Title	英文ワード・プロセッサーの作成 : 研究活動におけるラボラトリー・オートメーションの試み
Author(s)	酒井, 源樹
Citation	北海道教育大学紀要. 第一部. C, 教育科学編, 34(2): 165-174
Issue Date	1984-03
URL	http://s-ir.sap.hokkyodai.ac.jp/dspace/handle/123456789/4948
Rights	

英文ワード・プロセッサーの作成

——研究活動におけるラボラトリー・オートメーションの試み——

酒　井　源　樹

はじめに

研究成果を外国語で書かれた論文として発表する分野では原稿をタイプライトすることが研究活動の実務面で大きな比重を占める．タイピストを近くにかかえているという比較的によい条件にある場合は別にして，多くは自分自身でタイプを打つことになるのだが，手書き原稿から完成原稿に仕上げるまでに普通は数次にわたる打ち直しをする．そこから生ずる思いがけない様々なミスに泣かされ，その修正に疲れ果てるという経験をもつ方も少なくないであろう．筆者の分野では，単著論文だけではなく，共同研究として成果を英文で発表することが多く，従って共著論文を作成する過程で著者間で幾度もの書き直しを繰り返しながら投稿原稿が作られていく．その間にもまたタイピングの作業が介在する．このように見ると，一つの論文をまとめあげるうえでタイプすることに投入されるエネルギーは，実は，膨大なものである．研究成果の公表という喜びに支えられて，研究活動をする者にとっての基本的な責務である論文作りのこのような努力がなされているといえよう．とはいえ，これに注がれるエネルギーが軽減できないものか，との感をいだくのも事実である．

筆者は1981年から研究室にパーソナル・コンピューター（以下，パソコン）を導入した．大型計算機の利用形態とはちがって，パソコンのパソコンたるゆえんであるパーソナル・ユースを追求する試みの一環として，本稿ではパソコンの英文ワード・プロセッサーとしての利用について述べる．これにより，上述したタイプライティングにまつわる負担を飛躍的に軽減することが可能となった．研究活動におけるラボラトリー・オートメーションの一側面とでもいうべきであろうか．英文ワード・プロセッサーについていうならば，それ自体は目新らしいものではなく，多少高価ではあっても商品化されているし，あるいは，それぞれの研究分野での用途に応じて研究者個々において開発されもしている．したがって本稿の目的も筆者が今回公開する英文ワード・プロセッサーのプログラムのオリジナリティの主張自体にあるのではなく，プログラムの公開により，英文のタイピングで煩わしい思いをしている多くの方に手許のパソコンを用いて手軽に利用していただくことを目的としている．

次節ではパソコンをワード・プロセッサーとして働かせるBASIC言語プログラムを呈示し，その発想の基本点ならびにそれに基くプログラムの構造を述べる．二節では実際に利用するうえでの英文の入力形式および出力結果について具体的に解説する．

一節　英文ワード・プロセッサー・プログラム

パソコンの機種と言語

使用したパソコン・システムは富士通 MICRO-8（略称ＦＭ８）本体とシリアル・ドットプリンター（EPSON　ＭＰ-80）である．印字品質を高めるにはデージ・ホイール型のプリンターがあればなおよい．使用言語はＦ－BASIC であるが，他の BASIC 言語，例えば，Ｎ－BASIC を使用したい場合には対応して容易に書きかえられる．なお，入力した英文を格納するための外部記憶装置は勿論何でもよく，フロッピー・ディスク，オーディオ・カセット・テープなど利用の容易なものを選べばよい．ここではバブル・メモリーを使った．

プログラムの基本発想

資料１に筆者の開発した英文ワード・プロセッサーのＦ－BASIC 言語によるプログラムを示す．

ワード・プロセッサーにたいする基本的な要請は，入力された文章の修正（例えば，字句の挿入・削除，文節の移動等）が容易であることである．日本語とちがう英語の事情として行末が単語単位で区切られねばからないことも挙げられる．自然科学分野では数式や記号を扱うから，例えば，p^2，^4He，v_{ij} などのように上つき，下つきの添字を容易に処理できることも要求されよう．

