詳細はINFO命令（ XIV . 2 ）で述べる。

テクスト・ファイル，サブテクスト・ファイル，単語アルファベット順ファイルについては，2 番目以降のレコードの構造は同じであるが，単語アルファベット順ファイルでは単語がテクストでの出現順ではなく，ファイルの先頭からアルファベット順に並べられている。この 3 種のファイルの 2 番目以降の各々のレコードには次のデータが記録されている。

単語の LEX 行中の位置
単語長
単　語
単語番号
LEX 行番号（行ファイルがなければ 0 ）
テクスト・ユニット識別値
テクスト・ユニットのシステム識別値

これに対して見出語ファイルの 2 番目以降の各レコードには次のデータが記録されている。

見出語
当該見出語の頻度
（この頻度は，当該見出語ファイルが作られたテクスト・ファイルまたはサブテクスト・ファイル中，あるいはいずれかにデータ置換選別命令を施したテクスト中の頻度）

LEX 行ファイルでは最初のレコードから，LEX 行が入力テクストでの出現順に各レコードに識別値とともに記録されている。

V．2　新しいファイルの作成 ——CREATE命令

CREATE命令は，CREATE SUBTEXT, CREATE KEYWORD, CREATE ALPHAの3つの命令の総称である。

書　式

```
CREATE△{SUBTEXT, KEYWŌRD, ALPHA}
[WŌRK=i ]   (i ≧10)   〈 i =100 〉
[データ置換・選別命令]
```

機　能

新しいファイルを作成する。この命令によって作成されたファイルは後にLEXで処理することができる。

CREATE SUBTEXT命令はテクスト・ファイルまたはサブテクスト・ファイルから新たなサブテクスト・ファイルを作成する。このファイルはPRINT RAW DATA命令を除く他のすべてのLEX実行命令によって処理することができる。またこのファイルには，データ置換・選別命令によって削除されなかったすべての単語が原テクストでの出現順に記録されている。

CREATE KEYWORD命令は見出語ファイルを作成する。入力ファイルはテクスト・ファイル，サブテクスト・ファイルのいずれであってもよいし，また，データ置換・選別命令を用いてもよい。このファイルは後に，FREQ, ALPHA等単語の頻度に関する実行命令の入力として用いることができる。一般に，単語の頻度に関する命令を同一テクストについて複数回用いる場合には，この命令を使って見出語ファイルを作成しておいたほうが処理時間の短縮ができる。

CREATE ALPHA命令は，テクスト・ファイルまたはサブテクスト・ファイルから単語アルファベット順ファイルを作成する。このファイルは，WORD INDEX等索引系の実行命令の入力として用いることができる。このファイルは一種の転置ファイルであって，単語はファイル中ではアルファベット順に配列されており，同一の単語については入力テクストでの出現順に配列されている。

作業領域定義命令

ソーティングのための作業領域の大きさを指定する。CREATE SUBTEXT 命令では，与えてもエラーにはならないが，実際には無意味である。

一般に作業領域定義命令では，WORK=に続けて，ソーティングに使う領域の大きさを，10以上の整数値で指定する。単位はKB(キロバイト)である。省略時には，WORK=100,即ち，ソーティングのための作業領域を100 KBとることを仮定する。ここで指定しうる最大値についてはLEX自体には特に制限はないが，計算機システムでは制限値が設けられているので，この制限の範囲内で指定しなければならない。また，実際のデータに対して，必要以上の値が指定されているときには，ソーティングを行う直前に，必要な領域だけを割りあてるから，利用者はこの点の無駄使いを心配する必要はない。

データ置換・選別命令

前章に述べたデータ置換・選別命令の規則や注意事項に従えば，データ置換・選別命令は自由に用いてよい。勿論まったく用いなくてもよい。CREATE SUBTEXT命令で，データ置換・選別命令をまったく用いなければ，もとのファイルをそのままコピーすることになる。従って計算機システムに同様の機能が備わっていなければ，テクスト・ファイル，サブテクスト・ファイルの複写のためにCREATE SUBTEXT命令を使うことができる。また，計算機システムに編集機能がない場合も，CREATE SUBTEXT命令のもとで，例えばREPLACE命令を用いることにより，テクストの修正，編集を行うことができる。

注意事項

a　CREATE命令の入力ファイルはテクスト・ファイル，サブテクスト・ファイルのいずれであってもよいが，ファイル名称(ｄｄ名)は常にLEXSYSでなくてはならない。(詳しくはXV章を参照)

b　出力用ファイル，即ち，新しく作成されるファイルのためのDD文(ファイル定義の命令)は利用者が与えなければならない。このファイルの名称は常にLEXSUBである。(詳しくはXV章を参照)

使 用 例

a 'ED'で終る単語のみからなるサブテクスト・ファイルを作成する。

```
〈例 5.1〉
CREATE△SUBTEXT
INCLUDE
POSITION=E
ED
%END
```

b 'AMD'を'AND', 'THR'を'THE'に修正して新しいサブテクスト・ファイル（場合により事実上のテクスト・ファイル）を作成する。

```
〈例 5.2〉
CREATE△SUBTEXT
REPLACE
LIST=Y
AMD
AND
THR
THE
%END
```

c 見出語ファイルを作成する。

```
〈例 5.3〉
CREATE KEYWORD
```

入力ファイルがテクスト・ファイルであれば，テクスト中のすべての見出語を含む見出語ファイルが作成されるが，サブテクスト・ファイルであれば，テクストの特定の

部分等についての見出語ファイルを作ることもできる。

d　単語アルファベット順ファイルを作る。テクストが長く，単語総数が多いので，まず，'A', 'B'で始まる単語についてのみのファイルを作成する。

```
〈例 5.4〉
CREATE△ALPHA
INCLUDE
POSITION=B△△C
A
B
%END
```

Ⅴ．3　単語の併合—— MERGE　WORD命令

書　式

```
 MERGE △WORD
[データ置換・選別命令]
[CONNECT=連結記号]
[BOUNDARY={LSQ, UN1, UN2, UN3, UN4}]
 単語11
 単語12
[単語21
 単語22
 ・・・
 ・・・
 単語n1
 単語n2]        (1 ≦ n ≦ 100)
 %END
```

機　　能

テクスト中の連続したふたつの単語を一語に併合した新しいファイルを作成する。

データ置換・選別命令

どのようなデータ置換・選別命令を用いてもよいが，併合の対称となるのは，単語番号の連続した2つの単語であるから，単語の選別を行うINCLUDE命令などを使うときには細心の注意を要する。また，データ選別命令を用いたときには，新しく作成されるファイルは，データ選別命令の処理結果と同様，入力テクストの一部からなるテクストである。この意味で，MERGE WORD命令で，データ選別命令を用いた結果は，CREATE SUBTEXT命令でデータ選別命令を用いた場合と同じである。しかし，MERGE WORD命令のときには，単語が併合されていることと，新しく作成されるサブテクスト・ファイルでは単語番号が1から始まる一連番号に新たに付け変えられていることの2点が異なる。CREATE SUBTEXT命令では，単語番号はテクスト・ファイルのままである。

従ってMERGE WORD命令で，まったくのダミーの単語リストを与え，つまり，単語の併合は行わないで，データ選別命令を実行すると，CREATE SUBTEXT 命令と同じ結果が得られ，かつ，単語番号を1から新しく付け変えることができる。

この命令でデータ選別命令を用いると，単語番号が1から始まる一連番号に付け変えられるので，後に，共出現や，単語列の処理のような単語の隣接性が問題になる実行命令を実行する可能性のあるときには十分な注意が必要である。

CONNECT明細指示

CONNECT= に続けて，併合されるふたつの単語を連結すべき記号をひとつ指定する。セミコロン以外のどのような文字，記号を用いてもよい。省略時には**空白**が仮定される。例えば，UNITEDとNATIONSを一語に併合するとき，

CONNECT=－

であれば，新しい単語は，

UNITED−NATIONS

となり，この明細指示を省略したときには，

UNITED NATIONS

となる。

BOUNDARY明細指示

テクスト構成要素を略称で指定する。但し，単語併合の定義からして，WSQは指定できない。与えられた2つの単語が，ここで指定されたテクスト構成要素の境界をまたがるときには併合が行われない。この明細指示を省略したときには，テクスト構成要素の境界に関わりなく，単語の併合が行われる。

単語リスト

ここで与える単語リストの書式は，REPLACE命令の，MODE=S，TYPE=P の場合の置換リストとほぼ同じである。（Ⅳ．2。）単語$_{i1}$と単語$_{i2}$という単語の対は最大100組まで与えることができる。リストの終りには%ENDというステートメントが必要である。

このリストで与えられた単語$_{i1}$と単語$_{i2}$の対につき，テクスト中で単語$_{i1}$，単語$_{i2}$の順で出現し，しかも，単語番号が連続しているとき，

単語$_{i1}$＊単語$_{i2}$

（＊はCONNECTで指定した連結記号）

という単語に変えられる。但し，この長さが32文字を越えるとエラーになる。

注意事項

a　この命令の入力ファイルはテクスト・ファイル，サブテクスト・ファイルのいずれ

であってもよいが，ファイル名称（dd名）は常にLEXSYSとしなければならない。また新しく作成されるファイル（出力ファイル）のためのファイル定義文（DD文）を利用者が定義しなければならない。出力ファイルのファイル名称は常にLEXSUBである。

b　この命令では，3語以上の併合を行うことができない。一般に3語以上の併合を行う場合は，REPLACE命令を使うか，入力テクスト自体を直接修正する方が便利である。勿論，MERGE WORD命令を使ったジョッブを何回か行えば，3語以上の併合も可能である。

c　この命令の実行後の単語の最大長は，新しく併合された単語の最大長と，INPUT命令で指定した仮想最大長の大きいほうとなる。

d　本命令実行後には，テクスト中の単語総数は変更される。

使 用 例

同一の第2ユニット中の‘DISARMAMENT’で始まる複合語を一語として扱うために併合する。連結はハイフンではなく空白とする。

```
例 5. 5 〉
MERGE△WORD
BOUNDARY=UN2
DISARMAMENT
MEASURE
DISARMAMENT
NEGOTIATION
DISARMAMENT
DECADE
%END
```

Ⅵ　テクストの印刷

どの計算機システムにもデータ（データセット）やファイルを印刷するためのプログラムが備わっているが，LEX を利用して処理するテクストに関しては，本章で述べる実行命令によるほうが，点検，保存に便利である。

Ⅵ．1　入力テクストの印刷 —— PRINT RAW DATA命令

書　式

```
  PRINT△RAW△DATA
[FIRST=i ]  (1≦i)  〈i=1〉
[LAST=j ]   (i≦j≦999999)  〈j=999999〉
  READ[△{SYSIN, LXDATA}]
[入力テクスト]
```

機　能

入力テクストの全体もしくは一部を点検，保存のために印刷する。

FIRST明細指示とLAST明細指示

入力テクストの一部だけを印刷したときに指定する。必要に応じ一方だけを指定してもよい。入力テクスト全体を印刷するときには指定不要である。省略時も含め，

FIRST=i

LAST=j

のとき，入力テクストの先頭からi番目のレコードから，j番目のレコードまでを印刷する。与える数値は整数で，かつ書式に与えた制限値を越えてはならない。また$i \leqq j$でなくてはならない。また，ここで与える数値は，計算機システムや利用者が識別のた

めに付けた所謂レコード番号あるいは行番号とは無関係である。

READ命令

このREAD命令はINPUT命令のREAD命令と同じである。入力テクストがこのREAD命令の後にあれば（正確には，PRINT RAW DATA命令のLEX文と，入力テクストが同一データセット上にあれば），

```
READ
```

または，

```
READ△SYSIN
```

としておけばよい。逆に入力テクストがLEX文と別のデータセット上にあれば，

```
READ△LXDATA
```

とし，入力テクストの存在するファイルを，ファイル名称（dd名）LXDATAとして利用者が定義しなければならない。（詳しくはXV章を参照）

注意事項

a　入力テクストは固定長80バイトのレコードから成っていなくてはならない。入力テクスト中にLEX文があってもよいが，計算機システムのジョッブ制御文を含んでいてはならない。

b　この命令は，本来LEXに入力されるテクストの点検，修正，保存を目的とするものであるが，上記aの制限のもとでは，LEXに入力されるテクストに限らず，どのようなデータ，プログラムでも印刷できる。

使 用 例

点検のため入力テクストの10番目から，199 番目までのレコードを印刷する。

```
〈 例 6.1 〉
PRINT△RAW△DATA
FIRST＝10
LAST ＝199
READ
入力テクスト
```

Ⅵ．2　単語の印刷——PRINT　WORD命令

書　　式

```
  PRINT△WORD
［データ置換・選別命令］
［PRINT＝《WSQ△△LSQ△△UN1△△UN2△△UN3△△UN4》］
```

機　　能

テクスト・ファイルまたはサブテクスト・ファイルの内容を印刷する。データ選別命令がなければ，処理対象となるテクスト（テクストT_i）中のすべての単語が印刷される。

データ置換・選別命令

任意。

PRINT明細指示

単語に続けて識別値を印刷すべきテクスト構成要素を略称で指定する。指定は任意の順序で最大 6 個まで行うことができるが，ふたつ以上を指定するときには，間にひとつ以上の空白が必要である。但し，存在しないテクスト・ユニットを指定してはならない。テクスト構成要素の識別値は指定順に左から印刷される。

省略された場合には，左から単語番号，LEX 行番号，第 1 ユニットの識別値，・・・，の順に印刷される。

注意事項

INCLUDE 命令と併用すれば特定の単語（群）についての簡単な単語索引を作ることもできる。単語の数，種類が多いときには，後述のWORD INDEX命令によるほうが出力も見やすく，便利である。

使 用 例

a 参考資料として保存しておくためテクスト・ファイルのすべての単語を印刷する。

```
〈例 6. 2 〉
PRINT△WORD
```

b 'DISARM'という文字列を含む単語の単語番号を知りたいので，単語と単語番号だけを印刷する。

```
〈例 6. 3 〉
PRINT△WORD
INCLUDE
POSITION=C△△B△△M△△E
DISARM
%END
PRINT=WSQ
```

Ⅵ. 3 LEX行の印刷 —— PRINT LINE命令

書 式

PRINT△LINE
［データ置換・選別命令］

機　能

LEX 行を印刷する。データ選別命令を用いなければすべての LEX 行が印刷される。LEX 行とともに，LEX 行番号と各テクスト・ユニットの識別値も印刷される。

データ置換・選別命令

REPLACE, POSITION, SAMPLEの各命令は使えない。また，SELECT 命令，REJECT命令のKEY明細指示ではWSQは指定できない。

INCLUDE命令，EXCLUDE命令ではPOSITION明細指示は使えない。INCLUDEまたはEXCLUDEに続けて，文字列を最大 100 まで与えることができるが，文字列の長さは空白も含め最大60文字まで許される。文字列中にはセミコロンを除くどのような記号，数値，文字を含んでいてもよい。与えられた文字列を含む（完全一致も含む）LEX 行が処理の対象となる（INCLUDEの場合）か，処理から除外される。（EXCLUDEの場合）。

