

4. 付 録

付録として、

1. 出現樹種リスト
2. 材積式計算プログラム
3. 使用データ を掲げた。

4-1 出現樹種リスト

今回の調査で対象となった樹種は、前述の主要6樹種を含め合計77種であった。この主要6樹種は現在エクアドル北東部において、合板用材として重点が置かれているものであるが、この6樹種以外にもいくつか合板用材として当地域において伐採されている樹種や、家具用材として商業的価値の高いもの、また自家消費用として地域住民が建築、カヌー等に利用しているものも見られる。

これら今回の調査で出現した樹種について、以下地方名、学名を掲げまとめた。

なお、学名については以下の参考文献により地方名からの推定を行なったため未だ不十分な点が多く、今後専門学者による同定作業が期待される。

(出現樹種リスト参考文献)

B I B L I O G R A F I A

=====

1. Algunos Nombres Botánicos de la FLORA ECUATORIANA : Ing. Victor E. Pasaca Ibra, QUITO - ECUADOR 1983.
2. Arboles Comunes de la Provincia de Esmeraldas : FO/SF/ECU 13/Vol.4/ Borrador , FAD
3. Catálogo de al Flora del Nororiente Ecuatoriano : Gonzalo Campuzano, CONOCOTO - ECUADOR, Mayo-1978
4. Informe de Actividades para la Elaboración de TABLAS DE VOLUMEN : Manuel Carrera B., 13 de junio 1982
5. Inventario Forestal de la Región Amazónica Ecuatoriana (Sector Central: Provincia de Pastaza); MAG-CLIRSEN, QUITO-ECUADOR 1981
6. Maderas y Maderables del Ecuador
7. Proyecto de Evaluación de los Recursos Forestales Tropicales (SIMUWIMA), Los Recursos Forestales de la América Tropical
8. Tablas de Volúmenes de Algunas Especies del Noroccidente Ecuatoriano, MAG-1975
9. The development of Ecuador's Forestry Resources and Institutions : Social Considerations, Theodore Macdonald Ph.D., Cultural Survival/
Harvard U.
10. Usos Probables de Algunas Maderas del Ecuador : Edgar Vasquez M., CONOCOTO-ECUADOR, 1981

(地方名) (他の地方名) (学 名) (科名)
 Nombre común Otros nombres Nombre científico Familia Observación

Agua caspi	Yacu caspi, Palo de agua	Guare-a sp.	Meliaceae	(Crias neuberthii ?)
Aguecatillo		Beilshamedia sulcata (R et?)	Lauraceae	
		Persea sp.	Lauraceae	
Ajua	(Ajo?)	Ocotea spp.	Lauraceae	
Amarillo	(zaramuyo?)	ocotea sp.	Lauraceae	
		Persea rigens	Lauraceae	
		Cryptocarya sp.	Lauraceae	
Ardilla caspi		Centrolobium patinense, Pit.	Leguminosae	
Arenillo		Centrolobium paraense	Leguminosae	
Azua	(Palo de chiche)	Pseudolmedia eggersii	Moraceae	
Balcilla				
Balsa	Balza	Ochroma lagopus	Bombacaceae	
Batia caspi	Batea caspi	Ochroma pyramidale	Bombacaceae	
Caimitillo		Chrysophyllum cainito	Fabaceae?	
		Chrysophyllum aurantium, Mig	Sapotaceae	
Caimito		Pouteria spp.	Sapotaceae	
Caimito de monte		Chrysophyllum exelcius	Sapotaceae	(Chrysophyllum cainito)
Calla caspi		Pouteria sp.	Sapotaceae	
Canelo		Endlicheria sp.	Lauraceae	
Canelo amarillo		Ocotea cernus	Lauraceae	
		Nectandra reticulata	Lauraceae	
		Urera sp.	Urticaceae	
Cacba		Platymiscium pinnatum	Papilionoideae	
Capirona		Calycophyllum spruceanum	Rubiaceae	
		Capirona decorticans, Spruce	Rubiaceae	
		Ladembergia sp.	Rubiaceae	

(地方名) (他の地方名)
 Nombre común Otros nombres

(学 名)
 Nombre científico

(科名)
 Familia

Observación

=====

Cara caspi			Boraginaceae	
			Dichapetalaceae	
Cauchero		Castilla tunu	Moraceae	
Cauchillo		Pseudolmedia laevigata	Moraceae	
		Ficus elastica	Moraceae	
Caucho		Castilla elastica Cerv	Moraceae	
		Castilla panamensis, O.F. Cook	Moraceae	
		Perebea guianensis	Moraceae	
		Sepium verum	Euphorbiaceae	
		Guarea guidonia	Meliaceae	
Cedrillo		Guarea rugosa	Meliaceae	
		Turpinia heterophylla	Staphyleaceae	
		Cedrela spp.	Meliaceae	
Cedro		Cedrela rosei, (MA)	Meliaceae	
		Cedrela fissilis, (MB) Vell	Meliaceae	
		Trichilia sp.	Meliaceae	
		Ceiba pentandra, (L) Gaertn	Bombacaceae	
Ceibo	Lagarto caspi	Ochroma tomentosa, var Ibar. Gilid	Bombacaceae	
		Turpinia heterophylla, Harms	Meliaceae	
Copal	Anime, Copalillo	Dacryodes occidentalis	Burseraceae	D. peruviana
		Dacryodes kuskachana	Burseraceae	
		Dacryodes copularis	Burseraceae	
		Protium colombianum	Burseraceae	
		Tetracastria sp.	Burseraceae	
		Compsoeura sp.	Myristicaceae	
		Hymenaea courbaril L.	Caesalpinoideae	
Cochillo				
Cruz caspi		Brownea sp.	Caesalpinoideae	
		Cespedezia spathulata?	Dchnaceae	
Chachaco de piedra				
Chonta caspi		Cespedezia sp.	Dchnaceae	

(地方名) (他の地方名) (学名) (科名)
 Nombre común Otros nombres Nombre científico Familia Observación

Choto				
Chuncho		Cedrelinga catenaeformis	Mimosoideae	
Felipo		Comsoneura sp.	Myristicaceae	
Guabillo		Inga spp.	Mimosoideae	
Guabo	Guaba	Inga sp.	Mimosoideae	
Guapa	Sangre de gallina, Coquito, Cuangare	Dialyanthera sp. Dialyanthera parvifolia Dialyanthera gordoniae-folia Dialyanthera gracilipes Virola sp. Couratari sp. Iryanthera sp. Vismia obtusa	Myristicaceae Myristicaceae Myristicaceae Myristicaceae Myristicaceae Lecythidaceae Myristicaceae Guttiferae	
Guarango		Caesalpinia tinctoria Caesalpinia spinosa Acacia macracantha Humb et Bonpl Acacia flexuosa Humb et Bon Coulteria sp. Tara spinosa (Mold) Bet Rose Inga sp. Parkia sp.	Caesalpinoideae Caesalpinoideae Mimosoideae Mimosoideae Leguminosae Leguminosae Mimosoideae Mimosoideae	
Guarumo		Cecropia sp. Cecropia sciadophylla Eugenia sp. Microphalis sp. Psidium guineense, Sw Psidium guajava L. Mimantia guianensis Guaiacum officinalis, L	Moraceae Moraceae Myrtaceae Sapotaceae Myrtaceae Myrtaceae Glacaceae Zygophyllaceae	Pourouma cecropiaefolia (Moraceae)
Guayabillo	(Guayabilla)			
Guayabo				
Guayacán pechiche				

(地方名)	(他の地方名)	(学 名)	(科名)	Observación
Nombre común	Otros nombres	Nombre científico	Familia	
(Guayaén pechiche)		Tabebuia guayacana	Bignoniaceae	
		Tabebuia sp.	Bignoniaceae	
Guión	Jigua	Pseudolmedia eggersii Standl	Moraceae	
Haya caspi		Ficus gwanensis	Moraceae	
Higuerrón	Leche brava	Ficus insipida	Moraceae	
		Ficus sp.	Moraceae	
Jigua		Ocotea floribunda	Lauraceae	
		Ocotea sp.	Lauraceae	
		Nectandra cissiflora	Lauraceae	
		Nectandra laevis	Lauraceae	
		Nectandra cossitiora	Lauraceae	
		Nectandra reticulata	Lauraceae	
		Beilschmiedia alliophylla Kosterm	Lauraceae	
		Endlicheria formosa	Lauraceae	
		Aniba sp.	Lauraceae	
		Pleurothyrium sp.	Lauraceae	
		Virola elongata	Myristicaceae	
	(Guión)	Pseudolmedia eggersii Standl	Moraceae	
Lacao		Cordia alliodora	Boraginaceae	
Laurel		Sapium verum Hemsf	Euphorbiaceae	
Lechero		Brosimum utile, (HBK) Pit	Moraceae	
		Euphorbia sp.	Euphorbiaceae	
Lotería			Myristicaceae	
Manduro caspi		Coussapoa sp.	Moraceae	
	(Achote)	Bixa orellana L.	Bixaceae	
Mangua caspi				
Maní de árbol	(Guachanzo)	Caryodendron amazonicum	Euphorbiaceae	
		Caryodendron orinocensis	Euphorbiaceae	

(地方名) (其他地方名) (学名) (科名)
 Nombre común Otros nombres Nombre científico Familia Observación

(Maní de árbol)

Ficus sp.
 Moreceae

Manzanillo

Flacourtiaceae

Manzano colorado

Hippomane mancinella

Euphorbiaceae

Mascarey

Guarea vellutina

Meliaceae

Hyeronima choacoensis

Euphorbiaceae

Hyeronima sp.

Euphorbiaceae

Clusia sp.

Clusiaceae (Guttiferae)

Matapalo

Ficus martini Mig

Moraceae

Ficus sp.

Moraceae

Coussapoa rotunda

Moraceae

Coussapoa oligoneura

Moraceae

Coussapoa eggersii

Moraceae

Clusia sp.

Guttiferae

Antidaphne colombiana, K

Loranthaceae

Phoradendron sp.

Loranthaceae

Mindal

Chimarrhis hookeri

Rubiaceae

Obo

Spondias mombin

Anacardiaceae

Palo de achote

(Achotillo)

Quapoya peruviana

Clusiaceae (Guttiferae)

Vismie vacillifera, ferruginea

Guttiferae

Apeiba membranacea

Tiliaceae

Bixa arborea ?

Bixaceae

Palo de maíz

Palo de tinta

Paparao

Peine de mono

(Nacuchacaspi)

Apeiba membranacea, Spruce

Tiliaceae

Pilchimuyo

Pili caspi

Pungara caspi

(Pungacaspi ?)

Apeiba aspera, Aubl

Tiliaceae

Raiz de palo

Tiliaceae

(地名) (其他地名) (学 名) (科名)
 Nombre común Otros nombres Nombre científico Familia Observación

Remo caspi			Leguminosae	
Rumi caspi	(Palo de piedra)		Polygonaceae	
Sande		Brosimum utile	Moraceae	
Saramuyo	(Piedra de palo)	Terminalia sp.	Combretaceae	
Shelipo				
Tocota	(Cedrillo ?)	Trichilia sp.	Meliaceae	
		Cedrela sp.	Meliaceae	
		Guarea sp.	Meliaceae	
Uba	(Ubilla)	Pourouma choocoana	Moraceae	
	(Uva, Uvilla)	Pourouma cecropiaefolia	Moraceae	
		Pourouma guanensis	Moraceae	
		Cecropia obtusifolia	Moraceae	
		Ficus trigonata	Moraceae	
Ubilla		Coussapoa sp.	Moraceae	
		Physalis subglabrata, Mac et Bush	Solanaceae	
Yuyún		Terminalia sp.	Combretaceae	
		Chimarrhis williamsii	Rubiaceae	
Zapote	(Sapote)	Matisia cordata	Bombacaceae	
		Matisia alata	Bombacaceae	
		Capparis angulata	Capparidaceae	
		Achras sp.	Sapotaceae	
		Chrysophyllum sp.	Sapotaceae	

付 録

4-2 材積式計算プログラム

今回作成したプログラムはNEC PC 9801 VM 0用で、プログラム用とデータ用のフロッピーディスクに分けて作成した。PC-9801の電源を入れた後、ドライブ1にプログラムのフロッピーディスクをセットし、ドライブ2にデータのフロッピーディスクをセットする。リセットボタンを押すとHow many files (0-15)?とメッセージされるので「5」を入力しておく。

以下順にプログラムの概要とプログラムリストを示す。

(1) menuuu. PRO

希望する仕事を表示し、選択したプログラムを実行するメニュープログラムである。run "menuuu.PRO" と入力すると次のメッセージが表示される。

```
Preparation of data 1 ( Species Dbh Chd Th Ch Bh ) ***** 1
Preparation of data 2 ( diameter with bark of each 2 m ) ** 2
Preparation of data 3 ( bark thick of each 2 m ) ***** 3
Preparation of data 4 ( branch data ) ***** 4
Sum up of data ( data1 - data 4 ) ***** 5
Check of sum up data ( read and list ) ***** 6
Calculation of volume formula ***** 7
Fitting test of volume formula ***** 8
Preparation of volume table ***** 9
End ***** E
```

この中から希望する仕事番号を入力するとそのプログラムを実行する。ただし5番を実行する時は1~4までのデータがすべて入力された時点で行う。また今回の材積表のデータは1~4のプログラムを実行して既にデータ用のフロッピーディスクに作成されている。

(2) dapalt. PRO (menuuu. PRO の 1)

このプログラムは、単木のデータ番号、樹種コード、胸高直径、利用高直径、全樹高、利用高、板根高を入力するプログラムである。

menuuu. PRO を実行させて1を入力するとdapalt. PRO が実行され、次のメッセージが表示される。

```
Preparation, Correction and Observacion of data file
Data are Number, Species No, DBH, Commercial height diameter
Total height, Commercial height, Buttress height
```

```
*** Enter floppy disk in Drive Number 2 ***
*** If you enter floppy disk, Put Y or y **
```

ここでドライブ2にデータ用のフロッピーディスクを入れた後、Y又はyを入力する。

次に、

```
Do you want to create new file? ***** 1
Do you want to correct data? ***** 2
```

とメッセージされるので新しいデータファイル名又は、修正したいデータファイル名を入力する。

◎ 1を入力した場合

データ用フロッピーディスクに既に入っているファイル名が表示される。

```
The data that you have in drive 2 are as follows.
```

それらのファイル名と重ならないように新しいファイル名を入力する。

```
Input new file name that you want to create.
```

新しいファイル名が入力されると確認のメッセージが表示され、

```
Are you sure create new file? ( y / n )
```

Yを入力するとプログラムが実行され、Nを入力するとメニュープログラムに戻る。ファイル名を入力する時に既にフロッピーディスクに入力されているファイル名と同じ名前のもを入力すると、

```
You have already same name file.
Computer kill the file, and then create new file.
Are you sure? ( y / n )
```

とメッセージされ、Yを入力すると古いファイル名は消され新しいファイルが作成されるので注意を要する。Nを入力すると古いファイルはそのまま残されメニュープログラムに戻る。

◎ 2を入力した場合

データ用フロッピーディスクに既に入っているファイル名が表示される。

The data that you have in drive 2 are as follows.

