「広島平和科学」14 (1991) pp.129－186 ISSN0386-3565
Hiroshima Peace Science 14 (1991)

漢字テクスト検索システムＫＲ

松尾 雅嗣
広島大学平和科学研究センター

KR: A Retrieval System for Japanese Texts

Masatsugu MATSUO
Institute for Peace Science, Hiroshima University

SUMMARY

The present paper is an interim report of KR, a simple concording and word counting microcomputer program for full texts of Japanese, that is, texts represented in kanji, kana and other 2 byte symbols. The program was developed as part of a research project on "Full Text Data Base of Documents of Atomic Bomb Damages", a report of which is also in this issue.

The program, written in the C language for portability, is intended, first of all, as an easy and quick tool of searching and retrieving parts of a text where a given word or string apprears. In view of this, the usually time-consuming procedure of preparing Japanese texts,, which requires the delimitation of every word in the texts and the manual lemmmatization or explicit specification of the rules of lemmatization, are all drastically simplified. Users are expected only to prepare an MS－DOS text (or ASCII) files. The process of

seraching conducted by a menu is also intended to be simple and quick. But, it cannot be so simple and easy because the program must satisfy many different reseach needs of users. Therefore, the program offers the following facilities as options.

searching of a pair of terms,

reordering, merging, and pairing of results of a search

limiting of the range of search to part(s) or subtext(s)

simultaneous counting and/or searching of more than one terms

目次

第１章　概説

１　概要

漢字テクスト検索システムＫＲは，漢字テクストの用例検索を主たる目的とするプログラム群である。ここで言う漢字テクストには，漢字仮名交じりのテクスト，正確には２バイトコードで構成されたテクストを含む。

本プログラムは，テクストの検索を主目的とするものであり，文字，語彙の数量的分析，文法的分析は対象としない。但し，補助的機能として，以下の機能を有する。

サブテクストの作成
文字の度数つきリスト作成
文字列の度数算出
グループ別文字列頻度リストの出力
テクストの外部出力
ＫＷＩＣ索引の一括作成

ＫＲは，以下の環境で実行可能である。

OS	MS－DOS　(2.xx 以前に関しては保証の限りでない)
対応機種	NEC PC 9801 シリーズ
メモリ	640K

ハードディスク，プリンタは必須ではないが，あることが望ましい。また，モニターは，カラー，モノクロいずれでも可。

また，その目的からして，日本語フロントエンド・プロセッサは不可欠である。画面２０行で使用でき，画面の最下行を使用した後は，以前表示されていた行を復活できることが望ましい。

導入は以下の手順で行なう。

ハードディスクで利用する場合，提供された実行用プログラムを（ルートを含む）特定のディレクトリにすべてコピーする。

データも提供を受けた場合は，適当なディレクトリを作成し，そこにコピーする。

このとき，異なるデータ（テクスト）は，別のディレクトリに置かなくてはならない。

なお，ＫＲを実行するための条件は正確には以下の通りであり，これが満足される限り，ハードディスクとフロッピィディスクの併用あるいはフロッピィディスクのみでも実行できる。

① ＫＲを構成するプログラム群はすべてカレントディレクトリにあること

② ひとつのテクストを構成するファイル群は，すべて同一のディレクトリにあること

2 実行

2．1 実行開始

ＫＲの実行開始には，MS－DOS のプロンプトに対し，

KRMN

というコマンドを投入する。すると，ＫＲのメインメニュが表示されるので実行するメニュを選択する。以後，ＫＲは原則としてすべてメニュにより実行する。なお，メニュによらないで実行する方法については，「参考」を参照。

ＫＲのメインメニュでは，以下のいずれかが選択可能である。

Ｓ：検索

D：テクスト定義
I：テクスト入力
T：サブテクスト
U：ユーティリティ
Q：終了

各メニュの機能は次のとおり。
＜検索＞は，テクストの検索処理を行なう。詳細は，第2章「検索」で述べる。
＜テクスト定義＞は新たに入力するテクストの諸属性を定義する。＜テクスト入力＞は，テクストのシステムへの入力処理を行なう。いずれについても，詳細は第3章「テクストの入力」で述べる。
＜サブテクスト＞は，サブテクストの作成，削除，一覧などの処理を行なう。詳細は第4章「補助機能」で述べる。
＜ユーティリティ＞は，テクスト情報の出力，テクストのディスク出力，文字，文字列の頻度つきリストの作成，KWIC索引の一括作成などの処理を行なう。詳細は第4章「補助機能」で述べる。
＜終了＞は，KRの実行を終了する。

メニュの選択は，次のいずれかの方法で行なう。

① カーソルを選択すべき項目に合わせ，リターンキーを押す。
② 各メニュの先頭に与えられている半角1文字を入力する。

この選択方法は，KRのすべてのメニュに共通である。

2．2　実行時のエラー

プログラム実行時のエラーには幾つかの種類がある。
プログラムそのもののエラーは，利用者による修正は不可能である。

メモリの不足，ディスク容量の不足，ファイル名の誤りなど実行中には回復不能のエラーが生じたときには，その旨のメッセージを表示して実行を中止する。

実行時のメモリの不足に対しては，可能ならば，フロントエンド・プロセッサを含めた常駐プログラムをはずして実行する，ディスクファイルからの入力があるときには，入力ファイルを分割するなどの方法があるが，いずれも決定的なものではない。

利用者の入力ミスなど実行中に回復可能なエラーの検出されたときにも，同様にメッセージを表示するが，実行は中断しない。

プログラムがエラーを検出しなくても，利用者の意図した結果が得られないというエラーもありうる。プログラムにその機能が本来ない場合も，やり方によっては可能な場合もありうる。

2．3　実行終了

ひとつのメニュの実行が正常に終了するとその旨のメッセージが表示され，その後当該メニュを呼出した親メニュに戻る。メインメニュに戻って，＜終了＞を選択すれば，ＫＲの実行終了となる。

3　テクストの選択

ＫＲの検索は，利用者が予めMS−DOSテクストファイルとして作成した入力テクストを，索引などを付加して変換した（これを入力と呼ぶ）データを使用して行なう。検索あるいは補助機能の実行に使用する形に変換されたデータを，以下単にテクストと呼ぶ。ＫＲで使用するテクストは，実際には幾つかのファイルから構成され，いずれも“.KTR”という拡張子が付されている。

ＫＲの実行に際しては，ひとつのテクストを構成するファイルは，同一のディレクトリになくてはならない。また，異なるテクストは常に異なるディレクトリになくてはならない。換言すれば，ＫＲは，ディレクトリによってテクストを区別する。

それゆえ，ＫＲのメニュ選択の後，当該メニュの実行が開始されると，処理対象とするテクストを選択しなければならない。この選択は以下の手順で行なう。一度使用されたテクストは，ＫＲが自動的に記憶，登録する。但し，自動的に登録できるテクストは最大５０までであり，これを越えるとエラーになる。この場合，他のテクストを削除するなどの対応が必要である。

① 直前に使用（あるいは定義）したテクストがあれば，その名称が表示される。まったくなければ，テクストの存在するディレクトリに関する問合せがある。これに対しては，④以下の手順による。

② テクストの名称の表示に続いて，このテクストを使用（入力，検索など）するか否かの問い合せがある。

表示されたテクストでよければ，その旨応答すれば，選択したメニュの実行が開始される。否と応答したときには，システムに登録されたテクストの有無によって，手順が異なる。

システムに登録されたテクストがないときには，④以下の手順により，テクストの存在するディレクトリを指定する。他の場合には，次の③による。

③ このときには，既存のテクストのリストから選択するか否かの問い合せがある。

選択すると答えたときには，既存の，即ち登録済のテクストの一覧が画面に表示される。カーソルを動かして適当なテクストを選択し，リターンキーを押せばよい。このとき，テクストを選択しないとエラーとなる。選択が終ると，②により，確認が行なわれる。

④ テクストのパス名を入力せよという指示に対しては，使用するテクストの存在するディレクトリ名を入力する。このときには，ドライブ名で始まる絶対パス名として与えることと，パス名の最後に¥記号を付けることを忘れてはならない。指定したパスにテクストがないときには，その旨のメッセージと処理を続行するかという問い合せがある。処理を続行すると応答したときには，再びパス名の入力が要求される。継続しないと応答したときには，ＫＲを終了する。

指定したパス名にテクストが存在するときには，②によりテクストの確認が行なわれる。

以上の記述から明らかなように，テクストがどのディレクトリにあるかを指示する必要があるのは，当該のテクストを初めて使用するとき，正確には，ディレクトリ名がＫＲに知られていないとき，だけであり，以後，使用するテクストの選択は，既存の登録済リストを使用して行なえばよい。

登録済のテクストを削除したときにも，ＫＲが自動的に対応する。

第二章　検索

1　ＫＲにおける検索の概要

ＫＲは，基本的には文字単位の処理システムであり，検索も単語ではなく，１文字以上の文字列をキーとして行なう。文字列には後述のワイルドカードを含むことができる。また，ふたつの文字列をを指定して，文字列の対として検索することもできる。

いずれの場合も，サブテクストが作成されていれば，特定のサブテクストだけを検索の対象とすることもできる。

検索に際しては，利用者がまず検索すべき文字列を与える。与えれた文字列が存在すれば，システムはその文字列のテクスト中の用例から成る集合を作成する。これを用例集合と呼ぶ。用例集合が作成されたときには，直ちに画面表示することもできる。また後に画面表示，ディスク出力，印刷することもできる。

出力の形式としては，文字列の対から成る用例集合を除き，ＫＷＩＣ形式と伝統的なコンコーダンス形式のいずれの形式でも出力可能である。また，出力の形式に関しては，幾つかのパラメータにより利用者がある程度まで制御できる。

作成された用例集合は，当該の検索処理実行中は何度でも出力が可能である。また，ペアリング，併合，並べ替えなどの簡単な演算も可能である。

１回の検索メニュ実行中には，最大２０個までの用例集合を作成できる。

検索メニュ終了時には作成されたすべての用例集合（に関する情報）は失われる。

2　検索メニュ

＜検索＞で使用可能なメニュとその機能の概要を以下に示す。

F：文字列検索	単一の文字列の検索
P：ペア検索	文字列のペア（対）の検索
M：用例ペア／併合	用例集合のペアリング，併合
R：並べ替え	用例集合中の用例の出力順の並べ替え
C：用例画面表示	用例集合の画面表示
O：ディスク出力	ひとつの用例集合全体をディスクファイルに出力
P：印刷	ひとつの用例集合全体を印刷
S：サブテクスト	サブテクスト，フルテクストの切替
T：テクスト情報	テクスト情報の出力
I：用例集合情報	既存の用例集合の一覧
Z：パラメータ	検索，出力パラメータの表示，変更
Q：終了	検索の終了

