On the tabulation of a “Japanese word list” by computer

電子計算機による

ワードリスト作成上の一問題

田中章夫

0 まえがき

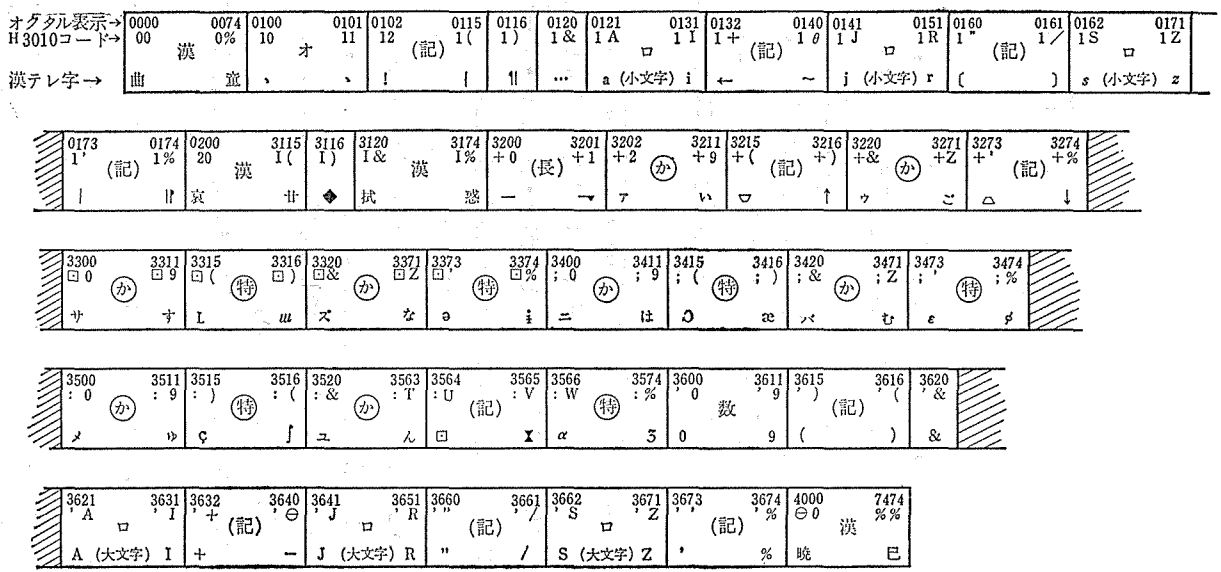
語彙調査や用語索引の作成を，Computerによって行なうとなると，その最終出力としてのWord-listの作成にはいろいろとなやみが多い。もちろん語彙調査における頻度順の語彙表や使用率順の語彙表などは，きわめてComputer向きのリストなので，ここには，ほとんど問題はない。しかし，この種の語彙表についても，ひとたび五十音順の索引などを用意するとなると，どのような五十音順を採用して，どんな順序で単語を並べていくかという問題が，すぐ生じてくる。

国立国語研究所において，現在進行中の語彙調査は，新聞の用語を対象として，入出力には，漢字鍵盤穿孔印刷機（以下「漢テレ」と呼ぶ）を用いている。この漢テレには，2110字の漢字のほか，ひらがな・カタカナ・ローマ字・特殊文字（ギリシア文字・発音記号など）・算用数字・各種記号類が全部で290字，合計2400字が収容され，そのコードは，コード順に示すと，ほぼ表1のようになっている。もし，このコードの順序に，データの中の単語や記号を配列すると，表2に示すように並ぶ。表2のような配列では，あまりに機械的で，この調子で数万の単語が並んでしまっては，とても，人間の頭で目ざす単語を捜し出すわけにはいかない。

そこで，すくなくとも，表3に示す程度の配列をComputerにやらせてみようというのが，われわれの目標である。

なお，表2，表3において，トランプのダイヤの形をした黒い菱形のマークが出てくるが，これは，漢テレに収容してない漢字を打つ場合の記号である。そして，この菱形記号とつぎに出てくる2つの漢字で，漢テレに収容していない漢字を1つ表わすことにしている。「成◆空両大学」の「◆空両」は「成蹊大学」の「蹊」の字を表わし，「◆投行脱」の「◆投行」は「剝脱」の「剝」の字

表1　漢テレコード順字種配列一覧



漢……盤内漢字　　ロ……ローマ字　　オ……かなオドリ字　　か……カクカナ・ひらがな　(長)…長音符

(記)…盤内記号　　◆……盤外漢字マーク　　特……特殊文字　　|¶……盤外記号マーク　　数……算用数字

表 2　　漢テレコード順配列

(* 印はエラーデータ)