次に修正したいファイル名を入力する。

Input file name that you want to correct

ファイル名が正しければ実行され、フロッピーディスクに入力したファイル名がなければ、

You have not the file yet. Return to first !!!

と表示されメニュープログラムに戻る。

◎プログラムが実行されると次のメッセージが表示され、

```
Change of data ***** C
Append and Preparation of data ***** A
Delete of data ***** D (Attention)
List ( in Display ) ***** L
List ( in Printer ) ***** P
End ***** E
```

Cはデータの修正

Aは # の入力

Dは / の削除

Lは / を画面上に表示し

Pは / をプリンタに表示する。

Eを入力するとメニュープログラムに戻る。

ここでDを入力する場合は、入力されたファイルのデータ順序が他のデータファイルのデータと異なってしまうため注意を要する。

なおデータは、dapalt. DAT としてデータ用フロッピーディスクに作成済である。

(3) dapcor. PRO (menuuu. PRO の 2)

2 m 毎の皮付きの直径を入力するプログラムで、操作は dapalt. PRO と同様である。

データは dapcor. DAT として作成済である。

(4) espcor. PRO (menuuu. PRO の 3)

2 m 毎の皮厚を入力するプログラムで、操作は dapalt. PRO と同様である。

データは espcor. DAT として作成済である。

(5) ramade. PRO (menuuu. PRO の 4)

枝データを入力するプログラムで、操作は dapalt. PRO と同様である。

データは ramade. DAT として作成済である。

(6) sumade. PRO (menuuu. PRO の 5)

上記の(2)~(5)のデータをまとめるプログラムで、操作は前記までと同様であり、(2)~

(5)までのファイル名を入力すると材積を計算した後

1. No
2. 樹種コード
3. 胸高直径
4. 利用高直径
5. 全樹高
6. 利用高
7. 板根高
8. 幹材積(皮付き)
9. 幹材積(皮無し)
10. 枝材積(皮付き)
11. 幹材積+枝材積(皮付き)

のデータが sumade. DAT にまとめられる。

(7) sumade. LEE (menuuu. PRO の 6)

(6)で作成した sumade. DAT のデータをプリンターに印刷するプログラムである。

これを出力してデータを確認し、修正するデータがあれば(2)~(6)のプログラムをくり返し、完全なデータとする。

(8) volcal. REG (menuuu. PRO の 7)

材積式 $\log V = a + b \log D + c \log H$ を作成するプログラムで、最初の操作方法は(2)の dapalt. PRO と同様である。使用するデータファイルは sumade. DAT であり、これを入力するとデータを読んだ後、次のように表示される。

Diameter at breath height	*****	1
Diameter at commercial height	*****	2
Total height	*****	3
Commercial height	*****	4
Buttress height	*****	5
Volume with bark	*****	6
Volume without bark	*****	7
Branch volume	*****	8
Volume with bark + Branch volume	*****	9

**** Select criterion variable (Y) ****

ここで従属変数にあたる変数の番号を入力する。

次に

```
**** Select explanatory variables X 1, X 2 ****
```

と表示されるので独立変数の番号を入力する。

次に

```
"Selection of data"  
"From what No until what No do you want to calculate?"  
  "From"  
  "Until"
```

と表示されるので、計算したいデータ番号を何番から何番までと入力する。

次に

```
Input exclusive data No  
If end, input 0
```

と表示されるので、異常データ等、棄却するデータ番号を入力する。ここで注意するのはデータ番号は、samade. DATの番号と対応しているので、くり返し棄却データを除いて計算する場合、データ番号をよく確認する。出力結果のモデルとして、主要6樹種の結果をプログラムリストの後に付した。

(9) fittit. REG (menuuu. PROの8)

適合性判別のプログラムであり、操作は(8)の volcal. REG とほぼ同様である。最初に

```
Input constant term  
Input regression coefficient X 1  
Input regression coefficient X 2
```

と表示されるので、決定した材積式の定数項と偏回帰係数を入力する。その後の操作は(8)の volcal. REG と同様に行う。すると、入力された材積式に希望するデータが入力され、実材積と推定材積の相関が計算され、各結果が出力される。主要6樹種の材積式に主要6樹種以外のデータを入力した打出し結果をプログラムリストの後にモデルとして付した。

(10) voltab. PRO (menuuu. PROの9)

材積表をプリンターに出力するプログラムである。最初に

6 especies principales (con corteza)	*****	1
Resto de especies (con corteza)	*****	2
6 especies principales (sin corteza)	*****	3
Resto de especies (sin corteza)	*****	4
Volumen de ramas (con corteza)	*****	5

と表示されるので、作成したい材積表の番号を入力する。次に

Input Constant term	(a)
Input Regression coefficient	(b)
Input Regression coefficient	(c)

と表示されるので、定数項と偏回帰係数を入力すると材積表を出力する。

使用プログラムリスト

言語 N₈₈ - BASIC (86)
機種 NEC PC-9801 VMO

(1) menuuu. PRO

(2) dapalt. PRO

(3) dapcor. PRO

(4) espcor. PRO

(5) ramade. PRO

(6) sumade. PRO

(7) sumade. LEE

(8) volcal. REG

(主要 6 樹種の計算結果)

(9) fittit. REG

(主要 6 樹種の材積式に主要 6 樹種以外のデータ
を入力した計算結果)

(10) voltab. PRO

(1) menuuu. PRO

```

100 '==== menuuu.PRO =====
110 '
120 ' PROGRAM save "menuuu.PRO"
130 '
140 ' Menu program
150 '
160 '=====
170 '
180 WIDTH 80,25 :CONSOLE 0,25,0,0
190 DEFINT A-Z
200 '
210 READ MENU
220 DIM MENU$(MENU),PROG$(MENU)
230 FOR I=1 TO MENU
240 READ MENU$(I),PROG$(I)
250 NEXT I
260 '
270 PRINT TAB(15)
280 PRINT "-----"
290 PRINT TAB(15)
300 PRINT "! Menu for preparation of volume table !"
310 PRINT TAB(15)
320 PRINT "-----"
330 PRINT
340 '
350 FOR I=1 TO MENU
360 LOCATE 10,I+3 :PRINT MENU$(I)
370 NEXT I
380 '
390 LOCATE 22,15
400 PRINT STRING$(28,"-")
410 LOCATE 22,16
420 PRINT "Input No. of work"
430 N$=INKEY$ : IF N$="" THEN GOTO 430
440 IF N$="E" OR N$="e" THEN GOTO 530
450 N=VAL(N$)
460 IF N>MENU OR N=0 THEN BEEP : GOTO 430
470 '
480 COLOR 4
490 LOCATE 10,N +3
500 PRINT MENU$(N)
510 COLOR 0
520 FOR I= 1 TO 200 : BEEP 1 : BEEP 0 : NEXT I
530 LOCATE 28,18
540 COLOR 6 : PRINT " cofirmation "
550 COLOR 0 : LOCATE 23,19
560 PRINT " excute(,) cancel(.)"
570 I$=INKEY$ :IF I$="" THEN GOTO 570
580 IF I$="," AND N$="e" OR N$="E" THEN CLS :END
590 IF I$="," THEN RUN PROG$(N)
600 IF I$<>"," THEN BEEP : GOTO 570
610 LOCATE 10,N+3 : PRINT MENU$(N)
620 LOCATE 10,18 : PRINT SPC(50)
630 LOCATE 10,19 : PRINT SPC(50)
640 GOTO 430
650 '-----
660 'menu data
670 '-----
680 DATA 10
690 DATA Preparation of data1 ( Species Dbh Chd Th Ch Bh ) ***** 1,dapalt.PRO
700 DATA Preparation of data2 ( diameter with bark of each 2m ) * 2,dapcor.PRO
710 DATA Preparation of data3 ( bark thickof each 2m ) ***** 3,espcor.PRO
720 DATA Preparation of data4 ( branch data ) ***** 4,ramade.PRO
730 DATA Sum up of data ( data1 ~ data4 ) ***** 5,sumade.PRO
740 DATA Check of sum up data ( read and list ) ***** 6,sumade.LEE
750 DATA Calculation of volume formula ***** 7,volcal.REG
760 DATA Fitting test of volume formula ***** 8,fittit.REG
770 DATA Preparation of volume table ***** 9,voltab.PRO
780 DATA End ***** E,dummy

```

(2) dapalt.PRO

```

10000 '==== dapalt.PRO =====
10010 '
10020 '   PROGRAM save "dapalt.PRO"
10030 '
10040 '   Preparation of data file
10050 '
10060 '=====
10070 '
10080 '   WIDTH 80,25 : CONSOLE 0,25,0,1
10090 PRINT " "
10100 PRINT "| Preparation , Correction and Observacion of data file |"
10110 PRINT "| Data are Number, Species No., DBH, Commercial height diameter |"
10120 PRINT "| Total height, Commercial height, Buttressheight |"
10130 PRINT " "
10140 PRINT
10150 PRINT " "
10160 PRINT "| *** Enter floppy disk in Drive Number 2 *** |"
10170 PRINT "| *** If you enter floppy disk, Put Y or y ** |"
10180 PRINT " "
10190     I$ = INKEY$ : IF I$ = "" THEN GOTO 10190
10200     IF I$="Y" OR I$="y" THEN GOTO 10220
10210     GOTO 10150
-----
10220 CLS
10230 MAXITEM =6
10240 TRUE =-1 : IN.USE = -1 : NIL = 0
10250 LLEN = 26           : ' LOGICAL RECORD LENGTH
10260 PLEN = 256        : ' PHYSICAL RECORD LENGTH
10270 BF = PLEN % LLEN  : ' BLOCKING FACTOR
10280 DIM F$(MAXITEM), ITEM.NAME$(MAXITEM), ITEM.FORM$(MAXITEM)
10290 DIM P$(BF-1)
10300 GOSUB *SELECT.FILE
10310 GOSUB *INIT.JOB
-----
10320 '
10330 '
10340 FOR CND=0 TO -1 STEP -1 : 'REPEAT
10350 CLS
10360 PRINT " "
10370 PRINT "| Change of data ***** C |"
10380 PRINT "| "
10390 PRINT "| Append and Preparation of data ***** A |"
10400 PRINT "| "
10410 PRINT "| Delete of data ***** D (Attention) |"
10420 PRINT "| "
10430 PRINT "| List ( in Display ) ***** L |"
10440 PRINT "| "
10450 PRINT "| List ( in Printer ) ***** P |"
10460 PRINT "| "
10470 PRINT "| End ***** E |"
10480 PRINT " "
10490     C$=INPUT$(1)
10500     C = ( INSTR("CcAaDdLlPpEe", C$)+1 ) %2
10510     ON C GOSUB *CHANGE , *APPEND , *DEL , *DISPLAY , *PRINTER
10520     CND = ( C = 6 )
10530 NEXT CND : 'UNTIL "e" or "E" is pressed
10540 GOSUB *END.JOB
10550 RUN "menuuu.PRO"
-----
11000 '
11010 *CHANGE
11020 GOSUB *GET.REC : IF OUT.RANGE THEN RETURN
11030 GOSUB *ITEM.LIST
11040 INPUT "Number of article = ";N
11050 N=N-1
11060 IF (N>MAXITEM) OR (N<0) THEN RETURN
11070 GOSUB *READ.ITEM
11080 PUT #1, PREC
11090 GOTO *DISP.DATA
-----
11100 '
11110 *APPEND
11120 CLS
11130 PRINT "Put data ( New or Append )"
11140 GOSUB *NEW.REC
11150 PRINT USING "(###) "; LREC;
11160 GOSUB *ITEM.LIST
11170 FOR N = 0 TO MAXITEM