3　標準出力ファイル

検索実行時のディスク出力に際しては，「標準出力ファイル」を使用することができる。これは，予め定められたファイル（標準では，カレント・ディレクトリのKTR.OUT）であり，利用者が特に指定しない限り，ディスクへの出力はすべてこのファイルに行なわれる。

標準出力ファイルへの出力は常に追加モードで行なわれる。即ち，新たな出力は，以前に出力された部分の最後に追加される。それゆえ，場合によっては，きわめて大きなファイルとなることもあるので，適宜削除する必要がある。標準出力ファイルが存在しければ，新たに作成されるので，不要になったとき削除することに何ら差し支えはない。

標準出力ファイルの名称は，後述の＜パラメータ＞機能を利用して自由に変更できる。また，この名称の設定はテクストごとに行なわれるので，異なるテクス

トに異なる標準出力ファイルを設定することもできる。

標準出力ファイルに自動的に出力されるときを除き，検索実行時等では，標準出力ファイルと利用者指定のファイルのいずれかを選択できる。後者を選択したときは，出力ファイル名を与えなければならない。このとき，標準出力ファイルと同名のファイル名を与えることは危険である。標準出力ファイルを利用者が選択したとき以外は，常に上書きモードで出力されるからである。

4　用例集合の作成

4．1　文字列検索

＜文字列検索＞は，指定された文字列の検索を行なう。サブテクストが指定されていれば，その範囲内が検索対象となる。それ以外の場合には，テクスト全体が検索対象となる。

文字列検索に際しては，指示に従い検索すべき文字列を入力する。文字列の長さは全角１文字以上，１０文字までである。

＜文字列検索＞では，ただひとつの文字列だけを指定できる。複数の文字列の用例集合を作成するには，この＜文字列検索＞と後述の＜用例ペア／併合＞を順次組合せて行なう。

文字列にはワイルドカード文字が含まれていてもよい。但し，ＫＲにおけるワイルドカード文字は，MS−DOS におけるそれと異なる。第一に，文字列の両端にあるワイルドカード文字は無視される。第二に，ワイルドカード文字は，任意の全角文字１文字（もしくは半角２文字）のみを表わす。従って，「山？？川」と「山？川」では，検索結果は異なる。ワイルドカード文字（標準では，全角の？）は，＜パラメータ＞により変更できる。

指定された文字列が存在しないときには，その旨のメッセージを表示して用例検索メニュに戻る。

指定された文字列が存在するときには，新たな用例集合を作成し，その番号と用例の件数を表示する。そして，これに続けて，直ちに画面表示するか否かとい

う問合せがある。ここで，表示すると応答したときには，検索された用例が表示される。表示の形式等については，＜用例の画面表示＞の項に述べる。表示しないと応答したときには，用例検索メニュに戻る。

4．2　ペア検索

＜ペア検索＞は，ふたつの文字列を指定して，それが共に出現する用例を検索する。サブテクストが指定されていれば，その範囲内が検索対象となる。それ以外の場合には，テクスト全体が検索対象となる。

検索に際しては，指示に従い，文字列，出現範囲，出現の順序などの検索条件を指定する。ＫＲのペア検索では，他の条件の如何に拘らず，ふたつの文字列が全角２００文字かつテクスト行１１行の範囲にある場合のみが検索の対象となる。検索の制約等に関しては，４．２．２も参照。

4．2．1　ペア検索の方法

まず，ふたつの文字列を順にひとつずつ与える。指定する文字列に課される制約は，＜文字列検索＞の場合とまったく同じである。ただ，検索条件によっては，ふたつの文字列のいずれを先に与えるかが意味をもつ。

これに次いで，検索の条件を指示する。まず，プロンプトに対し，与えたふたつの文字列に順序を想定するか否かを指示する。

「順序あり」を選択したときには，「最初に与えた文字列」，「ふたつめの文字列」という順序で出現する場合だけを検索の対象とする。これに対し，「順序なし」を選択したときには，このような制約は課されない。

次いで，ふたつの文字列が共に出現する境界を設定する。まず境界そのものを設定するか否かを指定する。

境界を設定しないと応答したときには，ふたつの文字列が前述の全角２００文字かつテクスト行で１１行の範囲内にある場合を検索の対象とする。

境界を設定すると応答したときには，ふたつの文字列が以後に指定されるテク

スト構成要素内で（正確には，同一のトークン内に）出現する場合のみが検索の対象となる。

この境界としては，利用者がテクスト入力時に設定したテクストユニットあるいはテクスト行１行が可能である。

後者を境界として指定したときには，ふたつの文字列が同一のテクスト行に共に出現する用例が検索される。

前者を境界とするときには，画面に表示されるテクストユニットのうちから，ひとつを番号で選択する。この場合，ふたつの文字列が選択されたテクストユニットの同一のトークンに共に出現する用例が検索される。

以上の検索条件の設定が終ると「作業中」のメッセージが出て，直ちに検索が開始される。テクストの大きさに応じて多少の時間を要する。

検索後の処理は，＜文字列検索＞の場合と同じである。条件を満足する用例がなければ，その旨のメッセージを表示して，検索メニュに戻る。あれば，用例集合の番号と件数を表示して直ちに画面表示するか否かを尋ねる。以下の処理は＜用例画面表示＞の項に述べる。

４．２．２　ペア検索に関する制約と補足

前述のように，ＫＲのペア検索では，ふたつの文字列が，他の条件の如何に拘らず，全角２００文字，テクスト行１１行の範囲にある用例のみを検索する。他の条件が指定されたときには，それに加えて，この制約が課される。

ペア検索の場合，一般には，ふたつの文字列の隣接／非隣接という条件を課すことができる。しかし，ＫＲは文字単位の検索システムであり，隣接する場合は，ひとつの文字列として検索可能である。それゆえ，検索条件としてはこの条件は設定しない。このことは，しかしながら，ペア検索の結果が，ふたつの文字列が隣接する場合を排除することを意味しない。検索結果には，利用者の指定したふたつの文字列が隣接する用例も含む。

また，非隣接の文字列ペアについても，ふたつの文字列間の距離が短く，かつ容易に特定できるときには，＜ペア検索＞ではなく，＜文字列検索＞のほうが効

率がよい場合もある。

複数の文字列から成るふたつの文字列群のペアの検索は，＜文字列検索＞と次項の＜用例ペア／併合＞を順次組み合せて行なう。

4．3　用例集合のペアリングと併合

＜用例ペア／併合＞は，既存の用例集合のペアあるいは併合を行ない，新た用例集合を作成する。

これは，既に作成された用例集合のうち，ペアでもなくかつ並べ替えも行なわれていない集合についてのみ可能である。逆に言えば，＜文字列検索＞の結果として作成された用例集合と，本コマンドの併合により作成された用例集合は，常に本コマンドによる処理が可能である。

実行には，まず，ペアか併合のいずれかを選択する。

「ペア」は，既存のふたつの用例集合に関して，＜ペア検索＞と同じ処理を行なう。このときには，ペアリングする用例集合を選択した後，ペアを作成するための条件を設定する。

ペアリングのための条件の設定，制約等は，＜ペア検索＞の場合とまったく同じである。従って，＜文字列検索＞を2度行ない，「核兵器」と「廃絶」それぞれの用例集合を作成した後，本コマンドにより，「核兵器」と「廃絶」を共に含む用例を作成する場合と，＜ペア検索＞で「核兵器」と「廃絶」を指定して用例集合を作成する場合とでは，検索条件等他の条件が同じであれば，まったく同一の結果，即ちまったく同一の用例集合が得られる。

これに対し，「併合」は，ふたつの用例集合を文字どおり併合して新しい用例集合を作成する。結果は，元のふたつの用例集合に含まれるすべての用例を含む集合である。「廃棄」，「廃絶」，「廃止」のいずれかを含む用例を検索するときには，この3つの文字列ごとに＜文字列検索＞を行なった後，本コマンドで順次併合すればよい。

その結果を，「核兵器」で＜文字列検索＞した集合とペアリングすれば，「核兵器」と，「廃棄」，「廃止」，「廃絶」の3つのうちのいずれかひとつとを共に含む

用例集合を作成することができる。

処理の方式を定めた後，処理対象とする用例集合を一度にひとつずつ合計ふたつ選択する。画面に既存の用例集合のリストが表示されるので，対象とする用例集合にカーソルを合せて，リターンキーを押せばよい。

前述の如く，既にペアリングされた用例集合と，並べ替えの行なわれた，即ち＜並べ替え＞の結果作成された用例集合は選択できない。また，言うまでないことながら，同一の用例集合を選択することは無意味である。

処理が終了すると，新たな用例集合の番号と用例の件数が表示され，直ちに画面表示するか否かを尋ねる。ペアリングの場合，条件を満たす用例が存在しないこともある。画面表示する際の詳細は，＜用例画面表示＞に述べる。

4．4　並べ替え

任意の用例集合においては，用例はテクストでの出現順に並べられており，この順に出力される。＜並べ替え＞は，用例集合の中の各用例を，検索キーとした文字列の直後の文字列あるいは直前の文字列の内部コード順に並べ替えて新たな用例集合を作成する。

まず，並べ替えの対象とする用例集合を，画面に表示された既存の用例集合のリストの中からひとつ選択する。対象とする用例集合にカーソルを合わせ，リターンキーを押す。

このとき，既に並べ替えの行なわれた用例集合とペアの用例集合は選択できない。選択できるのは，＜文字列検索＞の結果作成された用例集合と，＜用例ペア／併合＞の併合処理によって作成された用例集合のみである。

用例集合が決定すると，次の検索キーの前の文字列によってソートするか，後の文字列によってソートするかを指示する。

いずれの場合も＜文字列検索＞時に利用者の与えたキーの直前もしくは直後の最大全角１０文字によってソートする。１１文字以上が同一である場合には，テクストでの出現順となる。

並べ替えは，全角文字のＪＩＳコードの大小による。直前の文字列によって並

べ替えるときには，所謂綴字逆順で，即ちテクストでの出現順では後の文字から，この場合検索キーに近い文字から比較を行なう。

処理が終了すると，新たな用例集合の番号と用例の件数が表示され，直ちに画面表示するか否かを尋ねる。画面表示する際の詳細は，＜用例画面表示＞に述べる。

5　用例集合の出力

5．1　出力先と出力形式

作成された用例集合は，画面，ディスクファイル，プリンタのいずれにも出力できる。但し，ＫＲの印刷機能はきわめて貧弱であり，見苦しくても早いほうがいい場合を除き，一度ディスクファイルに出力した後，他のソフトを用いて印刷することを奨める。