結局	ウンーと	マーク
東京駅	ウンカ	まえ
* ゝゞ	ウンと	⊡
「	コーヒー	α 星
¶M○10	コイ	β—ray
¶X2	この	2月分
・・・	ござ	3000l
atom	↓	5mg
⟶	さゝやき	ATOM
～	さくら	AX—2000
memo	ささあめ	AZ錠
〔	ざくろ	＋
xyz氏	すゞめ	・
〕	すいか	MEMO
愛し	すすむ	PTA
成◈空両大学	すず	/
内閣	* タ中ー章夫	Z 革命
* 々々々	トロッコ	,
夫	ドッーと	%
別冊	ドック	内閣
産業	ドッと	準備
毎日	ə	
◈投行脱	æ	
◈供組川	フーン	
* ー ー ー	ファン	
アラさん	ふあん	
あたり	ベアリング	
ある	ペーパー	
↑	ペア	

表 3　　配列情報作成プログラムによる配列

愛し	ファン	β—ray
あたり	ふあん	
アラさん	フーン	2 月
ある	ふーん	3000 l
ウンカ	ペア	5 mg
ウンーと	ベアリング	
ウンと	ペーパー	「
結局	別冊	・・・
コイ	マーク	⟶
コーヒー	毎日	～
ござ	まえ	〔
この		〕
さくら	◆投行脱	↑
ざくろ	（剝脱）	↓
ささあめ	◆供組川	⊡
さゝやき	（芥川）	＋
産業		・
準備	atom	／
すいか	ATOM	,
すず	AX—2000	%
すゞめ	AZ 錠	
成◆空両大学	memo	¶ M○10
（成蹊大学）	MEMO	（¥10）
東京駅	PTA	¶ X 2
ドック	xyz 氏	（X^2）
ドッーと	z 革命	
ドッと		ゝゞ
トロッコ	ə	々々々
内閣	æ	ー ー ー
夫	α 星	タ中ー章夫

を表わす。また，「◈供組川」の「◈供組」は「芥川」の「芥」の字を表わすものである。

もう一つ，音楽の四分音符のわきに一本棒が加わった「¶」のマークは，漢テレに収容していない記号を打つ場合のしるしで，このマークと後続の2字で，漢テレの文字盤にない記号を表わす。「¶M○10」の「¶M○」は，「¥10」の「¥」を表わすものであり，「¶X2」は「X^2」を表わすものである。

1 配列情報の作成

今回われわれが試みた word-list 作成のためのプログラムは，漢テレのコード順に機械的に配列されているデータ(単語)を，一応人間の扱いやすい形に組みかえることを目的としたものである。そして，その配列の順序は，現行の国語辞典の見出し語の配列になるべく近づくことを目標とした。

しかし，考えてみると，国語辞典の見出語の配列システムは，一見きわめて単純な常識的な配列に見えながら，これを Computer によって実行するとなると，思いもよらないほど複雑な処理になってしまい，とても不可能である。その結果，現段階で，一応実用にたえる線でまとめたのが，表3に示した配列システムである。

以下，この配列システムの性格を，国語辞典における見出語の配列と対比しつつ，説明してみることにする。

① カタカナとひらがな

普通，国語辞典の配列システムにおいて，カタカナで書いた「サクラ」と，ひらがなで書いた「さくら」とが出てくれば，これは，ほとんどまちがいなく「サクラ→さくら」の順に並ぶ。ということは，カタカナで書かれていようとひらがなで書かれていようと，同音のものは，一つのものとして，まずまとめた上で，「カタカナ書き→ひらがな書き」の序列をつけている。しかし，漢テレコードに限らず，機械的なコードの場合には，ほとんど，こんな高級なことは，できない。カタカナの「サ」で始まる単語が全部並び終ってから，ひらがなの「さ」で始まる単語が並ぶというシステムが普通である。したがって，これを，国語辞典的な配列に近づけるためには，各データに，カタカナか，ひらがなか，どちらかに統一した配列のための情報をつけておいて，それ

によって配列を進めなくてはならない。われわれが扱う一般の国語資料の場合には，ひらがな書きのデータの方が，カタカナ書きのデータよりも出現頻度が高いので，配列情報は，すべて「ひらがな書き」で記入することにした。したがって，さきの「サクラ→さくら」の例について言えば，カタカナ書きの「サクラ」の方に，ひらがな書きに変換した配列情報を加えることにした。ひらがな書きの「さくら」の方の配列情報は，データをそのまま転写しておくことになる。