```

```

11180          GOSUB *READ.ITEM
11190      NEXT N
11200      FREE = CVI(FP$)
11210      RSET FP$ = MKI$( IN.USE ) : PUT #1, PREC
11220      GOTO *DISP.DATA
11230  -----
11240 *DEL
11250      GOSUB *GET.REC : IF OUT.RANGE THEN RETURN
11260      RSET FP$ = MKI$( FREE ) : PUT #1,PREC
11270      FREE = LREC
11280      RETURN
11290  -----
11300 *DISPLAY
11310      GOSUB *GET.REC : IF ( LREC <> 0 ) THEN RETURN
11320      LREC = 0
11330      FOR PREC = 1 TO LOF (1)
11340          GET #1, PREC
11350          FOR K = 0 TO BF-1
11360              FIELD #1,K*LLEN AS DUMMYS, 2 AS FPS, 2 AS F$(0), 2 AS F$(1), 4 AS
                F$(2), 4 AS F$(3), 4 AS F$(4), 4 AS F$(5), 4 AS F$(6)
11370              IF ( LREC <> 0 ) THEN GOSUB *DISP.DTDOS
11380              LREC = LREC + 1
11390          NEXT K
11400      NEXT PREC
11410      PRINT "If end put Y or y "
11420      IS = INKEYS : IF IS = "" THEN GOTO 11420
11430      IF IS="Y" OR IS="y" THEN GOTO 11450
11440      GOTO 11410
11450      RETURN
11460  -----
11470 *GET.REC
11480      INPUT "Number of data = " ;LREC
11490      PREC = 1 + LREC % BF : COLM = LREC MOD BF
11500      OUT.RANGE = ( PREC > LOF(1) ) OR ( LREC < 1 )
11510      IF OUT.RANGE THEN RETURN
11520      GET #1, PREC
11530      FIELD #1, COLM*LLEN AS DUMMYS, 2 AS FPS, 2 AS F$(0), 2 AS F$(1), 4
                AS F$(2), 4 AS F$(3), 4 AS F$(4), 4 AS F$(5), 4 AS F$(6)
11540  -----
11550 *DISP.DATA
11560      IF( CVI(FP$) <> IN.USE ) THEN OUT.RANGE = TRUE : RETURN
11570      PRINT USING "(###) " ; LREC;
11580      FOR I= 0 TO MAXITEM
11590          ON I+1 GOTO 11610, 11610
11600              PRINT USING ITEM.FORM$(I); CVS(F$(I)); : GOTO 11620
11610              PRINT USING ITEM.FORM$(I); CVI(F$(I)); : GOTO 11620
11620      NEXT I
11630      PRINT : FOR I=1 TO 5000 : NEXT I
11640      RETURN
11650  -----
11660 *DISP.DTDOS
11670      IF( CVI(FP$) <> IN.USE ) THEN OUT.RANGE = TRUE : RETURN
11680      PRINT USING "(###) " ; LREC;
11690      FOR I= 0 TO MAXITEM
11700          ON I+1 GOTO 11720, 11720
11710              PRINT USING ITEM.FORM$(I); CVS(F$(I)); : GOTO 11730
11720              PRINT USING ITEM.FORM$(I); CVI(F$(I)); : GOTO 11730
11730      NEXT I
11740      PRINT
11750      RETURN
11760  -----
11770 *PRINTER
11780      GOSUB *GET.REC : IF ( LREC <> 0 ) THEN RETURN
11790      LREC = 0
11800      FOR PREC = 1 TO LOF (1)
11810          GET #1, PREC
11820          FOR K = 0 TO BF-1
11830              FIELD #1,K*LLEN AS DUMMYS, 2 AS FPS, 2 AS F$(0), 2 AS F$(1), 4 AS
                F$(2), 4 AS F$(3), 4 AS F$(4), 4 AS F$(5), 4 AS F$(6)

```

```

11840         IF ( LREC <> 0 ) THEN GOSUB *PRINT.DATA
11850         LREC = LREC + 1
11860         NEXT K
11870     NEXT PREC
11880     RETURN
11890 -----
11900 *PRINT.DATA
11910     IF( CVI(FPs) <> IN.USE ) THEN OUT.RANGE = TRUE : RETURN
11920     LPRINT USING "(###) "; LREC;
11930     FOR I= 0 TO MAXITEM
11940         ON I+1 GOTO 11960,11960
11950         LPRINT USING ITEM.FORMS(I); CVS(Fs(I)); : GOTO 11970
11960         LPRINT USING ITEM.FORMS(I); CVI(Fs(I)); : GOTO 11970
11970     NEXT I
11980     LPRINT
11990     RETURN
12000 -----
12010 *READ.ITEM
12020     PRINT "( "; N+1; " ) "; ITEM.NAMES(N); " = ";
12030     INPUT Ds
12040     ON N+1 GOTO 12060, 12060
12050     LSET Fs(N) = MKSs( VAL(Ds) ) : RETURN
12060     LSET Fs(N) = MKIs( VAL(Ds) ) : RETURN
12070 -----
12080 *NEW.REC
12090     IF( FREE = NIL ) THEN *ADD.NEW.REC
12100     LREC = FREE
12110     PREC = 1 + LREC ¥ BF : COLM = LREC MOD BF
12120     GET #1, PREC
12130     FIELD #1, COLM*LLEN AS DUMMYS, 2 AS FPs, 2 AS Fs(0), 2 AS Fs(1),
4 AS Fs(2), 4 AS Fs(3), 4 AS Fs(4), 4 AS Fs(5), 4 AS Fs(6)
12140     FREE = CVI( FPs )
12150     RETURN
12160 -----
12170 *ADD.NEW.REC
12180     LREC = LOF(1) * BF
12190     PREC = LOF(1) + 1
12200     FIELD #1, 2 AS Ps(0), 24 AS D0s, 2 AS Ps(1), 24 AS D1s, 2 AS Ps(2), 24
AS D2s, 2 AS Ps(3), 24 AS D3s, 2 AS Ps(4), 24 A
S D4s, 2 AS Ps(5), 24 AS
D5s, 2 AS Ps(6), 24 AS D6s, 2 AS Ps(7), 24 AS D7s, 2 AS Ps(8)
12210     LSET Ps( BF-1) = MKIs( NIL)
12220     FOR I = BF-2 TO 0 STEP -1
12230         LSET Ps(I) = MKIs( LREC + I + 1 )
12240     NEXT I
12250     FREE = LREC
12260     FIELD #1, 2 AS FPs, 2 AS Fs(0), 2 AS Fs(1), 4 AS Fs(2), 4 AS Fs(3),
4 AS Fs(4), 4 AS Fs(5), 4 AS Fs(6)
12270     RETURN
12280 -----
12290 *ITEM.LIST
12300     FOR I = 0 TO MAXITEM
12310         IF( (I MOD 5) = 0 ) THEN PRINT : PRINT TAB(7);
12320         PRINT USING "##)8      8"; I+1; ITEM.NAMES(I);
12330     NEXT I
12340     PRINT
12350     RETURN
12360 -----
12370 *INIT.JOB
12380     FOR I = 0 TO MAXITEM
12390         READ ITEM.NAMES(I)
12400     NEXT
12410     FOR I = 0 TO MAXITEM
12420         READ ITEM.FORMS(I)
12430     NEXT
12440     IF( LOF(1) = 0 ) THEN *PUT.1ST.REC
12450     FIELD #1, 2 AS FPs.
12460     GET #1, 1 : FREE = CVI( FPs )
12470     RETURN

```

```

12480 -----
12490 *PUT.1ST.REC
12500 FIELD #1, 2 AS FP$, 24 AS D0$, 2 AS P$(1), 24 AS D1$, 2 AS P$(2), 24
S D4$, 2 AS P$(5), 24 AS AS D2$, 2 AS P$(3), 24 AS D3$, 2 AS P$(4), 24 A
D5$, 2 AS P$(6), 24 AS D6$, 2 AS P$(7), 24 AS D7$, 2 AS P$(8)
12510 LSET P$( BF-1) = MKI$(NIL)
12520 FOR I = BF-2 TO 1 STEP -1
12530 LSET P$(I) = MKI$( I+1)
12540 NEXT I
12550 FREE = 1
12560 PUT #1, 1
12570 RETURN
12580 -----
12590 *END.JOB
12600 FIELD #1, 2 AS FPS
12610 GET #1 ,1
12620 RSET FP$ = MKI$( FREE ) : PUT #1,1
12630 CLOSE #1
12640 RETURN
12650 -----
12660 DATA "Number....", "No. Specis", "D B H.....", "D.Com.Heit","Total.Heit"
12670 DATA "Com.Heit..", "Butt.Heit."
12680 DATA "#####", "#####", "#####.#", "#####.#", "#####.#"
12690 DATA "#####.#", "#####.#"
13000 -----
13010 *SELECT.FILE
13020 COLOR 5
13030 PRINT "-----"
13040 PRINT " | Do you want to create new file ? ***** 1 |"
13050 PRINT " |-----"
13060 PRINT " | Do you want to correct data ? ***** 2 |"
13070 PRINT " |-----"
13080 COLOR 4
13090 PRINT " Input No. of work " : INPUT X
13100 IF X=1 THEN GOSUB *CREATE.FILE : RETURN
13110 IF X=2 THEN GOSUB *CORRECT.DATA : RETURN
13120 GOTO 13030
13130 -----
13140 *CREATE.FILE
13150 PRINT "-----"
13160 PRINT " | The data that you have in drive 2 are as follows. |"
13170 PRINT " |-----"
13180 PRINT "( Drive 2 )" : FILES 2
13190 COLOR 7
13200 PRINT "-----"
13210 PRINT " | Input new file name that you want to create. |"
13220 PRINT " |-----"
13230 INPUT FILE$
13240 XS="2:"
13250 FILENAMES=X$+FILES$
13260 IF ( FILES$ = "" ) THEN GOTO 13200
13270 OPEN FILENAMES AS #1
13280 IF LOF(1)<>0 THEN GOTO 13350
13290 PRINT "-----"
13300 PRINT " | Are you sure create new file ? ( y / n ) |"
13310 PRINT " |-----"
13320 INPUT B$
13330 IF NOT ((B$ = "Y") OR (B$ = "y")) THEN *GOBACK
13340 RETURN
13350 FOR I=1 TO 500 : BEEP 1 : BEEP 0 : NEXT I
13360 PRINT "-----"
13370 PRINT " | You have already same name file. |"
13380 PRINT " | Computer kill the file, and then create new file. |"
13390 PRINT " | Are you sure ? ( y / n ) |"
13400 PRINT " |-----"
13410 INPUT A$
13420 IF NOT ((A$ = "Y") OR (A$ = "y")) THEN *BACK
13430 CLOSE : KILL FILENAMES$
13440 OPEN FILENAMES AS #1 :RETURN
13450 -----
13460 *CORRECT.DATA

```

```

13470 PRINT "
13480 PRINT "| The data that you have in drive 2 are as follows.
13490 PRINT "
13500 PRINT "( Drive 2 )" : FILES 2
13510 COLOR 7
13520 PRINT "
13530 PRINT "| Input file name that you want to correct
13540 PRINT "
13550 INPUT FILES
13560 X$="2:"
13570 FILENAMES=X$+FILES
13580 IF ( FILES$ = "" ) THEN RETURN
13590 OPEN FILENAMES AS #1
13600 IF LOF(1)<>0 THEN RETURN
13610 FOR I=1 TO 500 : BEEP 1 : BEEP 0 : NEXT I
13620 PRINT "
13630 PRINT "| You have not file yet. Return to first!!!
13640 PRINT "
13650 GOTO *GOBACK
13660 -----
13670 *GOBACK
13680 CLOSE
13690 KILL FILENAMES
13700 PRINT "Back to menu!!!"
13710 FOR I=1 TO 6000
13720 NEXT I
13730 RUN "menuuu.PRO"
13740 -----
13750 *BACK
13760 CLOSE
13770 PRINT "Back to menu!!!"
13780 FOR I=1 TO 6000
13790 NEXT I
13800 RUN "menuuu.PRO"

```

(3) dapocr. PRO

```

10000 '==== dapcor.PRO =====
10010 '
10020 ' PROGRAM save "dapcor.PRO"
10030 '
10040 ' Preparation of data file, diameter with bark of each 2m
10050 '
10060 ' -----
10070 '
10080 ' WIDTH 80,25 : CONSOLE 0,25,0,1
10090 PRINT "
10100 PRINT "| Preparation , Correction and Observacion of data file
10110 PRINT "|
10120 PRINT "| Data are each 2 m diameter with bark
10130 PRINT "
10140 PRINT
10150 PRINT "
10160 PRINT "| *** Enter floppy disk in Drive Number 2 *** |"
10170 PRINT "| *** If you enter floppy disk, Put Y or y ** |"
10180 PRINT "
10190 IS = INKEY$ : IF IS = "" THEN GOTO 10190
10200 IF IS="Y" OR IS="y" THEN GOTO 10220
10210 GOTO 10150
10220 -----
10230 CLS
10240 MAXITEM = 22
10250 TRUE = -1 : IN.USE = -1 : NIL = 0
10260 LLEN = 48 : ' LOGICAL RECORD LENGTH
10270 PLEN = 256 : ' PHYSICAL RECORD LENGTH
10280 BF = PLEN ¥ LLEN : ' BLOCKING FACTOR
10290 DIM FS(MAXITEM), ITEM.NAME$(MAXITEM), ITEM.FORM$(MAXITEM)
10300 DIM P$(BF-1)
10310 GOSUB *SELECT.FILE
10320 GOSUB *INIT.JOB