また，用例集合を画面表示するときには，表示された個々の用例をディスクファイルに出力したり，印刷することもできる。

用例集合の出力先の選択は，＜用例画面表示＞，＜ディスク出力＞，＜印刷＞の３つのメニュのいずれかを選択することにより行なう。

また，いずれの場合も，出力する用例集合を，画面に表示された用例集合のリストの中から，ひとつ選択しなければならない。

次いで，ＫＷＩＣ形式か，コンコーダンス形式かの選択を行なう。出力先の如何に拘らず，出力の形式は，ＫＷＩＣ形式とコンコーダンス形式が可能である。但し，ペアの用例集合の場合は，ＫＷＩＣ形式での出力は当然のことながら不可能であり，コンコーダンス形式の出力のみが可能である。

このふたつの出力形式に加えて，用例の文脈として出力する範囲，行数などを利用者が制御することができる。このような用例出力のパラメータは，出力先，出力形式により異なるものも，共通のものもある。出力はこれらのパラメータに従って行なわれる。パラメータの変更は，＜パラメータ＞によって行なう。

5．2　用例出力のパラメータ

用例を出力するときには，以下のパラメータに従って出力が行なわれる。確認，変更の方法に関しては，「パラメータ」の項を参照。

① 標準出力ファイル

本章の3を参照。検索メニュでの＜ディスク出力＞では，標準出力ファイル以外のファイルも指定できるが，画面表示中に用例を1件だけ出力するときには，常に標準出力ファイルに出力される。

② 索引項目

用例の出現個所を示すために個々の用例に付ける巻，章の番号，名称などの項目。使用者が指定したテクストユニットの識別値が指定した順に与えられる。

用例に付けれられる索引項目は，固定長ではなく，当該の用例の識別値に従い常に可変長で出力される。

③ 用例出力境界

用例の前後に文脈として付ける部分をテクストユニット単位で指定する。指定することも指定しないこともできる。任意のテクストユニットを出力境界として指定したときには，そのテクストユニットの境界を越えない範囲が用例の文脈として出力されるが，この範囲内でも，用例の出現する行の前後5行以内に制約される。画面表示の場合には，さらに，画面に表示できる範囲内に制約される。

このパラメータは，後述の文脈行数に優先する。また，このパラメータで指定した出力境界と，ペア検索を行なうとき指定する検索範囲とは独立である。

④ 検索キー表示色，反転表示，点滅表示

この3個のパラメータは，コンコーダンス形式での画面表示に関してのみ有効である。それぞれ検索キーの色，反転表示，点滅表示を行なうか否かを指定する。

カラーディスプレイの場合，色だけで変えれば検索キーは容易に識別できるが，モノクロの場合は，反転表示にしなければ識別は面倒である。点滅表示も必要なら可能である。

⑤ 文脈行数

コンコーダンス形式の出力の場合，用例を含む行の前後に文脈として何行ずつ出力するかを指定する。但し，上述の用例出力境界が指定されているときには，

この指定はできない。また，画面表示のときには，指定の範囲内であっても画面に表示できないない部分が生じうる。

ペアの用例集合の場合，行数は，最初の検索キーの出現する行の前と二番目の検索キーの出現する行の後にそれぞれ指定された行数が付加される。画面表示での制約は同様である。

⑥　検索キー標識

コンコーダンス形式のディスク出力と印刷では，検索キーを迅速に認識することが場合により困難である。それゆえ，検索キーの前後に特定の１文字を付けて出力することができる。この標識を設定するときには，全角１文字（あるいは半角２文字）で設定する。設定しなければ，検索キーの前後には何のマーカーも入らない。

⑦　行境界標識

ＫＷＩＣ形式の出力に際して，テクスト行の境界を示す全角の斜線（／）を入れるか否かを指定する。設定しなければ，出力では行の境界は表示されない。

⑧　ＫＷＩＣ行長

ディスク出力，印刷するときのＫＷＩＣの１行の文字数を指定する。この文字数は，上述の索引項目を含まない長さであり，出力の１行の実際の長さは，この長さに索引として出力される部分の長さを加えたものとなる。但し，行境界，前方，後方分離記号も，この文字数に含まれる。

画面表示の場合には，この値は固定されており，変更できない。

⑨　ＫＷＩＣ前方分離記号と後方分離記号

ＫＷＩＣ形式での出力時に，検索キーの前後にそれぞれ全角空白を入れるか否かを指定する。

５．３　用例の画面表示

５．３．１　用例表示メニュ

＜用例画面表示＞は，選択された用例集合に属する用例を１件ずつ画面に表示

する。

表示に先立ってまず，表示する用例集合を，画面に表示されたリストの中から，ひとつ選択する。用例集合がひとつしかないとき，または用例集合作成後直ちに画面表示を行なうときには，この選択は不要である。

出力形式を選択すると，直ちに画面が切り替わり，用例表示画面となる。用例表示画面では以下の「用例表示」メニュが使用可能である。但し，KWIC形式では＜ジャンプ＞機能は使用できない。

D：次の用例	次の用例を画面に表示する
U：前の用例	直前の用例を画面に表示する
O：ディスク出力	画面に表示された用例をディスクに出力する
P：印刷	画面に表示された用例を印刷する
T：最初の用例	最初の用例を画面に表示する
B：最後の用例	最後の用例を画面に表示する
J：ジャンプ	指定した用例を画面に表示する
Q：終了	用例の表示を終了する

画面表示の場合，用例の表示される行の上に，その用例の用例集合中での番号と，テクストでの出現個所がパラメータ「索引項目」に従って表示される。

画面に表示される用例の数と形式は，出力形式により異なる。

画面表示の終了後には，検索メニュに戻る。

5．3．2　KWIC形式

この形式では，画面に最大１３個の用例が表示され，そのうちの１行だけが反転表示される。用例表示部の最上段の用例番号とテクストでの出現個所は，反転表示された用例のものである。反転表示された，即ち，カレントの用例は，上記用例表示メニュにより変更，移動することができる。

用例の画面表示中にディスク出力，印刷の対象となるのは，この反転表示され

た用例ひとつだけである。またディスク出力ファイルは，標準出力ファイルのみである。出力の細目は，パラメータの設定により画面表示のそれとは異なりうる。

5．3．3　コンコーダンス形式

この形式では，用例を1件ずつ表示する。用例表示部の最上段には，表示されている用例の番号とテクストでの出現個所が表示される。

用例の移動，変更は，KWIC形式の場合と同じく，上記用例表示メニュによるが，この形式では，用例番号を指定したジャンプが可能である。この場合，指示に従ってジャンプ先の用例番号を入力する。

用例の画面表示中にディスク出力，印刷の対象となるのは，画面表示されている用例ひとつだけである。またディスク出力ファイルは，標準出力ファイルのみである。出力の細目は，パラメータの設定により画面表示のそれとは異なりうる。

5．4　用例集合のディスク出力

検索メニュの＜ディスク出力＞は選択された用例集合に属するすべての用例をディスクファイルに出力する。

このときには，出力する用例集合の選択，出力形式の選択とともに，出力先を標準出力ファイルとするか否かも指定する。標準出力ファイル以外に出力するときには，出力ファイル名を与える。ディスク出力の内容は，パラメータの設定次第で，画面に表示される内容と異なることが多い。

出力終了後は，検索メニュに戻る。

5．5　用例集合の印刷

検索メニュの＜印刷＞は選択された用例集合に属するすべての用例を印刷する。

このときには，前述のように，出力する用例集合の選択，出力形式の選択を行なう。印刷される内容は，パラメータの設定次第で，画面に表示される内容と異

なることが多い。

出力終了後は，検索メニュに戻る。

なお，KRの印刷機能は，貧弱であり，確認のために汚くても早いプリントアウトでよいときのほかには，ディスクに出力しその結果を編集して印刷するほうが望ましい。KRは，実際に出力されるデータと，プリンタそのものとの整合性，プリンタの1行の最大長，1頁の最大行数などとの整合性は一切考慮しない。

6　検索補助機能

6．1　サブテクスト

＜サブテクスト＞は，テクスト全体を検索対象とするか，特定のサブテクストだけを検索対象とするかを指定する。言うまでもなく，サブテクストが作成されていなければ，このメニュを選択しても無意味である。

サブテクストは，KRのメインメニュの＜サブテクスト＞により作成することができる。

このメニュを選択すると，現在テクスト全体が検索範囲となっているか，特定のサブテクストが検索範囲となっているかが示される。なお，検索開始時には，検索範囲は自動的にテクスト全体となっている。

ここで，変更するか否かの問合せがあるので，適宜応答する。

変更すると答えたときには，さらに，テクスト全体とサブテクストのいずれかを選択する。サブテクストを選択したときには，画面に表示されるサブテクストのリストの内から，検索範囲として指定するものをひとつ選ぶ。サブテクストがひとつだけのときには，この選択は行なわれず，当該のサブテクストを選択したものと仮定する。

以後＜文字列検索＞と＜ペア検索＞は，ここで指定した範囲を対象として行なわれる。なお，検索された用例の文脈として出力される部分に関しては，この制約は課されない。

サブテクストを指定した検索は，必ずしも高速化されない。これは，KRが，

検索時には速度を犠牲にしても使用メモリとディスク容量を節約するよう設計されているためである。

6．2　テクスト情報

現在使用中のテクストの概要を画面に出力する。

テクストに関する詳細な情報は，KRのメインメニュの＜ユーティリティ＞を選択し，＜ユーティリティ＞のメニュである＜テクスト情報＞により，画面表示，ディスク出力，印刷できる。

6．3　用例集合情報

現在実行中の検索処理においてこれまでに作成された用例集合に関する簡単な情報を出力する。出力されるのは，集合番号，作成の経緯，当該の集合に含まれる用例の件数である。

出力先は，画面，ディスクファイル，プリンタのいずれかであり，ひとつを選択する。ディスク出力では，標準出力ファイルあるいは利用者指定のファイルいずれも使用できる。後者の場合，出力ファイル名を与えなければならない。

出力終了後は，検索メニュに戻る。

7　パラメータ

7．1　パラメータの表示と変更の方法

＜パラメータ＞は検索用の諸パラメータの表示と変更を行なう。変更，表示の方法は以下のとおりである。なお，パラメータの設定は，テクストに固有であり，テクストごとに異なってよい。また，パラメータの変更は，恒久的である。即ち，KRの実行が終了しても記憶され，次に変更されるまで設定した値が保持される。

画面にパラメータのリストが表示されるので，まず表示あるいは変更したいパ

ラメータをひとつしかもひとつだけ選択する。

次に選択されたパラメータの現在の値が表示されるので，変更するのであれば，「イエス」，その必要がなければ「ノー」と答える。

変更の方法については，パラメータにより異なるので，詳細は後述する。新たな値が設定されると，確認の問い合せがあるので必要な応答を行なう。

表示だけの場合，変更を行なった場合（新たに設定した値をキャンセルした場合も含む）いずれの場合にも，他に表示・変更する項目があるかという問い合せがあるので，適宜応答する。必要なら，「ある」と応答して幾つかのパラメータを順次変更する。