以上の要請にどのように応えるのか．

BASIC 言語で動くパソコンの一大特徴はプログラムが行番号で管理され，しかも CRT ディスプレイ上で画面編集されるところにあるといってよい．カーソルを画面上で移動させ必要な箇所で文字を挿入あるいは削除すればそれで修正される．また行番号をつけ変えればステートメントの順序を逆転させることもできる．BASIC の持っているこの機能そのものをワード・プロセッサーの持つべき編集機能として利用する．すなわち，入力データであるはずの英文自体を BASIC 言語の DATA ステートメントで与え，行番号がついているプログラムの一部として扱えばよいということである．（資料２の行番号 1000 以降を参照のこと．）

次に行末の区切り方であるが，単語の途中のどこにハイフンを置くかを計算機自身に判断させるのは難かしい。通常考えられることは，区切り位置を人間が指定する方式であろう．そうだとしても，日本語のように行末をきちんと揃えることは一般には面倒である（どこかで空白を調節する必要がある）．さらに不都合なのは，かりに，一行の文字数を変更する（例えば一行 65 文字で処理していたのを 60 文字にする）と，ハイフンの置かれる位置そのものが変わることになろう．以上の事情を勘案すると，行末の区切り位置を外から人間が指示する方式はかえって煩雑となる．ここでは，残りのスペースが不足する場合は，単語全体を次行の先頭に移すものと割り切ることにする．

第三の添字に関する要請は次の考えで処理する．全く上つき，下つきの添字が現われない行であっても，計算機の処理としては三行とみなすのである．すなわち，第一行目は全文字が空白で構成され，第二行目が本来の文章から，第三行目はやはり空白からなると考える．たまたま p^2 のような場合には第一行目の適当な場所に２が位置し，v_{ij} のときは第三行目のどこかに ij があるという具合になる．最近普及のめざましい電子タイプライターには，上つき，下つきの制御コードを備えたものもあるが，ドット・プリンター等，紙送りを戻す機能のないものが多い現状では，この処理が好ましく思われる．

なお，行間隔および文字ピッチはタイプライターの用語でのダブル・スペースおよび 10 ピッチに統一する．

プログラムの構造

資料1の内容のフローチャートを示せば図1のようになる．図中の数字はプログラムでの行番号を示す．

行番号20－40はプリンターの一頁長を11インチに固定したり，パソコンのファンクション・キーを適宜設定する．最初のRUNコマンドでファンクション・キー3も設定されるので，2度目以降のRUNはキー3を使って実行する．

行番号70－140は，70番でREADされた文章A$が段落で開始されるか，文字をゴチックにするかなどを調べる．

行番号150：配列C$(I)はプリンターに出力される第I番目の行の内容を貯める．なお配列U$(I)およびL$(I)はI番目の行についての上つき添字および下つき添字の行にそれぞれ対応する．第I行について，あと何文字分の余白があるか（設定はM＝68文字）をチエックする．

行番号160：A$が余白の中に収まる場合の処理．

行番号170－230：行末が単語の途中で切れる場合に単語の先頭位置をとり出す．あらかじめハイフンが指定されているときはハイフンで区切ることも可能．

行番号240－260：この行の英文に上つき，下つき添字が混っているか否かを文番号390－490のサブルーチンで調べ，その結果に応じて一行の文字数を調節する．

行番号270：68文字分が埋められたので新しい行に移って処理を続行する．このとき行数が指定のN（＝25）を超えたら，一頁分を打ち出す．（サブルーチン行番号500－590）