こうしておいて，配列情報を第 1key にとり，データそのものを第 2key にとってソートすれば，国語辞典の配列が実現するわけである。

② 清音・濁音・半濁音

普通の国語辞典の配列システムによれば，清音・濁音・半濁音は，「清音→濁音→半濁音」の順に並ぶ。「ハス」「バス」「パス」は一カ所に集まり，「ハス→バス→パス」の順に並ぶ。ということは，このシステムにおいては，同一のかな文字に，濁音符号や半濁音符号のついたものは，まず，これらの符号を無視して，同一のカテゴリーのものとしてまとめ，その上で，「清音→濁音→半濁音」の序列を与えているわけである。

しかし，漢テレコードでは，清音・濁音・半濁音は，一応，別ものになっている。したがって，漢テレコード順で機械的に配列すると，「ひ」なら「ひ」の清音で始まる単語が全部並び終ってから，濁音「び」で始まる単語が並び，それが全部並び終ってから，半濁音「ぴ」で始まる単語が並ぶという順序になる。さきの「ハス」「バス」「パス」の例で言えば，「ハス」と「バス」の間には，「ハタ／ハリ／バカ／バケツ」などの単語が，たくさん，はいってくる。また，「バス」と「パス」の間には，「バタヤ／バナナ／パイプ／パクパク」というような単語が並んでしまう。これを，国語辞典的な配列に並べかえるには，データ(単語)の中に出てくる濁音・半濁音を，すべて清音に変換した配列情報を，各データにつけておいて，それによって配列を実行することになる。したがって，さきの例で言えば，「ハス」には，単にひらがな化しただけの形「はす」を配列情報としてつけておく。「バス」には，清音化した配列情報「はす」をつけ，「パス」にも，同じく「はす」という配列情報をつける。そして，配列情報を第1key とし，データを第2key としてソートすれば，「ハス→バス

→パス」の順，すなわち，国語辞典的な配列が実現することになる。ただし，これに，ひらがな書きの「はす」が加わると，前述の「カタカナ→ひらがな」の序列からして，「ハス→バス→パス→はす」の順になり，カタカナ書きの「ハス」と，ひらがな書きの「はす」とは，離れて並ぶ。

③ おどり字「ゝ」「〳〵」

国語辞典には，おどり字を含む見出語というのは，まず見当らないが，語彙調査のデータには，数は少ないかもしれないが，出現する可能性がある。漢テレコードでは，かなのおどり字「ゝ」と「〳〵」は，かな文字の仲間には，はいっていない。これらは，記号類の仲間にはいり，きわめて若いコードをもっている。したがってコード順に並べると，「すゞ→すゝ→すあし→すし」となってしまう。

「すあし→すし」のあとに「すゝ」や「すゞ」を配列するためには，おどり字を，おどり字の一つの前の文字で埋めた形を配列情報として，これによって配列していけばよい。そのさい，おどり字の前の文字が濁音・半濁音の場合には，それを清音化しておどり字を埋めることになる。たとえば「ばゝあ」の配列情報は「ははあ」，「じ〳〵い」は「ししい」となるわけである。こうしておいて，配列情報を第1key，見出語を第2keyとしてソートすると，「すあし→すし→すゞ→すゝ→すす→すず」の順に並ぶ。「すゞ」と「すず」が，隣合わせにはならないが，プログラムテクニックの方からいって，これ以上，手をかけても，あまり利益にならないので，このへんで，あきらめておいた。

④ 長音符

現行の国語辞典においても，長音符号を含む語のとり扱いは，辞典によって，かなりまちまちになっている。しかし，多くの辞典に共通している扱い方は，長音符号を母音にかえて，すなわち「コーヒー」は「コオヒイ」として並べるというやり方である。これは、人間にとっては，きわめて自然な配列方式であり，ことばをさがすときは，前の語をのばして発音して，そこに現われる母音で検索するのだから，さがす手間もかからない。

しかし，この配列を機械にやらせるとなると，ちょっと面倒なことになる。今回のシステムでは，記憶装置の中に，五十音図のすべてのカナ文字を母音に変換するテーブルを用意しておいて，かながきのデータ(単語)の中に長

音符号がでたら，すぐ，このテーブルをひく方式をとった。たまたま国研の漢テレには，カタカナの「アイウエオ」についても，ひらがなの「あいうえお」についても，小文字の「ァィゥェォ」「ぁぃぅぇぉ」が含まれており，これらの小文字はコードの上では，大文字よりも一つ若いコードになっている。カタカナについていえば「ァアィイゥウ…」，ひらがなについていえば「ぁあぃいぅう…」の順になっているわけである。