7．2　パラメータの意味と変更の詳細

各項目の詳細に関しては，5．2も参照。

① 標準出力ファイル

本章の3を参照。変更するときには，新たなファイル名を入力する。

② 索引項目

用例の出現個所を示すために個々の用例に付ける巻，章の番号，名称などの項目。使用者が指定したテクストユニットの識別値が指定した順に与えられる。

変更するときには，画面に表示されたテクストユニットのうちから，その識別値を索引項目として出力したいものを，ひとつずつ，指示に従い順次選択する。すべてのテクストユニットを指定する必要はない。選択したテクストユニットの識別値が選択した順に用例などの出現個所としてとして出力される。

指定が終了すると，新たな索引項目が表示されるので，それでいいかどうか確認する。ここで否と応答したときには，元の設定が残る。

③ ワイルドカード文字

ワイルドカード文字として使用する全角の（通常は）記号を定義する。変更に際しては，新たなワイルドカード文字を全角1文字（半角2文字）で定義する。

④ 用例出力境界

用例の前後に文脈として付ける部分をテクストユニット単位で指定する。指定することも指定しないこともできる。

変更するときには，まず，用例出力境界を定義するか否かという問い合せがあるから，いずれか応答する。

定義しないと応答したときには，用例出力境界は，設定されない。また，文脈行数は自動的にゼロに再設定されるので，必要に応じて，「文脈行数」を変更する。

定義すると応答したときには，画面に表示されたテクストユニットのうちから，用例出力境界とするものをひとつ選択する。このとき，識別モードが行モードであるテクストユニットを選択すると用例の出現する1行のみが出力対象となる。

⑤ 検索キー表示色，反転表示，点滅表示

それぞれ検索キーの色，反転表示，点滅表示を行なうか否かを指定する。

検索キーの表示色を変更するときには，画面に表示された色の名称を番号により選択する。

反転表示，点滅表示では，

オン：指定の表示を行なう
オフ：指定の表示を行なわない

に従い，表示形式を指定する。

⑥ 文脈行数

コンコーダンス形式の出力の場合，用例を含む行の前後に文脈として何行ずつ出力するかを指定する。但し，上述の用例出力境界が指定されているときには，この指定はできない。変更するときには，前後に出力する行数を0から5までの整数で指示する。

⑦ 検索キー標識

コンコーダンス形式のディスク出力と印刷では，検索キーの前後に特定の1文字を付けて出力することができる。この標識を設定するときには，全角1文字（あるいは半角2文字）（常識的には記号）で設定する。

⑧ 行境界標識

KWIC形式の出力に際して，テクスト行の境界を示す全角の斜線（／）を入れるか否かを指定する。設定しなければ，出力では行の境界は表示されない。

⑨　KWIC行長

ディスク出力，印刷するときのKWICの1行の文字数を指定する。

変更に当たっては，行の長さを文字数で指定する。この文字数は半角文字数であり，全角文字では半分の文字数となる。指定は，50から120までの偶数で行なう。奇数の値を与えた場合の結果は保証されない。KRは，また，ここで指定した長さが出力媒体，とくにプリンタの1行の長さを越えるか否かはチェックしない。

⑩　KWIC前方分離記号と後方分離記号

KWIC形式での出力時に，検索キーの前後にそれぞれ全角空白を入れるか否かを指定する。

第三章　テクストの入力

1　テクスト入力の概要

KRによる検索その他の処理を行なうためには，対象とするテクストをKRが処理できる形に変換する必要がある。これを以下単にテクストの入力と言う。テクストの入力は実際には，以下のステップに分かたれる。

①　処理対象とするテクストの選定

利用者はまず対象とするテクストを選定する。ここで言うテクストは，一般には，書物，印刷物など，文字に記録されたものであるが，インタビュー・データのように文字の形で記録されていないものであってもよいし，本来特別のまとまりをもったものでなくてもよい。実際，KRで言うテクストとは，利用者にとって意味をもつ限り，文字の連鎖であれば，それ以外の制約はない。この段階でのテクストを原テクストと呼ぶ。

② 原テクストは，コンピュータが処理できる形に変えなければならない。これを機械可読化と言う。機械可読化に際しては，後述のＫＲが許容する形式に従わなければならない。機械可読化されたテクストを入力テクストと呼ぶ。

③ 入力テクストをＫＲが高速かつ効率的に処理できる形式に変換する処理が，テクストの入力である。このステップはＫＲではふたつに分かたれる。

第一に，テクストの名称，構造，使用する記号など，入力テクストに関する情報をＫＲに知らせる。これは，メインメニュの＜テクスト定義＞で行なう。

第二に，ＫＲに入力テクストを読み込ませ入力処理を行なう。これは，メインメニュの＜テクスト入力＞で行なう。テクストが大きい場合，この意味での狭義のテクスト入力は何回かに分けて行なうことになる。

このテクスト入力が終了してはじめて，ＫＲにより，検索などの処理が可能となる。実際には，テクスト全体の入力が完了しなくても，入力の終った部分については検索等すべての処理が可能である。

実際のテクスト処理では，上記②，③と検索が繰り返されるのが通常である。まず試験的に入力し，様々なリストや索引を出力して，誤りを修正し，最終的に完全な入力テクストを作成，これを入力するというのが，通常の手順であろう。

2 テクストの構造

入力テクストを作成するためには，利用者は，入力テクストに課される構造上その他の制約を知る必要がある。ＫＲは，この意味での制約の緩やかなプログラムではあるが，それでも幾つかの制約をなしで済ますことはできない。

こような制約は主として，ＫＲが前提とするテクストの構造に由来するものである。それゆえ，ここではまず，ＫＲがどのようなテクスト構造を仮定しているかを述べる。

ＫＲの場合，テクスト（入力テクストではない）は，第一義的には全角文字の連鎖である。文字には，句読点など記号も含む。半角文字に関しては，後述する。

第二のレベルでは，テクストは全角文字の空でない連鎖であるテクスト行（以

下単に行と呼ぶ場合もある）の連鎖である。

第三のレベルにおいて，テクストはひとつ以上のテクスト行から成るテクストユニットの連鎖である。例えば，あるテクストは，複数のテクスト行から成る詩というテクストユニットの集合である。常識的には，巻，節，章，段落，頁といったテクストユニットが考えられよう。勿論，行の連鎖であるという条件を満足する限り，何をテクストユニットとするかは自由である。

テクストユニットは少なくともひとつ定義しなくてはならない。それゆえ，テクストユニットが不要であれば，ダミーのテクストユニットを設定しておけばよい。

他方，ＫＲは，最大６種類までのテクストユニットを認める。複数のテクストユニット間には特定の関係は想定されていない。勿論，章と段落のように，一方が他方の構成要素になるといった関係が存在することは差し支えない。

ＫＲによる様々な処理は，（全角）文字，テクスト行，テクストユニットのいずれかを単位として行なわれる。

3　入力テクスト

3．1　入力テクストの構造

前述のテクスト構造からして，入力テクストでは，テクスト行，テクストユニットを識別する必要がある。入力テクストに課される制約は主としてこの目的のためのものである。

入力テクストは，まずMS−DOSテクストファイル（アスキーファイル）でなくてはならない。正確には，行の終りは１６進の0D0Aであること，ファイルの終りにはEOFコード（１Ａ）があることが必要である。EOFコードがないファイルについては，結果は保証されない。

入力テクスト，入力ファイルの大きさにプログラム上の制限はないと考えて差し支えない，入力テクストの大きさは，全角で約１０億文字までであるが，現実には，ディスク容量など使用機器の容量に制約される。

入力テクストは，複数の入力ファイルに分れていてよい。その場合，後述の入

力処理を何回か順に行なえばよい。ただ，経験的には，１００Ｋバイト程度のファイルに分割しておくほうがＫＲへの入力の効率がよい。

入力ファイルの１行の長さは，最大4096バイト（全角文字で2048文字）までである。これには，行末の空白および後述の識別部をも含む。また，これは，テクストの１行の最大長でもある。

入力ファイルでは，前述のように，テクスト行，テクストユニットの境界を識別する必要がある。それゆえ，入力テクストは，「識別部」と「テクスト本文」のふたつの部分から構成される。そして，入力の１行は，

① 識別部のみから成る行
② テクスト本文のみから成る行
③ テクスト本文と識別部から成る行
④ 空行あるいは全角空白のみから成る行

の４種類に分けられる。

このうち，最後の④は，テクスト入力時には無視されるから，入力テクストを作成あるいは修正する際，テクストを見やすくする手段として用いることもできる。

ＫＲで言うテクスト行とは，入力行中のテクスト本文部分である。
このとき，本文部分の後にある全角および半角の空白は無視される。従って，テクスト行１行の長さは，入力の１行を越えることはない。また，その内容については，半角文字の扱い以外何ら制約はない。

識別部に関しては，後に詳細を述べるが，識別部と本文が同一行内に存在するとき，識別部は常に本文部分の後になければならない。逆の場合，識別部の後に本文部分があっても，それは無視される。このような間違いを避けるために，識別部と本文部分とは常に別の入力行とするのもひとつの方法である。

３．２　識別値

入力テクスト中では，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。識別部は，この目的のための部分である。ひとつの識別部は，

テクストユニット識別記号＋テクストユニット識別値

という組み合わせを少なくともひとつもつ。この組み合わせは，１種類のテクストユニットに対応するものであり，識別部には，異なるテクストユニットに対応するこの識別記号と識別値の組み合せが複数個含まれてもよい。ひとつの識別部に，同一のテクストユニットに対するこの組み合わせが誤って複数個与えられたときには，結果は保証されない。プログラムが誤りを検出する場合もあるが，そうでない場合もありうる。慎重を期するときには，＜ユーティリティ＞の＜テクスト情報＞により，後述の識別値の一覧を出力して点検する。

テクストユニット識別記号は，テクストユニットに固有の全角１文字あるいは半角２文字の記号であり，当該テクストユニットのトークン，即ちテクストユニットの個々の現われの境界，例えば，章と章，あるいは頁と頁の境界を示す。識別記号は一般に省略できない。

テクストユニット識別値は，テクストユニットの個々の現われ（トークン）に与える名称もしくは番号である。識別値は場合により与えなくてもよい。これについては，後述する。識別値が省略された場合，これを暗黙指定の識別値と呼ぶ。

識別記号と識別値の間に，全角もしくは半角の空白がある場合，処理は後述の識別値の型によって異なる。但し，任意の識別記号と，入力行の行末もしくは同一行中の後にある他の識別記号との間に何もないとき，あるいは半角もしくは全角の空白のみが存在するときには，識別値は与えられていないものと見なす。