行番号290－320：文章A$をC$(I)に収めてもまだC$(I)に余白ができたときに，次の文章A$をREADし処理を続行する．

行番号330：新たにRAEDした文章が頁がえを要求しているか否かのチエック．

行番号360：文章A$の区切り位置として，コンマ(,)，ピリオド(.)，ハイフン(-)を検出する．

行番号370：文章A$の，空白ではない，単語の先頭位置を検出する．

行番号380：文章A$の先頭に空白があるとき，空白を削除する．

行番号400－440：記号#で両側からはさみつけられた文字列を上つき添字とみて配列U$(I)に収める．それに応じてC$(I)の並びをつめる．

行番号450－490：記号@で両側からはさみつけられた文字列を下つき添字とみて配列L$(I)に収める．それに応じてC$(I)およびU$(I)の並びをつめる．

行番号500－520：ディスプレイ上への編集結果の表示．

行番号530－570：PTの値に応じて，編集結果を一頁分まとめて打ち出す．

行番号600－830：編集する際の条件（一頁あたり何行，一行あたり何文字，どの文章から編集を開始するか．プリンターへの打ち出しは必要か，必要なら何頁目が必要か等々）の設定．

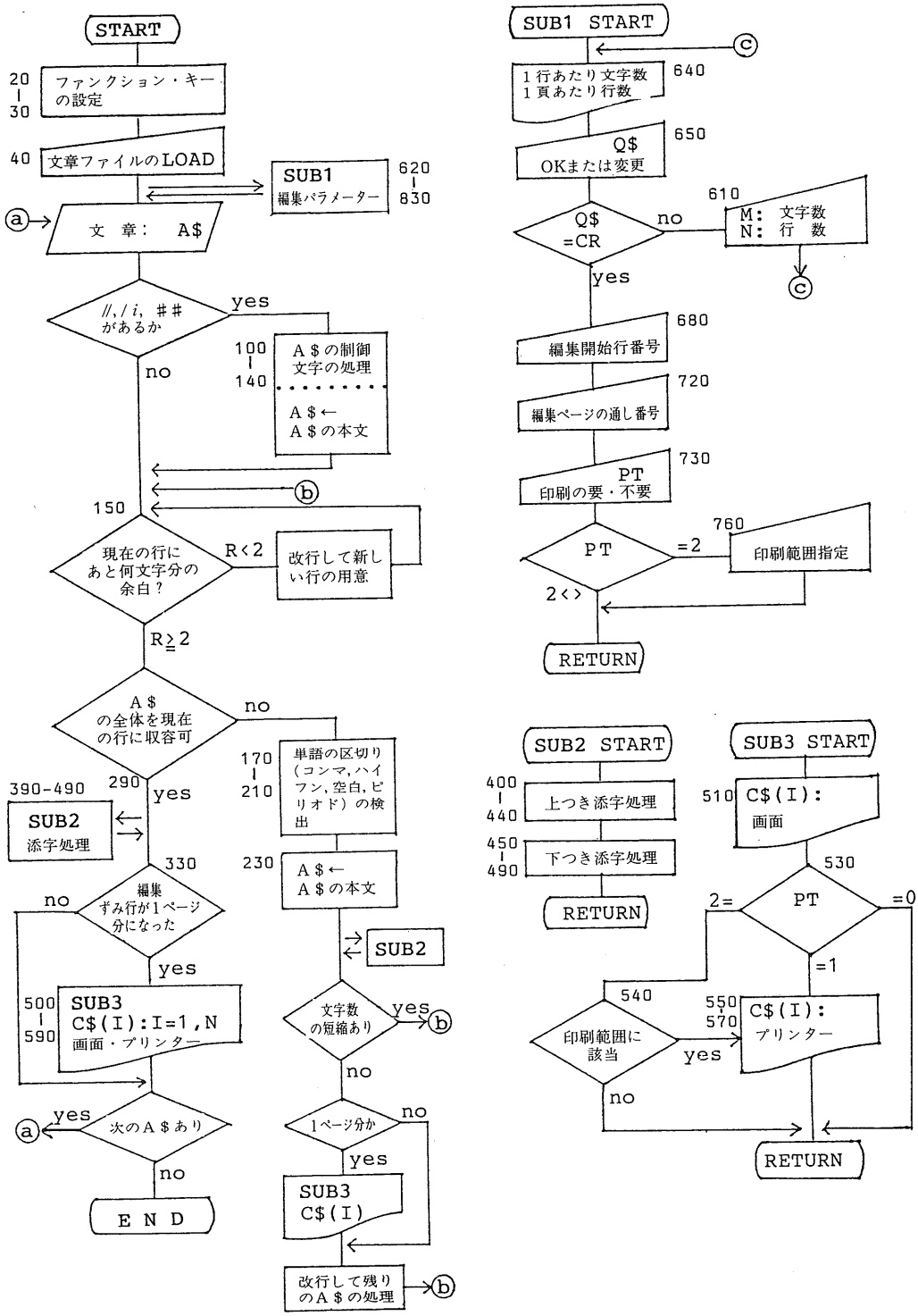
二節　英文の入力形式と編集結果の出力例

入力形式と出力形式の対応

前節で説明したプログラム自体にはまだ何も英文がDATA文で与えられていないがとりあえず，RUNさせてみよう．ディスプレイ上に

on BUB 1：

が表示されたところで強制的にSTOPキーを押して実行を中断する．次にキーボードから



（図）ワード・プロセッサー，プログラムのフロー・チャート

AUTO 1000 とキーインすることにより，ディスプレイ上に

1000

が表示され，ステートメントの入力待ち状態となる．自分の入力したい文章を DATA 文の形式で〝英文〟のように引用符ではさみつけて次々入力すればよい．その一例を資料 2 に示す．行番号は 1000 からはじめて 10 きざみで増加させておくと便利である．なおパソコンのキーボード上のファンクション・キー 5 を利用すれば毎回のキー操作が簡略化される．

さて資料 2 では例えば行番号 1000 の DATA 文が//## の記号で，また行番号 1010 の DATA 文が/ 8 の記号で，さらに行番号 1020, 1050, 1080 の DATA 文はそれぞれ//の記号で書き始められているのに対して，行番号 1030, 1040, 1060, 1070 は何もそれらしき記号なしに DATA 文が書き始められている．一方資料 3 は資料 2 のデータのもとで RUN 50 のコマンドで実行させた結果を示したものである．資料 3 には資料 2 で見られた記号//, ##, /8, /&等が姿を消している．そのかわり，行番号 1000 の文章はプリンターに打ち出されるときには行の先頭から 5 文字の空白がとられて第 6 桁目からゴチックとなって現われる．行番号 1010 は文字は太さは通常のものだが，しかし先頭から 8 文字の空白がとられたのち印字がはじまっている．また，行番号 1020, 1050, 1080 の文章は段落がおかれて 5 文字の空白が行の先頭にある．ところが行番号 1030, 1040, 1060, 1070, 1090 の文章では行かえもなされず，それぞれ，直前の行番号の文章にひき続いて打ち出されている．すなわち, DATA ステートメントの文字列の先頭に//あるいは/ 8 等がある場合には，それらの記号は印刷あるいは様式を整えるための制御コードとして扱われ，それら自身は印字されない．まとめるならば，DATA 文の先頭の 2 文字が

/ / のときは段落をおく（自動的に 5 文字の空白をとる).

/ i のときは改行ののち i 個の空白をおいてから文章がはじまる．（$i = 0 \sim 9$）

/& のときは用紙のページ替えをする．（ページ替え後の文章は次の DATA 文で与える.）

の機能をもち，それ以外の文字列があるときには，直前の行番号の文章にそのまま続く．また，DATA 文が//または/ i で始まっている場合に限り，ひきつづく 2 文字が

のときには，その行番号の文字列をゴチックで印字する.