```

```

10330 '-----
10340 FOR CND=0 TO -1 STEP -1      :'REPEAT
10350 CLS
10360 PRINT "-----"
10370 PRINT "| Change of data      ***** C"
10380 PRINT "|
10390 PRINT "| Append and Preparation of data ***** A"
10400 PRINT "|
10410 PRINT "| Delete of data      ***** D ( Attention )"
10420 PRINT "|
10430 PRINT "| List ( in Display ) ***** L"
10440 PRINT "|
10450 PRINT "| List ( in Printer ) ***** P"
10460 PRINT "|
10470 PRINT "| End      ***** E"
10480 PRINT "-----"
10490   C$=INPUT$(1)
10500   C = ( INSTR("CcAaDdLlPpEe", C$)+1) ¥2
10510   ON C GOSUB *CHANGE ,*APPEND , *DEL      , *DISPLAY ,*PRINTER
10520   CND = ( C = 6 )
10530 NEXT CND      :'UNTIL "e" or "E" is pressed
10540 GOSUB *END.JOB
10550 RUN "menuuu.PRO"
11000 '-----
11010 *CHANGE
11020 GOSUB *GET.REC : IF OUT.RANGE THEN RETURN
11030 GOSUB *ITEM.LIST
11040 INPUT "Number of article = ";N
11050 N=N-1
11060 IF (N>MAXITEM) OR (N<0) THEN RETURN
11070 GOSUB *READ.ITEM
11080 PUT #1, PREC
11090 GOTO *DISP.DATA
11100 '-----
11110 *APPEND
11120 CLS
11130 PRINT "Put data ( New or Append )"
11140 GOSUB *NEW.REC
11150 PRINT USING "(###)" ; LREC;
11160 GOSUB *ITEM.LIST
11170 FOR N = 0 TO MAXITEM
11180 GOSUB *READ.ITEM
11190 NEXT N
11200 FREE = CVI(FP$)
11210 RSET FP$ = MKI$( IN.USE ) :PUT #1, PREC
11220 GOTO *DISP.DATA
11230 '-----
11240 *DEL
11250 GOSUB *GET.REC : IF OUT.RANGE THEN RETURN
11260 RSET FP$ = MKI$( FREE ) : PUT #1,PREC
11270 FREE = LREC
11280 RETURN
11290 '-----
11300 *DISPLAY
11310 GOSUB *GET.REC : IF ( LREC <> 0 ) THEN RETURN
11320 LREC = 0
11330 FOR PREC = 1 TO LOF (1)
11340 GET #1, PREC
11350 FOR K = 0 TO BF-1
11360 FIELD #1,K*LLEN AS DUMMY$, 2 AS FP$
11370 FOR I= 0 TO MAXITEM
11380 FIELD #1, K*LLEN+(I+1)*2 AS DUMMY$, 2 AS F$(I)
11390 NEXT I
11400 IF ( LREC <> 0 ) THEN GOSUB *DISP.DTDOS
11410 LREC = LREC + 1
11420 NEXT K
11430 NEXT PREC
11440 PRINT "If end put Y or y"
11450 I$ = INKEY$ ; IF I$ = "" THEN GOTO 11450
11460 IF I$="Y" OR I$="y" THEN GOTO 11480
11470 GOTO 11440
11480 RETURN

```



```

11490 !-----
11500 *GET.REC
11510 INPUT "Number of data = " ;LREC
11520 PREC = 1 + LREC ¥ BF : COLM = LREC MOD BF
11530 OUT.RANGE = ( PREC > LOF(1) ) OR ( LREC < 1 )
11540 IF OUT.RANGE THEN RETURN
11550 GET #1, PREC
11560 FIELD #1, COLM*LLEN AS DUMMYS, 2 AS FPs
11570 FOR I=0 TO MAXITEM
11580 FIELD #1, COLM*LLEN+(I+1)*2 AS DUMMYS, 2 AS F$(I)
11590 NEXT I
11600 !-----
11610 *DISP.DATA
11620 IF( CVI(FP$) <> IN.USE ) THEN OUT.RANGE = TRUE : RETURN
11630 PRINT USING "(###) "; LREC;
11640 FOR I= 0 TO MAXITEM
11650 ON I+1 GOTO 11670
11660 PRINT USING ITEM.FORM$(I); CVI(F$(I))/10; : GOTO 11680
11670 PRINT USING ITEM.FORM$(I); CVI(F$(I)); : GOTO 11680
11680 NEXT I
11690 PRINT : FOR I=1 TO 6000 : NEXT I
11700 RETURN
11710 !-----
11720 *DISP.DTDOS
11730 IF( CVI(FP$) <> IN.USE ) THEN OUT.RANGE = TRUE : RETURN
11740 PRINT USING "(###) "; LREC;
11750 FOR I= 0 TO MAXITEM
11760 ON I+1 GOTO 11780
11770 PRINT USING ITEM.FORM$(I); CVI(F$(I))/10; : GOTO 11790
11780 PRINT USING ITEM.FORM$(I); CVI(F$(I)); : GOTO 11790
11790 NEXT I
11800 PRINT
11810 RETURN
11820 !-----
11830 *PRINTER
11840 GOSUB *GET.REC : IF ( LREC <> 0 ) THEN RETURN
11850 LREC = 0
11860 FOR PREC = 1 TO LOF (1)
11870 GET #1, PREC
11880 FOR K = 0 TO BF-1
11890 FIELD #1,K*LLEN AS DUMMYS, 2 AS FPs
11900 FOR I=0 TO MAXITEM
11910 FIELD #1,K*LLEN+(I+1)*2 AS DUMMYS, 2 AS F$(I)
11920 NEXT I
11930 IF ( LREC <> 0 ) THEN GOSUB *PRINT.DATA
11940 LREC = LREC + 1
11950 NEXT K
11960 NEXT PREC
11970 RETURN
11980 !-----
11990 *PRINT.DATA
12000 IF( CVI(FP$) <> IN.USE ) THEN OUT.RANGE = TRUE : RETURN
12010 LPRINT USING "(###) "; LREC;
12020 FOR I= 0 TO MAXITEM
12030 ON I+1 GOTO 12050
12040 LPRINT USING ITEM.FORM$(I); CVI(F$(I))/10; : GOTO 12060
12050 LPRINT USING ITEM.FORM$(I); CVI(F$(I)); : GOTO 12060
12060 NEXT I
12070 LPRINT
12080 RETURN
12090 !-----
12100 *READ.ITEM
12110 PRINT "(; N+1; ) "; ITEM.NAMES(N); " = ";
12120 INPUT D$
12130 IF D$="0" THEN GOSUB *ZERO
12140 ON N+1 GOTO 12160
12150 LSET F$(N) = MKI$( VAL(D$)*10) : RETURN
12160 LSET F$(N) = MKI$( VAL(D$)) : RETURN
12170 !-----
12180 *ZERO
12190 LSET F$(N) = MKI$( VAL(D$))
12200 FOR I=N TO MAXITEM-2 : N=N+1

```

```

12210         PRINT "("; N+1; ")" "; ITEM.NAME$(N); " = 0"
12220         LSET F$(N) = MKI$( VAL(D$))
12230         NEXT I :RETURN
-----
12240
12250 *NEW.REC
12260     IF( FREE = NIL ) THEN *ADD.NEW.REC
12270     LREC = FREE
12280     PREC = 1 + LREC % BF : COLM = LREC MOD BF
12290     GET #1, PREC
12300     FIELD #1, COLM*LLEN AS DUMMY$, 2 AS FP$
12310     FOR I=0 TO MAXITEM
12320         FIELD #1, COLM*LLEN +(I+1)*2 AS DUMMY$, 2 AS F$(I)
12330     NEXT I
12340     FREE = CVI( FP$)
12350     RETURN
-----
12360
12370 *ADD.NEW.REC
12380     LREC = LOF(1) * BF
12390     PREC = LOF(1) + 1
12400     FIELD #1, 2 AS P$(0), 46 AS D0$, 2 AS P$(1), 46 AS D1$, 2 AS P$(2), 46
12410     AS D2$, 2 AS P$(3), 46 AS D3$, 2 AS P$(4)
12420     LSET P$( BF-1) = MKI$( NIL)
12430     FOR I = BF-2 TO 0 STEP -1
12440         LSET P$(I) = MKI$( LREC + 1 + 1 )
12450     NEXT I
12460     FREE = LREC
12470     FIELD #1, 2 AS FP$
12480     FOR I=0 TO MAXITEM
12490         FIELD #1, (I+1)*2 AS DUMMY$, 2 AS F$(I)
12500     NEXT I
12510     RETURN
-----
12520 *ITEM.LIST
12530     FOR I = 0 TO MAXITEM
12540         IF( (I MOD 5) = 0 ) THEN PRINT :PRINT TAB(7);
12550         PRINT USING "##)8      &"; I+1; ITEM.NAME$(I);
12560     NEXT I
12570     PRINT
12580     RETURN
-----
12590
12600 *INIT.JOB
12610     FOR I = 0 TO MAXITEM
12620         READ ITEM.NAME$(I)
12630     NEXT
12640     FOR I = 0 TO MAXITEM
12650         READ ITEM.FORM$(I)
12660     NEXT
12670     IF( LOF(1) = 0 ) THEN *PUT.1ST.REC
12680     FIELD #1, 2 AS FP$
12690     GET #1, 1 : FREE = CVI( FP$ )
12700     RETURN
-----
12710
12720 *PUT.1ST.REC
12730     FIELD #1, 2 AS FP$, 46 AS D0$, 2 AS P$(1), 46 AS D1$, 2 AS P$(2), 46 AS
12740     D2$, 2 AS P$(3), 46 AS D3$, 2 AS P$(4)
12750     LSET P$( BF-1) = MKI$(NIL)
12760     FOR I = BF-2 TO 1 STEP -1
12770         LSET P$(I) = MKI$( I+1)
12780     NEXT I
12790     FREE = 1
12800     PUT #1, 1
12810     RETURN
-----
12820 *END.JOB
12830     FIELD #1, 2 AS FP$
12840     GET #1 ,1
12850     RSET FP$ = MKI$( FREE ) : PUT #1,1
12860     CLOSE #1
12870     RETURN
-----
12880
12890 DATA "Number.," "Dim. 0m", "Dim. 2m", "Dim. 4m", "Dim. 6m", "Dim. 8m", "Dim.10m"
12900 DATA "Dim.12m", "Dim.14m", "Dim.16m", "Dim.18m", "Dim.20m", "Dim.22m", "Dim.24m"

```

```

12910 DATA "Dim.26m","Dim.28m","Dim.30m","Dim.32m","Dim.34m","Dim.36m","Dim.38m"
12920 DATA "Dim.40m","Lon.Lst"
12930 DATA "#####", "###.##", "###.##", "###.##", "###.##", "###.##", "###.##", "###.##"
12940 DATA "###.##", "###.##", "###.##", "###.##", "###.##", "###.##", "###.##", "###.##"
12950 DATA "###.##", "###.##", "###.##", "###.##", "###.##", "###.##", "###.##", "###.##"
13000 '-----
13010 *SELECT.FILE
13020     COLOR 5
13030 PRINT " "
13040 PRINT "| Do you want to create new file ? ***** 1 |"
13050 PRINT " "
13060 PRINT "| Do you want to correct data ? ***** 2 |"
13070 PRINT " "
13080     COLOR 4
13090 PRINT " Input No. of work " : INPUT X
13100 IF X=1 THEN GOSUB *CREATE.FILE : RETURN
13110 IF X=2 THEN GOSUB *CORRECT.DATA : RETURN
13120 GOTO 13030.
13130 '-----
13140 *CREATE.FILE
13150 PRINT " "
13160 PRINT "| The data that you have in drive 2 are as follows. |"
13170 PRINT " "
13180     PRINT "( Drive 2 )" : FILES 2
13190     COLOR 7
13200 PRINT " "
13210 PRINT "| Input new file name that you want to create. |"
13220 PRINT " "
13230     INPUT FILE$
13240     X$="2:"
13250     FILENAMES=X$+FILE$
13260     IF ( FILE$ = "" ) THEN GOTO 13200
13270     OPEN FILENAMES AS #1
13280     IF LOF(1)>>0 THEN GOTO 13350
13290 PRINT " "
13300 PRINT "| Are you sure create new file ? ( y / n ) |"
13310 PRINT " "
13320     INPUT B$
13330     IF NOT ((B$ = "Y") OR (B$ = "y")) THEN *GOBACK
13340     RETURN
13350     FOR I=1 TO 500 : BEEP 1 : BEEP 0 : NEXT I
13360 PRINT " "
13370 PRINT "| You have already same name file. |"
13380 PRINT "| Computer kill the file, and then create new file . |"
13390 PRINT "| Are you sure ? ( y / n ) |"
13400 PRINT " "
13410     INPUT A$
13420     IF NOT ((A$ = "Y") OR (A$ = "y")) THEN *BACK
13430     CLOSE : KILL FILENAMES
13440     OPEN FILENAMES AS #1 : RETURN
13450 '-----
13460 *CORRECT.DATA
13470 PRINT " "
13480 PRINT "| The data that you have in drive 2 are as follows. |"
13490 PRINT " "
13500     PRINT "( Drive 2 )" : FILES 2
13510     COLOR 7
13520 PRINT " "
13530 PRINT "| Input file name that you want to correct |"
13540 PRINT " "
13550     INPUT FILE$
13560     X$="2:"
13570     FILENAMES=X$+FILE$
13580     IF ( FILE$ = "" ) THEN RETURN
13590     OPEN FILENAMES AS #1
13600     IF LOF(1)>>0 THEN RETURN
13610     FOR I=1 TO 500 : BEEP 1 : BEEP 0 : NEXT I
13620 PRINT " "
13630 PRINT "| You have not file yet. Return to first!!! |"
13640 PRINT " "
13650 GOTO *GOBACK
13660 '-----

```

```

13670 *GOBACK
13680 CLOSE
13690 KILL FILENAME$
13700 PRINT "Back to menu!!!"
13710 FOR I=1 TO 6000
13720 NEXT I
13730 RUN "menuuu.PRO"
-----
13740
13750 *BACK
13760 CLOSE
13770 PRINT "Back to menu!!!"
13780 FOR I=1 TO 6000
13790 NEXT I
13800 RUN "menuuu.PRO"