テクストユニットの識別部は，当該のテクストユニットのトークンより前の入力行になくてはならない。従って，入力テクスト（幾つかのファイルに分割された場合には，最初のファイル）の（空行でない）最初の行は，後掲（３．５）の実例からも明らかなように，識別部のみから成る行である。

識別値には，数値型と文字型の２種類がある。

数値型の場合，識別値は，１から３２７６７までの正整数でなくてはならない。これは，後述の暗黙指定の場合にも妥当する。

数値型の識別値を与えるときには，半角で与えなくてはならない。全角の数字で与えたときには，文字型の識別値と見なす。

数値型の識別値には全角，半角の空白を含んではならない。全角あるいは半角の空白は，数値の終りと見なす。数値型の識別値は，後述のように一般に省略可能である。識別記号と識別値の間には半角の空白があってもよいが，半角文字数と数値の文字数の合計が8バイトを越えてはならない。

識別記号と識別値の間に全角空白がある場合，常に文字型の識別値と見なされ，数値型とは解されない。

文字型の場合，識別値は8バイトを越えてはならない。文字は全角，半角を問わない。文字型の識別値の種類は，テクストユニットごとに32767種類を越えてはならない。文字型の識別値には全角，半角の空白を含んでもよい。但し，識別値の末尾の空白は，全角，半角ともに無視される。

これは，識別記号と（空白以外の）識別値の間に半角，全角の空白があってもよいことを意味する。しかし，この場合，空白も識別値と解するので，半角文字を加えたバイト数が8バイト以下でなくてはならない。

KRの出力で，KWIC索引など，用例その他の出現個所を示すレファレンスとして出力される識別値は常に可変長で出力される。それゆえ，索引部分を固定長で出力したいときには，識別値を与えるときに，本来の識別値の前に，空白を適宜用いて識別値そのものを固定長にしてしまうこともできる。但し，この方法は，数値型の識別値には適用できない。

識別値は，型の如何を問わず，トークンごとに固有である必要はない。また，同一の識別値が連続するトークンに与えられたとしても，識別記号がある限り，別のトークンと見なす。

識別値は，用例の出現個所を示す項目として出力することを第一義的目的とするが，サブテクストもこの識別値をキーとして作成されるので，用例の出現個所として出力しないテクストの構成要素あるいは分割単位を，テクストユニットとして適当な名称や番号を与えておくこともできる。

以上述べた識別値は利用者の与える識別値であるが，KRはこれとは別に，テクストユニットごとに先頭のトークンから順に一連番号を与えてトークンを区別

する。これをシステム識別値と言う。

3．3　識別モード

テクストユニットの識別値の与え方（以下識別モードと言う）には次の3種類がある。識別モードはテクストユニットごとに定められる。

① 行型
② 連番
③ 文字型

行型と連番のふたつの識別モードは，基本的には数値型の識別値をもつ。

しかしながら，いずれの場合も，文字型の値を与えてもよい。このふたつの識別モードで，半角数字以外の文字，記号が識別値として与えられた場合常に文字型の識別値が与えられたものと見なす。

任意のトークンの前に識別値が与えられていないとき，これは，前のトークンの数値型の値を1増やすという指示と解される。従って，文字型の値を与えた次のトークンの識別値が与えられていないとき，このふたつの識別モードでは，エラーとなる。文字型の値を1増やすことはできないからである。

数値型の値を与えるとき上述の制限の範囲内であれば，その値は任意である。数値は，不連続であってもよいし，トークンの出現順と数値の大小には何の制約もない。ただ，整数値が不連続であるときには必ず数値を与える必要がある。

行型と連番というふたつの識別モードの違いは，識別記号の与え方にある。

行型の場合，テクストユニットのトークンの境界は，その名称の示すように，入力行である。従って，特に値を変更するとき以外は，識別記号も識別値も与える必要はない。但し，入力テクストの先頭では，必ず識別記号と識別値，例えば，1を与えて，初期値を設定しておかなくてはならない。

連番の場合，テクストユニットのトークンの境界は，識別記号のみによって認識される。従って，新しいトークンが出現する度に，（空白行を除く，その直前

の行で）識別記号により，そのことを示しておく必要がある。

文字型の識別モードの場合，識別値としては，文字型のみが可能である。即ち，数値が与えられたときでも，これを数値としては認識しない。文字型の識別モードの識別値が省略されたときには，半角空白１文字が与えられたものとして，これを識別値と見なす。

3．4　外字

ＫＲは，独自の外字処理機能をもたない。また，ＫＲは入力テクストに原則として半角文字を認めない。これは具体的には，識別値を除いて，入力テクスト中のすべての文字を全角文字であるものとして，つまり２バイト単位で処理することを意味する。この二点から，外字は半角文字２文字によって代用することができる。検索その他の処理に際しても同様である。

3．5　入力テクストの例

次の例は，「手記略称」，「筆者」，「頁」をテクストユニットとする被爆手記の先頭の一部である。＊，＃，％の３つの記号は，それぞれのテクストユニットの識別記号である。

「手記略称」と「筆者」の識別モードは文字型，「頁」の識別モードは連番である。

例では，入力の１行が印刷の数行にわたることがあるので，入力行の終りに¥記号を付して行の終りを示す。これは勿論元の入力テクストにはない。

＊爆心　＃ 野村英三¥
％1¥
廣島市中島本町，丁度元安橋南詰に現在燃料會館がある。¥
當時廣島縣燃料配給統制組合の本部であった。¥
この建物は地上三階地下一階で鐵骨鐵筋コンクリ―ト建ての丈夫なもので爆心点

から西南約百メートルに位置している。¥

組合は當時毎朝八時に全員を二階に集めて國民儀禮をするのが例であった。¥
その朝も河合業務部長の音頭で濟まし全出勤者三十七名は各階各自の机にかえって仕事前の一服をやっていた。¥
さて「仕事」だと私は机上を見たところいつもの書類がまだ置いてない。¥
いつも課長が地下室から持ってくるのを今朝にかぎって忘れていたのだった。¥
そこで自分の隣の廣瀬女事務員に取りに行って貰うつもりでその方を見たら忙がしそうにしていたので私は二階を下りて地下室へ行った。¥

下りる前に私はめがねを外し財布をズボンのポケットから出しそしてズボンのバンドに巻いてある鎖を解いて懐中時計を出し机上にこの三点を揃えて地下室へ下りて行った。¥
この品は勿論みな焼いてしまったが何故そんなことをしたのかは五年後の今日どうしてもわからない。¥
%

地下室は建物の三分の一位の廣さで十坪餘りの狭まいもので常に電灯が灯してある。¥
(以下略)

4　テクスト定義

テクストを入力する前には，入力テクストに関する情報を予めＫＲに知らせておく必要がある。これは，メインメニュの＜テクスト定義＞で行なう。＜テクスト定義＞では，テクストの名称，テクストユニットの名称，識別記号，識別モード，テクストのパスなどを定義する。

ひとつのテクストに関する定義は１回で済むが，必要に応じ何度行なってもよい。しかし，＜テクスト定義＞で指定したパスに，既に入力したテクストあるいはテクスト定義だけ行なったテクストが存在するときには，新たなテクスト定義により，それ以前に作成されたテクストおよびその定義はすべて消去される。

メインメニュの＜テクスト定義＞を選択すると，以下の順でテクストの定義を

行なう。実行中にエラーが発生したときには，その旨のメッセージを表示して，KR自体の実行を終了する。このときには，当該の＜テクスト入力＞の対象となった入力テクストは無視され，それ以前の状態が保持される。

4．1　テクスト名称

テクストの名称を，最大64文字（全角で32文字）までで与える。入力は，全角，半角を問わない。空白が含まれてもよい。

4．2　テクストユニットの定義

テクストの名称に続けてテクストユニットに関する定義が要求される。画面にテクストユニット番号が表示されるので，以下の項目を定義する。

テクストユニットは最小限ひとつだけは定義しなければならない。テクストユニットが不要な場合には，適当な名称と記号を与え，識別モードを文字型とするダミーのテクストユニットをひとつ定義しておけばよい。このときには，入力テクストの先頭に，ここで定義した識別記号だけから成る行を挿入しておく。

ひとつのテクストユニットの定義が終ると，次のテクストユニットの定義を行なうか否かの問い合せがあるので，適宜応答する。最大6個までのテクストユニットを定義できる。

ひとつのテクストユニットに関して定義すべき項目は次のとおり。

① テクストユニットの名称

最大8バイト（全角で4文字）までの長さの名称を与える。全角半角を問わない。空白があってもよい。但し，既に定義したテクストユニットと同一の名称を与えることはできない。

② 識別記号

入力テクスト中でテクストユニットのトークンの区切を示す記号，即ち識別記号を与える。

全角１文字，または半角２文字（の通常は記号）を与える。この識別記号はテクストユニットごとに異なっていなければならない。

③　識別モード

行型，連番，文字型のいずれかひとつを選択する。識別モードの意味に関しては，３．２および３．３を参照。このときの応答は，半角でなければならない。

４．３　テクストパス

任意のテクスト（を構成する諸ファイル）が存在するディレクトリをテクストパスと呼ぶ。ここでは，この<テクスト定義>で定義し，次に入力処理を行なおうとするテクストをどのディレクトリに置くかを，指定する。

パス名は，画面に表示される入力要求で例示されているように，ドライブ名から始まる完全なパス名でなくてはならない。また，パス名の最後には必ず半角の¥記号がなくてはならない。パス名に誤りがあるとその旨のメッセージが表示され，再試行するかどうかの問い合せがある。否と答えたときには，ＫＲの実行を即時終了する。

ここで指定したパスに，（テクスト定義，入力の失敗による場合も含め）既存のテクストが存在するときには，その旨の警告が表示され，既存のテクストを無視してよいかどうかの確認が行なわれる。無視しないと応答したときには，別のテクストパスを指定するか否かを指示する。指定しないと応答したときには，ＫＲの実行を終了する。指定すると応答したときには，指示に従い，再びパス名を指定する。

無視すると応答したときには，次項に述べるように新たなテクストを作成する。

テクストパスが設定されると，若干の時間の経過後「テクスト定義正常終了」のメッセージが出る。これで，テクスト定義の正常な終了である。その後，メインメニュに戻る。

４．４　テクスト定義に関する注意

KRは，同一のテクストを構成する諸ファイルは同一のディレクトリにあり，かつ，ひとつのディレクトリには，(他のファイルは別として，KRのテクストに関する限り) ひとつのテクストを構成するファイルしか存在しないものと仮定する。従って，<テクスト定義>を実行し，そこで指定したパスに，同一のものであれ，異なるものであれ，既に入力されたテクストが存在するとき，そのテクストは無条件に消去される。