そこで，長音符号の母音変換では，この小文字を使用することにした。したがって，「コーヒー」の配列情報は，「こぉひぃ」の形になり，「ペーパー」は「へぇはぁ」になる。そしてこの配列情報を第1key，データを第2keyとしてソートすると，ほぼ国語辞典と同じ配列が実現する。同じ語について，たとえば，「コピー／コピイ」とか「ヘヤー／ヘヤア」というように，長音符号による表記と母音を記した表記との2種類の表記があると，その前後関係は，さきにのべたコードの順からいって，常に，「長音符号表記→母音表記」の順，すなわち「コピー→コピイ」「ヘヤー→ヘヤア」の配列になる。

かなTABLE

アあ	ああ	アあ	ああ	イい	いい	イい	いい	ウう	うう	ウう	うう
ヴう	うう	エえ	ええ	エえ	ええ	オお	おお	オお	おお	カあ	かあ
ガあ	かあ	キい	きい	ギい	きい	クう	くう	グう	くう	ケえ	けえ
ゲえ	けえ	コお	こお	ゴお	こお	サあ	さあ	ザあ	さあ	シい	しい
ジい	しい	スう	すう	ズう	すう	セえ	せえ	ゼえ	せえ	ソお	そお
ゾお	そお	タあ	たあ	ダあ	たあ	チい	ちい	ヂい	ちい	ツう	つう
ヅう	つう	テえ	てえ	デえ	てえ	トお	とお	ドお	とお	ナあ	なあ
ニい	にい	ヌう	ぬう	ネえ	ねえ	ノお	のお	ハあ	はあ	バあ	ばあ
パあ	はあ	ヒい	ひい	ビい	ひい	ピい	ひい	フう	ふう	ブう	ふう
プう	ふう	ヘえ	へえ	ベえ	べえ	ペえ	へえ	ホお	ほお	ボお	ほお
ポお	ぽお	マあ	まあ	ミい	みい	ムう	むう	メえ	めえ	モえ	もお
ヤあ	やあ	ヤあ	やあ	ユう	ゆう	ユう	ゆう	ヨお	よお	ヨお	よお
ラあ	らあ	リい	りい	ルう	るう	レえ	れえ	ロお	ろお	ワあ	わあ
ワあ	わあ	ヰい	ゐい	ヱえ	ゑえ	ヲお	をお	ⓐ		ッ	っ
ン	ん	＊									
がか	ぎき	ぐく	げけ	ごこ	ざさ	じし	ずす	ぜせ	ぞそ	だた	ぢち
づつ	でて	どと	ばは	ぱは	びひ	ぴひ	ぶふ	ぷふ	べへ	ぺへ	ぼほ
ぽほ	うう	＊									

国語辞典では，まず問題にならないが，実際の書きことばデータでの，長音符号の使われ方には，かなりおかしなものがある。たとえば，「ドッーと」「ウンーと」式の表記である。一般的な長音符の使い方からすれば，「ドーッと」「ウーンと」であろうが，マンガなどには，「ドッーと」「ウンーと」式もかなり現われる。そうすると，これらを，さきにのべた処理方式で処理すると，「ドッーと」の方は，まだ「とっぅと」という変換ができるが，「ウンーと」については，お手あげになってしまう。そこで，このように，促音と撥音のあとに出てきた長音符号については，この長音符号を無視した形を配列情報とすることにした。すなわち「ドッーと」については「とっと」を，「ウンーと」については「うんと」を配列情報としたわけである。こうすると，「ドッーと」は表3に示したように，「ドッと」と並び，「ウンーと」は「ウンと」と並ぶことになる。

⑤ 盤内漢字

国語辞典においては，いうまでもなく，単語の中に出てくる漢字は，すべて解読して，その読みにしたがって単語の配列位置を決めている。もし，コンピュータが，このまねをするとなると，データ(単語)の中に含まれている漢字のすべてに，よみがなをつけてから，配列情報の作成にとりかかることになる。現在，われわれの間では，こうした処理方式について研究中であるが，今回報告する配列情報作成のプログラムにおいては，つぎのような方法をとった。

○漢字で始まるデータ（単語）は，その第一字めの漢字の代表的な読み(音訓)，のみによって配列位置を決める。

○そのさい使用する代表的な読みは，それぞれの漢字について，一種類のよみ方だけを採用する。

簡単にいえば，データの2字め以降に出てくる漢字については，一切の処理をあきらめ，単語をさがすときには，第1字めの漢字の代表的な一種類のよみだけでその単語をさがすということである。「人」という漢字ではじまる単語は，「人づくり」も「人間」も「人力車」も，すべて一カ所に集まり，「人」の代表音を，「ジン」なら「ジン」にきめておけば，これらの単語は，「しん」のところに並ぶことになる。したがって，かながきの「ひとづくり」と漢字ではじまる