```

(4) espcor. PRO

```

10000 '==== espcor.PRO =====
10010 '
10020 ' PROGRAM save "espcor.PRO"
10030 '
10040 ' Preparation of data file , bark thick of each 2m
10050 '
10060 ' =====
10070 '
10080 ' WIDTH 80,25 : CONSOLE 0,25,0,1
10090 PRINT " "
10100 PRINT "| Preparation , Correction and Observacion of data file |"
10110 PRINT "| " |"
10120 PRINT "| Data are bark thick of 2m diameter |"
10130 PRINT "| " |"
10140 PRINT " "
10150 PRINT " "
10160 PRINT "| *** Enter floppy disk in Drive Number 2 *** |"
10170 PRINT "| *** If you enter floppy disk, Put Y or y ** |"
10180 PRINT " "
10190 I$ = INKEY$ : IF I$ = "" THEN GOTO 10190
10200 IF I$="Y" OR I$="y" THEN GOTO 10220
10210 GOTO 10150
-----
10220
10230 CLS
10240 MAXITEM =21
10250 TRUE =-1 : IN.USE = -1 : NIL = 0
10260 LLEN = 46 : LOGICAL RECORD LENGTH
10270 PLEN = 256 : PHYSICAL RECORD LENGTH
10280 BF = PLEN ¥ LLEN : BLOCKING FACTOR
10290 DIM F$(MAXITEM), ITEM.NAME$(MAXITEM), ITEM.FORM$(MAXITEM)
10300 DIM P$(BF-1)
10310 GOSUB *SELECT.FILE
10320 GOSUB *INIT.JOB
-----
10330
10340 FOR CND=0 TO -1 STEP -1 : REPEAT
10350 CLS
10360 PRINT " "
10370 PRINT "| Change of data ***** C |"
10380 PRINT "| " |"
10390 PRINT "| Append and Preparation of data ***** A |"
10400 PRINT "| " |"
10410 PRINT "| Delete of data ***** D ( Attention ) |"
10420 PRINT "| " |"
10430 PRINT "| List ( in Display ) ***** L |"
10440 PRINT "| " |"
10450 PRINT "| List ( in Printer ) ***** P |"
10460 PRINT "| " |"
10470 PRINT "| End ***** E |"
10480 PRINT " " |"
10490 C$=INPUT$(1)
10500 C = ( INSTR("CcAaDdLlPpEe", C$)+1) ¥2
10510 ON C GOSUB *CHANGE , *APPEND , *DEL , *DISPLAY , *PRINTER
10520 CND = ( C = 6 )
10530 NEXT CND : UNTIL "e" or "E" is pressed

```

```

10540 GOSUB *END.JOB
10550 RUN "menuuu.PRO
11000 -----
11010 *CHANGE
11020 GOSUB *GET.REC : IF OUT.RANGE THEN RETURN
11030 GOSUB *ITEM.LIST
11040 INPUT "Number of article = ";N
11050 N=N-1
11060 IF (N>MAXITEM) OR (N<0) THEN RETURN
11070 GOSUB *READ.ITEM
11080 PUT #1, PREC
11090 GOTO *DISP.DATA
11100 -----
11110 *APPEND
11120 CLS
11130 PRINT "Put data ( New or Append )"
11140 GOSUB *NEW.REC
11150 PRINT USING "(###)" ; LREC;
11160 GOSUB *ITEM.LIST
11170 FOR N = 0 TO MAXITEM
11180 GOSUB *READ.ITEM
11190 NEXT N
11200 FREE = CVI(FP$)
11210 RSET FP$ = MKI$( IN.USE ) :PUT #1, PREC
11220 GOTO *DISP.DATA
11230 -----
11240 *DEL
11250 GOSUB *GET.REC : IF OUT.RANGE THEN RETURN
11260 RSET FP$ = MKI$( FREE ) : PUT #1,PREC
11270 FREE = LREC
11280 RETURN
11290 -----
11300 *DISPLAY
11310 GOSUB *GET.REC : IF ( LREC <> 0 ) THEN RETURN
11320 LREC = 0
11330 FOR PREC = 1 TO LOF (1)
11340 GET #1, PREC
11350 FOR K = 0 TO BF-1
11360 FIELD #1,K*LLEN AS DUMMY$, 2 AS FP$
11370 FOR I= 0 TO MAXITEM
11380 FIELD #1, K*LLEN+(I+1)*2 AS DUMMY$, 2 AS F$(I)
11390 NEXT I
11400 IF ( LREC <> 0 ) THEN GOSUB *DISP.DTDS
11410 LREC = LREC + 1
11420 NEXT K
11430 NEXT PREC
11440 PRINT "If end put Y or y "
11450 I$ = INKEYS : IF I$ = "" THEN GOTO 11450
11460 IF I$="Y" OR I$="y" THEN GOTO 11480
11470 GOTO 11440
11480 RETURN
11490 -----
11500 *GET.REC
11510 INPUT "Number of data = " ;LREC
11520 PREC = 1 + LREC ¥ BF : COLM = LREC MOD BF
11530 OUT.RANGE = ( PREC > LOF(1) ) OR ( LREC < 1 )
11540 IF OUT.RANGE THEN RETURN
11550 GET #1, PREC
11560 FIELD #1, COLM*LLEN AS DUMMY$, 2 AS FP$
11570 FOR I=0 TO MAXITEM
11580 FIELD #1, COLM*LLEN+(I+1)*2 AS DUMMY$, 2 AS F$(I)
11590 NEXT I
11600 -----
11610 *DISP.DATA
11620 IF( CVI(FP$) <> IN.USE ) THEN OUT.RANGE = TRUE : RETURN
11630 PRINT USING "(###)" ; LREC;
11640 FOR I= 0 TO MAXITEM
11650 PRINT USING ITEM.FORM$(I); CVI(F$(I));
11660 NEXT I
11670 PRINT : FOR I=1 TO 6000 : NEXT I
11680 RETURN
11690 -----

```

```

11700 *DISP.DTDOS
11710     IF( CVI(FP$) <> IN.USE ) THEN OUT.RANGE = TRUE : RETURN
11720     PRINT USING "(###) "; LREC;
11730     FOR I= 0 TO MAXITEM
11740         PRINT USING ITEM.FORM$(I); CVI(F$(I));
11750     NEXT I
11760     PRINT
11770     RETURN
-----
11780
11790 *PRINTER
11800     GOSUB *GET.REC : IF ( LREC <> 0 ) THEN RETURN
11810     LREC = 0
11820     FOR PREC = 1 TO LOF (1)
11830         GET #1, PREC
11840         FOR K = 0 TO BF-1
11850             FIELD #1,K*LLEN AS DUMMY$, 2 AS FP$
11860             FOR I=0 TO MAXITEM
11870                 FIELD #1, K*LLEN+(I+1)*2 AS DUMMY$, 2 AS F$(I)
11880             NEXT I
11890             IF ( LREC <> 0 ) THEN GOSUB *PRINT.DATA
11900             LREC = LREC + 1
11910         NEXT K
11920     NEXT PREC
11930     RETURN
-----
11940
11950 *PRINT.DATA
11960     IF( CVI(FP$) <> IN.USE ) THEN OUT.RANGE = TRUE : RETURN
11970     LPRINT USING "(###) "; LREC;
11980     FOR I= 0 TO MAXITEM
11990         LPRINT USING ITEM.FORM$(I); CVI(F$(I));
12000     NEXT I
12010     LPRINT
12020     RETURN
-----
12030
12040 *READ.ITEM
12050     PRINT "(:; N+1; " ) "; ITEM.NAME$(N); " = ";
12060     INPUT D$
12070     IF D$="0" THEN GOSUB *ZERO
12080     LSET F$(N) = MKI$( VAL(D$)) : RETURN
-----
12090
12100 *ZERO
12110     LSET F$(N) = MKI$( VAL(D$))
12120     FOR I=N TO MAXITEM-1 : N=N+1
12130     PRINT "(:; N+1; " ) "; ITEM.NAME$(N); " = 0"
12140     LSET F$(N) = MKI$( VAL(D$))
12150     NEXT I :RETURN
-----
12160
12170 *NEW.REC
12180     IF( FREE = NIL ) THEN *ADD.NEW.REC
12190     LREC = FREE
12200     PREC = 1 + LREC ¥ BF : COLM = LREC MOD BF
12210     GET #1, PREC
12220     FIELD #1, COLM*LLEN AS DUMMY$, 2 AS FP$
12230     FOR I=0 TO MAXITEM
12240         FIELD #1, COLM*LLEN *(I+1)*2 AS DUMMY$, 2 AS F$(I)
12250     NEXT I
12260     FREE = CVI( FP$)
12270     RETURN
-----
12280
12290 *ADD.NEW.REC
12300     LREC = LOF(1) * BF
12310     PREC = LOF(1) + 1
12320     FIELD #1, 2 AS P$(0), 44 AS D0$, 2 AS P$(1), 44 AS D1$, 2 AS P$(2), 44
AS D2$, 2 AS P$(3), 44 AS D3$, 2 AS P$(4)
12330     LSET P$( BF-1) = MKI$( NIL)
12340     FOR I = BF-2 TO 0 STEP -1
12350         LSET P$(I) = MKI$( LREC + I + 1 )
12360     NEXT I
12370     FREE = LREC
12380     FIELD #1, 2 AS FP$
12390     FOR I=0 TO MAXITEM
12400         FIELD #1, (I+1)*2 AS DUMMY$, 2 AS F$(I)

```



```

13250 FILENAME$=X$+FILE$
13260 IF ( FILE$ = "" ) THEN GOTO 13200
13270 OPEN FILENAME$ AS #1
13280 IF LOF(1)<>0 THEN GOTO 13350
13290 PRINT " "
13300 PRINT "| Are you sure create new file ? ( y / n ) "
13310 PRINT " "
13320 INPUT B$
13330 IF NOT ((B$ = "Y") OR (B$ = "y")) THEN *GOBACK
13340 RETURN
13350 FOR I=1 TO 500 : BEEP 1 : BEEP 0 : NEXT I
13360 PRINT " "
13370 PRINT "| You have already same name file. "
13380 PRINT "| Computer kill the file, and then create new file . "
13390 PRINT "| Are you sure ? ( y / n ) "
13400 PRINT " "
13410 INPUT A$
13420 IF NOT ((A$ = "Y") OR (A$ = "y")) THEN *BACK
13430 CLOSE : KILL FILENAME$
13440 OPEN FILENAME$ AS #1 :RETURN
13450 -----
13460 *CORRECT DATA
13470 PRINT " "
13480 PRINT "| The data that you have in drive 2 are as follows. "
13490 PRINT " "
13500 PRINT "( Drive 2 ) : FILES 2
13510 COLOR 7
13520 PRINT " "
13530 PRINT "| Input file name that you want to correct "
13540 PRINT " "
13550 INPUT FILES
13560 X$="2:"
13570 FILENAME$=X$+FILE$
13580 IF ( FILE$ = "" ) THEN RETURN
13590 OPEN FILENAME$ AS #1
13600 IF LOF(1)<>0 THEN RETURN
13610 FOR I=1 TO 500 : BEEP 1 : BEEP 0 : NEXT I
13620 PRINT " "
13630 PRINT "| You have not file yet. Return to first!!! "
13640 PRINT " "
13650 GOTO *GOBACK
13660 -----
13670 *GOBACK
13680 CLOSE
13690 KILL FILENAME$
13700 PRINT "Back to menu!!!"
13710 FOR I=1 TO 6000
13720 NEXT I
13730 RUN "menuuu.PRO"
13740 -----
13750 *BACK
13760 CLOSE
13770 PRINT "Back to menu!!!"
13780 FOR I=1 TO 6000
13790 NEXT I
13800 RUN "menuuu.PRO"

```

(5) ramade.PRO

```

10000 '==== ramade.PRO =====
10010 '
10020 ' PROGRAM save "ramade.PRO"
10030 '
10040 ' Preparation of data file of branch
10050 '
10060 ' =====
10070 '