5 テクスト入力

5．1 テクスト入力の概要

メインメニュの<テクスト入力>は，テクストの入力を行なう。テクスト入力の前には，<テクスト定義>によりテクストが定義されていなければならない。

テクスト入力に当たっては，指示に従い，まずテクストパスを指定し，次いで入力ファイル名を与える。その後，直ちに入力処理が開始される。入力テクストの読み込み処理中には，読み込んだテクスト行２００行ごとに，その旨のメッセージとともに行数が表示される。この行数は，既に入力した部分があるときには，そのテクストの通算の行数である。テクストの読み込みが終了すると，索引処理を行なう。これは，検索を高速化するための索引作成作業である。

このふたつの処理が終了するとテクストの入力の正常終了である。その後，メインメニュに戻る。

何らかのエラーのため，入力が正常に終了しなかった場合，当該の<テクスト入力>の処理結果は，すべて無視され，<テクスト入力>が行なわれる前の状態が保持される。それゆえ，この場合には，入力テクストの修正，変更の後，あるいはテクストの再定義の後，再度テクストの入力を行なえばよい。

ひとたび正常に入力したテクストの修正，変更，削除は一切不可能である。必ず<テクスト定義>からやり直す必要がある。

5．2 追加入力

テクストの入力が正常に終了した場合，このテクストには新たな入力テクストを追加することができる。実際，既述のように，入力テクストは幾つに分割されていてもよいし，追加入力を何度行なってもよい。

しかしながら，これに関しては次の点に注意する必要がある。

第一に，入力テクストの同一の部分を，例えば，既に入力したファイル名を入力ファイル名として与えるなど，誤って二度以上入力する危険がある。ＫＲはこれに関しては一切チェックしない。このときには，再度＜テクスト定義＞から始めて，入力をやり直す必要がある。大規模なテクストを分割して順次追加入力するときの唯一の安全な対策は，＜テクスト入力＞が正常終了する都度，別のディレクトリにバックアップを取ることだけである。".KTR"という拡張子をもつファイルをコピーすればよい。付録「参考」を参照。

第二に，異なったテクストに追加入力する危険も皆無ではない。これについても，テクストの名称による確認を求める以外，一切のチェックをしない。対策としては，前項と同様バックアップを取っておくのが唯一の安全なものである。

第三に，データを追加入力するときに，既成のサブテクストがあれば，それはすべて廃棄される。これに関しては，警告のメッセージが出るので，そのときに判断する。

5．3　入力時のエラー

テクスト入力中にエラーが発生したときには，その旨のメッセージと当該の行が表示され，異常終了となるので，必要な修正，変更を行なった後再度入力を行なう。以下，テクスト入力時のエラーについて述べる。エラーメッセージからその意味と対策が明らかなものは一部省略する。また，メモリの不足に関しては次の５．４を参照。

①　半角文字

ＫＲはテクストには原則として半角文字を認めない。しかし，ふたつの半角文字の並びは，全角１文字に相当するものとして処理する。これに対し，半角１文

字だけ出現したときには，エラーとなる。正確には，入力行のテクスト本文部分の先頭を１として，先頭からの奇数バイトが半角文字の場合，その次のバイトが半角文字でなければ，エラーとする。

② ライトエラー

稀な例外を除き，ハードディスクなどの記憶媒体の容量の不足による書込の失敗を意味する。不要なファイルを削除するか，より大きな記憶媒体を用意する。

③ 識別値

「識別値長不正」は識別値が長すぎるときに発生する。実際には，識別記号の誤りが原因の場合もある。識別モードが行型，連番であるとき，入力テクストの最初の識別値は，明示的に文字型で与える場合を除き，省略できない。与えられないと「識別値初期値未設定」のエラーとなる。識別モードが文字型であれば，この問題は起こらない。

数値型の識別値が１より小であるか，３２７６７より大であるときには，ＫＲの数値型の値の範囲を越えるので，「数値型識別値不正」となる。これは，暗黙指定による増加の結果をも含む。

通常は考えられないが，ひとつのテクスト・ユニットにつき，文字型の識別値の種類数が３２７６７を越えると「識別値数超過」となる。

５．４　メモリ不足による中断とその後の処理

テクストの入力は大量の内部メモリを必要とする。それゆえ，メモリの不足のため実行が中断することは十分に想定されることである。それゆえ，ＫＲ自体が残りの使用可能メモリを一定の間隔でチェックし，使用可能メモリが乏しくなったときには，処理を中断し，それまでに入力された部分についてのみ正常な処理を行なう。

しかし，１行の長さが極端に長く，例えば，全角で２０００字，かつ文字の種類がきわめて多いテクストでは，このような防止策を講じても防ぎ切れない場合もある。このときには，真のメモリ不足が発生し，実行は直ちに中断される。

他方，プログラムによるチェックにかかったときには，まずその旨のメッセー

ジを表示し，入力ファイルのうちまだ読み込みの済んでいない部分を，利用者の指定する別のファイルに出力する。このときには，利用者は，読み込みの済んでいない部分を出力する（入力ファイルあるいは他のファイルとは異なる）ファイル名を与えなければならない。ここで指定したファイルに，読み込みの終っていない部分が出力されるので，次はこのファイルを入力ファイルとして，追加入力すればよい。

このように未処理部分を別ファイルに書き出したときには，それまでに読み込みの終った部分については，完全に正常な入力処理が行なわれる。書き出しが終ると索引処理を行ない，他のエラーがない限り，テクスト入力の正常終了となる。

この場合，利用者が実行を強制終了して，入力ファイルを自分で分割して，改めて入力をやり直すことも勿論可能である。

第四章　補助機能

1　ユーティリティ

ＫＲは検索機能のほか，以下の補助的機能をもっている。

サブテクストの作成，併合，削除，一覧
テクスト情報の出力
テクスト出力
文字リストの出力
文字列リストの出力
グループ別文字列頻度リストの出力
ＫＷＩＣ一括出力

これらの機能は，メインメニュのひとつであるサブテクスト処理を除き，メインメニュの＜ユーティリティ＞を選択すれば次のユーティリティメニュとして表示されるので，適当なものを選択して実行する。

Ｉ：テクスト情報
Ｔ：:テクスト出力
Ｌ：文字リスト
Ｓ：文字列リスト
Ｂ：グループ別リスト
Ｋ：ＫＷＩＣ
Ｑ：終了

ユーティリティメニュのいずれかが正常に終了したときには，再びユーティリティメニュに戻る。メインメニュに戻るためには，ユーティリティメニュの＜終了＞を選択すればよい。

＜サブテクスト＞は正常終了後，メインメニュに戻る。

2　サブテクスト

2．1　サブテクストの概要

テクスト全体に対して，テクストの特定の部分をサブテクストと呼ぶ。テクストユニットの利用者識別値，システム識別値（ＫＲが与えた一連番号）のいずれかを用いてサブテクストを定義し，検索などの処理の範囲をテクストの特定の部分に限定することができる。

サブテクストは，ひとつのテクストにつき最大２０個まで作成できる。これを越えるときには，不要になったサブテクストを削除すればよい。

サブテクストは，常にテクストユニットの識別値あるいはテクスト行単位で定義される。従って，サブテクストは，テクストのテクスト行を単位とする部分集合である。

サブテクストを構成するテクスト行はテクスト中で連続している必要はない。即ち，サブテクストはテクスト中の不連続なテクスト行の集合であってもよい。

サブテクストは，言うまでもなく，テクストごとに作成される。また，データを追加入力するときに，既成のサブテクストがあれば，利用者の確認を経た後，すべて消去される。

サブテクストの作成は，検索などの処理を高速化する点では，効果を期待できない。これは，サブテクストに関し，ＫＲがメモリ，ディスク容量の節約に主眼をおいているからである。

＜サブテクスト＞では，以下の処理が可能である。

Ｃ：新規作成	サブテクストの新規作成
Ｍ：併合	サブテクストの併合
Ｄ：削除	既存のサブテクストの削除
Ｓ：一覧表示／出力	既存のサブテクストの一覧出力
Ｑ：終了	サブテクスト処理の終了

２．２　サブテクストの新規作成

サブテクストは最大２０個まで作成できる。これを越えるときには，既存のサブテクストのうち不要なものを削除する。

サブテクストの作成に当たっては，以下の項目をこの順に指定する。

① サブテクストの名称

サブテクストに与える名称を入力する。名称の長さは，最大６４バイトまで，全角，半角文字を問わない。空白を含んでいてよい。

② 選択形式

後述のように，サブテクストは，テクスト行番号，テクストユニットの識別値，システム識別値を手掛りとして作成する。その際，作成するサブテクストが，利用者が選択あるいは指定した値をもつテクスト行から成るか，逆にそれ以外のテクスト行から成るかを指示する。

「選択」の場合は，サブテクストは，利用者が指定した条件に合致するテクス

ト行のみから成る。逆に「除外」の場合，サブテクストは，利用者が指定した条件に合致しないテクスト行だけから成る。

③　選択のキー

テクストユニットの識別値を選択除外のキーとするか，システムがテクスト行に付けた一連番号を選択除外のキーとするかを指示する。後者は，テクストを，先頭から１０００行ごとの単位に分割して語彙や文字の度数を調べるといった場合に使用することになろう。

「テクスト行」を選択した場合，後述の範囲指定の方法により，選択除外する範囲をテクスト行の行番号で定める。範囲指定が終れば，直ちにサブテクスト作成の処理に移る。

「テクストユニット」を選択したときには，どのテクストユニットの識別値を使用するかを，画面に表示されたテクストのうちから番号で選ぶ。この場合，さらに利用者の与えた識別値とシステム識別値のいずれを使用するかを指示する。

④　利用者識別値とシステム識別値

すべてのテクストユニットについて，すべてのトークンは，利用者識別値とシステム識別値のふたつの値をもつ。

利用者識別値は，利用者がテクスト入力時に，暗黙指定も含め与えた文字列あるいは数値である。これに対し，システム識別値は，ＫＲが個々のトークンに出現順に与えたトークンに固有の一連番号であり，ひとつのテクスト・ユニットに関してはトークンごとに異なる。同様に，テクスト行番号もシステム識別値の一種と見ることができる。

サブテクストを作成するキーとしては，このふたつのいずれを使用してもよいが，併用はできない。

システム識別値を使用するときには，後述の範囲指定の方法により，選択除外する範囲を指定する。範囲指定が終れば，サブテクストの作成を開始する。

利用者識別値を選択したときには，処理がやや複雑になる。選択されたテクストユニットが，文字型と数値型両方の識別値を有するときには，その双方あるいは一方だけによる選択除外が可能である。このときには，プロンプトに対し，文字型，数値型それぞれについて選択除外のキーとするか否かを応答する。