「人づくり」とは，まったく別のところに並んでしまう。一般の国語辞典に，これに類する配列システムをとっているものは，ちょっと見当らないが，用字辞典あるいは，用字用語字典といった種類の辞書では，これと同じ配列システムをとっているものもある。

ところで，この配列をコンピュータで実行するためには，まず，盤内漢字の各漢字についての代表的なよみをコンピュータに記憶させておく必要がある。

今回の処理においては，盤内漢字と，その代表音は，磁気テープに収め，漢字テーブルテープと名づけた。この漢字テーブルの中には，たとえば「新聞」の「新」という字は，「新しんァァ」の形ではいっている。また，この漢字テーブルの各漢字の代表音(読み)は，それを，そのまま配列情報として使用できるように，すべて清音に変換してある。したがって「人」という字は，「人しんァァ」の形で収められている。そして，漢字テーブルの中の配列は，はいってくるデータが，漢テレコード順にソートしてあるので，テーブルの方も，各レコードのあたまの漢字のコードによって，漢テレコード順に並べてある。

実際の操作においては，漢字ではじまるデータ(単語)が，はいってくると，その漢字と，この漢字テーブルテープの漢字とを，つき合わせる。そして同じ漢字がさがし出されると，テーブルの方の第2字目以下が，そのまま，配列情報として転写されるわけである。

さきの例でいえば，「新聞」についても，「人づくり」についても，配列情報は，ともに「しんァァ」となる。この二つのデータ(単語)の間の前後関係，すなわち，「漢字ではじまる単語で，配列情報が同一のもの」の配列順は，漢テレコード順となる。もし，ここに，かながきの「しん(例．しんのあるメシ)」とか「ジン(gin)」などの単語が存在すると，これらと「新」「人」との前後関係は，「ジン→しん→新→人」となる。これらのデータの配列情報のあたま2字は，すべて「しん」だが，3字め以降が，かな書きのデータではスペース，漢字の方は「ァァ」となっている。コード的に，「㊿＜ァ」なので，上記のような場合には，データ(単語)は，すべて，「かながき単語→漢字」の順に並ぶ。また，さらに，「ァ」は，すべてのかなのうちで，もっとも若いコードをもっている。したがって，助詞の「や(山や川)」と感動詞の「ヤー」と「やあ」，それ

に，漢字の「矢」が，どんな順に並ぶかというと，「や」の配列情報は「やSPSP…」，「やー」の配列情報は「やぁSPSP…」，「やあ」の配列情報は「やあSPSP…」，そして，「矢」の配列情報は「やァァァSPSP…」となる。配列情報の大小関係は，「や＜矢＜やー＜やあ」になるので，いうまでもなく，「や→矢→やー→やあ」の順に並ぶ。このような例については，国語辞典の配列と一致する。ということは，漢字処理についても，配列という点だけにしぼれば，国語辞典の配列システムの基本的なところは，実現したことになる。したがって，今回のプログラムで，はぶいたプロセス，すなわち「データ(単語)に含まれている，すべての漢字に適切な読みがなを与えるプロセス」が完成すれば，カタカナ・ひらがな・盤内漢字の範囲のデータについては，国語辞典的な配列が，一応できるようになる。

漢字 TABLE

曲きょくァ	区くァァァ
計けいァァ	決けつァァ
巾きんァァ	脈かけるァ
形けいァァ	結けつァァ
勤きんァァ	隅くうァァ
型けいァァ	月けつァァ
錦きんァァ	屈くつァァ
経けいァァ	建けんァァ
緊きんァァ	訓くんァァ
芸けいァァ	見けんァァ
堂とうァァ	群くんァァ
頭とうァァ	⋮

なお，今回のプログラムにおいて，さきにのべた漢字テーブルに収めた，各漢字の代表的な読みは，国立国語研究所報告22「雑誌九十種の用字用語」の第2分冊「漢字表」の索引を参考にして定めた。

⑥ 盤外漢字

盤外漢字で始まるデータ(単語)は，いままで述べてきた「カタカナ・ひらがな・盤内漢字などで始まるデータ」が全部並び終ったあとに配列することにした。これら「カタカナ・ひらがな・盤内漢字ではじまるデータ」がもちうる配列情報のうちで，コード的に，もっとも大きなもの，すなわち，いちばん最後に並ぶものは，「ん…」の形である。しかし，「ん」ではじまる単語というものは，そうはない。ましてや配列情報の頭の2字が「んん」となりうる単語となると，ほとんど出現の可能性が考えられない。

したがって，盤外漢字を，「カタカナ・ひらがな・盤内漢字の類」のあとに並べたければ，その配列情報のあたまに「んんんん」と，いくつか「ん」をつけておけばよいということになる。今回のプログラムは，安全性を見こんで，「んんん」と「ん」を3つつけることにした。そして，そのあとに，カタカナの