```



```

10080 WIDTH 80,25 : CONSOLE 0,35,0,1
10090 PRINT "
10100 PRINT "| Preparation , Correction and Observacion of data file |"
10110 PRINT "|
10120 PRINT "| Data are branches |"
10130 PRINT "
10140 PRINT
10150 PRINT "
10160 PRINT "| *** Enter floppy disk in Drive Number 2 *** |"
10170 PRINT "| *** If you enter floppy disk, Put Y or y ** |"
10180 PRINT "
10190 I$ = INKEY$ : IF I$ = "" THEN GOTO 10190
10200 IF I$="Y" OR I$="y" THEN GOTO 10220
10210 GOTO 10150
10220
-----
10230 CLS
10240 MAXITEM =62
10250 TRUE =-1 : IN.USE = -1 : NIL = 0
10260 LLEN = 128 : ' LOGICAL RECORD LENGTH
10270 PLEN = 256 : ' PHYSICAL RECORD LENGTH
10280 BF = PLEN / LLEN : 'BLOCKING FACTOR
10290 DIM F$(MAXITEM), ITEM.NAMES$(MAXITEM), ITEM.FORM$(MAXITEM)
10300 DIM P$(BF-1)
10310 GOSUB *SELECT.FILE
10320 GOSUB *INIT.JOB
10330
-----
10340 FOR CND=0 TO -1 STEP -1 : 'REPEAT
10350 CLS
10360 PRINT "
10370 PRINT "| Change of data ***** C |"
10380 PRINT "|
10390 PRINT "| Append and Preparation of data ***** A |"
10400 PRINT "|
10410 PRINT "| Delete of data ***** D ( Attention ) |"
10420 PRINT "|
10430 PRINT "| List ( in Display ) ***** L |"
10440 PRINT "|
10450 PRINT "| List ( in Printer ) ***** P |"
10460 PRINT "|
10470 PRINT "| End ***** E |"
10480 PRINT "
10490 C$=INPUT$(1)
10500 C = ( INSTR("CcAaDdLlPpEe", C$)+1) / 2
10510 ON C GOSUB *CHANGE , *APPEND , *DEL , *DISPLAY , *PRINTER
10520 CND = ( C = 6 )
10530 NEXT CND : 'UNTIL "e" or "E" is pressed
10540
10550 GOSUB *END.JOB
10560 RUN "menuuu.PRO"
-----
11000
11010 *CHANGE
11020 GOSUB *GET.REC : IF OUT.RANGE THEN RETURN
11030 GOSUB *ITEM.LIST
11040 INPUT "Number of article = ";N
11050 N=N-1
11060 IF (N>MAXITEM) OR (N<0) THEN RETURN
11070 GOSUB *READ.ITEM
11080 PUT #1, PREC
11090 GOTO *DISP.DATA
-----
11100
11110 *APPEND
11120 CLS
11130 PRINT "Put data ( New or Append ) "
11140 GOSUB *NEW.REC
11150 PRINT USING "(###)" ; LREC ;
11160 GOSUB *ITEM.LIST
11170 FOR N = 0 TO MAXITEM
11180 GOSUB *READ.ITEM
11190 NEXT N
11200 FREE = CVI(FP$)
11210 RSET FP$ = MKIS( IN.USE ) :PUT #1, PREC
11220 GOTO *DISP.DATA

```

```

11230 -----
11240 *DEL
11250 GOSUB *GET.REC : IF OUT.RANGE THEN RETURN
11260 RSET FP$ = MKI$( FREE ) : PUT #1,PREC
11270 FREE = LREC
11280 RETURN
11290 -----
11300 *DISPLAY
11310 GOSUB *GET.REC : IF ( LREC <> 0 ) THEN RETURN
11320 LREC = 0
11330 FOR PREC = 1 TO LOF (1)
11340 GET #1, PREC
11350 FOR K = 0 TO BF-1
11360 FIELD #1,K*LLEN AS DUMMY$, 2 AS FP$
11370 FOR I= 0 TO MAXITEM
11380 FIELD #1, K*LLEN+(I+1)*2 AS DUMMY$, 2 AS F$(I)
11390 NEXT I
11400 IF ( LREC <> 0 ) THEN GOSUB *DISP.DTDO$
11410 LREC = LREC + 1
11420 NEXT K
11430 NEXT PREC
11440 PRINT "If end put Y or y "
11450 I$ = INKEY$ : IF I$ = "" THEN GOTO 11450
11460 IF I$="Y" OR I$="y" THEN GOTO 11480
11470 GOTO 11440
11480 RETURN
11490 -----
11500 *GET.REC
11510 INPUT "Number of data = " ;LREC
11520 PREC = 1 + LREC % BF : COLM = LREC MOD BF
11530 OUT.RANGE = ( PREC > LOF(1) ) OR ( LREC < 1 )
11540 IF OUT.RANGE THEN RETURN
11550 GET #1, PREC
11560 FIELD #1, COLM*LLEN AS DUMMY$, 2 AS FP$
11570 FOR I=0 TO MAXITEM
11580 FIELD #1, COLM*LLEN+(I+1)*2 AS DUMMY$, 2 AS F$(I)
11590 NEXT I
11600 -----
11610 *DISP.DATA
11620 IF( CVI(FP$) <> IN.USE ) THEN OUT.RANGE = TRUE : RETURN
11630 PRINT USING "(###) "; LREC;
11640 FOR I= 0 TO MAXITEM
11650 ON I+1 GOTO 11670
11660 PRINT USING ITEM.FORM$(I); CVI(F$(I))/10; : GOTO 11680
11670 PRINT USING ITEM.FORM$(I); CVI(F$(I)); : GOTO 11680
11680 NEXT I
11690 PRINT : FOR I=1 TO 6000 : NEXT I
11700 RETURN
11710 -----
11720 *DISP.DTDO$
11730 IF( CVI(FP$) <> IN.USE ) THEN OUT.RANGE = TRUE : RETURN
11740 PRINT USING "(###) "; LREC;
11750 FOR I= 0 TO MAXITEM
11760 ON I+1 GOTO 11780
11770 PRINT USING ITEM.FORM$(I); CVI(F$(I))/10; : GOTO 11790
11780 PRINT USING ITEM.FORM$(I); CVI(F$(I)); : GOTO 11790
11790 NEXT I
11800 PRINT
11810 RETURN
11820 -----
11830 *PRINTER
11840 GOSUB *GET.REC : IF ( LREC <> 0 ) THEN RETURN
11850 LREC = 0
11860 FOR PREC = 1 TO LOF (1)
11870 GET #1, PREC
11880 FOR K = 0 TO BF-1
11890 FIELD #1,K*LLEN AS DUMMY$, 2 AS FP$
11900 FOR I=0 TO MAXITEM
11910 FIELD #1,K*LLEN+(I+1)*2 AS DUMMY$, 2 AS F$(I)
11920 NEXT I
11930 IF ( LREC <> 0 ) THEN GOSUB *PRINT.DATA
11940 LREC = LREC + 1

```

```

11950         NEXT K
11960         NEXT PREC
11970         RETURN
11980 -----
11990 *PRINT DATA
12000         IF( CVI(FP$) <> IN.USE ) THEN OUT.RANGE = TRUE : RETURN
12010         LPRINT USING "(###) "; LREC;
12020         FOR I= 0 TO MAXITEM
12030             ON I+1 GOTO 12050
12040             LPRINT USING ITEM.FORM$(I); CVI(F$(I))/10; : GOTO 12060
12050             LPRINT USING ITEM.FORM$(I); CVI(F$(I)); : GOTO 12060
12060         NEXT I
12070         LPRINT
12080         RETURN
12090 -----
12100 *READ ITEM
12110         PRINT "( "; N+1; " ) "; ITEM.NAME$(N); " = ";
12120         INPUT D$
12130         IF D$="0" THEN GOSUB *ZERO
12140         ON N+1 GOTO 12160
12150         LSET F$(N) = MKI$( VAL(D$)*10) : RETURN
12160         LSET F$(N) = MKI$( VAL(D$)) : RETURN
12170 -----
12180 *ZERO
12190         LSET F$(N) = MKI$( VAL(D$))
12200         FOR I=N TO MAXITEM-1 : N=N+1
12210         PRINT "( "; N+1; " ) "; ITEM.NAME$(N); " = 0"
12220         LSET F$(N) = MKI$( VAL(D$))
12230         NEXT I :RETURN
12240 -----
12250 *NEW.REC
12260         IF( FREE = NIL ) THEN *ADD.NEW.REC
12270         LREC = FREE
12280         PREC = 1 + LREC ¥ BF : COLM = LREC MOD BF
12290         GET #1, PREC
12300         FIELD #1, COLM*LLEN AS DUMMY$, 2 AS FP$
12310         FOR I=0 TO MAXITEM
12320             FIELD #1, COLM*LLEN +(I+1)*2 AS DUMMY$, 2 AS F$(I)
12330         NEXT I
12340         FREE = CVI( FP$)
12350         RETURN
12360 -----
12370 *ADD.NEW.REC
12380         LREC = LOF(1) * BF
12390         PREC = LOF(1) + 1
12400         FIELD #1, 2 AS P$(0), 126 AS D0$, 2 AS P$(1)
12410         LSET P$( BF-1) = MKI$( NIL)
12420         FOR I = BF-2 TO 0 STEP -1
12430             LSET P$(I) = MKI$( LREC + I + 1 )
12440         NEXT I
12450         FREE = LREC
12460         FIELD #1, 2 AS FP$
12470         FOR I=0 TO MAXITEM
12480             FIELD #1, (I+1)*2 AS DUMMY$, 2 AS F$(I)
12490         NEXT I
12500         RETURN
12510 -----
12520 *ITEM.LIST
12530         FOR I = 0 TO MAXITEM
12540             IF( (I MOD 5) = 0 ) THEN PRINT :PRINT TAB(7);
12550             PRINT USING "##)&      8"; I+1; ITEM.NAME$(I);
12560         NEXT I
12570         PRINT
12580         RETURN
12590 -----
12600 *INIT.JOB
12610         READ ITEM.NAME$(0)
12620         FOR I = 1 TO MAXITEM-2 STEP 2
12630             READ ITEM.NAME$(I)
12640             READ ITEM.NAME$(I+1)
12650             RESTORE 12990
12660         NEXT

```

```

12670          READ ITEM.NAMES$(MAXITEM-1)
12680          READ ITEM.NAMES$(MAXITEM)
12690          READ ITEM.FORM$(0)
12700          FOR I = 1 TO MAXITEM STEP 2
12710              READ ITEM.FORM$(I)
12720              READ ITEM.FORM$(I+1)
12730              RESTORE 13010
12740          NEXT
12750
12760          IF ( LOF(1) = 0 ) THEN *PUT.1ST.REC
12770              FIELD #1, 2 AS FP$
12780              GET #1, 1 : FREE = CVI( FP$ )
12790              RETURN
-----
12810 *PUT.1ST.REC
12820     FIELD #1, 2 AS FP$, 126 AS DO$, 2 AS P$(1)
12830     LSET P$( BF-1) = MKI$(NIL)
12840     FOR I = BF-2 TO 1 STEP -1
12850         LSET P$(I) = MKI$( I+1)
12860     NEXT I
12870     FREE = 1
12880     PUT #1, 1
12890     RETURN
-----
12910 *END.JOB
12920     FIELD #1, 2 AS FP$
12930     GET #1 ,1
12940     RSET FP$ = MKI$( FREE ) : PUT #1,1
12950     CLOSE #1
12960     RETURN
-----
12980 DATA "Number."
12990 DATA "Dim.base","Length.."
13000 DATA "####"
13010 DATA "###.#","###.#"
14000 '-----
14010 *SELECT.FILE
14020     COLOR 5
14030 PRINT " "
14040 PRINT "| Do you want to create new file ? ***** 1 |"
14050 PRINT "| "
14060 PRINT "| Do you want to correct data ? ***** 2 |"
14070 PRINT " "
14080     COLOR 4
14090 PRINT " Input No. of work " : INPUT X
14100 IF X=1 THEN GOSUB *CREATE.FILE : RETURN
14110 IF X=2 THEN GOSUB *CORRECT.DATA : RETURN
14120 GOTO 14030
14130 '-----
14140 *CREATE.FILE
14150 PRINT " "
14160 PRINT "| The data that you have in drive 2 are as follows. |"
14170 PRINT " "
14180     PRINT "( Drive 2 )" : FILES 2
14190     COLOR 7
14200 PRINT " "
14210 PRINT "| Input new file name that you want to create. |"
14220 PRINT " "
14230     INPUT     FILES
14240     X$="2:"
14250     FILENAME$=X$+FILES
14260     IF ( FILES = "" ) THEN GOTO 14200
14270     OPEN FILENAME$ AS #1
14280     IF LOF(1)>>0 THEN GOTO 14350
14290 PRINT " "
14300 PRINT "| Are you sure create new file ? ( y / n ) |"
14310 PRINT " "
14320     INPUT B$
14330     IF NOT ((B$ = "Y") OR (B$ = "y")) THEN *GOBACK
14340     RETURN
14350     FOR I=1 TO 500 : BEEP 1 : BEEP 0 : NEXT I

```

```

14360 PRINT "
14370 PRINT "| You have already same name file.
14380 PRINT "| Computer kill the file, and then create new file .
14390 PRINT "| Are you sure ? ( y / n )
14400 PRINT "
14410 INPUT AS$
14420 IF NOT ((AS$ = "Y") OR (AS$ = "y")) THEN *BACK
14430 CLOSE : KILL FILENAME$
14440 OPEN FILENAME$ AS #1 :RETURN
14450 -----
14460 *CORRECT.DATA
14470 PRINT "
14480 PRINT "| The data that you have in drive 2 are as follows.
14490 PRINT "
14500 PRINT "( Drive 2 )" : FILES 2
14510 COLOR 7
14520 PRINT "
14530 PRINT "| Input file name that you want to correct
14540 PRINT "
14550 INPUT FILE$
14560 XS="2:"
14570 FILENAME$=X$+FILES
14580 IF ( FILE$ = "" ) THEN RETURN
14590 OPEN FILENAME$ AS #1
14600 IF LOF(1)<>0 THEN RETURN
14610 FOR I=1 TO 500 : BEEP 1 : BEEP 0 : NEXT I
14620 PRINT "
14630 PRINT "| You have not file yet. Return to first!!!
14640 PRINT "
14650 GOTO *GOBACK
14660 -----
14670 *GOBACK
14680 CLOSE
14690 KILL FILENAME$
14700 PRINT "Back to menu!!!"
14710 FOR I=1 TO 6000
14720 NEXT I
14730 RUN "menuuu.PRO"
14740 -----
14750 *BACK
14760 CLOSE
14770 PRINT "Back to menu!!!"
14780 FOR I=1 TO 6000
14790 NEXT I
14800 RUN "menuuu.PRO"