テクストユニットが文字型あるいは数値型の識別値しかもたないときには，それぞれの型の識別値が自動的に選択除外のキーとなる。

いずれの場合も数値型の値は後述の範囲指定の方法により，文字型の場合は文字型識別値の選択により，選択除外する範囲ないしは識別値を指定する。

⑤　数値による範囲指定

利用者識別値であれ。システム識別値であれ数値によって範囲を指定するときには，起点と終点により範囲を指定する。起点と終点の組合せは最大２０個まで指定できる。ひとつの範囲指定が終ると次の範囲指定を行なうか否かについて問い合せがあるので，適宜応答する。

範囲指定に際しては，まず起点を指定する。起点は，選択除外の対象となる番号の最小値である。このとき，画面には，

起点（ret=デフォルト，最小値－最大値）

という形で（ ）内に数字が示される。デフォルトは，リターンキーのみ入力した場合に仮定される値であり，最小値，最大値は入力可能な最小値と最大値である。

起点に次いで，終点の指定も同様に行なう。（ ）内に示される数字の意味も同じである。起点と終点の値は同じあってもよい。

選択形式が「選択」の場合，起点以上かつ終点以下の数値を識別値としてもつテクストユニットあるいはテクスト行がサブテクストを構成することになる。この場合，起点と終点は含まれる。逆に選択形式が「除外」の場合，起点より小あるいは終点より大である数値を識別値として有するテクストユニットあるいはテクスト行がサブテクストを構成する。起点と終点は含まれない。

起点と終点による範囲指定では，可能な最大値と最小値が表示されるが，テクストユニットの利用者識別値の場合，この値は，理論的なものであって，当該のテクストユニットの識別値の現実の最小値と最大値ではない。それゆえ，ここで指定する値が現実の最大値あるいは最小値を越えることもありうるが，特に意に介する必要はない。

起点と終点による範囲指定は上述のように最大２０組まで与えることができる

が，「選択」の場合，個々の範囲指定のいずれかを満たす要素がサブテクストを構成する。逆に「除外」の場合は，個々の範囲指定のいずれをも満たさない要素がサブテクストを構成する。ＫＲは，複数の範囲指定の相互的整合性，例えば，範囲の重複などはチェックしない。

⑥　文字型識別値の選択

文字型の識別値による指定を行なうときには，画面に表示された識別値のリストからひとつずつ順に選択する。識別値をひとつ選択すると次の選択を行なうか否かの問い合せがあるから，適宜応答して必要なだけ選択する。

識別値は最大２０個まで選択できる。これ以上の識別値を指定したいときには，２０個ずつに分割してサブテクストを作成しておき，後述の併合処理を行なう。

選択形式が「選択」のときには，選択された識別値のいずれかを識別値とするテクストユニットがサブテクストを構成する。逆に「除外」のときには，選択された識別値のいずれをも識別値としないテクストユニットがサブテクストを構成する。

範囲指定あるいは文字型識別値の選択が終ると，サブテクスト作成の作業を始める。これには，相当の時間を有する。

指定された条件を満たすテクストユニットが存在するときには，新たなサブテクストを作成した旨のメッセージを表示する。逆の場合は，新たなサブテクストは作成されない。

２．３　サブテクストの併合

サブテクストメニュの＜併合＞は，ふたつのサブテクストの論理和あるいは論理積からなる新たなサブテクストを作成する。

併合に際しては，まず，対象とするふたつのサブテクストを，指示に従い，画面でひとつずつ選択する。選択の順序に意味はない。

次に，論理積をとるか論理和をとるかを指示する。論理積の場合は，ふたつのサブテクストのいずれにも含まれるテクストユニットあるいはテクスト行が新たなサブテクストを構成する。論理和の場合は，ふたつのサブテクストの少なくと

もいずれか一方に含まれるテクストユニットあるいはテクスト行が新たなサブテクストを構成する。

指定された条件を満たすテクストユニットが存在するときには，新たなサブテクストを作成した旨のメッセージを表示する。逆の場合は，新たなサブテクストは作成されない。

2．4　サブテクストの削除

＜削除＞は既に作成されたサブテクストを削除する。

画面にサブテクストのリストが表示されるので，削除するサブテクストをひとつ選択する。削除していいか否かの確認に対しては適宜応答する。

次に削除するサブテクストがまだあるかという問い合せがあるので，適宜応答する。あると答えたときには，上記の作業を繰り返す。

2．5　サブテクストの一覧

＜一覧表示／出力＞は，既存のサブテクストの一覧を画面表示，ディスク出力，あるいは印刷する。なお，ほぼ同じ情報は，ユーティリティメニュの＜テクスト情報＞によっても出力できる。

この場合，まず，出力先を選択する。ディスクを選択したときには，標準出力ファイルにも利用者の指定したファイルにも出力できる。画面表示は，サブテクストの名称だけのリストであるが，ディスク出力，印刷では，それぞれのサブテクストについて，名称，総文字数，総（テクスト）行数も出力される。

正常に終了すればサブテクストメニュに戻る。

3　テクスト情報

ユーティリティメニュの＜テクスト情報＞は，テクストに関する詳細な情報を出力する。出力先は，画面，ディスクファイル，プリンタのいずれかである。

実行に当たっては，まず，出力先を選択する。ディスク出力を選択したときには，標準出力ファイルあるいは利用者指定のファイルいずれにも出力できる。

出力される内容は以下のとおりである。画面表示の場合，問い合せに対し，否と答えることにより，以後の情報の表示や識別値に関する情報の表示を止めることができる。

テクスト名称
文字トークン総数
文字タイプ数
テクスト行数
行の最大長
サブテクスト数
テクストユニット数

サブテクスト情報　（これは2．5と同じ内容である）
テクストユニット情報
　各テクストユニットについて次の情報を出力する

　　テクストユニット番号
　　名称
　　識別モード
　　識別記号
　　ユニット（のトークン）総数
　　文字型識別値の種類数
　　文字型識別値があれば，その一覧

正常に終了すれば，ユーティリティメニュに戻る。

4　テクスト出力

ユーティリティの＜テクスト出力＞は，テクストを利用者の指定したディスクファイルに出力する。出力における行単位あるいはテクスト・ユニット単位で索引を付けることもできる。また，サブテクスト単位の出力も可能である。

終了すると，ユーティリティのメニュに戻る。

出力はディスクファイルだけであり，利用者が出力ファイル名を入力する。

以下，次の項目を指定する。

① 索引項目

出力したテクストにレファレンスを付けるか否かを指定する。付けないと応答したときには，テクストだけが出力される。付けると応答したときには，以下このレファレンスの位置などを指定する。

② 索引項目の変更

付けると応答したときには，現在検索パラメータとして設定された索引項目が表示され，変更するかどうかの確認を行なう。変更するときには，第二章７．２に従って新たな索引項目を設定する。

なお，ここでの変更は，１回限りのものであり，検索パラメータとして設定された索引項目は変更されない。

③ 索引項目の位置

まず，索引を本文と同一の行に出力するか別の行に出力するかを指示する。同一の行に出力するときには，さらに行頭に出力するか，行末に出力するかを指示する。

索引項目の出力は，位置の如何に拘らず，常に可変長で行なう。また，テクスト本文，つまり出力される行自体も常に可変長である。それゆえ，行頭に置く場合には，本文部分の開始位置が一定しないし，索引を行末に置く場合は，索引の位置が一定しない。

④ テクストユニット単位の出力

本文の１行とは別の行に索引を出力するときには，当該の行の前に出力するか，後に出力するかを指示する。この場合，さらに，任意のテクストユニットの境界で，例えば，頁が変る度に，１行改行（つまり空白行を１行挿入）するか否かを

指示する。

ここで，テクストユニットごとには改行しないとと指示したときには，テクストの１行ごとに，指定に従って，前か後の行に索引が付けられる。

逆に，テクストユニットごとの改行を指示したときには，改行の単位とするテクストユニットを，画面表示されたテクストユニットのうちから，ひとつ番号で選ぶ。このときには，テクストユニットが変る度に１行空行が挿入され，索引もテクストの１行ごとではなく，テクストユニット単位で前か後の行に付けられる。前に付ける場合は，直後のテクストユニットの最初の行の識別値が索引として与えられる。逆に，後に付ける場合には，当該テクストユニットの最後の行のそれが索引となる。なお，サブテクストを指定して出力するとき，ひとつのテクストユニットが完全には出力されないこともある。このときには，索引として与えられるのは，本来のテクストユニットの最初あるいは最後の行のそれではなく，出力された中での最初あるいは最後の行のそれである。

テクストユニット単位での出力を指示したときには，索引項目として指定したテクストユニットとの整合性の問題が生じうる。例えば，頁単位での出力を指定して，段落や文のように一般には頁と一致しない単位で索引を付けたり，逆に頁番号を付けなかったりすれば，索引の価値は半減しよう。ＫＲはこのチェックは行なわない。

⑤　サブテクスト

サブテクストを指定して，出力の範囲を限定することもできる。このときには，画面に表示されたサブテクストのうちからひとつを選択する。

以上の設定が終ると出力が開始される。終了すれば，ユーティリティのメニュに帰る。

5　文字リスト

ユーティリティの＜文字リスト＞は，テクストに出現する文字のリストを出力する。出現度数を付けて，度数順に出力することもできる。サブテクストを指定して，テクストの特定の部分についても同様の処理ができる。いずれの場合も，

テクスト全体あるいはサブテクストに出現するすべての文字を対象とする。特定の文字あるいは，特定の順に並べた文字に関して同様のリストを得るためには，次節の＜文字列リスト＞を用いる。

ＫＲでは漢字の音訓による配列はできないので，利用者がテクストに出現する文字や文字列を再配列したリストを作成するには，まず，この＜文字リスト＞で出力し，出力結果を利用者が並べ替え，それを次の＜文字列リスト＞で処理するという方法しかない。

出力先は，ディスクファイルだけであり，まず，出力ファイル名を与える。標準出力ファイルは使用できない。出力は，各文字（＋出現度数）を１行として行なう。

次のような順で出力の条件を設定する。

① 文字の出現度数

出現度数を付けるかどうかを指示する。出現度数を付けないときには，出力は１行２バイトの固定長ファイルである。また，文字は内部コード順，正確にはＪＩＳコードの昇順に出力される。この出力を利用者の必要に応じて，編集，並べ替えを行なった結果を次の＜文字列リスト＞の入力とすることができる。