という機能をもたせている．なお，最終の行番号の DATA 文において

/&, /&

を指定すれば編集データの末尾を意味する．（資料 2 行番号 1100 参照）

上つき，下つき添字の制御は，添字が現らわれる箇所で随時行われる．例えば，文中に p^2 と表現したければ DATA ステートメントの該当箇所に，p # 2 # の形式で，また，v_{ij} としたければ，v@ij@の形式で指定するものとする．すなわち，上つき添字は 2 つの # 記号により，下つき添字は 2 つの@記号により，それぞれ左右からはさみつけられる．資料 2 行番号 1090 と資料 3 の打ち出し結果を比較すれば一目瞭然であろう．

以上の制約，つまり，DATA 文の開始に関するルールと文中における添字に関する制限を除くとあとは一切の制約なく，意図する文章をキー・インすればよい．極端な用例では一つの単語を 2 つの行番号にまたがって入力しても構わないのである．（資料 2 行番号 1050 と 1060 のつながりを参照.）

編集機能および編集メニュー

ファンクション・キー 3 には RUN 50 のコマンドがセットされているからキー 3 を押しさえすれば編集を開始する．間もなく一頁毎に編集された結果が画面上に出る．もし，そこに誤りを見い出

せば，STOP キーで実行を中断し，LIST 1000－を投入すればもともとのデータが表示され，通常の BASIC プログラムの修正手順でデータを変更すればよい．また文章を追加する際には，使われていない中間の行番号で新たな DATA ステートメントをキー・インするとよい．その場合には，RENUM コマンドで，もう一度，1000 を先頭とする増分 10 の通し番号につけかえるのが便利となる．（なぜなら，ディスプレイ上に文章が表示される際，画面の左端に，その文章が何番目の行番号にあたるかの数字が書き出され，該当文章の検出を容易にしているから．）

なお，標準的にはＡ４版の印刷を想定し，一頁 25 行，一行 68 文字をセットしている（行番号 60 のＮおよびM）．RUN させると最初にこれらのパラメターの変更を問い合わせてくるので任意に変更することができ，勿論，それに応じた編集テキストが打ち出されることになる．

種々の編集についての選択メニュー中，CR とあるのはキャリッジ・リターン，つまり送信キーを押すの意味である．

文章の修正にあたっては，文章の先頭からやり直さずとも該当箇所の付近だけ見ればよいということが多い．そのための編集メニューとして

Sentence : CR (top), or Number ?

が用意されている．途中から見たい時はその付近の適当な行番号をキー・インすることにより当該番号をもつ文章が表示され，再度，イエスかノーかをたずねてくる．イエスなら CR を，ノーならピリオドを投入することにより求める文章に辿りつくことができる．

Page assign : CR ($p1$), or ?

が表示されたならば，編集する最初のページのナンバーリングを指定する．CR キーを入れると自動的に先頭ページに１頁の番号が割りあてられ，もし４をキー・インすれば先頭ページが４頁と順序づけられる．

Copy : 1 (all), 2 (in part), CR (nothing) ?

のメニューに対して，CR を送信すると，プリンターには何も出力されず，ディスプレイ上にのみ編集結果が出される．１をキー・インすれば，先頭ページから最終ページまで全ページがプリンターに打ち出される．２を選択すると，つづいて

Print pages : p1, p2

が表示され入力待ちとなる．p1 は印刷したい始まりのページであり，p2 は終りのそれである．

文章の保存

行番号 10 から 830 の部分と行番号 1000 以降の部分とをそれぞれ別のプログラム・ファイル名で外部記憶装置（このプログラムではバブル・メモリ）に保存してもよいし，合わせて一つのプログラム・ファイルとして保存するのも構わない．後者の方式の場合，再び LOAD して実行させるときには，この節の冒頭に述べたように

on BUB 1:

の表示に対して STOP キーをいれる必要があり，その後はファンクション・キー３で RUN をさせる必要がある．

前者の方式では，この表示に対しては行番号 1000 以降の DATA ステートメントの格納されているファイル名を指定すれば自動的に MERGE される．

なお，フロッピー・ディスクあるいはカセット・テープにより保存している場合は行番号 40 の BUB 1:を対応して，それぞれのファイル・デスクリプタに変更しておかなければならない．