```

(6) sumade. PRO

```

10000 '==== sum up.PRO =====
10010 '
10020 ' PROGRAM save "sumade.PRO"
10030 '
10040 ' Preparation of data file
10050 '
10060 '====
10070 '
10080 ' WIDTH 80,25 : CONSOLE 0,25,0,1
10090 PRINT "
10100 PRINT "| Preparation of datafile of sum up
10110 PRINT "| File 1 (Spe No, DBH, C Heit Dim. T.Heit. Com.Heit. butt Heit)
10120 PRINT "| File 2 (Diameter with bark of each 2m)
10130 PRINT "| File 3 (Bark thik of each 2m)
10140 PRINT "| File 4 (Branch data)
10150 PRINT "
10160 PRINT
10170 PRINT "
10180 PRINT "| *** Enter floppy disk in Drive Number 2 ***
10190 PRINT "| *** If you enter floppy disk, Put Y or y **
10200 PRINT "
10210 I$ = INKEY$ : IF I$ = "" THEN GOTO 10210

```

```

10220             IF I$="Y" OR I$="y" THEN GOTO 10250
10230             GOTO 10170
10240 -----
10250             CLS
10260 MAX1=6      : LLEN1=26 : PLEN1=256 : BF1=PLEN1¥LLEN1
10270 MAX2=22   : LLEN2=48 : PLEN2=256 : BF2=PLEN2¥LLEN2
10280 MAX3=21   : LLEN3=46 : PLEN3=256 : BF3=PLEN3¥LLEN3
10290 MAX4=62   : LLEN4=128: PLEN4=256 : BF4=PLEN4¥LLEN4
10300 MAX5=10   : LLEN5=58 : PLEN5=256 : BF5=PLEN5¥LLEN5
10310 DEFDBL V,R,E,F
10320 DIM A$(MAX1),B$(MAX2),C$(MAX3),D$(MAX4),E$(MAX5),F$(MAX5)
10330 DIM AAA(200,MAX1),BBB(200,MAX2),CCC(200,MAX3),DDD(200,MAX4)
10340 DIM EEE(200,MAX5),FFF(200,MAX5)
10350 DIM V(200),ESP(200,MAX3),VESP(200),RAMA(200)
10360 DIM P$(BF5-1)
11000 -----
11010 GOSUB *SELECT.FILE
11020 LREC1=0 : LREC2=0 : LREC3=0 : LREC4=0 : LREC5=0 : PAI#=3.14159265358979#
11030 ----- read dbh height data -----
11040 FOR PREC1 = 1 TO LOF (1)
11050     GET #1, PREC1
11060     FOR K = 0 TO BF1-1
11070         FIELD #1,K*LLEN1 AS DUMMY$, 2 AS AP$, 2 AS A$(0), 2 AS A$(1), 4 AS
            A$(2), 4 AS A$(3), 4 AS A$(4), 4 AS A$(5), 4 AS A$(6)
11080             IF( CVI(AP$)<>-1) THEN GOTO 11150
11090             LREC1 = LREC1 + 1
11100             FOR I=0 TO MAX1
11110                 ON I+1 GOTO 11130 ,11130
11120                 AAA(LREC1,I)=CVS(A$(I)) : GOTO 11140
11130                 AAA(LREC1,I)=CVI(A$(I)) : GOTO 11140
11140             NEXT I
11150     NEXT K
11160 NEXT PREC1
11170 ----- read diameter data -----
11180 FOR PREC2 = 1 TO LOF (2)
11190     GET #2, PREC2
11200     FOR K = 0 TO BF2-1
11210         FIELD #2,K*LLEN2 AS DUMMY$, 2 AS BP$
11220         FOR I=0 TO MAX2
11230             FIELD #2,K*LLEN2+(I+1)*2 AS DUMMY$, 2 AS B$(1)
11240         NEXT I
11250             IF( CVI(BP$)<>-1) THEN GOTO 11320
11260             LREC2 = LREC2 + 1
11270             FOR I=0 TO MAX2
11280                 ON I+1 GOTO 11300
11290                 BBB(LREC2,I)=CVI(B$(I))/10 : GOTO 11310
11300                 BBB(LREC2,I)=CVI(B$(I)) : GOTO 11310
11310             NEXT I
11320     NEXT K
11330 NEXT PREC2
11340 ----- calculation -----
11350 FOR I=1 TO LREC2
11360     FOR J=1 TO 21
11370         IF BBB(I,J+2)=0 THEN GOTO 11400
11380         V(I)=V(I)+(BBB(I,J)^2+BBB(I,J+1)^2)*PAI#/40000#
11390     NEXT J
11400     V(I)=V(I)+((BBB(I,J)^2+BBB(I,J+1)^2)*PAI#)*BBB(I,22)/80000#
11410 NEXT I
11420 ----- read bark data -----
11430 FOR PREC3 = 1 TO LOF (3)
11440     GET #3, PREC3
11450     FOR K = 0 TO BF3-1
11460         FIELD #3,K*LLEN3 AS DUMMY$, 2 AS CP$
11470         FOR I=0 TO MAX3
11480             FIELD #3,K*LLEN3+(I+1)*2 AS DUMMY$, 2 AS C$(1)
11490         NEXT I
11500             IF( CVI(CP$)<>-1) THEN GOTO 11550
11510             LREC3 = LREC3 + 1

```

```

11520          FOR I=0 TO MAX3
11530          CCC(LREC3,I)=CVI(Cs(I))
11540          NEXT I
11550          NEXT K
11560          NEXT PREC3
11570          ----- calculation -----
11580          FOR I=1 TO LREC3
11590          FOR J=1 TO 21
11600          ESP(I,J)=BBB(I,J)-CCC(I,J)/10
11610          NEXT J
11620          NEXT I
11630          FOR I=1 TO LREC3
11640          FOR J=1 TO 21
11650          IF BBB(I,J+2)=0 THEN GOTO 11680
11660          VESP(I)=VESP(I)+(ESP(I,J)^2+ESP(I,J+1)^2)*PAI#/40000#
11670          NEXT J
11680          VESP(I)=VESP(I)+((ESP(I,J)^2+ESP(I,J+1)^2)*PAI#)*BBB(I,22)/80000#
11690          NEXT I
11700          ----- read branch data -----
11710          FOR PREC4 = 1 TO LOF(4)
11720          GET #4, PREC4
11730          FOR K = 0 TO BF4-1
11740          FIELD #4,K*LLEN4 AS DUMMY$, 2 AS DP$
11750          FOR I=0 TO MAX4
11760          FIELD #4,K*LLEN4+(I+1)*2 AS DUMMYS, 2 AS D$(I)
11770          NEXT I
11780          IF( CVI(DP$)<>-1) THEN GOTO 11850
11790          LREC4 = LREC4 + 1
11800          FOR I=0 TO MAX4
11810          ON I+1 GOTO 11830
11820          DDD(LREC4,I)=CVI(D$(I))/10 : GOTO 11840
11830          DDD(LREC4,I)=CVI(D$(I)) : GOTO 11840
11840          NEXT I
11850          NEXT K
11860          NEXT PREC4
11870          ----- calculation -----
11880          FOR I=1 TO LREC4
11890          FOR J=1 TO MAX4 STEP 2
11900          IF DDD(I,J)=0 THEN GOTO 11920
11910          RAMA(I)=RAMA(I)+((DDD(I,J)^2+100)*PAI#)*DDD(I,J+1)/80000#
11920          NEXT J
11930          NEXT I
12000          ----- sum up -----
12010          FOR I=1 TO LREC4
12020          EEE(I,7)=V(I)
12030          EEE(I,8)=VESP(I)
12040          EEE(I,9)=RAMA(I)
12050          EEE(I,10)=V(I)+RAMA(I)
12060          FOR J=0 TO MAX1
12070          REE(I,J)=AAA(I,J)
12080          NEXT J
12090          NEXT I
12100          -----
12110          IN.USE=-1 : NIL =0
12120          GOSUB *INIT.JOB
12130          FOR Z= 1 TO LREC1
12140          GOSUB *NEW.REC
12150          FOR N=0 TO MAX5
12160          ON N+1 GOTO 12170,12170,12180,12180,12180,12180,12180,12190,12190,
12190,12190
12170          LSET E$(N)=MKI$(EEE(Z,N)) :GOTO 12200
12180          LSET E$(N)=MKS$(EEE(Z,N)) :GOTO 12200
12190          LSET E$(N)=MKD$(EEE(Z,N)) :GOTO 12200
12200          NEXT N
12210          FREE=CVI(EP$)
12220          RSET EP$=MKIS(IN.USE) :PUT #5,PREC5
12230          NEXT Z
12240          GOSUB *END.JOB
12250          GOTO 13000
12260          -----
12270          *INIT.JOB
12280          IF (LOF(5)=0 ) THEN *PUT.1ST.REC
12290          FIELD #5, 2 AS EP$

```

```

12300 GET #5,1
12310 FREE = CVI(EP$)
12320 RETURN
12330 -----
12340 *PUT.1ST.REC
12350 FIELD #5, 2 AS EP$ ,56 AS D0$,2 AS P$(1),56 AS D1$,2 AS P$(2),56 AS
D2$,2 AS P$(3)
12360 LSET P$(BF5 -1)=MKI$(NIL)
12370 FOR I=BF5 -2 TO 1 STEP -1
12380 LSET P$(I)=MKI$(I+1)
12390 NEXT I
12400 FRER=1
12410 PUT #5,1
12420 RETURN
12430 -----
12440 *NEW.REC
12450 IF (FRER=NIL) THEN *ADD.NEW.REC
12460 LREC5=FREE : PREC5=1+LREC5*BF5 : COLM=LREC5 MOD BF5
12470 GET #5,PREC5
12480 FIELD #5, COLM*LLEN5 AS DUMMY$,2 AS EP$,2 AS E$(0),2 AS E$(1),
4 AS E$(2),4 AS E$(3),4 AS E$(4),4 AS E$(5),4 AS E$(5),8 AS E$(7)
,8 AS E$(8),8 AS E$(9),8 AS E$(10)
12490 FREE=CVI(EP$)
12500 RETURN
12510 -----
12520 *ADD.NEW.REC
12530 LREC5=LOF(5)*BF5 : PREC5=LOF(5)+1
12540 FIELD #5, 2 AS P$(0),56 AS D0$,2 AS P$(1),56 AS D1$,2 AS P$(2)
,56 AS D2$,2 AS P$(3)
12550 LSET P$(BF5 -1) =MKI$(NIL)
12560 FOR I= BF5 -2 TO 0 STEP -1
12570 LSET P$(I)=MKI$(LREC5+I+1)
12580 NEXT I
12590 FREE=LREC5
12600 FIELD #5,2 AS EP$,2 AS E$(0),2 AS E$(1),4 AS E$(2),4 AS E$(3),
4 AS E$(4),4 AS E$(5),4 AS E$(6),8 AS E$(7),8 AS E$(8),8 AS E$(9)
,8 AS E$(10)
12610 RETURN
12620 -----
12630 *END.JOB
12640 FIELD #5, 2 AS EP$
12650 GET #5,1
12660 RSET EP$ = MKI$(FREE): PUT #5,1
12670 CLOSE #1
12680 CLOSE #2
12690 CLOSE #3
12700 CLOSE #4
12710 CLOSE #5
12720 RETURN
13000 -----
13010 OPEN FILENAME5$ AS #5
13020 PREC5=0 :LREC5=0
13030 FOR PREC5=1 TO LOF(5)
13040 GET #5 ,PREC5
13050 FOR K=0 TO BF5-1
13060 FIELD #5,K*LLEN5 AS DUMMY$, 2 AS FP$ ,2 AS F$(0),2 AS F$(1)
,4 AS F$(2),4 AS F$(3),4 AS F$(4),4 AS F$(5),4 AS F$(6),8 AS F$(7)
),8 AS F$(8),8 AS F$(9),8 AS F$(10)
13070 IF (CVI(FP$)<> -1) THEN GOTO 13150
13080 LREC5=LREC5+1
13090 FOR I=0 TO MAX5
13100 ON I+1 GOTO 13110,13110,13120,13120,13120,13120,13120,13130,13130
,13130,13130
13110 FFF(LREC5 ,I)=CVI(F$(I)) : GOTO 13140
13120 FFF(LREC5 ,I)=CVS(F$(I)) : GOTO 13140
13130 FFF(LREC5 ,I)=CVD(F$(I)) : GOTO 13140
13140 NEXT I
13150 NEXT K
13160 NEXT PREC5
13170 FOR I=1 TO LREC5
13180 LPRINT USING "#####" ;FFF(I,0) ;
13190 LPRINT USING "#####" ;FFF(I,1) ;
13200 LPRINT USING "####.#" ;FFF(I,2) ;
13210 LPRINT USING "####.#" ;FFF(I,3) ;