出現度数を付ける場合は，さらに以下のパラメータを設定する。出力は，この場合，各文字につき，１行１２バイトの固定長である。

② 出現度数によるソート

出現度数を付けるときには，出現度数の大小によってソートするか否かを指示する。ソートしないときの文字の出力順は，前項の出現度数を付けない場合と同じである。

③ タイプ数とトークン数の出力

出現度数を付けるときには，さらに出力された文字のタイプ数，トークン総数を出力するかどうかも指示する。出力するときには，文字リストの末尾に出力の対象となった文字のタイプ数と総トークン数が与えられる。

④ 昇順と降順

ソートすると応答したときには，出現度数の大きい順（降順）に並べるか小さい順（昇順）に並べるかを指示する。

⑤　サブテクスト

サブテクストを指定して，出力の範囲を限定することもできる。このときには，画面に表示されたサブテクストのうちからひとつを選択する。

以上の設定が終ると出力が開始される。終了すれば，ユーティリティのメニュに帰る。

度数による並べ替えを指示したときには，実行中にメモリ不足が生ずる場合もありうる。この場合は，「概説」に述べたメモリ拡張手段を講ずるか，度数を付けないリストを出力し，その結果を適宜分割して，次項＜文字列リスト＞に入力する。後者の場合，出力結果を利用者が再度編集する必要が生じうる。

6　文字列リスト

ユーティリティの＜文字列リスト＞では，上述の＜文字リスト＞がテクスト内部の情報のみを利用するのみ対し，利用者が文字あるいは文字列を与える。＜文字列リスト＞では，利用者が与えた文字列リストを読み込み，これにもとづいて度数つきリストを作成する。度数によるソート，サブテクストの指定が可能である。入出力ともディスクファイルのみが可能である。

実行に際して設定すべき項目は以下のとおりである。

①　入力ファイル

度数付きリスト作成する文字列を記録したファイルを指定する。ＫＲはこのファイルに与えられた文字列の出現度数を算出し，必要なら並べ替えて，出現度数とともに出力する。与えられた文字列が出現しないときにも，出現度数ゼロとして出力される。

入力ファイルは，MS－DOS のテクストファイルであり，入力の１行は，ひとつの文字列から成るものとする。単独の文字であってもよいが，ワイルドカード，全角半角の空白など，文字列以外のものが存在してはならない。ひとつの文字列の長さは，２０バイト（全角で１０文字）を越えてはならない。

与える文字列の数に制限はないが，万を越えるような場合，ソートを行なうとメモリが不足する可能性もある。このときには，何らかの方法でメモリを拡張するか，入力ファイルを分割する。後者の場合，出力結果を編集する必要が生じうる。

入力ファイル中の空行は無視される。

② 出力ファイル

出力ファイル名を与える。標準出力ファイルは使用できない。出力は，ひとつの文字列とその出現度数を１行として行なう。出力の１行は，文字列部分２０バイト，度数部分１０バイトの計３０バイトの固定長である。

③ 度数順ソート

与えられた文字列の出現度数順にソートするか否かを指定する。ソートしないときには，利用者が入力ファイルに与えた順に出力される。

④ 昇順と降順

ソートすると応答したときには，出現度数の大きい順（降順）に並べるか小さい順（昇順）に並べるかを指示する。

⑤ トークン数の出力

トークン総数，この場合与えられた文字列の出現度数の総和を出力するかどうかを指示する。出力するときには，文字リストの末尾に総トークン数が与えられる。

⑥ サブテクスト

サブテクストを指定して，出力の範囲を限定することもできる。このときには，画面に表示されたサブテクストのうちからひとつを選択する。

以上の設定が終ると出力が開始される。終了すれば，ユーティリティのメニュに帰る。

7 グループ別頻度リスト

ユーティリティの＜グループ別リスト＞は，利用者の与えた文字列について，指定されたグループごとに出現度数を算出して出力する。グループとしては，サブテクストあるいは利用者の指定したテクストユニットの識別値によるグルーピ

ングが可能である。

入出力ともディスクファイルのみ可能である。入力としては，＜文字リスト＞の出力あるいはそれを編集したものも可能である。

実行に際して設定すべき項目は以下のとおりである。

① 入力ファイル

グループ別頻度リストを作成する文字列を記録したファイルを指定する。

ＫＲはこのファイルに与えられた文字列のすべてに関し，指定されたグループ別の出現度数を算出し出力する。

与えられた文字列が出現しないときには，その旨のメッセージを画面に表示して，利用者の確認を求めた後，処理を続行する。

入力ファイルと文字列に関する制約は，前項＜文字列リスト＞の場合とまったく同様である。与える文字列の数に制限はない。入力ファイル中の空行は無視される。

② 出力ファイル

出力ファイル名を与える。標準出力ファイルは使用できない。出力は，ひとつの文字列とそのグループごと出現度数及びテクスト全体での出現度数を１行として行なう。出力の１行は，文字列部分２０バイト，度数部分各９バイトの固定長である。グループ数が多い場合，１行は数百バイトになる。

③ グループの選択

上述のように，サブテクストあるいはテクストユニットの識別値がそれぞれひとつのグループを構成する。サブテクストが２つ以上ある場合には，サブテクストごとの度数をとるか，テクストユニットの識別値ごとの度数をとるかをまず指示する。サブテクストがないか，ひとつだけのときには，テクストユニットの識別値が自動的にグループとなる。

サブテクストを選択したときには，作成されたすべてのサブテクストごとに，文字列の度数を計算する。

テクストユニットを選択したときには，さらにどのテクスト・ユニットの識別値をグループ分けの規準にするかを指示する。画面に表示されたテクストユニッ

トの一覧のうちから，ひとつを番号で選択する。

このとき，文字型であれ，数値型であれ，異なる識別値を有するテクストユニット（のトークン）はすべて異なるグループを成すものとする。他方，同一の識別値を有するトークンは，連続していなくても，同一のグループに属するものとする。

グループ数は，最大２００までであり，これを越える場合は，処理できない。

以上の設定が終ると出力が開始され，正常に終了したときには，ユーティリティメニュに戻る。

出力の詳細は以下のとおりである。

出力の先頭には，見出しが付けられる。この見出しには，各グループの名称が与えられるが，テクストユニットの場合，これは識別値である。サブテクストの場合は，サブテクストの作成された順に，“GRP n”という名称が与えられる。サブテクストの正式な名称は，上述の＜サブテクスト＞あるいは＜テクスト情報＞により知ることができる。

見出し行の最後には，「総計」の欄があるが，この欄には，当該の文字列のテクスト全体での出現度数が与えられる。サブテクストによるグルーピングの場合，この出現度数は，各グループの出現度数の総和とは必ずしも一致しない。

以下，与えられた文字列について，入力ファイルに与えられた順に，出力される。ひとつの文字列については，文字列，各グループの出現度数，テクスト全体での出現度数が出力される。

グループごとの出現度数に関しては，与えられた文字列が，例外的ではあるが，ふたつのグループの境界を跨がる場合，先頭の文字の属するグループに属するものとして算出する。

出力の最後に，各グループごとの出現度数を示す。これは，与えられた文字列の各グループの出現度数の総和である。この場合の「総計」に関しては，上述の注意を参照。

簡単な出力例を次に示す。例は，幾つかの語彙の広島市平和宣言における市長ごとの出現度数を示すものである。

	浜井信三	渡辺忠雄	山田節男	荒木　武	total
核兵器	10	1	21	68	100
世界	45	11	58	77	191
平和	38	12	24	86	160
悲惨	1	3	2	2	8
死	7	4	2	6	19
column total	101	31	107	239	478

8　KWIC一括出力

ユーティリティの＜KWIC一括出力＞は，利用者が与えたすべての文字列についてKWIC形式の索引を出力する。サブテクストの指定も可能である。

出力対象となる文字列のリストは＜文字列リスト＞の場合と同じくファイルから与える。テクストに出現する文字に関しては，このリストは，＜文字リスト出力＞を利用して作成することができる。

出力はディスクファイルだけである。

KWIC出力に関しては，行の長さ，索引項目，行境界の表示などすべて＜検索＞で設定されパラメータに従う。それゆえ，実行する前に，メインメニュの＜検索＞により，予め出力の形式に関わる諸パラメータを確認あるいは変更しておく必要がある。

実行に際して設定すべき項目は以下のとおりである。

① 入力ファイル

KWICを作成する文字列を記録したファイルを指定する。KRはこのファイルに与えられたすべての文字列についてKWIC索引を出力する。

与えられた文字列が出現しないときには，その旨のメッセージを画面に表示して，利用者の確認を求めた後，処理を続行する。

入力ファイルと文字列に関する制約は，前項＜文字列リスト＞の場合とまったく同様である。

与える文字列の数に制限はない。しかし，文字あるいは文字列の数が多いと出力がメガ単位になる可能性もある。

入力ファイル中の空行は無視される。

② 出力ファイル

出力ファイル名を与える。標準出力ファイルは使用できない。出力は，ひとつの文字列につき，そのすべての用例をKWIC形式で出力する。文字列にはその出現度数，ここでは用例数を付ける。

③ サブテクスト

サブテクストを指定して，出力の範囲を限定することもできる。このときには，画面に表示されたサブテクストのうちからひとつを選択する。

以上の設定が終ると出力が開始される。終了すれば，ユーティリティのメニュに帰る。出現度数の多い文字列が多い場合，処理には相当の時間を必要とする。また，出力が膨大になる可能性もある。特にフロッピィディスクに出力するときには注意が必要である。

参考

1 コマンドによる実行

KRのメニュはすべてMS−DOSのコマンドとして実行できる。メニュに対応するコマンド名は以下のとおりである。

KRMAIN	検索（メインプログラム）
KRDEF	テクスト定義
KRIN	テクスト入力
KINF	ユーティリティのテクスト情報
KTXT	ユーティリティのテクスト出力
KLST	ユーティリティの文字リスト出力

KSTR	ユーティリティの文字列リスト出力
KBRK	ユーティリティのグループ別度数出力
KKWC	ユーティリティのKWIC一括出力

コマンドとして実行したときにも，正常終了後には，当該コマンドを呼出すべき親コマンドに必ず戻る。それゆえ，コマンド名を変更した場合，当該コマンドの実行は可能であるが，実行後メニュに戻ることはできないし，メニュでの実行も不可能になる可能性がある。コマンドラインのオプションはない。また，リダイレクションの結果は保証されない。

2 バックアップについて

KRで使用するテクストは，場合により非常に大きなファイルになりうる。テクストを構成するファイルはバックアップ時には，言うまでもなく，すべて同一フロッピィディスク上にある必要はない。またひとつのファイルも複数のフロッピィディスクに分割して差し支えない。このときには必要に応じMS−DOSのCOPY2コマンドを使用すればよい。

KRのテクストを構成するファイルは，すべて“.KTR”という拡張子をもつ。書込エラーが出て，異常終了となり，誤った追加入力は，結果的に防止できる。

これ以外のファイルを書込禁止にしたときの結果は，保証されない。