おわりに

現在のシステム構成ではシリアル・ドット・プリンターの制約があり，印字品質は資料3に見る通りである．筆者の使い方としては，共著者間の日常的な原稿のやりとりには，このままで十分に用が足りているし，実際，学会誌への投稿もそのままでも受理されると聞いている．ただ現実には，タイプライターの印字に見劣りがしたり，他の機会への投稿に際してはタイプライター使用が指定されたりといった事情があり，完成原稿はもう一度タイプライターで打ち直すといった中途半端なラボラトリー・セミ・オートメーションになっている．そうではあっても，パソコン上で簡単に修正編集ができ，殆んど労せずして第何次目かの原稿が清書されて出てくるという利点は存分に味わっている．

最近タイプライターも電子化されパソコンと接続して出力機として使えるのはもとよりとして，オフラインでの機能としても数千文字のメモリーをもち編集可能なものが出てきている．タイプライターという従来のイメージからは考えられないような種々の機能が具ってきている．しかしながら修正機能という点で見たとき，パソコンによるワード・プロセッサーにははるかに及ばない印象をもつ．最大の難点は，電子タイプライターでは，メモリーからの情報の検索が一次元的にしか行えないことである．例えば文章の末尾を修正したいときでも，先頭から順にアクセスせざるをえないのであり，パソコンの場合のようにLISTコマンドで直接見たいステートメントを呼びだすのとでは，実用上は大変な差になることをこの間経験する機会があった．筆者もいずれは，パソコンの端末出力機としての電子タイプライターを導入する計画ではあるが，当分はパソコン・ワード・プロセッサーの首座は揺がないように思われる．