注意事項

この命令は LEX 行ファイルが作成されていなければ実行できない。また，印刷すべき LEX 行ファイルをLXLINEというファイル名称で定義しておかなければならない。（XV章）

この命令は，INCLUDE命令を用いれば，後述のCONCORDANCE命令，あるいはWORD SET CONCORDANCE命令の簡単な代用品として用いることができる。但し，この場合，この命令では，LEX 行を越える単語列は取扱えないし，特定のLEX 行の前後の LEX 行を出力することはできない。

使 用 例

a　点検または保存用にすべての LEX 行を印刷する。

```
〈例 6.4〉
PRINT△LINE
```

この出力とPRINT RAW DATA命令の出力の最も大きな相違は，この出力では，LEX行とともにテクスト構成要素の識別値が印刷されることである。

b 'DEVELOP'という文字列（単語である必要はない）を含むすべてのLEX行を印刷する。

```
〈例 6.5〉
PRINT△LINE
INCLUDE
DEVELŌP
%END
```

c 'DFVELOP'と'POLITIC'というふたつの文字列をともに含むLEX行を印刷する。

```
〈例 6.6〉
PRINT△LINE
INCLUDE
DEVELŌP
%END
INCLUDE
PŌLITIC
%END
```

この例では，ふたつのINCLUDE命令の順序は任意である。INCLUDE 命令をひとつだけ用いて，

```
INCLUDE
```

```
DEVELŌP
PŌLITIC
%END
```

とすると，'DEVELOP'，'POLITIC'のうち少なくともひとつ以上を含むLEX行が出力される。

Ⅶ　単語の頻度に関する出力

Ⅶ. 1　見出語の頻度順リストと度数分布表 ── FREQ命令

書　式

```
  FREQ
[WORK=i ]  (i≧10)  <i=100>
[READ△KEYWORD]
[WRITE△OUTPUT]
[データ置換・選別命令]
[TYPE={L, T}]
[ORDER={A, D}]
[HBOUND=j ]  (1≦j≦999999)  <j=999999>
[LBOUND=k ]  (1≦k≦j)  <k=1>
```

機　能

処理の対象となるテクスト（T_i）中のすべての見出語について，頻度順リストもしくは見出語頻度の度数分布表を作成する。

作業領域定義命令

CREATE命令における説明（Ⅴ.2）を参照。

READ KEYWORD命令

CREATE KEYWORD命令によって作成された見出語ファイルを使って処理を行うときに与える。

この命令を与えるときには，見出語ファイルが既に作成されていなければならないし，また，見出語ファイルを，ファイル名称LEXSUBとして，定義しておかなければならない。

WRITE OUTPUT命令

実行結果をラインプリンターやTSS端末以外の媒体，例えば磁気ディスク，に出力するときに指定する。省略時には実行結果はラインプリンターまたはTSS端末に出力される。但し，この命令の有無に拘らず，LEX文や他のメッセージはラインプリンターか端末に出力され，外部記憶媒体には出力されない。この命令を与えるときには，出力用ファイルを，ファイル名称LEXOUTとして定義しなければならない。（XV章参照）

データ置換・選別命令

任意。但し，READ KEYWORD命令を与えたときには，与えてはならない。データ選別命令を用いたときには，選別された見出語だけが処理の対象となる。

テクスト・ファイル中のすべての見出語ではなく，一部の見出語について処理を行いたいときには，FREQ命令のもとでデータ置換・選別命令を用いて処理する方法と，まずCREATE KEYWORD命令で，データの置換・選別をした見出語ファイルを作成しておき，次にFREQ命令を実行する方法のふたつがある。

TYPE明細指示

出力の種類を指定する。

L：頻度順リスト

T：度数分布表。正確には，見出語のテクスト中での出現度数を変数値とし，変数値をとる見出語数を度数とする度数分布表。

省略時にはLが仮定される。

ORDER明細指示

頻度順リストのとき，見出語の配列を頻度の上昇順にするか，下降順にするかを指定する。

A：頻度の上昇順

D：頻度の下降順

省略時にはAが仮定される。

この明細指示はTYPE＝Tを指定したときには指定してはならない。

HBOUND明細指示とLBOUND明細指示

処理対象となったすべての見出語ではなく，特定の頻度をもつ見出語のリストや度数分布表が必要なときに，一方もしくは両方を指定する。省略時も含め，

HBOUND＝j

LBOUND＝k

であれば，頻度j以下でかつk以上の見出語についてリストもしくは度数分布表が出力される。j，kの値域は書式の通りであるが，$j \geq k$でなくてはならぬことに注意する必要がある。また，$j = k$であれば頻度jの見出語についての出力となるが，この時にはFREQ命令よりも，次のALPHA命令のほうが便利である。

出力の詳細

a　リストや度数分布表の印刷もしくは外部媒体への出力に先立って次の数値が印刷される。外部媒体には出力されない。

テクスト・ファイル中の単語総数（T）

処理対象となった単語総数（S）

処理対象となった見出語数（Sに対応する見出語数）（K）

LBOUNDまたはHBOUND明細指示が与えられているときには，この3つの数値に続けて，

印刷された見出語数（P）

が印刷される。

b　頻度順リストでは，見出語はその頻度の上昇順または下降順に印刷あるいは出力される。また，同一頻度の見出語についてはアルファベット順に印刷あるいは出力され

る。各見出語につき，同一行の先頭（左）から順に次の値が出力される。（　）には標準出力で与えられる見出しを示す。また〈　〉には，WRITE OUTPUT でディスク等に出力したときの書式を示す。

○ 見出語（INDEX WORD）〈A(w*)〉

○ 見出語の頻度（FREQUENCY）

絶対頻度（ABS）〈F(6)〉**

処理対象となった単語総数（S）に対する百分比（REL(S)）〈F(8, 4)〉***

テクスト中の単語総数（T）に対する百分比（REL(T)）〈F(8, 4)〉

○ 当該見出語の頻度を含む頻度順上位または下位からの単語累積頻度（CUM WORD FREQ）

絶対累積頻度（ABS）〈F(6)〉

前掲Sに対する累積百分比（REL(S)）〈F(8, 4)〉

前掲Tに対する累積百分比（REL(T)）〈F(8, 4)〉

○ 見出語の累積度数（CUM INDEX WORD FREQ）

頻度順上位または下位からの印刷対象となった見出語の累積度数（ABS）〈F(6)〉

処理対象となった見出語総数（K）に対する累積百分比****（REL(S)）〈F(8, 4)〉

* 単語の最大長

** FORTRAN流の書式であれば，F6.0 または I6，F8.4 等となる。以下同様。

*** 外部媒体出力のときには，1レコードが120バイトであるから，レコードの終りに，(120−58−w)の空白ができる。単語の最大長が予めわかっていれば，出力ファイルLEXOUTのレコード長を(w+58)として，記憶域を節約することもできる。

**** 前掲のK＞Pであるとき，つまりHBOUND, LBOUND明細指示の効果があるときには，最後の見出語についても100％にはならない。

c　度数分布表（TYPE=T）の場合は見出語の度数ごとに次の値が与えられる。（ ）と〈 〉の意味は前項bの場合と同じである。

○ 見出語の頻度（FREQUENCY）〈F(6)〉

○ 当該頻度の見出語数（NUMBER OF INDEX WORDS）〈F(6)〉

○ 処理対象となった見出語総数（前掲a項のK）に対する百分比（PERCENT）〈F(8, 4)〉

外部媒体出力ではレコード長を20バイトとしてもよい。

注意事項

テクスト全体の単語を対象とするとき，テクストが大きければ，計算機システムの時間の制約に触れることがある。このような場合には，CREATE KEYWORD命令で予め見出語ファイルを作るなり，データ選別命令を使うなりして，幾つかのステップに分けて行う工夫が必要である。初心者向けではないが，最も確実な方法は，まず'A'で始まる単語について見出語ファイルを作り，次で'B'で始まる単語についての見出語ファイルを作るという形で，必要な見出語ファイルを作っておき，最後にこれらの見出語ファイルをひとつに併合するという方法である。但し，この時，各見出語ファイルの先頭の1レコードは，併合後に先頭に位置するものを除き，削除しなければならないし，併合後の見出語ファイルの先頭1レコードの見出語総数の項は，併合されたファイル中の見出語総数に修正しておかなければならない。（XIV.2参照）

使 用 例

テクスト・ファイルまたはサブテクスト・ファイル中のすべての単語を対象とするときの例は，I章2節の例1.1，例1.4，例1.5を参照。

Ⅶ. 2 アルファベット順（アイウエオ順）リスト——ALPHA命令

書 式

```
ALPHA
[WORK=i]  (i≧10)  〈i=100〉
```

［READ△KEYWORD］
［WRITE△OUTPUT］
［データ置換・選別命令］
［HBOUND=j］（$1 \leqq j \leqq 999999$）〈$j=999999$〉
［LBOUND=k］（$1 \leqq k \leqq j$）〈$k=1$〉

機　能

処理の対象となるテクスト（T_i）中のすべての見出語についてアルファベット順リストを作成する。但し，LEXでアルファベット順というのは正確には内部コードの上昇順であって，上昇順は一般に記号，カナ，アルファベット，数字の順になる。従って，カナのみからなるデータの場合はアイウエオ順リストとなる。

作業領域定義命令（WORK），READ KEYWORD命令，WRITE OUTPOT命令，**データ置換・選別**命令，HBOUND明細指示，LBOUND明細指示についてはすべて前節FREQ命令に準ずるのでこれを参照。

出力の詳細

a　リストの出力に先立って，テクスト・ファイル中の総単語数などがプリンターか端末に出力される。前節FREQ命令の出力の詳細a項参照。

b　見出語はアルファベット順に配列される。各見出語につき，次の値が一行（外部媒体出力では1レコード）に先頭から順に出力される。（　）内はプリンター印刷のときの見出し，〈　〉は外部媒体出力の書式である。この書式については，前節の出力の詳細の項を参照。

○見出語（INDEX WORD）〈A(w)〉（wは単語の最大長）
○見出語の頻度（FREQUENCY）
　絶対頻度（ABS）〈F(6)〉
　処理対象となった単語総数（S）に対する百分比（REL(S)）〈F(8, 4)〉
　テクスト中の単語総数（T）に対する百分比（REL(T)）〈F(8, 4)〉

使 用 例

Ⅰ章2節の例1．2を参照。

注意事項

テクストが大きいときは，まず'A'で始まる単語，といった形で順次処理すればよい。

Ⅶ．3　綴字逆順アルファベット順（アイウエオ順）リスト——BACK命令

書　式

```
  BACK
[WŌRK=i ]  (i≧10)  〈i=100〉
[READ△KEYWŌRD]
[WRITE△ŌUTPUT]
[データ置換・選別命令]
[HBŌUND=j ]  (1≦j≦999999)  〈j=999999〉
[LBŌUND=k ]  (1≦k≦j)  〈k=1〉
```

機　能

処理対象となったテクスト（T_i）中のすべての見出語の綴字逆順アルファベット順リストを出力する。ここで，綴字逆順アルファベット順とは，見出語の末尾の文字からのアルファベット順であり，見出語

ABCDEF

があるとき，そのミラー・イメージである

FEDCBA

によるアルファベット順である。

作業領域定義命令その他

BACK命令のLEX文，出力形式等は前節のALPHA命令とまったく同じである。ALPHA命令，また必要に応じ，FREQ命令（本章1節）についての記述を参照。

注意事項

ALPHA命令の場合と同じ。但し，分割処理をするときには，'A'で終る単語，等々という形で，語末の文字によって分けなければならない。

使 用 例

Ⅰ章2節の例1.3を参照。

Ⅶ．4　エントロピーの計算——ENTROPY命令

書　　式

```
  ENTRŌPY
[WŌRK=i]
[READ△KEYWŌRD]
[データ置換・選別命令]
[LŌG={E, 2, X}]
```

機　　能

テクストの語彙のサイズを計るひとつの尺度として，見出語の出現比率にもとづくエントロピーを計算する。エントロピーの計算法は次の通りである。見出語$_i$の出現度数をf，処理対象となった単語総数（本章1節のb）をtとすると，見出語$_i$の出現比率P_iは，

$P_i = f / t$

である。このとき，エントロピーHは，

$$H = -\Sigma_i \ (P_i \log P_i)$$

ただし，対数の底はLOG明細指示によって指定できる。

作業領域定義命令

READ KEYWORD命令を与えるときには与えても無意味である。実際に与える場合については，V.2のCREATE命令における説明を参照。

READ KEYWORD命令

見出語ファイルを使って処理を行うときに与える。一般的注意については，Ⅶ．1のFREQ命令における説明を参照。

データ置換・選別命令

READ KEYWORD命令を与えたときには使用できない。それ以外は任意である。

LOG明細指示

エントロピーの計算に用いる対数の底を指定する。省略時にはEが仮定される。記号の意味は次の通りである。

E：自然対数，底はe.

2：底は2

X：常用対数，底は10.