「ア」を置いて配列情報を構成することにした。この「ア」は，以下に述べるローマ字データなどとの位置関係を決めるためのものである。

したがって，表3にある「◈投行脱」すなわち「剝脱」の配列情報も，「◈供組川」すなわち「芥川」の配列情報も，ともに「んんんア」となる。これでソートすると，盤外漢字で始まるデータは「かな・盤内漢字で始まるデータ」の直後に並ぶが，盤外漢字のデータ同士の前後関係は，漢テレコード順になる。したがって，盤外漢字データの中から，「芥川」なら「芥川」，「剝脱」なら「剝脱」という単語をさがし出すときには，別に作成してある「盤外漢字のコード表」を参照しなくてはならない。

⑦ ロ ー マ 字

国語辞典においては，たとえば「COMPUTER」と書いてある単語を収めようという場合には，これを「コンピューター」と読んでしまい，この読みによって配列位置が決まってくる。ということは，「COMPUTER」に「コンピュウタア」という一種のふりがなをつけて，そのふりがなによって配列しているわけである。ところが，このふりがなも決して，一つの文字について，一種類に決まるわけではない。「A」なら「A」という文字が，「A クラス」というときには，「エー」と読まれ，「ATOM」で出てくると，「アトム」すなわち「ア」と読まれる。そのうえ，さらに，国語辞典では，これらの単語が，大文字で書かれていようと，小文字で書かれていようと，まったく問題にならない。

これだけの複雑な手順を機械に任せるのは，きわめてむずかしい。そこで，今回のプログラムでは，大文字と小文字の統合，すなわち，大文字で「ATOM」と書いてあっても，小文字で「atom」と書いてあっても，これがとなり合わせに並ぶという点だけの処理にとどめた。そして，ローマ字で始まるデータ(単語)は，すべて一まとめにして，さきの「盤外漢字データ」のつぎに，アルファベット順に並ぶようにした。

大文字・小文字の統合だけは，しておかないと，漢テレコードのまま，ローマ字書きのデータをソートすると，小文字で書かれたものが，「a～z」まで全部並んでから，大文字の「A」がはじまるというアルファベット順になってしまう。これを避けるためには，配列情報を，小文字なら小文字，大

文字なら大文字に統一してつけてやればいいわけである。われわれが扱う国語資料では大文字の出現頻度の方が高いとは思ったが，コード上の対応からいって，大文字と小文字の間の変換は，きわめて手軽で，処理時間への影響も，ほとんどない程度なので，配列情報には，小文字をえらんだ。そして，盤外漢字データのうしろに並べるために，配列情報のあたまには，「んんんイ」をつけることにした。したがって，「atom」の配列情報は「んんんイ atom」，「ATOM」の配列情報も「んんんイ atom」となる。これを第1key，データを第2key として，ソートすると，「atom→ATOM」の順に並ぶことになる。

⑧ 特 殊 文 字

特殊文字というのは，はじめにのべたように，「α，β」などのギリシャ文字や発音記号などである。これらが，もし出て来た場合には，ローマ字データのあとに，漢テレコード順に配列することにしてある。特殊文字相互の間の配列順は，漢テレコードのままだから，配列情報は，単に「んんんウ」のみとなる。

⑨ 算 用 数 字

算用数字は，特殊文字ではじまるデータ(単語)のあとに，並べる。そしてその配列順は，データのいちばん最初の数字について上昇順，すなわち「0・1・2・3…9」の順に配列する。「漢テレコードそのものが，算用数字については，すでに上昇順なので，配列情報は，特殊文字データのあとに並べるというだけの情報を与える「んんんエ」となる。

⑩ 盤 内 記 号

「んんんオ」の配列情報を与えて，漢テレコード順に，算用数字データのあとに並べる。

⑪ 盤 外 記 号

「んんんカ」の配列情報を与えて，漢テレコード順に，盤内記号データのあとに並べる。

⑫ エラーデータ(表2の＊印)

エラーデータには，H—3010 コードで，最も大きい「＝＝＝～＝」の配列情報を与えてある。したがって，もし，アウトプットを行なえば，表3に出て

いるように，いちばん最後に並べられる。現在のところ，エラーデータとしては，表2・表3からわかるように，おどり字で始まるデータ，長音符号で始まるデータ，それと，かな以外の文字のあとに長音符号が来ているデータ(表3のいちばん最後に挙げてあるデータで，これは漢字「中」のあとに長音符号が出てきている)の3種類を，はじき出すようにしてある。こうした形式は，日本語の単語には，ありえない形式であり，パンチミスか，語彙調査なら単位切りのミスに違いないということになる。しかし，この程度のエラーチェックでは，語彙調査の実行プログラムとしては，あまり実際的な効果がないので，今後，単位切りのさいの，各種のルール違反なども，はじき出せるように改良していくつもりである。