（本学講師　釧路分校）

(資料1) 英文ワード・プロセッサー, プログラム・リスト

```
10 'English editor: <version 16>
20    CLEAR 3000:OPEN "O",1,"LPTO:":PRINT#1,CHR$(27)"C"CHR$(0);CHR$(11):CLOSE1
30    DEFSTR A-F:A=CHR$(13):C=CHR$(34):KEY3,"R.50"+A:KEY4,"W.,20:L.1000-"+A
40    KEY5,C+A+"data "+C:INPUT "on BUB1:",F:F="BUB1:"+F:KEY8,"L."+C+F+C:MERGE F
50  DEFINT A-Z:DIM C$(30),U$(30),L$(30),T$(30),S$(30)
60    M=68:N=25:NN=5:OPEN "O",1,"LPTO:":GOSUB 620
70  READ A$:S=S+10
80    IF LEFT$(A$,2)="/&" THEN CLOSE 1:WIDTH ,20:END
90  S$(I)=MID$(STR$(S)+H$,2,LN)+" ":M=M1:X$=H$
100     IF LEFT$(A$,1) <> "/" THEN 150
110     IF LEFT$(A$,2) = "//" THEN D=NN ELSE D=VAL(MID$(A$,2,1))
120     IF MID$(A$,3,2)<>"##" THEN A$=SPACE$(D)+MID$(A$,3):GOTO 140
130        A$=SPACE$(D)+MID$(A$,5):MID$(S$(I),LN)="#":M=M2:X$=LR$:GOTO 150
140     IF A$=SPACE$(LEN(A$)) THEN READ A$:S=S+10:GOTO 320
150   G=LEN(C$(I)):R=M-G+1:IF R<2 THEN GOSUB 380:GOTO 270
160     IF MID$(A$,R,1)= "" THEN 290
170     J=R:GOSUB 360:IF J>R THEN 210
180     FOR J=R TO 2 STEP -1
190       IF MID$(A$,J-1,1)=" " OR MID$(A$,J-1,1)="-" THEN 210
200     NEXT J:  IF G=0 THEN GOSUB 600 ELSE GOSUB 380:GOTO 270
210       IF J>2 THEN IF MID$(A$,J-2,2)=" -" THEN J=J-1
220       IF LEN(A$)<J THEN 290
230     C$(I)=C$(I)+LEFT$(A$,J-1):A$=MID$(A$,J)
240       GOSUB 390:IF K=0 THEN 270 ELSE IF K>0 THEN 150
250         J=INSTR(A$,R$)+1:IF J=1 THEN PRINT "*****";S;C$(I):BEEP:STOP
260         GOSUB 360:GOTO 230
270     T$(I)=X$:I=I+1:IF I>N THEN GOSUB 500
280   GOTO 150
290   C$(I)=C$(I)+A$:GOSUB 390:READ A$:S=S+10
300     IF LEFT$(A$,1)="/" THEN 320
310     IF LEN(C$(I)) <M   THEN 90 ELSE GOSUB 380
320   T$(I)=X$:I=I+1
330     IF LEFT$(A$,2)="/&" THEN GOSUB 500:PRINT "To next page":GOTO 70
340     IF I>N THEN GOSUB 500
350 GOTO 90
360   WHILE MID$(A$,J,1)="," OR MID$(A$,J,1)="." OR MID$(A$,J,1)="-":J=J+1:WEND
370   WHILE MID$(A$,J,1)=" ":J=J+1:WEND:RETURN
380   WHILE LEFT$(A$,1) =" ":A$=MID$(A$,2):WEND:RETURN
390 K=0:Q=G:' upper("#") and lower("@") suffixes.
400   Q=INSTR(Q+1,C$(I),"#"):IF Q=0 THEN Q=G:GOTO 450
410     K=INSTR(Q+1,C$(I),"#")-1:IF K<0 THEN R$="#":RETURN
420     U$(I)=U$(I)+SPACE$(Q-LEN(U$(I))-1)+MID$(C$(I),Q+1,K-Q):P$=SPACE$(K-Q)
430     M$=MID$(C$(I),K+2,1):IF M$="," OR M$="." THEN MID$(P$,1)=M$:K=K+1