使 用 例

テクスト全体を対象としてエントロピーを底を2として計算する。見出語ファイルは

既に作成されているので，これを使うものとする。

〈例 7.1〉

```
ENTRŌPY
READ△KEYWŌRD
LŌG=2
```

Ⅷ　単語列の頻度に関する出力

ＬＥＸでは，テクスト中で連続したふたつ以上の単語から成る列（string）を**単語列**と称する。より具体的には，テクスト・ファイル（T_i）*中で，連続した単語番号（ＷＳＱ）をもつふたつ以上の単語の連鎖が単語列である。

ＬＥＸには，単語列の頻度順，アルファベット順リストを出力するふたつの実行命令がある。**WORD STRING1** と，**WORD STRING2** のふたつの命令である。前者は，特定の単語(群)を含む単語列を対象とするものであり，後者はテクスト中のすべての可能な単語列を対象とするものである。

＊**MERGE WORD** 命令実行後のテクスト・ファイルも含む。

Ⅷ. 1.　特定の単語(群)を含む単語列のリスト――WORD STRING 1 命令

書　式

WORD△STRING 1
[**WORK** $= i$]　　($i \geqq 10$) < $i = 100$ >
[**WRITE△OUTPUT**]
[データ置換・選別命令]
[**LENGTH** $= j$]　($2 \leqq j \leqq 8$) < $j = 2$ >
[**MODE** $=\{$ **E**, **I** $\}$]
[**BOUNDARY** $=\{$ **LSQ**, **UN1**, **UN2**, **UN3**, **UN4** $\}$]
[**POSITION** $=$ ((**B△△M△△E**))]
[**TYPE** $=\{$ **A**, **F** $\}$]
[**HBOUND** $= k$]　($1 \leqq k \leqq 999999$) < $k = 999999$ >
[**LBOUND** $= l$]　($1 \leqq l \leqq k$) < $l = 1$ >
単語$_1$
[単語$_2$ ……　……　単語$_m$]　　($1 \leqq m \leqq 100$)

```
%END
[%STOPWORD
 ストップワード1
[ストップワード2]
 .....
 .....
[ストップワード n]        ( 1 ≦ n ≦ 100 )
 %END                                    ]
```

機　　能

利用者の指定した単語（群）を含む単語列の頻度上昇順リスト，あるいはアルファベット順リストを作成する。単語列の長さ，単語列中の指定された単語（群）の位置等について，利用者の選択が可能である。

作業領域定義命令

CREATE命令における説明（V. 2）を参照。

WRITE OUTPUT命令

処理結果をラインプリンターまたは端末以外の外部記憶媒体に出力するときに指定する。この命令を与えたときには，出力用ファイルをLEXOUTという名称で定義しておく必要がある。（XV章参照）

データ置換・選別命令

任意である。しかし，何らかの形で単語単位のデータの選別を行うときには次の注意が必要である。

a　POSITION命令，KEY＝WSQとしたSAMPLE命令もしくはSELECT命令，REJECT命令，またINCLUDE命令，EXCLUDE命令の少なくともひとつを実行したとき，当該の命令の実行によって除去された単語を含む単語列は一際処理の対象とならない。逆に言えば，処理（出力）の対象となるのは，このような命令の実行に

よって選ばれた（除去されなかった）単語から成る単語列のうち，利用者が指定した単語を含む単語列だけである。

b　このことから，INCLUDE命令の単語もしくは文字列リストは，このWORD STRING1命令の単語リストを含む（あるいは部分集合とする）ものでなくてはならない*。

c　EXCLUDE命令のリストに与えられた単語を含む単語列は一際処理の対象とならない。このことを逆用した例については，本章2節の注意事項を参照。また%STOP WORDの効果との相違については後述する。

*厳密に言えば，bの2種のリストによって定義される2種の単語の集合の論理積が空でなければよい。

LENGTH明細指示

単語列の長さを単語数で指定する。指定は2から8までの正整数で行う。省略時には2が仮定される。従って，最大8語までの単語列を取扱えるが，単語数に関係なく単語列の長さが101文字以上となったときには*，先頭から数えて101文字以降は削除され，100文字からなる文字列として処理される。

*単語列を構成する単語$_i$の長さをiとすると，単語列においては，単語と単語の間にひとつの空白が置かれるから，単語列の長さは，

$$\sum(i+1)-1$$

となる。

MODE明細指示

単語列の長さが3語以上のとき，それより短かい単語列も処理するか否かを指定する。従って，この明細指示は，省略時も含め，LENGTH＝2のとき与えても無意味である。今，

LENGTH＝j

とするならば，キーワードの意味は次のようになる。

E：長さjの単語列のみを処理する。

I ：長さ2以上，j以下の単語列をすべて処理する。

省略時の標準値はEである。

BOUNDARY明細指示

テクスト構成要素の境界をまたがる単語列，例えばふたつの文にまたがる単語列を処理から除くときに指定する。指定されたテクスト構成要素の境界をまたがる単語列は処理対象とならない。LSQ等のキーワードの意味については，SAMPLE命令のKEY明細指示の項（Ⅳ. 3）を参照。

この明細指示が省略されたときには，テクスト構成要素の境界をまたがる単語列も処理の対象となる。

POSITION明細指示

利用者の指定した単語(群)が単語列中の特定の位置にあるケースだけを処理対象としたいときに指定する。POSITION＝に続けて，位置を示すキーワードB，M，Eの，少なくともひとつ，最大3つを任意の順序で指定できる。ふたつ以上指定するときには，間に少なくともひとつの空白が必要である。

B ：指定された単語（群のひとつ）が単語列の先頭にある場合を処理対象とする。

M ：指定された単語（群のひとつ）が単語の列の先頭でも末尾でもない位置にある場合を処理対象とする。

E ：指定された単語（群のひとつ）が単語列の末尾にある場合を処理対象とする。

この明細指示が省略されたときには，

POSITION＝B△△M△△E

が仮定される。即ち，指定された単語が単語列中のどこにあっても処理対象となる。

省略時も含め，LENGTH＝2のときには，Mを指定してはならない。

TYPE 明細指示

頻度上昇順リストか，アルファベット順リストかを指定する。省略時解釈は A である。

A：アルファベット順リスト

F：頻度上昇順リスト

HBOUND 明細指示，LBOUND 明細指示

単語列の出現度数によって処理する単語列を選別するときに，一方または両方を指定できる。今，省略時も含め，

HBOUND $= k$

LBOUND $= l$

とすれば，頻度 k 以下，l 以上の単語列だけが処理の対象となる。$k = l$ であれば，頻度 k の単語列だけが処理の対象となる。

省略時の標準値については書式を参照。

単語リスト

リストには少なくともひとつ，最大 100 個までの単語を与えることができる。リストの終りには％END の明細指示が必要である。ここに与えられた単語のうちの少なくともひとつを含む単語列が処理の対象となる。BOUNDARY 明細指示，POSITION 明細指示，HBOUND 明細指示，LBOUND 明細指示のいずれかひとつが与えられている場合には，このような単語列のうち，条件に合うものだけが処理の対象となる。

ストップワードリスト

％STOPWORD ステートメントで始まり，％END ステートメントで終る単語のリストがストップワードリストである。このリストにはひとつ以上，最大 100 個までの単語を指定できる。このリストに与えられた単語のうちのひとつが，単語列の先頭もしくは末尾にあるとき，この単語列は処理から除外される。例えば，リスト中に単語 'AND'

が与えられていれば，‘AND’で始まるか，‘AND’で終わるすべての文字列が処理の対象から除外される。これに対して，**EXCLUDE**命令中に単語‘AND’を指定したときには，ANDを含むすべての単語列が処理から除外されることになる。

ストップワードリスト中に，単語リストで与えられた単語を与えることは，上述の定義からして，エラーにこそならないが一般には無意味である。

出力の詳細

a　出力される単語列は，利用者が単語リストで指定した単語と，データ置換・選別命令，**LENGTH**，**MODE**，**BOUNDARY**，**POSITION**，**HBOUND**，**LBOUND**の各明細指示，それにストップワードリストによって決まる。

b　リストの出力に先立ち，次の4種の数値が印刷される。外部媒体出力の場合も，ライン・プリンターか端末に出力される。

抽出された単語列（トークン）の総数（T）
実際に処理の対象となった単語列（トークン）の総数（S）
Sに含まれる単語列の種類（タイプ）数（K）
実際に出力される単語列の種類（タイプ）数（P）

ここで，T，S，K，Pは実際には次のように求められる。

T：（データ選別命令が実行されている場合も含め）テクストT_i中で，単語番号が連続し，かつ，**BOUNDARY**明細指示と，**LENGTH**明細指示，**MODE**明細指示の条件をすべて満たす単語の列の総数

S：上述T個の単語列のうち，単語リストに与えられた単語を少なくともひとつ含み，かつ省略時も含め，**POSITION**明細指示とストップワードリストで与えられる条件を満たす単語列の総数

K：上述S個の単語列に含まれる単語列の種類数（タイプ数）

P：K種類の単語列のうち，**HBOUND**，**LBOUND**明細指示の規定する条件を満たす単語列の種類数。このふたつの明細指示が省略されるか，標準値が指定さ

れていれば，P＝Kとなる。

c　リストでは標準出力，外部媒体出力とも単語列（タイプ）とその出現度数が出力される。外部媒体出力のフォーマットは各レコードの先頭から

単語列（タイプ）	A（M）
出現度数	F（6）

となる。但し，Mは単語列の最大長で，リストの出力に先立ち，そのバイト数が印刷される。

注意事項

a　この命令は特定の単語(群)を含む単語列を処理するためのものであり，テクスト中のほとんどすべての単語列を対象にするときには次節のWORD STRING2命令のほうが便利である。なお，これについては，次節注意事項も参照。

b　出力される単語列は機械的に抽出されるため，相当数の無意味な単語列（所謂ノイズ）の混入は避けられない。また場合によっては，ノイズか否かの判定が微妙なケースもある。この点はWORD SET CONCORDANCE命令など用例索引系の実行命令を用いて確認することができる。

使 用 例

第1ユニットの境界を越えない範囲で，'DEVELOPMENTを'含む，2語から成る，頻度8以上の単語列の頻度順リストを出力する。但し，'AND'，'THAT'，'ON'，'IN'，'TO'，'OF'，'FOR'を含むものは除外する。

```
＜例 8. 1 ＞
WŌRD△STRING1
BŌUNDARY＝UN1
TYPE＝F
LBŌUND＝8
DEVELŌPMENT
```

```
%END
%STŌPWŌRD
AND
THAT
ŌN
IN
TŌ
ŌF
FŌR
%END
```

この例では，単語列の長さが2であるからストップワードリストの代りに，**EXCLUDE**命令を用いてもよい。一般に，長さ2語の単語列だけを対象とするのであれば，ストップワードリストと**EXCLUDE**命令は等価である。

Ⅷ．2　テクスト中のすべての単語列のリスト——WORD　STRING2命令

書　式

```
WŌRD△STRING2
[ WŌRK＝i ]      ( i≧10 ) < i＝100 >
[ WRITE△ŌUTPUT ]
[ データ置換・選別命令 ]
[ LENGTH＝j ]      ( 2≦j≦5 ) < j＝2 >
[ MŌDE＝{ E, I } ]
[ BŌUNDARY＝{ LSQ, UN1, UN2, UN3, UN4 } ]
[ TYPE＝{ A, F } ]
[ HBŌUND＝k ]      ( 1≦k≦999999 ) < k＝999999 >
[ LBŌUND＝l ]      ( 1≦l≦k ) < l＝1 >
```

機　　能

テクスト（T_i）中の指定された長さのすべての単語列について，頻度上昇順リストもしくはアルファベット順リストを作成する。

作業領域定義命令

CREATE命令における説明（V.2）を参照。

WRITE OUTPUT命令，データ置換・選別命令，LENGTH明細指示，MODE明細指示，BOUNDARY明細指示，TYPE明細指示，HBOUND明細指示，LBOUND明細指示

前節WORD STRING1命令の説明を参照。但し，LENGTH明細指示では，最大値が5であることに注意。また，この命令では，POSITION明細指示，単語リスト，ストップワードリストは与えることができないことに注意。

また，データ選別命令が与えられ（実行され）たときには，データ選別命令によって選択された（削除されなかった）単語のみから成る単語列が処理対象となる。

出力の詳細

a　出力される単語列は，データ置換・選別命令，LENGTH，MODE，BOUNDARY，HBOUND，LBOUNDの各明細指示によって決まる。

b　出力の形式は前節WORD STRING1命令のそれに準ずる。但し，前節出力の詳細bのふたつの数値TとSに関しては，この命令では常にT＝Sとなる。

注意事項

テクストが大きい場合，この命令の実行には，かなりの処理時間と大きな作業用一時ファイルを必要とする。このような場合，テクスト中のすべての単語列をリストする必要があれば，次のような工夫が必要である。

ア．頻度の多い単語を含む単語列はまず前節のWORD STRING1命令で必要に応じ何回かに分けて処理し，次にその他の単語を含む単語列をこの命令で処理する。この命令で処理するときには，既にWORD STRING1命令で処理した単語はEX

CLUDE命令で削除する。

イ．MODE＝Eとして，2語の場合，3語の場合等に分けて処理する。

ウ．可能ならば，LBOUND明細指示を用いる。但し，この方法は処理時間，作業用一時ファイルの容量の節約にはあまり寄与しない。

エ．必要に応じ，ア，イ，ウを組合せる。参考までに処理される単語列の総数の計算法を述べておく。長さk語の単語列の総数T_kは，テクスト（T_i）中の単語総数をtとし，このテクスト中では，すべての単語の単語番号が連続しているものとすると，

i） BOUNDARY明細指示の省略時には，

$$T_k = t - k + 1$$

ii） BOUNDARY明細指示が与えられた時には，

$$T_k = t - n(k - 1)$$

（但し，nは，BOUNDARY明細指示で指定されたテクスト構成要素のテクスト（T_i）中での総数*）

また特定の単語を含む単語列の数Wは，BOUNDARY明細指示がないものとすれば，ごく大ざっぱに，

$$W = f k$$

（但し，fはT_i中でのこの単語の出現度数）

によって近似できる。

*nは，UNIT LENGTH命令，LINE LENGTH命令により知ることができる。

使用例

テクスト中の第1ユニットを越えない範囲の長さ2語のすべての単語列のアルファベット順リストを出力する。但し，“AND”，“A”，“THE”およびbe動詞を含む単語列と，頻度3以下の単語列は除外する。

```
<例 8. 2>
WORD△STRING2
EXCLUDE
AND
A
THE
BE
IS
AN
ARE
WAS
WERE
BEEN
%END
BOUNDARY=UN1
LBOUND=4
```

Ⅸ　LEX行のリスト——LINE LIST命令

書　式

```
LINE△LIST
[ WORK=i ]    ( i≧10 ) < i = 100 >
[ WRITE△OUTPUT ]
[ データ置換・選別命令 ]
[ JUSTIF={Y, N} ]
[ TYPE={A, F} ]
[ HBOUND=j ]    ( 1≦j≦999999 ) < j = 999999 >
[ LBOUND=k ]    ( 1≦k≦j ) < k = 1 >
```

機　能

LEX行の頻度上昇順リスト，またはアルファベット順リストを作成する。

作業領域定義命令

CREATE命令における説明（Ⅴ. 2）参照。

WRITE OUTPUT命令

EREQ命令における説明（Ⅶ. 1）参照。

データ置換・選別命令

PRINT LINE命令における説明（Ⅵ. 3）と，INCLUDE命令，EXCLUDE命令の注意事項（Ⅳ. 5）参照。

JUSTIF明細指示

LEX行の先頭（左端）に，ひとつ以上の空白（所謂インデンテーション）があるとき，この空白をつめて，左づめ（left−justify）するか否かを指定する。省略時はYが仮定される。

Y ：空白をつめて左づめにする

N ：空白をつめない

TYPE明細指示

頻度順リストかアルファベット順リストかを指定する。省略時解釈はAである。

A ：アルファベット順リスト

F ：頻度上昇順リスト

HBOUND明細指示，LBOUD明細指示

FREQ命令における説明（Ⅶ. 1）参照。

出力の詳細

a　リストの出力に先立ち，次の情報がラインプリンターまたは端末に出力される。（外部記憶媒体には出力されない。）

処理対象となったLEX行（トークン）の総数
（データ置換・選別命令が与えられていれば，その実行後のLEX行総数）
処理対象となったLEX行の種類（タイプ）数
出力されるLEX行の種類（タイプ）数

b　リストでは各行（各レコード）の先頭から次の内容が出力される。＜　＞は外部媒体出力の書式を示す。

LEX行　＜A（l）*＞
当該LEX行（タイプ）の頻度　＜F（6）＞
＊LEX行の最大長。INFO命令によって知ることができる。

注意事項

a　この命令はLEX行ファイルが作成されていなければ実行できない。また，LEX行ファイルをLXLINEという名称で定義しておく必要がある。（XV章参照）。

b　引用符等の記号は左づめできない。必要があれば，計算機システムのユーティリティを用いてLEX行ファイルを編集し，当該の記号を置換しておく。

c　LEX行の総数はINPUT命令の出力，INFO命令の出力によって知ることができる。

使用例

すべてのLEX行を対象として，頻度順リストを作成する。LEX行の頭の空白はつめ，しかも頻度2以下のものは除外する。

```
＜例 10.1 ＞
LINE△LIST
TYPE=F
LBOUND=3
```