これまで国語研究所の漢テレのコードにしたがって入力されたデータについて，その配列法をのべてきたが，配列システムの基本的な考え方は，もちろん特定なコードだけに通用するというものではない。漢字・かな・ローマ字・数字などによる実際の国語表記のデータを，そのまま機械で処理する場合には，必ず解決を迫られる問題である。

2. むすび

表4は，以上述べてきた，今回のプログラムによる，配列情報の記入方法を，表にまとめたものである。そして，この配列情報によって，ソートした結果，データがどう配列されるかを示したのが表5である。これらについては，すでに説明したので，表6のゼネラルフローについて，述べることにする。

いちばん問題なのは，Dのステップである。ここでは，データの第1字めしか，字の種類の判別をしていない。この第1字めの字種によって，データは，すべてふり分けられてしまうことになる。盤外漢字・ローマ字・特殊文字・算用数字・あるいは記号類を頭にもつデータのように，集まる位置だけを指定し，その中での前後関係については，漢テレコードに依存してしまう場合は，そう困らない。また盤内漢字で始まるデータも，2字め以降については，はじめから無視しているという，この約束さえ知っていれば，検索のさい，そう不便はない。

表 4　　このプログラムによる配列情報の記入

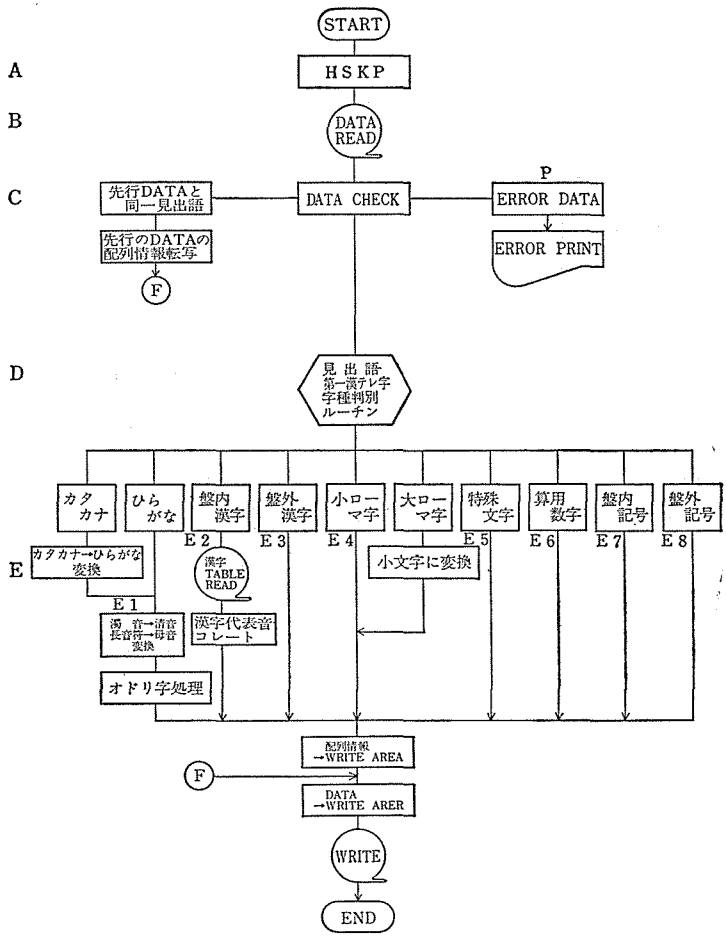
<DATA の第一字の字種>	<処　理>	<配列情報> 漢テレ10字(20桁)
ひらがな カタカナ	ひらがな清音に統合 (おどり字・長音符処理)	ひらがな清音(Sp)～(Sp)
盤内漢字	→代表音	ひらがな清音 ア ～ ア
盤外漢字		ん ん ん ア(Sp)～(Sp)
小ローマ字 大ローマ字	小ローマ字に統合	ん ん ん イ　小ローマ字で DATA 6字
特殊文字		ん ん ん ウ(Sp)～(Sp)
算用数字		ん ん ん エ(Sp)～(Sp)
盤内記号		ん ん ん オ(Sp)～(Sp)
盤外記号		ん ん ん カ(Sp)～(Sp)
エラーデータ		＝ ＝ ＝ ～ ＝ ＝ ＝

表 5　　見出語配列順

(配列情報を key としてソートとすると①～⑧の順に並ぶ)