440     C$(I)=LEFT$(C$(I),Q-1)+P$+MID$(C$(I),K+2):GOTO 400
450   Q=INSTR(Q+1,C$(I),"@"):IF Q=0 THEN RETURN
460     K=INSTR(Q+1,C$(I),"@")-1:IF K<0 THEN R$="@":RETURN
470     U$(I)=LEFT$(U$(I),Q-1)+MID$(U$(I),Q+2)
480     L$(I)=L$(I)+SPACE$(Q-LEN(L$(I))-1)+MID$(C$(I),Q+1,K-Q)
490     C$(I)=LEFT$(C$(I),Q-1)+SPACE$(K-Q)+MID$(C$(I),K+2):GOTO 450
500 TS=TIME-TS:PG=PG+1:I=I-1:CLS:' On display.
510   FOR J=1 TO I:PRINT H$;U$(J);CL$;S$(J);C$(J);CL$;H$;L$(J):NEXT J
520                 PRINT TAB(37);"-";PG;"-";TAB(69);"<";TS;"sec>"
530   ON PT+1 GOTO 580,550,540
540   IF PG<P1 OR P2<PG THEN 580:' On printer
550     PRINT#1,STRING$(31-N,LF$);ESA$;CHR$(6): FOR J=1 TO I
560     PRINT#1,LF$;T$(J);U$(J);CL$;T$(J);C$(J);CL$;T$(J);L$(J);D2$:NEXT J
570     PRINT#1,ESA$;FF$;STRING$(2*(N-I)+3,LF$);TAB(37)"-"PG"-"FF$
```

```
580   FOR J=1 TO I:C$(J)="":U$(J)="":L$(J)="":T$(J)="":S$(J)=H$:NEXT J
590   I=1:TS=TIME:RETURN
600 PRINT LF$;"Now, increase (";M$;") M !":BEEP:RESTORE
610 PRINT LF$;M$;" =<76, ";N$;" <30 !";LF$:INPUT "Number M ";M
620   LF$=CHR$(10):FF$=CHR$(12):CR$=CHR$(13):D2$=CHR$(18):ESA$=CHR$(27)+"A"
630   CL$=LF$+CR$:C29$=" ? "+CHR$(29):M$="letters/line":N$="lines/page"
640 WIDTH 80,20:PRINT STRING$(35,"-");TAB(3);"Parameters:";M;M$;TAB(14);N;N$
650             PRINT STRING$(35,"_");LF$:GOSUB 790:IF Q$<>CR$ THEN 610
660   LN=(80-M)/2:H$=SPACE$(LN):M1=M:M2=(M+LN)*66/77-LN:IF M2=<0 THEN 600
670   LR$=SPACE$(LN*132/77)+CHR$(15)+CHR$(14):LN=LN-1:S=990:S0=S+10
680 INPUT "Sentence: CR (top), or Number ";Q$:IF Q$=""THEN PRINT:GOTO 720
690   NS=VAL(Q$):IF NS<S0 THEN BEEP:GOTO 680
700   WHILE NS>S:READ A$:S=S+10:WEND:NS=NS+10:PRINT LF$;S;A$;LF$
710     GOSUB 790:IF Q$<>CR$ THEN 700
720 INPUT "Page assign: CR (p1), of ";Q$:PG=VAL(Q$)+1/VAL("1"+Q$)-1
730 PRINT LF$;"Copy:  1 (all),  2 (in part),  CR (nothing)";C29$;
740     Q$=INKEY$:IF Q$="" THEN 740
750   PT=VAL(Q$):IF PT>2 THEN BEEP:GOTO 740 ELSE PRINT " ";Q$
760     PRINT:IF PT=2 THEN INPUT "Print pages:  p1, p2 ";P1,P2
770   I=N:GOSUB 580:WIDTH ,25
780   IF S>=S0 THEN RETURN 80 ELSE RETURN 70
790 PRINT "input:  . (no),  CR (ok)";C29$;:' yes  or  no
800       Q$=INKEY$:IF Q$="" THEN 800
810   IF Q$=CR$ THEN PRINT "ok";LF$:RETURN
820   IF Q$="." THEN PRINT "no":RETURN
830 BEEP:GOTO 800
```