X　単語と単語列の索引

索引には，単語，句など用語の出現個所を示す用語索引と，出現個所だけでなく前後の文脈も示す用例索引とがある。本章ではLEXの用語索引について述べ，次章で用例索引について述べる。LEXの用語索引には単語索引と，単語列索引の2種類がある。

X. 1　単語索引—— WORD INDEX命令

書　式

```
WORD△INDEX
[WORK=i]
[READ△ALPHA]
[WRITE△OUTPUT]
[データ置換・選別命令]
[PRINT=((WSQ△△LSQ△△UN1△△UN2△△UN3△△UN4))]
```

機　能

処理の対象となったすべての見出語について索引を作成する。

作業領域定義命令

CREATE命令における説明（V. 2）参照。但し，この命令では，READ ALPHA命令を与えるときに作業領域定義命令を与えても無意味である。

READ ALPHA命令

CREATE ALPHA命令によって単語アルファベット順ファイル（V. 1，V. 2）が作成されており，このファイルからデータを読み込むとき与える。このときには，単語アルファベット順ファイルをLEXSUBという名称で定義しておく必要がある。（XV章参照）

WRITE OUTPUT 命令

FREQ命令における説明（Ⅶ. 1）を参照。

データ置換・選別命令

自由に任意的に用いてよいが，READ ALPHA命令を用いたときには用いてはならない。

PRINT明細指示

一般的注意に関しては，PRINT WORD命令における説明（Ⅵ. 2）参照。但し，WORD INDEX命令では，テクスト構成要素の略称，WSQ等，は最大4個までしか指定できない。また，省略時には，

PRINT＝UN 1△△UN 2 …… UN_n

と仮定される。即ち，第1ユニットの識別値，第2ユニットの識別値，……の順で印刷される。

出力の詳細

外部媒体出力で各レコードの項目と書式は次のようになる。まず，見出語が直前のレコードと異なる場合は，

空白	4バイト
見出語	wバイト*
ダッシュ	4バイト

識別値（PRINT明細指示で指定されたテクスト構成要素の識別値）

(m)(X(2), (n)(X(1), A(6)))バイト

（但し，nはPRINT明細指示で指定されたテクスト構成要素の数，mは，

$w + 10 + m(2 + 7n) \leqq 120$

を満たす最大の整数）

また，見出語が直前のレコードのそれと同じ場合には，

空白　　（w＋8）バイト

識別値　　(m)(X(2)，(n)(X(1)，A(6)))バイト

＊単語の最大長。INFO命令で知ることができる。

注意事項

テクストが長い場合には，必要に応じ，Aで始まる単語，等々の形で分割処理をする。CONCORDANCE命令やKWIC命令をも用いる可能性があれば，CREATE ALPHA命令で，単語アルファベット順ファイルを作っておいたほうが効率がよい。

使用例

a　テクスト中のすべての見出語についての索引を作成する。単語アルファベット順ファイルを作成してあるので，これを用いる。

```
＜例 10. 1＞
WORD△INDEX
READ△ALPHA
PRINT=UN2△△UN1
```

b　テクストを分割処理する。まず(A)で始まる見出語についての索引を外部媒体上に出力する。

```
＜例 10. 2＞
WORD△INDEX
WRITE△OUTPUT
INCLUDE
POSITION=B△△C
A
%END
PRINT=UN3△△UN1△△UN2
```

このように分割処理して，すべての見出語についての索引を作成できたら，別個に出した索引を計算機システムのユーティリティを用いて，併合（マージ）すれば，テクスト中のすべての見出語に関しての索引が作成できる。

X. 2　単語列索引── WORD SET INDEX命令

書　式

```
WORD△SET△INDEX
[ WRITE△OUTPUT ]
[ データ置換・選別命令 ]
[ BOUNDARY={ LSQ, UN1, UN2, UN3, UN4 } ]
[ ORDER={ Y, N } ]
[ CONTIG={ Y, N } ]
[ RANGE=i ]   ( i≦500 ) < i = 20 >
[ PRINT=( ( LSQ△△UN1△△UN2△△UN3△△UN4 ) ) ]
  単語1
  単語2
 [単語3
  .....
  単語j ]          ( 2≦j≦5 , j<i )
%END
```

機　能

利用者の指定した条件を満たす単語列もしくは単語の集合について索引を作成する。

WRITE OUTPUT命令

FREQ命令における説明（Ⅶ. 1）参照。

データ置換・選別命令

自由に用いてよい。しかし，データ選別命令により削除された単語を後述の単語リスト中に指定しても無意味である。単語間の距離はすべて単語番号によって測られるので，これ以外の場合はデータ選別処理による影響は特に考える必要はない。

BOUNDARY 明細指示

任意のテクスト構成要素の境界にまたがる単語列もしくは単語の集合を処理から除くとき，そのテクスト構成要素をLSQ等の略称で指定する。略称の意味については，SAMPLE命令における説明（Ⅳ. 3）を参照。この明細指示が省略されたときには，テクスト構成要素の境界に関係なく処理が行われる。

ORDER明細指示とCONTIG明細指示

後述の単語リストに与えられた単語相互の関係を規定する。テクスト中の単語に関しては，その出現順序，隣接性にもとづき，次の4つの区分が可能である。

順序あり，隣接（ORDERED, CONTIGUOUS）
順序あり，非隣接（ORDERED, NONCONTIGUOUS）
順序なし，隣接（UNORDERED, CONTIGUOUS）
順序なし，非隣接（UNORDERED, NONCONTIGUOUS）

単語が隣接している場合がLEXでいう単語列であり，非隣接の場合はむしろ単語集合というほうが正確な言い方である。また後者の場合順序を無視すれば，ある一定の範囲で共出現する単語群ということもできる。

ORDER明細指示は，順序の有無を指定する。省略時は，Yが仮定される。

Y：順序あり
N：順序なし

CONTIG明細指示は隣接性の如何を指定する。省略時解釈はYである。ただし，こ

こで言う隣接性はテクスト・ファイルにおける隣接性，即ち単語番号の隣接性である。

Y：隣接
N：隣接を条件としない

今テクストファイル中に単語W_1とW_2のふたつの単語があるとき，このふたつの単語の順序と隣接の関係には次の4通りが考えられる。

$W_1 - W_2$　　（A）
$W_1 - X - W_2$　（B）
$W_2 - W_1$　　（C）
$W_2 - X - W_1$　（D）
（但し，Xは空でなく，かつ，W_1，W_2のいずれをも含まぬ単語の列）

単語リストにW_1とW_2がこの順で与えられているとき，ORDER明細指示とCONTIG明細指示の条件に合うのは，

ORDER=Y，CONTIG=Yであれば，A
ORDER=Y，CONTIG=Nであれば，A，B
ORDER=N，CONTIG=Yであれば，A，C
ORDER=N，CONTIG=Nであれば，A，B，C，D

となる。単語リストに3語以上与えられている場合も同様に考えればよい。

RANGE明細指示

CONTIG=Nのときにのみ指定可能。単語列もしくは単語集合を構成する最初の単語と最後の単語が最大何語離れていてもよいかを単語数で指定する。指定は正整数で，最初の単語から最後の単語までの単語数を与える。値の範囲は書式の通りであるが，単語リストに与える単語の数より大きくなくてはならない。

PRINT明細指示

一般的説明，注意事項については，**PRINT WORD**命令における説明(VI. 2)を参照。但し，本命令では，テクスト構成要素の略称**LSQ**等は最大4個までしか指定できない。また省略時解釈は，前節**WORD INDEX**命令における説明を参照。

単語リスト

単語列もしくは単語集合を構成する単語をそれぞれひとつのステートメントとして，2個以上，5個まで与える。**ORDER=Y**のときには，与える順序が意味をもつ。**CONTIG=N**のときには，ここで与える単語数は，**RANGE**明細指示によって決まる単語数よりも小さくなければならない。

出力の詳細

a 出力の対象となる，即ち出現個所を示す識別値が出力される単語列もしくは単語集合は，**BOUNDARY**以下の各明細指示によって決まる条件のもとで，単語リストに与えられたすべての単語を含む単語列もしくは単語集合である。出力に先立ち，出力媒体の如何に拘らず，このような単語列ないしは単語集合の総数が，ラインプリンターもしくは端末に出力される。

b 識別値は，このような単語列ないしは単語集合が一回出現するごとに‘……’で結んで，ふたつ与えられる。これは，単語列ないしは単語集合の最初と最後の単語の識別値である。外部媒体出力での書式は各レコードの先頭から，

空白	3バイト
最初の単語の識別値	(n)(X(1), A(6))バイト
ダッシュ	5バイト
最後の単語の識別値	(n)(X(1), A(6))バイト

（但し，nは**PRINT**明細指示で指定されたテクスト構成要素数）

注意事項

a 単語リストに同一の単語を与えても構わない。このときには，同一の単語の繰返し

個所の索引となる。

b　具体的文脈や用例が必要なときには，この命令ではなくWORD SET CONCORDANCE命令を用いればよい。

使 用 例

a　‘A …… OR TWO’という句の出現個所を捜す。‘A’と‘OR’の間には任意の一語が来るものとし，第2ユニットの境界はまたがらないものとする。

```
＜例 10. 3＞
WORD△SET△INDEX
BOUNDARY=UN2
CONTIG=N
RANGE=4
PRINT=UN1△△UN2
A
OR
TWO
%END
```

この例では，テクスト中にもし，‘A OR TWO’，‘A OR… TWO’といった単語の列が存在すれば出力されることになる。ORDER=NもしくはCONTIG=Nのときには，このようなノイズが混入している可能性があるので注意を要する。

b　‘DISARM’という文字列を含む単語と‘DEVELOP’という文字列を含む単語が同一の第1ユニット（例えば文）内で共出現するケースについて索引を作る。第1ユニットの長さは，UNIT LENGTH命令によって既に最大35語であることがわかっているものとする。

```
＜例 10. 4＞
WORD△SET△INDEX
REPLACE
```

```
TYPE=G
PŌSITIŌN=C△△B△△M△△E
DISARM
%BY
X
%END
REPLACE
TYPE=G
PŌSITIŌN=C△△B△△M△△E
DEVELŌP
%BY
Y
%END
BŌUNDARY=UN1
ŌRDER=N
CŌNTIG=N
RANGE=35
PRINT=UN1△△UN2
X
Y
%END
```

最初ふたつのREPLACE命令は，‘DISARM’‘DEVELOP’を含む単語をそれぞれ，‘X’，‘Y’という単語に変えるものである。混同の余地があれば，他の形に変えればよいし，混同の余地がなければどのような形の単語でもよい。

結果として得られるのは，単語Xと単語Yの共出現索引ということになる。

c　WORD SET CONCORDANCE命令の使用例（XI. 3）も参照。

XI 用例索引

LEXの用例索引は見出しの種類，出力の形式によって次のように分けられる。但しここで見出しというのは，用例検索の手掛り（キー）というほどの意味である。

単語KWOC索引（コンコーダンス）

見出し：単語

出　力：LEX行1行もしくは複数行

形　式：KWOC

単語KWIC索引

見出し：単語

出　力：LEX行1行もしくは最大3行（但し，印刷では1行に収める）

形　式：KWIC

単語列用例索引

見出し：単語列または単語集合

出　力：LEX行1行または複数行

形　式：KWOC

文字列用例索引

見出し：文字列（1文字以上60文字以内）

出　力：LEX行1行

形　式：KWIC

LEX行索引

見出し：LEX行

出　力：LEX行1行

形　式：KWOC

本章で述べる用例索引はすべてLEX行（入力レコード）を単位として出力される。（LEX行については，Ⅲ章を参照。）LEX行に関しては既に述べたように，長さがカード1枚分以下という制約があるため，LEXでは，必要に応じ，用例としてLEX行を複数行出力できる命令を用意している。また，LEX行単位で処理が行われるので，本章の各命令の実行にはLEX行ファイルが必要である。このためには，**INPUT**命令において**LINE FILE=Y**という指定をしておき，また各命令の実行時にLEX行ファイルをLXLINEというファイル名称で定義しておく必要がある。

本章で述べる実行命令のほか，**PRINT LINE**命令が簡単な用例索引作成の機能をもつことは既に述べた通りである。（Ⅵ. 3）

Ⅺ. 1　単語のKWOC索引（コンコーダンス）――CONCORDANCE命令

書　式

```
CONCOR DANCE
[ WORK=i ]      ( i≧10 ) < i = 100 >
[ READ△ALPHA ]
[ WRITE△OUTPUT ]
[ データ置換・選別命令 ]
UNIT={ LSQ, UN1, UN2, UN3, UN4 }
[ LINES=j ]     ( 1≦j≦5 ) < j = 1 >
[ PRINT=( ( LSQ△△UN1△△UN2△△UN3△△UN4 ) ) ]
```

機　能

処理対象となったすべての見出語についてコンコーダンス（KWOC索引）を作成する。

作業領域定義命令

CREATE命令における説明（Ⅴ. 2）参照。

READ ALPHA命令

WORD INDEX命令における説明（X. 1）参照。

WRITE OUTPUT命令

FREQ命令における説明（Ⅶ. 1）参照。

データ置換・選別命令

テクスト中の特定の単語(群)，テクスト中の特定の部分についてのコンコーダンスを作成するとき任意に用いてよい。但し，READ ALPHA命令を用いたときには，用いてはならない。

UNIT明細指示

出力するLEX行をどのテクスト構成要素の範囲にとどめるかを指定する。この明細指示は省略できない。

LSQ ：見出語のトークンが出現するLEX行1行だけを出力する。

UN_i ：見出語のトークンが出現するLEX行と，その前後のLEX行を，第iユニットの境界を越えず，しかも次項LINES明細指示で指定された行数を越えない範囲で出力する。（この場合，一般にひとつの文，段落等が出力されることになる。）

LINES明細指示

UNIT明細指示で，LSQ以外を指定したときに，見出語トークンの出現するLEX行の前後に最大何行出力するかを指定する。UNIT=LSQのときは指定できない。UNIT=LSQ以外のときには，ここで指定された行数の範囲で，しかも指定されたテクスト・ユニットの境界を越えないLEX行がひとつの用例として出力される。