配列順位・DATA の第一字めの字種	<配列順>
① ひらがな カタカナ	ひらがな清音に変換 → 50音順
盤内漢字	→(漢字代表音) → 50音順
② 盤外漢字	→ 漢字テレコード順
③ 小ローマ字	→ アルファベット順
大ローマ字	→小ローマ字に変換 → アルファベット順
④ 特殊文字	→ 漢テレコード順
⑤ 算用数字	→ 漢テレコード順(上昇順)
⑥ 盤内記号	→ 漢テレコード順
⑦ 盤外記号	→ 漢テレコード順
⑧ エラーデータ	→ 漢テレコード順

表 6


START
HSKP
DATA READ
DATA CHECK
先行DATAと同一見出語
先行のDATAの配列情報転写
F
P
ERROR DATA
ERROR PRINT
見出語第一漢テレ字字種判別ルーチン
カタカナ
ひらがな
盤内漢字
盤外漢字
小ローマ字
大ローマ字
特殊文字
算用数字
盤内記号
盤外記号
E 2
E 3
E 4
E 5
E 6
E 7
E 8
カタカナ→ひらがな変換
E 1
漢字TABLE READ
小文字に変換
濁音→清音 長音符→母音 変換
漢字代表音コレート
オドリ字処理
配列情報→WRITE AREA
F
DATA→WRITE ARER
WRITE
END
A
B
C
D
E

不都合が，もっとも予想されるのは，かな文字ではじまるデータである。たとえば「お母さん」とか「パン屋」の配列情報は「(お母さん)」「(はん屋)」のように，漢字部分には，その漢字の漢テレコードが，そのままはいってしまう。これでソートすると，「お母さん」が「お」の部に，「パン屋」が，「は」の部に収められることは，まちがいない。しかし，配列の位置は，「お母さん」では，漢字「母」の，また「パン屋」では，漢字「屋」の漢テレコードが，大きな決定権をにぎってしまう。したがって，「お母さん」が，「お」の部のどの辺に並んでいるか，「パン屋」が「ハンモック」よりも前に出るか，あとに出るかは，漢テレコードを調べなくては，わからないことになる。この点が今回のプログラムの最大の欠陥であり，われわれも，ずいぶん不安をいだいた。しかし結果的には，こうした表記形式をとりうるものが，現代の新聞などでは，やはり数が少なかったということと，今回のテストランでは，こんなことよりも，もっと基本的な点での処理，たとえば同表記異語（例，工夫…くふう／こうふ）の判別や活用形の終止形変換などが進んでいなかったことのために，あまり目立たなかったようである。

この欠点も，結局は，何回ものべたように，今回のプログラムでは漢字の解読をしていないという点から生じたものである。語彙表の作成に限らず，漢字かなまじりのデータを扱う場合には，その第一条件として，漢字というものは，一応，自動的に解読すなわち「よみがなつけ」が，できるようになっていなくてはならない。

最後に，今回のプログラムの処理時間にふれておく。表6のフローチャートのCのステップの左側からわかるように，先行データと同一のデータは，同一の配列情報が書きこまれるため，まったく処理を行なわずに，先行データの配列情報を転写するだけのシステムになっている。ということは，頻度の高い単語，たえば「する」なら「する」が1000回出てきても，これは，1回分の処理と時間的には，そうかわらないことになる。したがって処理するデータの量がふえればふえる程，時間がかかるという比例関係ではない。大ざっぱに言えば，入力データの量と，処理時間との関係は，延べ語数と異なり語数との関係に近いものになる。すなわち入力データが，ある一定の量に達するとそれから先は，異なり語の現われ方が落ちるので，処理時間もあまり延びな

くなってくる理くつである。現在のところ，そう大きなデータは流していないが，20516語について約4時間，47638語について，5時間であった。これは，もちろん，現在われわれが扱っている長い単位(複合語を1語とする)についての結果である。

まだ，実際に処理してはいないが，短かい単位や国語辞典の見出語程度の長さの単語の場合には，処理時間は大幅に短かくなるはずである。その第1の理由は，単位が短かいほど，同一語形のデータすなわち同じ単語が，多くなり，異なり語の数が減少するからである。第2の理由は，表6のE1のあたりの処理すなわちカタカナのひらがなへの変換，濁音・半濁音・長音符号・おどり字の処理は，データ(単語)のあたまから一字ずつ扱っていくため，データ(単語)の長さが，処理時間と密接に関係しているからである。

注) ソート： データを，指定した一定の順序に電子計算機で並べさせる処理。

(参照) 田中章夫「電子計算機による漢字の自動解読とその問題点」(計量国語学・37)