(資料2) 文章の入力形式

```
1000 DATA "//##English word-processor system"
1010 DATA "/8(coded by M. Sakai)"
1020 DATA "//This word processor system is suit for researchers who publish pape
rs in foreign journals."
1030 DATA "  Energy devoted to typewriting is reduced to extreme extent when the
y use this system."
1040 DATA "  The computer program for the word processor is described in a versi
on of BASIC language, i.e., F-BASIC as listed in Materials 1 of the present pape
r."
1050 DATA "//In this program sentences to be typewitten are given in a form of D
ATA statement.  When you wish to modify a part of sentence, you can call it on C
RT display by keying in LIST i (i means a line number) com"
1060 DATA "mand and e.g., may delete some characters, or may insert characters b
y moving the cursor onto any word at whose position you try make a modification.
"
1070 DATA "  When you wish to change an ordering of some sentence and another on
e, you exchange line numbers of two sentences directly."
1080 DATA "//You can specify the superscript something like p#2#, or #4#He etc..
  The subscript is specified in a similar way, e.g., v@ij@, or A@xyz@ etc.."
1090 DATA "  Therefore we are able to use such an expression as p#2#=p@x@#2#+p@y
@#2#+p@z@#2#."
1100 DATA /&,/&
```

(資料3) 印刷例

English word-processor system

(coded by M. Sakai)

This word processor system is suit for researchers who publish papers in foreign journals. Energy devoted to typewriting is reduced to extreme extent when they use this system. The computer program for the word processor is described in a version of BASIC language, i.e., F-BASIC as listed in Materials 1 of the present paper.

In this program sentences to be typewitten are given in a form of DATA statement. When you wish to modify a part of sentence, you can call it on CRT display by keying in LIST i (i means a line number) command and e.g., may delete some characters, or may insert characters by moving the cursor onto any word at whose position you try make a modification. When you wish to change an ordering of some sentence and another one, you exchange line numbers of two sentences directly.

You can specify the superscript something like p^2, or ^4He etc.. The subscript is specified in a similar way, e.g., v_{ij}, or A_{xyz} etc.. Therefore we are able to use such an expression as $p^2=p_x^2+p_y^2+p_z^2$.