PRINT明細指示

一般的注意に関しては，PRINT WORD命令における説明（Ⅵ. 2）参照。但し，こ

の命令では，テクスト構成要素は最大4つまでしか指定できない。省略時については，**WORD INDEX**命令における説明（X. 1）参照。

出力の詳細

a　見出語はすべてアルファベット順に出力される。

b　外部記憶媒体の出力は次の書式で行われる。見出語のひとつのトークンにつき，複数のレコードが使用されるが，最初のレコードの先頭から，

#	1バイト
空白	2バイト
見出語	wバイト*

2番目以降のレコードでは，**UNIT**と**LINES**によって決まるLEX行が1行ないし複数行出力される。各レコードにLEX行が1行分記録されるが，その書式は，先頭から，

空白	6バイト
LEX行	lバイト**

また，見出語のトークンが出現するLEX行そのものには，88カラム以降に，

（n）（X（1），A（6））
（但し，nはPRINT明細指示で指定されたテクスト構成要素数）

の書式で，当該LEX行の識別値が出力される。

なお，この命令の外部媒体出力では，同一の見出語は，**WORD INDEX**命令のように省略されることはない。

*最大単語長。**INFO**命令で知ることができる。
最大LEX行長。INFO**命令で知ることができる。

c　ラインプリンターまたは端末出力で複数行が出力される場合，識別値が与えられる

のは当該見出語の出現するLEX行についてだけである。

注 意 事 項

a　この命令は，LEX行ファイルが作成されていなければ実行できない。また，利用者は行ファイルをLXLINEというファイル名称で定義しておく必要がある。(XV章参照。)

b　テクストが長いときは適宜分割処理をする。**WORD INDEX**命令の注意事項と使用例bを参照。(X. 1)

c　**INCLUDE**命令を用いて特定の見出語に関してのみ，**UNIT=LSQ**の場合のコンコーダンスを出力するのであれば，同様に**LINE INDEX**命令，**PRINT LINE**命令を用いてもよい。但し，**LINE INDEX**命令では，LEX行はアルファベット順に出力され，**PRINT LINE**命令では，この命令と同様テクストでの出現順に出力されるが，印刷する識別値の種類を自由に選べない。

使 用 例

a　テクスト中のすべての見出語についてのコンコーダンスを作成する。出力があまり大きくならないように，LEX行1行だけを出力する。

```
<例 11. 1>
CŌNCŌR
UNIT=LSQ
PRINT=UN1
```

b　テクスト中の“PEACE”を含むすべての第2ユニットを印刷する。第2ユニットの長さが不揃いなので，最大限の文脈が与えられるようにする。

```
<例 11. 2>
CŌNCŌRDANCE
INCLUDE
PEACE
```

```
%END
UNIT=UN2
LINES=5
```

XI. 2　単語のKWIC索引——KWIC　INDEX命令

書　式

```
KWIC△INDEX
[ WORK=i ]　　( i≧10 )<i=100>
[ READ△ALPHA ]
[ WRITE△OUTPUT ]
[ データ置換・選別命令 ]
UNIT={ LSQ, UN1, UN2, UN3, UN4 }
[ MARKER={ LSQ, UN1, UN2, UN3, UN4 } ]
[ PRINT=( ( LSQ△△UN1△△UN2△△UN3△△UN4 ) ) ]
```

機　能

処理対象となったすべての見出語についてKWIC索引を作成する。

作業領域定義命令

CREATE命令における説明（V. 2）参照。

READ　ALAPHA命令

WORD　INDEX命令における説明（X. 1）参照。

WRITE　OUTPUT命令

FQEQ命令における説明（Ⅶ. 1）参照。

データ置換・選別命令

前節CONCORDANCE命令における説明を参照。

UNIT明細指示

前節CONCORDANCE命令における説明を参照。但し，本命令で，UN_iを選択したときには，当該テクスト・ユニットの境界を越えない範囲で，前後1行ずつが出力の対象となる。即ち，見出語のトークンの出現するLEX行と，（第iユニットの境界を越えない場合）その前後のLEX行1行が，追い込み形式で，KWIC索引の1行として出力される。勿論KWICの定義からして，1行分の文字数（LEXでは80バイト）を越える部分は出力されない。これに対して，LSQを指定した場合は，トークンの出現するLEX行1行だけがKWICの出力の対象となる。

MARKER明細指示

UNIT=LSQのときには与えてはならない。UNIT=LSQ以外のとき，即ち，出力の1行に複数行のLEX行が出力される可能性があるとき，テクスト構成要素の境界を表示する必要があれば，そのテクスト構成要素の略称で指定する。略称LSQ等の意味については，SAMPLE命令のKEY明細指示における説明（Ⅳ. 3）を参照。指定されたテクスト構成要素の境界が出力の1行中に存在するときには，斜線（／）で示される。但し，出力行の両端にあるときには示されない。省略時には，境界は勿論表示されない。

PRINT明細指示

前節CONCORDANCE命令に準ずる。

出力の詳細

a　見出語の配列はアルファベット順である。

b　複数のLEX行が出力される場合も含め，与えられる識別値は，見出語のトークンが出現するLEX行のそれである。

c　外部媒体出力のとき，見出語のひとつのトークンに関する出力は2レコードから成る。最初のレコードには，先頭から順に，

#	1バイト
空白	2バイト
見出語	wバイト*

次のレコードには，先頭から順に，

空白	6バイト
LEX行	80バイト
空白	1バイト
識別値	7nバイト

（書式は，（n）（X（1），A（6））

（但し，nは，**PRINT**明細指示で指定されたテクスト構成要素数）

*最大単語長。**INFO**命令で知ることができる。

注意事項

a　本命令の実行にはLEX行ファイルが必要である。またLEX行ファイルを**LXLINE**というファイル名称で定義しておかなくてはならない。（XV章参照）

b　テクストが長いときの分割処理については**WORD INDEX**命令の注意事項と使用例b（X. 1）参照。

c　**INCLUDE**命令を使い特定の見出語についてのみ，**UNIT=LSQ**として本命令を実行したときには，後述**KLIC**命令とほぼ同じ結果が得られるが，次の点が異なる。

ア．**KWIC**命令では，出力行中の単語の先頭の文次の位置が固定されるが，**KLIC**命令では，利用者の与えられた文字列の先頭の文字の位置が固定される。この文字列の先頭の文字は必ずしも単語の先頭の文字とは限らない。

イ．**KLIC**命令で利用者の与えた文字列はあくまで文字列として処理されるので，その文字列を含むすべてのLEX行が出力される。従って，これと同じ結果が得られるのは，**KWIC**命令の**INCLUDE**命令において，

POSITION=C△△B△△M△△E

としたときである。

ウ．KWIC命令では，イの場合，単語を見出しとして，出力が単語ごとに分けられるのに対し，KLIC命令では，LEX行がテクストでの出現順に出力される。

また，その他の用例索引作成の命令，CONCORDANCE，LINE INDEX 等とは出力の形式が異なる。

d　出力の1行に複数行が出力されるとき，LEX 行の先頭に空白が幾つあってもつめられることはない。

使 用 例

a　テクスト中のすべての見出語についてのKWIC索引を作成する。出力はLEX行1行だけとする。単語アルファベット順ファイルを使用する。

```
＜例 11. 3＞
KWIC
READ△ALPHA
UNIT=LSQ
```

b　文字列‘DEVELOP’を含むすべての見出語についてKWIC索引を作成する。第2ユニット（例えば文）の境界を越えない範囲で出力するが，他のテクスト構成要素の境界は表示しない。

```
＜例 11. 4＞
KWIC△INDEX
INCLUDE
PO͞SITIO͞N=C△△B△△M△△E
DEVELO͞P
%END
UNIT=UN2
```

c　文字列‘LOVE’を含むすべての見出語についてKWIC索引を作成する。原テクストが韻文で，原テクストの1行がLEX行に対応しているので前後の1行ずつ出力し，

LEX行の境界を表示する。

```
＜例 11. 5＞
KWIC
INCLUDE
POSITION=C△△B△△M△△E
LOVE
%END
UNIT=UN1
MARKER=LSQ
```

この例で，UNIT=UN1としたのは，LSQとすると，1行だけしか出力できないからである。もし第1ユニットとLEX行が1対1に対応するとすれば，UN1ではなくUN2，UN3，等を指定しなければならない。このようなテクスト・ユニットが存在しなければ，INPUT命令実行時にダミーのテクスト・ユニットを指定して，入力テクストの先頭に，

@

などのUBMをひとつ入れておけばよい。

XI. 3　単語列の用例索引——WORD SET CONCORDANCE命令

書　式

```
WORD△SET△CONCORDANCE
[WORK=i]　　(i≧10)＜i＝100＞
[WRITE△OUTPUT]
[データ置換・選別命令]
[BOUNDARY={LSQ, UN1, UN2, UN3, UN4}]
[ORDER={Y, N}]
```

```
[ CONTIG={Y, N} ]
[ RANGE=j ]    ( j≦500 ) < j = 20 >
[ PRINT=( ( LSQ△△UN1△△UN2△△UN3△△UN4 ) ) ]
  単語1
  単語2
 [単語3
  .....
  単語n]          ( 2≦n≦5 , n< j )
%END
```

機　　能

利用者の指定した単語列あるいは単語集合についての複数行出力可能なコンコーダンス（KWOC索引）を作成する。

作業領域定義命令

CREATE命令における説明（V. 2）を参照。

WRITE OUTPUT命令

FREQ命令における説明（Ⅶ. 1）を参照。

データ置換・選別命令

WORD SET INDEX命令における説明（X. 2）を参照。

BOUNDARY明細指示

WORD SET INDEX命令における説明（X. 2）を参照。但し，本命令では，この明細指示では，単語列，単語集合の範囲を規定するだけでなく，出力されるLEX行の範囲をも規定する。

ORDER明細指示，CONTIG明細指示，RANGE明細指示

WORD SET INDEX命令における説明（X. 2）を参照。

PRINT明細指示

一般的注意に関してはPRINT WORD命令における説明（VI. 2）を参照。但し，本命令では指定できるテクスト構成要素数は最大4個である。省略時解釈については，WORD INDEX命令における説明（X. 1）を参照。

単語リスト

WORD SET INDEX命令における説明（X. 2）を参照。

出力の詳細

a　用例が出力される単語列もしくは単語集合は，BOUNDARY以下の各明細指示によって決まる条件と，単語リストに与えられた単語によって決まる。この単語列もしくは，単語集合の総数は用例の出力に先立ちラインプリンターもしくは端末に出力される。このような単語列もしくは単語集合の個々のトークンにつき，そのトークンをすべて含む範囲のLEX行1行ないし複数行が，対応する用例として出力される。

b　識別値は出力されるすべてのLEX行に与えられる。

c　外部媒体出力の書式は，aで述べた各用例につき，次のようになる。まず，最初のレコードの先頭から，

#	1バイト
空白	119バイト

2番目以降のレコードから，用例が始まるが，各レコードの先頭から，

空白	5バイト
LEX行	lバイト（l＝LEX行の最大長）
識別値	$7n$バイト

（書式は，（n）（X（1），A（6））。nはPRINT明細指示で与えられたテクスト構成要素数）

先頭のレコードは言うまでもなく，個々の用例を区別するためのものである。

注意事項

a　本命令の実行にはLEX行ファイルが必要である。またLEX行ファイルをLXLINEというファイル名称で定義しておく必要がある。（XV章参照）

b　単語列や単語集合の数が多いと，作業領域が不足することがある。このときには，

WORK AREA INSUFFICIENT

というメッセージに続けて，作業領域定義命令で指定すべき領域の最小値がバイト単位で示される。（作業領域定義命令ではキロバイト単位で与えることに注意）

c　その他WORD SET INDEX命令における注意事項a，使用例も参照。

使用例

a　‘GENERAL AND COMPLETE DISARMAMENT’という単語列の用例を検索する。第1ユニットの境界は越えないものとする。

```
＜例 11. 6＞
WORD△SET△CONCORDANCE
BOUNDARY=UN1
GENERAL
AND
COMPLETE
DISARMAMENT
%END
```

b　‘TECHNOLOGY TRANSFER’という単語と，‘MNC’，‘MNCS’，

'MULTINATIONAL ENTERPRISE' のいずれかの単語とが第1ユニット内で共出現する用例を検索する。件数が多いと思われるので，作業領域を400 Kバイトとる。

```
<例 11. 7 >
WORD△SET△CONCORDANCE
WORK=400
REPLACE
TYPE=G
MNCS
MULTINATIONAL△ENTERPRISE
%BY
MNC
%END
BOUNDARY=UN1
ORDER=N
CONTIG=N
RANGE=100
TECHNOLOGY△TRANSFER
MNC
%END
```

c　WORD SET INDEX命令の使用例（X. 2）も参照。

XI. 4　文字列の用例索引——KLIC命令

書　式

```
KLIC
[ WRITE△OUTPUT ]
[ データ置換・選別命令 ]
  文字列1
```

```
[文字列2 ]
 ……
[文字列n ]        ( 1≦n≦100 )
%END
[PRINT=((LSQ△△UN1△△UN2△△UN3△△UN4))
```

機　　能

利用者の指定した文字列を含むLEX行のKWIC索引を作成する。この命令の特長は，単語あるいは単語列として定義できない文字列をキーとしてLEX行のKWIC形式索引を作成するところにある。

＊KLICは本来，1文字をキーとするKWIC形式の索引の意であるが，LEXでは，文字列のKWIC索引の意で用いている。

WRITE　OUTPUT命令

FREQ命令における説明（Ⅶ. 1）を参照。

データ置換・選別命令

PRINT　LINE命令における説明（Ⅵ. 3）と，INCLUDE命令，EXCLUDE命令の注意事項（Ⅳ. 5）を必ず参照。

本命令のもとでINCLUDE命令を用いた場合，INCLUDE命令に与えられた文字列のうちの少なくともひとつを含み，かつ本命令の文字列リストに与えられた文字列のうちの少なくともひとつを含むLEX行が出力される。EXCLUDE命令の場合は，EXCLUDE命令に与えられた文字列を一際含まず，かつ本命令の文字列リストに与えられた文字列のうちの少なくともひとつを含むLEX行が出力される。

文字列リスト

このリストに与えた文字列の少なくともひとつを含むLEX行が出力の対象となる。文字列は少なくともひとつ以上，最大100個まで与えることができる。文字列の長さは1文字以上，60文字までで，セミコロン以外のどのような記号，文字，数字を含んでい

てもよいが，先頭と末尾の空白は無視される。

リストの終りを示す%ENDステートメントは省略できない。

PRINT明細指示

一般的注意に関しては，PRINT WORD命令における説明（VI. 2）を参照。但し，本命令で指定できるテクスト構成要素数は4個までである。省略時の扱いについては，WORD INDEX命令における説明（X. 1）を参照。

出力の詳細

a 与えられた条件に合うLEX行が1行ずつテクストでの出現順に出力される。

b 外部媒体出力の書式は各レコードの先頭から，次のようになる。

LEX行　89バイト

識別値　7nバイト

（書式は，（n）（X（1），A（6））。nは，PRINT 明細指示で指定したテクスト構成要素数）

注意事項

a 本命令の実行にはLEX行ファイルが必要である。LEX行ファイルはLXLINEというファイル名称で定義しておかなければならない。（XV章参照）

b 本命令では，複数行出力はできない。KWIC形式でのLEX行複数行出力はKWIC INDEX命令を用いる。

c PRINT LINE命令，LINE INDEX命令でINCLUDE命令を用いた場合との相違は，次の通りである。

ア．PRINT LINE命令，LINE INDEX命令ではKWIC形式の出力はできない。

イ．LINE INDEX命令ではLEX行はそのアルファベット順に出力される。

d KWIC INDEX命令との相違については，KWIC INDEX 命令の注意事項c（XI. 2）を参照。

使 用 例

文字列‘, WHICH’, ‘, WHO’を含むLEX行を検索する。

```
<例 11. 8 >
KLIC
, WHICH
, WHO
%END
```

KWIC形式でなくてもよければ，PRINT LINE命令のもとでINCLUDE 命令を使っても同様の出力が得られる。

XI. 5 LEX行の索引 — LINE INDEX命令

書 式

```
LINE△INDEX
[ WORK=i ]    ( i≧10 )<i=100 >
[ データ置換・選別命令 ]
[ JUSTIF={Y, N} ]
[ PRINT=((LSQ△△UN1△△UN2△△UN3△△UN4)) ]
```

機 能

処理対象となったすべてのLEX行に関し，索引を作成する。

作業領域定義命令

CREATE命令における説明（V. 2）を参照。

WRITE OUTPUT命令

FREQにおける説明(Ⅶ. 1)を参照。

データ置換・選別命令

PRINT LINE命令における説明(Ⅵ. 3)と，INCLUDE命令，EXCLUDE命令の注意事項(Ⅳ. 5)を必ず参照。

JUSTIF明細指示

LINE LIST命令における説明(Ⅸ)を参照。

PRINT明細指示

PRINT WORD命令における説明(Ⅵ. 2)を参照。 但し，本命令で指定できるテクスト構成要素数は4個までである。省略時の扱いについてはWORD INDEX命令における説明(Ⅹ. 1)を参照。

出力の詳細

a LEX行は先頭の文字からのアルファベット順に配列(出力)される。

b 外部媒体出力の項目は各レコードの先頭から順に次のようになる。

空白	4バイト
LEX行	lバイト(l＝LEX行の最大長)
識別値	$8n$バイト (書式は，(n)(X(2)，A(6))。nは， PRINT明細指示で与えたテクスト構成要素数)

LEX行が直前のレコードのLEX行と同じ場合は，LEX行自体は出力されず次のような項目が出力される。

空白	(l＋4)バイト

識別値　8 nバイト

（ l , n及び書式は前述の通り）

注意事項

a　INCLUDE命令を用いたときのPRINT LINE命令等との相違については各命令を参照。

b　LINE LIST命令の注意事項a , bを必ず参照のこと。

使用例

a　テクストのすべてのLEX行を対象として索引を作成する。空白は左づめにする。

```
＜例 11. 9＞
LINE△INDEX
PRINT=UN1△△UN2
```

b　テクストが長いので，分割処理をする。まず‘A’で始まるLEX行を処理する。出力は後に併合するため外部媒体に行う。

```
＜例 11. 10＞
LINE△INDEX
WRITE△OUTPUT
INCLUDE
A
%END
```

このような分割処理でアルファベット以外，あるいは他の数字，文字以外，特に空白，で始まるLEX行があるときには，EXCLUDE命令を用いる。

Ⅻ　単語の共出現に関する出力

ふたつの単語XとYがあり，XとYが同一のテクスト・ユニット内にともに出現するとき，XとYは共出現するという。共出現関係はふたつの言語要素の親疎，遠近，距離を測るひとつの尺度であり，LEXでは，共出現関係に関する次のような3種類の出力がある。

共出現リスト：特定の単語（群）と共出現する単語のリスト
共出現マトリックス：与えられた単語相互間の共出現指標の算出
共出現素データ出力：共出現指標算出のためのデータの外部媒体出力

Ⅻ．1　共出現リスト ── COLIST命令

書　式

```
  COLIST
[ WORK = i ]    ( i ≧ 10 )    < i = 100 >
[ WRITE△OUTPUT ]
  UNIT = { LSQ, UN1, UN2, UN3, UN4 }
[ MODE = { R, M } ]
┌ 単　語1      ┐
│ 単　語2      │
│ .....        │
│ .....        │
└ 単　語n      ┘ ( 1 ≦ n ≦ 100 )
 %END
```

機　能

単語リストに与えられた単語のうちのひとつと指定されたテクスト構成要素内で共出現する単語のアルファベット順リストを，共出現指標を付けて作成する。

作業領域定義命令

CREATE 命令における説明（Ⅴ．２）を参照。

WRITE OUTPUT 命令

FREQ 命令における説明（Ⅶ．１）を参照。

UNIT 明細指示

この明細指示で指定されたテクスト構成要素内における共出現を問題にする。この明細指示は省略できない。テクスト構成要素の略称 LSQ 等についてはSAMPLE命令のKEY 明細指示（Ⅳ．３）を参照。

MODE 明細指示

各ユニットでの単語（正確には見出語）の出現度数の計算方法を指定する。省略時の解釈はRである。

今記号を次のように定める。

k_i ：単語Xが，UNIT 明細指示で指定されたテクスト構成要素の i 番目のものの中で実際に出現する回数。

X_i ：単語Xの当該テクスト構成要素中の出現度数（必ずしも実際の出現回数とは限らない。）

U_i ：当該テクスト構成要素中の単語総数。

M_x ：単語Xの平均出現率

N ：UNIT 明細指示で指定されたテクスト構成要素の（処理の対象となった）総数

キーワードR，Mの意味は次の通りである。

R：単語Xが出現すれば１とする。即ち，

$k_i = 0$ ならば $X_i = 0$

$k_i \geqq 1$ ならば $X_i = 1$

M：平均出現率を用いて出現度数を決める。単語Xの平均出現率M_xが，

$$M_x = \frac{1}{N} \Sigma (k_i / U_i)$$

で与えられるとき，出現度数X_iを

$k_i / U_i < M_x$ ならば $X_i = 0$

$k_i / U_i \geqq M_x$ ならば $X_i = 1$

とする。

単語リスト

少なくともひとつ，最大100個まで単語を与えることができる。リストの終りには%ENDステートメントが必要である。

ここで与えられた単語はひとつのグループないしはカテゴリー，あるいは何らかの意味をもつ単語の集合と見なされ，この集合と，共出現する単語のリストが作成される。ここで与えられた個々の単語と他の単語との共出現リストが個別的に作成されるわけではない。このため，幾つかの単語をグルーピングするためのREPLACE命令は使ってはならないし，また一般に使う必要もない。使う必要がもしあれば，CREATE SUB TEXT命令のもとでREPLACE命令を使ってサブテクスト・ファイルを作成しておき，これを本命令の入力ファイルとすればよい。

出力の詳細

a　リストの出力に先立ち，次の数値が，ラインプリンターまたは端末に出力される。

UNIT明細指示で指定されたテクスト構成要素のうち処理対象となった数（本節

の記号を用いればN）

単語リストに与えられた単語の総出現度数（必ずしも総出現回数とは限らない）

（ΣX_i）

MODE＝M であれば，単語リストに与えられた単語（群）の平均出現率

単語リストに与えられ単語と共出現する単語の種類数（見出語数）

b　リストでは，共出現する単語はアルファベット順に配列され，出力の1行（1レコード）に先頭から順に次の各項目が出力される。（　）は標準出力での見出し，＜　＞は外部媒体出力での書式を示す。

単語（見出語）＜A（w）＞（wは単語の最大長）

この単語のテクスト中での出現度数（FREQ）＜F(6)＞

この単語とリストに与えられた単語（群）との共出現度数（COFREQ）＜F(6)＞

この単語とリストに与えられた単語（群）とのユール（Yule）のQ（Q）

＜F(7，4)＞

この単語とリストに与えられた単語（群）との共出現係数（C）＜F(7，4)＞

この単語とリストに与えられた単語（群）との共出現率（R）＜F(7，4)＞

MODE＝M のときはこの単語の平均出現率（MOR）　＜F(7，4)＞

今，記号を次のように定める。

F_X，F_Y：単語X，Yそれぞれのテクスト中での出現度数。即ち，

$$F_X = \Sigma X_i,\quad F_Y = \Sigma Y_i$$

C_{XY}　：単語XとYのテクスト中での共出現度数

単語リストに与えられた単語の集合を単語X，Xと共出現する任意の単語をYとすると，出力される共出現度数C_{XY}は，

$$C_{XY} = \Sigma X_i\, Y_i$$

によって求められる。（X_i，Y_iはともに，前項の説明から明らかなように，1と0の2値である。） またユールのQは，

$$Q=\frac{N\cdot C_{XY}-F_X F_Y}{N\cdot C_{XY}-F_X F_Y+2(F_X-C_{XY})(F_Y-C_{XY})}$$

によって与えられる。共出現係数Cは，

$$C=2\left[\sum_{i=m}^{n}\left(\frac{(N-F_X)!F_X!(N-F_Y)!F_Y!}{i!(F_X-i)!(F_Y-i)!(N-F_X-F_Y+i)!N!}\right)\right]-1$$

$$\left(\begin{array}{c}\text{但し，}m=\max(0,\ F_X+F_Y-N),\\ n=\max(C_{XY}-1,\ m)\end{array}\right)$$

によって求められる。ユールのQ，共出現係数ともに－1と1の間の値をとり，正であれば正の相関，負であれば負の相関（排反）の関係があると言える。特に共出現係数の場合，

$C\geqq 0.98$ なら　有意水準0.01で関連
$C\geqq 0.90$ なら　有意水準0.05で関連
$C\geqq 0.80$ なら　有意水準 0.1で関連
$C<-0.80$ なら　有意水準 0.1で排反
$C<-0.90$ なら　有意水準0.05で排反
$C<-0.98$ なら　有意水準0.01で排反

という関係がある。一般に共出現度数がC_{XY}以上となる確率Pは，

$$P=\frac{1-C}{2}$$

によって与えられる。

また共出現率Rは，次の式によって与えられる。

$$R = C_{XY} / \sqrt{F_X F_Y}$$

共出現率は，一般的には concomitance index と呼ばれる統計指標であり，理論的には最小値0，最大値1をとるが，実際にとりうる値の範囲は，F_X，F_Y，Nによって決まる。

注意事項

a　テクストが長いときには，処理時間，記憶容量とも相当必要になる。特に作業用一時ファイルは大容量を要する。このような時には，CREATE SUBTEXT 命令とSAMPLE命令を使って*サブテクスト・ファイルを作成し，これを本命令の入力ファイルとして，全体の傾向を見，それに基づいてCOMATRIX命令を使う方法がある。（本項Cも参照）

＊ SAMPLE命令のkEY明細指示と本命令のUNIT明細指示には同じテクスト構成要素を指定するのが無難である。

b　本命令ではデータ置換・選別命令は一際使えない。必要があればCREATE SUBTEXT命令のもとで使用し，作成されたサブテクスト・ファイルを本命令の入力ファイルとする。但し，複数個のサブテクスト・ファイルを作成して，それを本命令で処理して結果を比較する場合，テクスト構成要素の総数（前項，前々項のN）が一定でなければ比較の意味が失くなるケースもあることを念頭に置く必要がある。勿論テクストの異なった部分における比較であれば問題はない。

c　本項aの場合，特定の単語群と共出現するすべての単語のリストが必要であれば，CREATE SUBTEXT命令のもとで，SAMPLE，SELECT(REJECT)，INCLUDE(EXCLUDE)等を用いてサブテクスト・ファイルを必要なだけ作り，本命令で処理すればよい。この場合，共出現度数は求められるが，他の共出現指標は，テクスト（T_i）全体をベースにして求めたものと異なる可能性があるので，必要に応じCOMATRIX命令を用いる。

使用例

〝ARMS RACE〟，〝ARMAMENTS RACE〟と第1ユニット（例えば文）で共出

現する単語のリストを作成する。出現度数は，第1ユニットに1回以上出現すれば1として計算する。

```
〈例 12. 1〉
COLIST
UNIT＝UNI
ARMS△RACE
ARMAMENTS△RACE
%END
```

Ⅻ. 2　共出現マトリックス——COMATRIX命令

書　式

```
  COMATRIX
[WRITE△OUTPUT]
  UNIT＝{LSQ, UN1, UN2, UN3, UN4}
[MODE＝{R, M}]
[LIST＝{A, 1}]
  単　語1
  単　語2
 [単　語3    ]
 […          ]
 [単　語n    ]  ( 2≦n≦100 )
  %END
```

機　能

指定された単語相互の共出現指標を算出する。出力はマトリックス形式ではなく，前節COLIST命令と同じリスト形式になる。

WRITE OUTPUT 命令

FREQ 命令における説明（Ⅶ 1 ）を参照。

UNIT 明細指示

前節 COLIST 命令における説明を参照。

MODE 明細指示

前節 COLIST 命令における説明を必ず参照のこと。

LIST 明細指示

共出現指標の計算を，単語リストに与えた単語相互に関して行うか否かを指定する。省略時には，A が仮定される。

A：リストに与えられたすべての単語間の共出現指標を計算する。

1：リストの最初の単語と，他の単語との共出現指標を計算する。

単語リスト

少なくとも 2 語，最大 100 語まで単語を与えることができる。最語には %END ステートメントが必要である。LIST＝1 の場合には，最初にどの単語を与えるかが意味をもつ。

出力の詳細

a　単語リストに与えられた個々の単語に関し，リスト中の他の単語との共出現指標が COLIST 命令の場合と同じリスト形式で個別に出力される。（前節 COLIST 命令の出力の詳細 b，c を参照）。但し，本命令では単語は単語リストに与えられた順に出力される。また，LIST＝1 のときには，リストの最初の単語についてのみ他の単語との共出現リストが出力される。

b　単語リストに与えられた単語の数を n とすると，外部媒体出力では，最初の n レコードに，最初の単語とリスト中の他の単語（最初の単語自体も含む）との共出現指標等が出力され，次の n レコードに 2 番目の単語と他の単語との共出現データが

出力される，という形で出力が行われる。勿論，LIST＝1のときは最初の単語に関する出力しか行われない。各レコードの内容は前節のCOLIST命令とまったく同じである。

注意事項

a　単語リストに与えられた単語の出現度数が0であるとエラーになる。

b　データ置換・選別命令はまったく使えない。必要があれば，CREATE SUBTEXT 命令のもとで使用してサブテクスト・ファイルを作成し，これを本命令の入力ファイルとする。この時の注意に関しては，前節の注意事項bを参照。

使 用 例

a　様々な観点からの開発概念の相互関係を調べるため，第2ユニットにおけるその共出現関係を求める。同一の第2ユニット内に複数回出現する可能性があるので，平均出現率を用いて標準化する。場合によっては，多変量解析等他のプログラムによる分析を行うので，外部媒体に出力しておく。

```
〈例 12. 2 〉
COMATRIX
WRITE△OUTPUT
UNIT＝UN2
MODE＝M
SOCIAL△DEVELOPMENT
POLITICAL△DEVELOPMENT
INDUSTRIAL△DEVELOPMENT
ECONOMIC△DEVELOPMENT
NATIONAL△DEVELOPMENT
REGIONAL△DEVELOPMENT
URBAN△DEVELOPMENT
RURAL△DEVELOPMENT
INTERNATIONAL△DEVELOPMENT
%END
```

b 〝PEACE〟と他の単語との共出現関係を調べる。

```
〈例 12. 3 〉
COMATRIX
UNIT=UN1
LIST=1
PEACE
DISARMAMENT
DEVELOPMENT
BASIC△HUMAN△NEEDS
ARMS
ARMS△RACE
%END
```

Ⅻ. 3 共出現素データ出力——CODATA 命令

書　式

```
 CODATA
 UNIT={LSQ, UN1, UN2, UN3, UN4}
[MODE={R, M, X}]
 単 語1
 単 語2
[単 語3
 …
 単 語n]      ( 2≦n≦100 )
 %END
```

機　　能

与えられた単語のテクスト構成要素ごとの出現度数を外部記憶媒体に出力する。

UNIT明細指示

ここで指定されたテクスト構成要素の個々の現われを単位として（あるいはひとつのケースないしはサンプルとして）与えられた単語の出現度数に関するデータが作成される。この明細指示は省略できない。キーワードLSQ等の意味についてはSAMPLE命令のKEY明細指示（Ⅳ．3）を参照。

MODE明細指示

COLIST命令における説明（本章1節）を参照。本命令では，R，Mのほか，Xという形式も許されるが，これは単語の実際の出現回数k_iを出現度数とする方法である。Xを指定したときには，後述のように出力の1ケースが複数のレコードに分割される。

単語リスト

少なくと2個，最大100個までの単語を与えることができる。最後に%ENDステートメントが必要である。このリストに与えられた単語のすべてについて，UNIT明細指示で指定されたテクスト構成要素のi番目の現われにおける出現度数X_iが求められ，データとして出力される。

出力の詳細

a　MODE＝RまたはMODE＝Mのときには，出力の1レコードの詳細な書式がライン・プリンターまたは端末に出力される。外部記憶媒体の1レコードはテクスト構成要素Uのi番目の現われU_iに対応し，先頭から順に，

U_iの識別値　　＜A(6)＞

U_i中の単語総数（U_iの長さ）　　＜F(6)＞

単語$_1$～単語$_n$のU_i中での出現度数

＜(n)(F(1))＞（nはリストに与えた単語数）

が出力される。

b　これに対して，MODE＝Xの場合は，U_iに関するデータはレコード長6バイトの（$n+2$）レコードに分割され，先頭のレコードから順に

U_iの識別値	（1レコード）
U_iの長さ	（1レコード）
単語$_1$～単語$_n$の出現度数	（nレコード）

という形で出力される。

c　a，bいずれの場合にも，出力されたレコード数とレコード長がライン・プリンターか端末に出力される。MODE＝RまたはMODE＝Mのとき，このレコード数はテクスト構成要素数に等しく，MODE＝Xのときは，テクスト構成要素数の（$n+2$）倍に等しい。但し，ここでのテクスト構成要素数からは，単語$_1$～単語$_n$の出現度数（MODE≠Xであれば実際の出現回数とは限らない）の合計が0となるテクスト構成要素は除かれている。

注意事項

a　出力用ファイルをLEXOUTという名称で定義しなければならない。このファイルのレコード長はMODE＝Xならば6バイト，他の場合は（$12+n$）バイトとしてもよい。他の点については通常のLEXOUTファイルと同様である。（XV章参照）

b　本命令は他の命令と異なり外部記憶媒体出力だけを目的としたものであるが，これは，出力したデータを他のプログラムで処理するためである。

c　データ置換・選別命令は一際使用できない。この点に関しては，本章COLIST命令，COMATRIX命令の関連個所を参照。

使用例

例12．2（COMATRIX命令の使用例a）と同じように，開発概念の相互関係を調べるため，まず素データを出力する。

```
〈例 12. 4〉
CŌDATA
UNIT=UN2
MŌDE=M
SŌCIAL△DEVELŌPMENT
PŌLITICAL△DEVELŌPMENT
INDUSTRIAL△DEVELŌPMENT
ECŌNŌMIC△DEVELŌPMENT
NATIŌNAL△DEVELŌPMENT
REGIŌNAL△DEVELŌPMENT
URBAN△DEVELŌPMENT
RURAL△DEVELŌPMENT
INTERNATIŌNAL△DEVELŌPMENT
%END
```

出力されたデータは例えば多変量解析のプログラムの入力データとすることができる。このとき，このデータは，PL／Ⅰならば，

A (6)， F (6)， (9) F (1)

FORTRANならば，

A 6，F 6.0，9F 1.0

という書式で読めばよい。

XIII　テクスト構成要素の長さ

XIII 1. 単語の長さ——WORD LENGTH命令

書　式

```
WORD△LENGTH
[WORK=i]      (i≧10)  <i=100>
[WRITE△OUTPUT]
[データ置換・選別命令]
```

機　能

テクスト（T_i）中のすべての単語の長さを文字数で計算し，その統計値と度数分布表を出力する。

作業領域定義命令

CREATE命令における説明（V．2）を参照。

WRITE OUTPUT命令

FREQ命令における説明（VII．1）を参照。

データ置換・選別命令

自由に用いてよい。

出力の詳細

a　単語の長さは文字数で測られる。これには，途中の空白，記号等も含む。

b　統計値としては次のものが求められる。

平　均

分　　散
標準偏差
最 大 値
最 小 値

これに加えて，処理対象となった単語の総数も出力される。

これらの値は常にライン・プリンターまたは端末に出力され，外部記憶媒体に出力されることはない。

c　長さの度数分布表が統計値に続けて出力される。長さは値の小さい順に出力され，ひとつの値に対するデータが1行（または1レコード）となる。1行（1レコード）の各項目は先頭から順に，次のようになる。（　）は標準出力の見出し，＜　＞は外部媒体出力の書式である。

長さ（LENGTH）　＜F(6)＞
度数（当該の長さの単語の総数）（FREQ）＜F(6)＞
度数の百分比（単語総数に対する）（PCT）＜F(7, 2)＞

注意事項（略）

使 用 例

テクスト中のすべての単語を対象に長さを求める。

```
〈例 13.  1〉
WORD△LENGTH
```

XIII. 2.　LEX行の長さ―― LINE　LENGTH命令

書　　式

```
 LINE△LENGTH
[WORK=i]     (i≧10)  <i=100>
[WRITE△OUTPUT]
[データ置換・選別命令]
[MEASURE={W, C}]
```

機　　能

テクスト（T_i）中のすべてのLEX行の長さを計算し，その統計値と度数分布表を出力する。

作業領域定義命令

GREATE命令における説明（Ⅴ．2）を参照。

WRITE　OUTPUT命令

FREQ命令における説明（Ⅶ．1）を参照。

データ置換・選別命令

任意。但し，次のMEASURE明細指示でCを指定するときには一切用いてはならない。

MEASURE明細指示

LEX行の長さを測る単位を指定する。省略時にはWが仮定される。

W：単語数で長さを測る。

C：文字数で長さを測る。このとき，LEX行の先頭，途中の空白，すべての区切り記号等も文字数に計算される。

Cを指定した時には，データ置換・選別命令は使用できない。またLEX行ファイルを使用するので，LXLINEという名称で定義しておく必要がある。

出力の詳細

前節WORD　LENGTH命令の出力の詳細b，cに準ずる。但し，「単語総数」とある個所は，「LEX行総数」と読み換える。

注意事項（略）

使 用 例

a　単語数で測ったLEX行の長さに関するデータを出力する。

```
〈例13. 2〉
LINE△LENGTH
```

b　文字数で測ったLEX行の長さに関するデータを出力する。

```
〈例13. 3〉
LINE△LENGTH
MEASURE=C
```

XIII. 3. テクスト・ユニットの長さ—— UNIT　LENGTH命令

書　式

```
  UNIT△LENGTH
[WORK=i]     (i≧10)   <i=100>
[WRITE△OUTPUT]
[データ置換・選別命令]
  UNIT={UN1, UN2, UN3, UN4}
  MEASURE={WSQ, LSQ, UN1, UN2, UN3, UN4}
```

機　　能

指定されたテクスト構成要素数で測ったテクスト・ユニットの長さに関する統計値と度数分布表を出力する。

作業領域定義命令

CREATE 命令における説明（Ⅴ．２）を参照。

WRITE　OUTPUT 命令

FREQ 命令における説明（Ⅶ．１）を参照。

データ置換・選別命令

任意。但し，次の UNIT 明細指示で与えるテクスト・ユニットが壊れるような命令を用いたときの出力結果は保証されない。

UNIT 明細指示

どのテクスト・ユニットの長さを求めるかを指定する。この明細指示は省略できない。キーワード UN1 等の意味については SAMPLE 命令の KEY 明細指示（Ⅳ．３）を参照。

MEASURE 明細指示

長さを測る単位となるテクスト構成要素を指定する。この明細指示は省略できない。キーワード WSQ 等の意味については，SAMPLE 命令の KEY 明細指示（Ⅳ．３）を参照。

UNIT 明細指示で指定されたテクスト・ユニットの長さが，ここで指定されたテクスト構成要素数で測られる。従って，UNIT 明細指示で指定したのと同じテクスト・ユニットを指定してはならないし，意味のある結果を得るためには，ここで指定したテクスト構成要素は，UNIT 明細指示で指定したテクスト・ユニットの直接構成要素と見なしうるものでなくてはならない。（Ⅲ．１参照）

出力の詳細

本章WORD　LENGTH命令の出力の詳細b，cに準ずる。但し，「単語総数」は「テクスト・ユニット総数」と読み換える。

使用例

a　第1ユニットの長さを単語数で測る。

```
〈例 13.　4 〉
UNIT△LENGTH
UNIT=UN1
MEASURE=WSQ
```

b　第2ユニットの長さをLEX行数で測る。

```
〈例 13.　5 〉
UNIT△LENGTH
UNIT=UN2
MEASURE=LSQ
```

c　第3ユニット（例えば段落）の長さを，第1ユニット（例えば文）の数で測る。

```
〈例 13.　6 〉
UNIT△LENGTH
UNIT=UN3
MEASURE=UN1
```

XIV　その他の実行命令

XIV. 1. 識別値のリストとテクスト構造 ―― VALUE△LIST 命令

書　式

```
  VALUE△LIST
[データ置換・選別命令]
[TYPE={L, A}
  UNIT={UN1, UN2, UN3, UN4}
[UNIT={UN1, UN2, UN3, UN4}
[UNIT={UN1, UN2, UN3, UN4}
[UNIT={UN1, UN2, UN3, UN4}
```

機　能

テクスト・ユニットの識別値の出現順もしくはアルファベット順リストを作成する。ふたつ以上のテクスト・ユニットを組合せれば，簡単なテクスト構造の表示となる。

データ置換・選別命令

任意。

TYPE 明細指示

出現順（ sequential ）リストかアルファベット順リストかを指定する。省略時には L が仮定される。

L：出現順リスト

A：アルファベット順リスト

UNIT 明細指示

TYPE＝Aのときにはひとつだけ与える。省略することも，ふたつ以上与えることもできない。どのテクスト・ユニットの識別値についてアルファベット順リストを作成するかを指定する。

TYPE＝Lのときには，少なくともひとつは，必ず与えなくてはならない。このときには，2つ以上，INPUT命令で宣言したテクスト・ユニットの数まで与えることができる。いずれの場合にも，存在しないテクスト・ユニットを指定してはならない。ふたつ以上のUNIT明細指示を与えるときに，同一のテクスト・ユニットを指定しても無意味である。

出力の詳細

a　TYPE＝Aのときには，識別値がアルファベット順に出力され，それぞれに度数（その識別値を有するテクスト・ユニット数）が付けられる。

b　TYPE＝LでUNIT明細指示をひとつだけ与えたときには，当該テクスト・ユニットの識別値がテクストでの出現順に印刷される。

c　TYPE＝Lでふたつ以上のUNIT明細指示を与えた場合には，各テクスト・ユニットの識別値が同一行にテクストでの出現順に印刷されることになるが，新たなユニットが現われないときには空白が印刷される。UNIT明細指示が，

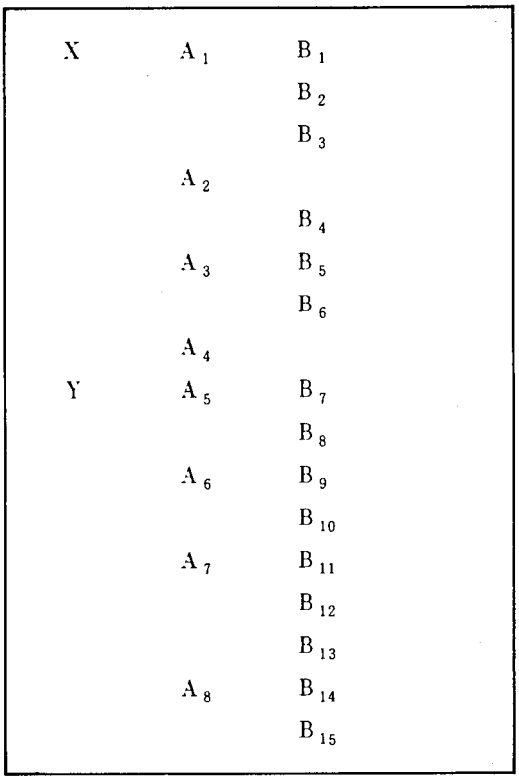
```
UNIT＝UN3
UNIT＝UN1
UNIT＝UN2
```

という形で与えられており，テクスト構造が，

<table>
<tr><td colspan="15">テ　　ク　　ス　　ト</td><td></td></tr>
<tr><td colspan="7">X</td><td colspan="8">Y</td><td>第3ユニット</td></tr>
<tr><td colspan="3">A_1</td><td>A_2</td><td colspan="2">A_3</td><td>A_4</td><td>A_5</td><td colspan="2">A_6</td><td colspan="2">A_7</td><td colspan="2">A_8</td><td>第1ユニット</td></tr>
<tr><td>B_1</td><td>B_2</td><td>B_3</td><td>B_4</td><td>B_5</td><td>B_6</td><td>B_7</td><td>B_8</td><td>B_9</td><td>B_{10}</td><td>B_{11}</td><td>B_{12}</td><td>B_{13}</td><td>B_{14}</td><td>B_{15}</td><td>第2ユニット</td></tr>
</table>

（X，Y，A_i，B_jは各ユニットの識別値を示す。）

のように表わされるものとするならば，出力は図式的には，次のようになる。



このように，TYPE＝L でふたつ以上の UNIT 明細指示を与える場合は，テクスト構造の大まかな表示と見なすことができる。

注意事項

a　本命令は，主として識別モードがAモード，即ち識別値が文字型のテクスト・ユニットを処理することを目的としているが，識別モードが他のモードであっても，識別値が正しく付けられているかどうかのチェックのために用いることもできる。

b　TYPE＝L で，複数の UNIT 明細指示を与える場合，指定されるテクスト・ユニットには，部分的な例外はあっても，階層関係があることが望ましい。勿論そうでない場合も，テクスト構造の表示という目的に特に拘泥しないのであれば，処理の対象として差支えない。

使 用 例

a　第1ユニットの識別値のアルファベット順リストを作成する。

```
〈例 14. 1〉
VALUE△LIST
TYPE=A
UNIT=UN1
```

b　第1～第3ユニットのリストを出力する。

```
〈例 14. 2〉
VALUE△LIST
UNIT=UN1
UNIT=UN2
UNIT=UN3
```

XIV. 2. テクストに関する情報——INFO命令

書　式

```
INFO
[READ△LEXSUB]
```

機　能

LEXテクスト・ファイル，サブテクスト・ファイル，見出語ファイル，単語アルファベット順ファイルに記録されているテクスト情報を印刷する。

READ LEXSUB命令

見出語ファイル，単語アルファベット順ファイルのテクスト情報を印刷するとき与える。この命令を与えたときには，当該ファイルをLEXSUBというファイル名称で定義

しておかなければならない。

出力の詳細

出力されるのは次の情報である。（　）は出力における略称を示す。

テクスト・ファイル中の単語総数（WTOTAL）

当該ファイル中の単語総数（FTOTAL）

（一般にFTOTAL≦WTOTAL）

当該ファイル中の見出語総数（KTOTAL）

（見出語ファイル以外ではKTOTAL＝0）

LEX行の総数（LTOTAL）

（行ファイルが作成されていなければ，LTOTAL＝0）

単語の最大長（MXWORD）

（INPUT命令のWORD LENGTH明細指示で与えられた長さ，あるいはMERGE WORD命令実行後であれば，実質最大長）

LEX行の最大長（MXLINE）

（行ファイルが作成されていなければ，MXLINE＝0）

宣言されたテクスト・ユニット数（UNUM）

テクスト・ユニットの名称（UNAME）

注意事項

a　外部記憶媒体出力の書式を確定するために，単語やLEX行の最大長を知る必要があるときには，この命令を使うことになるが，この命令の入力ファイルは外部記憶媒体出力を行った実行命令の入力ファイルとしたファイルでなくてはならない。

b　前項で特に，READ ALPHA，READ KEYWORDを用いた場合には，本命令でREAD LEXSUB命令を用い，見出語ファイルまたは単語アルファベット順ファイルを入力ファイルとしなければならない。

使 用 例（略）

XV　LEX入出力ファイルとJCL（ジョッブ制御言語）

本章ではLEXの実行に必要なJCL（ジョッブ制御言語）について述べる。JCLは，機種，設置機関によって異なるので，ここでは，広島大学総合情報処理センターのHITAC M−180およびM−200H，即ちVOS3システムにおけるJCLを説明する。また，ここでは所謂バッチ処理の例のみについて論ずる。以下の例はTSS端末からサブミットすることもできるが，TSSジョッブであれば変更する必要があることは言うまでもない。TSSジョッブとして実行する場合については別稿に述べる。

JCLで主として問題となるのは，LEXに関して言えば，ファイル（VOS3ではデータセット）の定義に関するものである。そこで，まず図15. 1にLEXの入出力ファイルの名称とその関係を図示しておく。図のファイル名称は，DD文（ファイル定義のJCL）でdd名として用いられるものである。

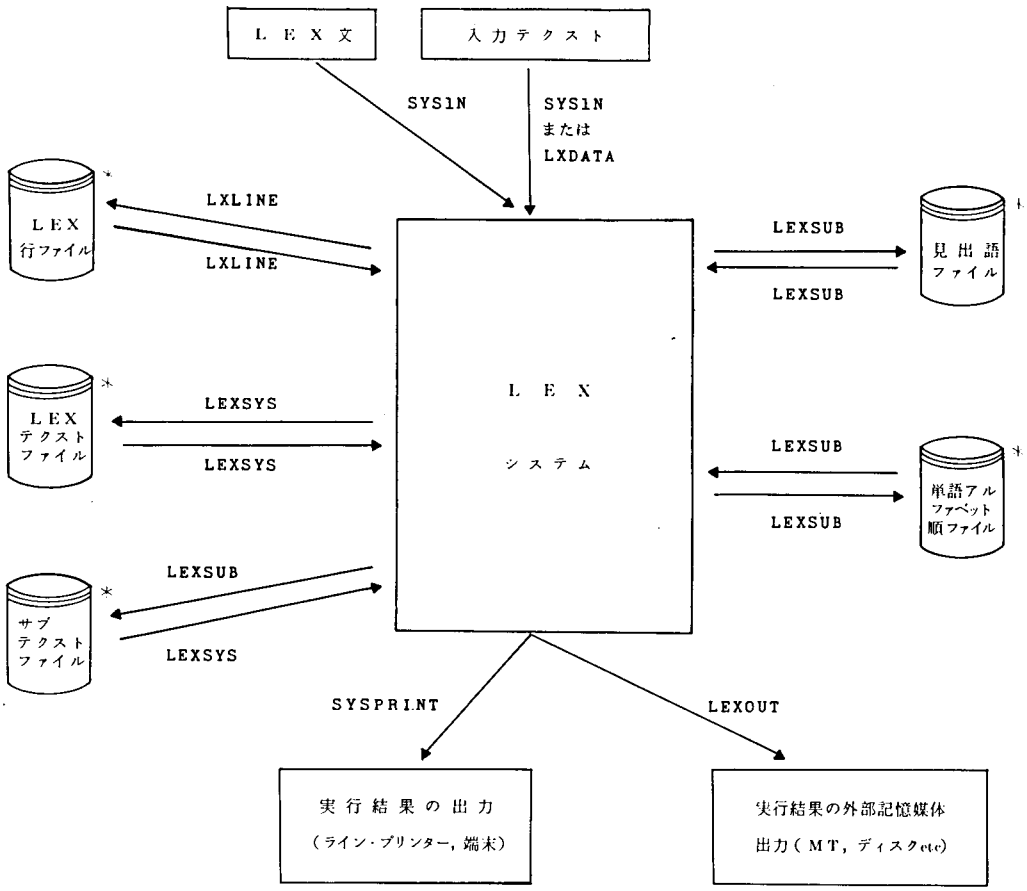
LEXの実行に必要なファイルは次の通りである。（＊印はLEXに固有のファイル名称である）

STEPLIB：LEXのロード・モジュール

LEXSYS*：INPUT命令の出力ファイル（LEXテクスト・ファイル）。
他の実行命令の入力ファイル（LEXテクスト・ファイル，サブテクスト・ファイル）

LXLINE*：INPUT命令の出力（LEX行ファイル）。PRINT LINE, LINE LIST, CONCORDANCE, KWIC INDEX, WORD SET CONCORDANCE, KLIC, LINE INDEX, LINE LENGTH（MEASURE＝C）各命令の入力ファイル（LEX行ファイル）

LEXSUB*：CREATE命令の出力ファイル（サブテクスト・ファイル，見出語ファイル，単語アルファベット順ファイル）。MERGE WORD命令の出力ファイル（サブテクスト・ファイル）。
READ KEYWORDを指定した命令の入力ファイル（見出語ファ



＊記憶媒体はディスクである必要はない。
矢印：：ＬＥＸへの入出力を示す。
ゴチック体はファイル名称（ｄｄ名）を示す。
作業用一時ファイルは省略されている。

図15. 1　ＬＥＸ入出力ファイルと名称（ｄｄ名）

イル），READ　ALPHAを指定した命令の入力ファイル（単語アルファベット順ファイル），INFO命令でREAD　LEXSUB　を指定したときの入力ファイル（見出語ファイル，単語アルファベット順ファイル）

LEXOUT*　：WRITE　OUTPUTを指定したときの出力用ファイル。（外部記

憶媒体出力用）

LXDATA* ：INPUT命令，PRINT RAW DATA命令で，入力データまたはテクストがLEX文と異なる媒体上にあるときの入力ファイル。

LXWRK1～LXWRK3* ：作業用一時ファイル。

SYSIN ：LEX文入力ファイル。INPUT命令，PRINT RAW DATA命令のデータまたはテクスト入力ファイル（LEX文と同一媒体上にあるとき）

SYSPRINT ：標準出力ファイル。

以下，いくつかの場合に分けてJCLの例を挙げる。例では次の点に注意を要する。

ア：dsnとあるところにはテクスト・ファイル，サブテクスト・ファイル等のデータセット名（ファイルの実体名）を指定する。

イ：SPACE，BLKSIZEは必要に応じて変更してよい。

ウ：LEXOUTファイルのレコード長（LRECL）も，各実行命令に合わせて変更してよい。

エ：その他LEXに固有の部分以外は変更してよい。例えばSYSPRINTをディスク出力にする，SYSINをディスクあるいはMT入力とする等。

INPUT命令のJCL

```
//JOB UiD …                                                    ①
//   EXEC PGM=MAIN                                             ②
//STEPLIB DD DSN=LX81LD, DISP=OLD                              ③
//LEXSYS DD DSN=dsn, DISP=(NEW, CATLG), SPACE=(TRK, (10, 10), RLSE), ④
//   DCB=(RECFM=VB, LRECL=140, BLKSIZE=18904)                  ⑤
//LXLINE DD DSN=dsn, DISP=(NEW, CATLG), SPACE=(TRK, (10, 10), RLSE), ⑥
//   DCB=(RECFM=VB, LRECL=140, BLKSIZE=18904)                  ⑦
//LXWRK1 DD DSN=&&LEX1, SPACE=(CYL, (1, 1)),                   ⑨
//   DCB=(RECFM=VB, LRECL=140, BLKSIZE=18904)                  ⑩
```

```
//SYSPRINT DD SYSOUT=*                                    ⑪
//SYSIN DD *                                              ⑫
LEX文(INPUT命令)                                          ⑬
入力テクスト                                              ⑭
//                                                        ⑮
```

行ファイルを作成しないのであれば，⑥，⑦は不要。

INPUT命令で，READ LXDATAを指定したときには，⑦と⑨の間に，

```
//LXDATA DD DSN=dsn, DISP=OLD                             ⑧
```

を入れ，さらに⑭をとる。この形のJCLになるのは，JCLとLEX文が同一媒体上にあり，入力テクストがLXDATAで定義される他の媒体上にあるときである。

PRINT RAW DATA命令のJCL

```
//JOB UiD …                                               ①
//   EXEC PGM=MAIN                                        ②
//STEPLIB DD DSN=LX81LD, DISP=OLD                         ③
//SYSPRINT DD SYSOUT=*                                    ⑤
//SYSIN DD *                                              ⑥
LEX文(PRINT RAW DATA命令)                                 ⑦
入力データ                                                ⑧
//                                                        ⑨
```

READ LXDATAを指定したとき，即ちLE文と入力データが別の媒体にあるときには，③と⑤の間に，

```
//LXDATA DD DSN=dsn, DISP=OLD                             ④
```

を入れ，さらに⑧をとる。

INFO命令のJCL

```
//JOB UiD …                                          ①
//   EXEC PGM=MAIN                                   ②
//STEPLIB DD DSN=LX81LD, DISP=OLD                    ③
//LEXSYS DD DSN=dsn, DISP=OLD                        ④
//SYPRINT DD SYSOUT=*                                ⑥
//SYSIN DD *                                         ⑦
LEX文                                                 ⑧
//                                                   ⑨
```

READ LEXSUBを指定したときには，④と⑥の間に，見出語ファイルまたは単語アルファベット順ファイルを定義する

```
//LEXSUB DD DSN=dsn, DISP=OLD                        ⑤
```

を与えなければならない。dsnには，見出語ファイルまたは単語アルファベット順ファイルのデータセット名を与える。

その他の命令のJCL

```
//JOB UiD …                                          ①
//   EXEC PGM=MAIN                                   ②
//STEPLIB DD DSN=LX81LD, DISP=OLD                    ③
//LEXSYS DD DSN=dsn, DISP=OLD                        ④
//LXWRK1 DD DSN=&&LEX1, SPACE=(CYL, (1, 1)),         ⑩
//   DCB=(RECFM=VB, LRECL=140, BLKSIZE=18904)        ⑪
```

```
//LXWRK2  DD  DSN=&&LEX2, SPACE=(CYL, (1, 1)),          ⑫
//    DCB=(RECFM=VB, LRECL=140, BLKSIZE=18904)          ⑬
//LXWRK3  DD  DSN=&&LEX3, SPACE=(CYL, (1, 1)),          ⑭
//    DCB=(RECFM=VB, LRECL=140, BLKSIZE=18904)          ⑮
//SYSPRINT DD SYSOUT=*                                  ⑯
//SYSIN  DD *                                           ⑰
 LEX 文                                                 ⑱
//                                                      ⑲
```

ア　④の dsn は，処理の対象となるテクスト・ファイルまたはサブテクスト・ファイルのデータセット名

イ　行ファイルが必要であれば，④の次に，

```
//LXLINE  DD  DSN=dsn, DISP=OLD                 ⑤
```

を与える。

ウ　**CREATE** 命令，**MERGE WORD** 命令では，④の次に，

```
//LEXSUB  DD  DSN=dsn, DISP=(NEW, CATLG), SPACE=(TRK, (10, 10), RLSE),  ⑥
//    DCB=(RECFM=VB, LRECL=140, BLKSIZE=18904 )                          ⑦
```

が必要。dsn はここでは，新規作成されるサブテクスト・ファイル，見出語ファイル，または単語アルファベット順ファイルのデータセット名。

エ　**WRITE OUTPUT** 命令を与えた場合には，④の次に，

```
//LEXOUT  DD  DSN=dsn, DISP=(NEW, CATLG), SPACE=(TRK, (10, 10), RLSE),  ⑧
//    DCB=(RECFM=FB, LRECL=120, BLKSIZE=3600 )                           ⑨
```

を与える。

オ　READ ALPHA, READ KEYWORD, READ LEXSUBのいずれかを指定したときには，④の次に

```
//LEXSUB DD DSN=dsn, DISP=OLD                    ⑩
```

を入れる。

カ　前述イ～オのふたつ以上が重複するときには，与える順序は任意である。但し，⑥と⑦，⑧と⑨はこの順序で与えなくてはならない。