```



```

13220 LPRINT USING "####.#" ;FFF(1,4) ;
13230 LPRINT USING "####.#" ;FFF(1,5) ;
13240 LPRINT USING "####.#" ;FFF(1,6) ;
13250 LPRINT USING "####.#####" ;FFF(1,7) ;
13260 LPRINT USING "####.#####" ;FFF(1,8) ;
13270 LPRINT USING "####.#####" ;FFF(1,9) ;
13280 LPRINT USING "####.#####" ;FFF(1,10);
13290 LPRINT
13300 NEXT I
13310 CLOSE #5
13320 RUN "menuuu.PRO"
20000 '-----
20010 *SELECT.FILE
20020 COLOR 5
20030 PRINT " "
20040 PRINT " | The data that you have in drive 2 are as follows. | "
20050 PRINT " "
20060 PRINT "( Drive 2 )" : FILES 2
20070 COLOR 7
20080 PRINT " "
20090 PRINT " | Input data file name that you have already made | "
20100 PRINT " | and input new data file name that you want to sum up | "
20110 PRINT " "
20120 X$="2:"
20130 INPUT "Data file of Spc,Dbh,Chd,Ch,Th,Bh ";FILE1$
20140 INPUT "Data file of diameter with bark of each 2m ";FILE2$
20150 INPUT "Data file of bark thik of each 2m ";FILE3$
20160 INPUT "Data file of branch data ";FILE4$
20170 INPUT "New data file for sum up ";FILE5$
20180 FILENAME1$=X$+FILE1$
20190 FILENAME2$=X$+FILE2$
20200 FILENAME3$=X$+FILE3$
20210 FILENAME4$=X$+FILE4$
20220 FILENAME5$=X$+FILE5$
20230 IF ( FILE1$ = "" ) THEN GOTO 20080
20240 IF ( FILE2$ = "" ) THEN GOTO 20080
20250 IF ( FILE3$ = "" ) THEN GOTO 20080
20260 IF ( FILE4$ = "" ) THEN GOTO 20080
20270 IF ( FILE5$ = "" ) THEN GOTO 20080
20280 OPEN FILENAME1$ AS #1
20290 OPEN FILENAME2$ AS #2
20300 OPEN FILENAME3$ AS #3
20310 OPEN FILENAME4$ AS #4
20320 OPEN FILENAME5$ AS #5
20330 IF LOF(1) =0 THEN GOSUB 20390
20340 IF LOF(2) =0 THEN GOSUB 20480
20350 IF LOF(3) =0 THEN GOSUB 20570
20360 IF LOF(4) =0 THEN GOSUB 20660
20370 IF LOF(5)<>0 THEN GOTO 20750
20380 RETURN
20390 '-----
20400 FOR I=1 TO 500 : BEEP 1 : BEEP 0 : NEXT I
20410 PRINT " "
20420 PRINT " | You have not file 1 (Spe No,DBH,CHD,TH,CH,BH) yet | "
20430 PRINT " | Return to first | "
20440 PRINT " "
20450 CLOSE #1
20460 KILL FILENAME1$
20470 RETURN
20480 '-----
20490 FOR I=1 TO 500 : BEEP 1 : BEEP 0 : NEXT I
20500 PRINT " "
20510 PRINT " | You have not file 2 (Diameter wirh bark of each 2m) yet | "
20520 PRINT " | Return to first | "
20530 PRINT " "
20540 CLOSE #2
20550 KILL FILENAME2$
20560 RETURN
20570 '-----
20580 FOR I=1 TO 500 : BEEP 1 : BEEP 0 : NEXT I

```

```

20590 PRINT "
20600 PRINT "| You have not file 3 (Bark thick of each 2m) yet |"
20610 PRINT "| Return to first |"
20620 PRINT "
20630     CLOSE #3
20640     KILL FILENAME3$
20650     RETURN
20660 '-----
20670     FOR I=1 TO 500 : BEEP 1 : BEEP 0 : NEXT I
20680 PRINT "
20690 PRINT "| You have not file 4 (Branch data) yet |"
20700 PRINT "| Return to first |"
20710 PRINT "
20720     CLOSE #4
20730     KILL FILENAME4$ : CLOSE #5 : KILL FILENAME5$
20740     GOTO *BACK
20750 '-----
20760     FOR I=1 TO 500 : BEEP 1 : BEEP 0 : NEXT I
20770 PRINT "
20780 PRINT "| You have already same name file. |"
20790 PRINT "| Computer kill the file, and then create new file . |"
20800 PRINT "| Are you sure ? ( y / n ) |"
20810 PRINT "
20820 INPUT A$
20830     IF NOT ((A$ = "Y") OR (A$ = "y")) THEN *BACK
20840     CLOSE #5 : KILL FILENAME5$
20850     OPEN FILENAME5$ AS #5 : RETURN
20860 '-----
20870 *BACK
20880 CLOSE
20890     PRINT "Back to menu!!!"
20900     FOR I=1 TO 6000
20910     NEXT I
20920 RUN "menuuu.PRO"

```

(7) sumade. LEE

```

100 '==== sumade.LEE =====
110 '
120 ' PROGRAM save "sumade.LEE"
130 '
140 ' List of sum up data
150 '
160 '=====
170 '
180 GOSUB *ATTENTION
190 GOSUB *SELECT.FILE
200 '-----
210 MAXITEM = 10 : LLEN = 58 : PLEN = 256 : BF = PLEN ¥ LLEN
220 DIM F$(MAXITEM)
230 DIM FFF(200,MAXITEM)
240 PREC = 0 : LREC = 0
250 FOR PREC = 1 TO LOF(1)
260 GET #1 ,PREC
270 FOR K=0 TO BF-1
280 FIELD #1,K*LLEN AS DUMMY$, 2 AS FP$, 2 AS F$(0),2 AS F$(1),
4 AS F$(2),4 AS F$(3),4 AS F$(4),4 AS F$(5),4 AS F$(6),8 AS F$(7),8 A
S F$(8),8 AS F$(9),8 AS F$(10)
290 IF (CVI(FP$)<> -1) THEN GOTO 370
300 LREC = LREC + 1
310 FOR I=0 TO MAXITEM
320 ON I+1 GOTO 330,330,340,340,340,340,350,350,350,350
330 FFF(LREC ,I)=CVI(F$(I)) : GOTO 360
340 FFF(LREC ,I)=CVS(F$(I)) : GOTO 360
350 FFF(LREC ,I)=CVD(F$(I)) : GOTO 360
360 NEXT I
370 NEXT K
380 NEXT PREC
390 FOR I=1 TO LREC
400 LPRINT USING "#####" ;FFF(I,0) ;
410 LPRINT USING "#####" ;FFF(I,1) ;

```

```

420 LPRINT USING "####.#" ;FFF(I,2) ;
430 LPRINT USING "####.#" ;FFF(I,3) ;
440 LPRINT USING "####.#" ;FFF(I,4) ;
450 LPRINT USING "####.#" ;FFF(I,5) ;
460 LPRINT USING "####.#" ;FFF(I,6) ;
470 LPRINT USING "####.#####" ;FFF(I,7) ;
480 LPRINT USING "####.#####" ;FFF(I,8) ;
490 LPRINT USING "####.#####" ;FFF(I,9) ;
500 LPRINT USING "####.#####" ;FFF(I,10);
510 LPRINT
520 NEXT I
530 CLOSE #1
540 RUN "menuuu.PRO"
550 '-----
560 *ATTENTION
570 WIDTH 80,25 : CONSOLE 0,25,0,1
580 PRINT " "
590 PRINT "| Preparation of volme formula |"
600 PRINT " "
610 PRINT
620 PRINT " "
630 PRINT "| *** Enter floppy disk in Drive Number 2 *** |"
640 PRINT "| *** If you enter floppy disk, Put Y or y ** |"
650 PRINT " "
660 I$ = INKEY$ : IF I$ = "" THEN GOTO 660
670 IF I$="Y" OR I$="y" THEN RETURN
680 GOTO 310
690 '-----
700 *SELECT.FILE
710 COLOR 5
720 PRINT " "
730 PRINT "| The data that you have in drive 2 are as follows. |"
740 PRINT " "
750 PRINT "( Drive 2 )" : FILES 2
760 COLOR 7
770 PRINT " "
780 PRINT "| Input data file name that you want to list |"
790 PRINT "| ( Data file of sum up ) |"
800 PRINT " "
810 INPUT FILES$
820 X$="2:"
830 FILENAMES=X$+FILES$
840 IF ( FILES$ = "" ) THEN GOTO 770
850 OPEN FILENAMES$ AS #1
860 IF LOF(1)<>0 THEN RETURN
870 FOR I=1 TO 500 : BEEP 1 : BEEP 0 : NEXT I
880 PRINT " "
890 PRINT "| You have not file yet. Return to first!!! |"
900 PRINT " "
910 GOTO *GOBACK
920 '-----
930 *GOBACK
940 CLOSE
950 KILL FILENAMES$
960 PRINT "Back to menu!!!"
970 FOR I=1 TO 6000
980 NEXT I
990 RUN "menuuu.PRO"

```

(8) volcal.REG

```

10000 '==== volcal.REG =====
10001 '
10002 ' PROGRAM save "volcal.REG"
10003 '
10004 ' Volume formula ( caluclation formula )
10005 '
10006 '=====

```

```

10010 'argument
10011 ' X(nc,nv) data matrix
10012 ' AVERAGE(nv) mean vector
10013 ' VARIANCE(nv) variance vector
10014 ' COV(nv,nv) covariance matrix
10015 ' RHO(nv,nv) correlation matrix
10016 ' SD(nv) standard deviation vector
10017 ' XLARGE(nv) max value vector
10018 ' XSMALL(nv) min value vector
10019 '
10100 DEFDBL F,X,A,V,C,R,S,Y,E
10110 GOSUB *ATTENTION
10120 GOSUB *SELECT.FILE
10130 GOSUB *READ.FILE
10140 GOSUB *INIT.JOB
10150 T01=2.596
10160 TLEVEL=.001
10170 '
10180 DIM AVERAGE(NV),VARIANCE(NV),RHO(NV,NV)
10190 DIM XLARGE(NV),XSMALL(NV)
10200 DIM SD(NV),MATX(NV,NV),SS(NV,NV),COV(NV,NV),PACO(NV,NV)
10210 DIM IX(NV),BETA(NV),BF(NV)
10220 DIM WM(NV)
10230 '
10240 OPEN "lpt1:" FOR OUTPUT AS #1
10250 '
10260 GOSUB 12000 ' *DATAIN
10270 GOSUB 13000 ' *BSTAT
10280 GOSUB 14000 ' *BSTAT.PRM
10290 GOSUB 15000 ' *MODEL
10300 GOSUB 16000 ' *MODEL.PR
10310 GOSUB 20000 ' *STEP.PRN
10320 GOSUB 21000 ' *RESIDUAL
10330 RUN "menuuu.PRO"
10500 '-----
10510 *READ.FILE
10520 CLS
10530 MAXITEM=10 : LLEN = 58 : PLEN = 256 : BF = PLEN ¥ LLEN
10540 DIM F$(MAXITEM)
10550 DIM FFF(200,MAXITEM)
10560 PREC = 0 :LREC = 0
10570 CLS : PRINT "Wait !!! Reading data !!!"
10580 FOR PREC = 1 TO LOF(2)
10590 GET #2 ,PREC
10600 FOR K=0 TO BF-1
10610 FIELD #2.K*LLEN AS DUMMYS, 2 AS FP$, 2 AS F$(0),2 AS F$(1)
,4 AS F$(2),4 AS F$(3),4 AS F$(4),4 AS F$(5),4 AS F$(6),8 AS F$(
7),8 AS F$(8),8 AS F$(9),8 AS F$(10)
10620 IF (CVI(FP$)<> -1) THEN GOTO 10710
10630 LREC = LREC + 1
10640 FOR I=0 TO MAXITEM
10650 ON I+1 GOTO 10660,10660,10670,10670,10680,10680,10680,10690,10690,
10690,10690
10660 FFF(LREC ,I)=CVI(F$(I)) : GOTO 10700
10670 FFF(LREC ,I)=CVS(F$(I))/100 : GOTO 10700
10680 FFF(LREC ,I)=CVS(F$(I)) : GOTO 10700
10690 FFF(LREC ,I)=CVD(F$(I)) : GOTO 10700
10700 NEXT I
10710 NEXT K
10720 NEXT PREC
10730 RETURN
11000 '-----
11010 *INIT.JOB
11020 CLS : WIDTH 80,25 : CONSOLE 0,25,0,1
11030 DIM BB(10) ,CC(20)
11040 LOCATE 10, 1: PRINT"-----"
11050 LOCATE 10, 2: PRINT"| Diameter at breatht height ***** 1 |"
11060 LOCATE 10, 3: PRINT"| Diameter at commercial height ***** 2 |"
11070 LOCATE 10, 4: PRINT"| Total height ***** 3 |"
11080 LOCATE 10, 5: PRINT"| Commercial height ***** 4 |"
11090 LOCATE 10, 6: PRINT"| Battress height ***** 5 |"
11100 LOCATE 10, 7: PRINT"| Volume with bark ***** 6 |"

```

```

11110 LOCATE 10, 8: PRINT"| Volume without bark ***** 7 |"
11120 LOCATE 10, 9: PRINT"| Branch volume ***** 8 |"
11130 LOCATE 10,10: PRINT"| Volume with bark + Branch volume ***** 9 |"
11140 LOCATE 10,11: PRINT"-----|"
11150 COLOR 5
11160 LOCATE 5,12 : PRINT "**** Select criterion variable (Y) ****"
11170 LOCATE 50,12 : INPUT Y :BB(1)=Y+1
11180 FOR I= 1 TO 2
11190 LOCATE 5,14 : PRINT "**** Select explanatry variables X1, X2 ****"
11200 LOCATE I*5+45,15 : INPUT XX :BB(I+1)=XX+1
11210 NEXT I
11220 CLS
11230 '-----
11240 PRINT "Selection of data"
11250 PRINT "From what No. until what No. do you want to calculate ?"
11260 INPUT "From" ; FRO
11270 INPUT "Until" ; UNT
11280 '-----
11290 CLS
11300 LOCATE 10, 1: PRINT"-----|"
11310 LOCATE 10, 2: PRINT"| Input exclusive data No. |"
11320 LOCATE 10, 3: PRINT"| If end , input 0 |"
11330 LOCATE 10, 4: PRINT"-----|"
11340 FOR I=1 TO 20
11350 LOCATE I*5,5: INPUT CC
11360 IF CC = 0 THEN GOTO 11390
11370 CC(I)=CC
11380 NEXT I
11390 CLS : PRINT "Wait !!! Calculatting !!!"
11400 FOR I=1 TO UNT-(FRO-1)
11410 FOR J=0 TO 20
11420 IF J>5 THEN GOTO 11440
11430 IF FFF(I+FRO-1,BB(J))=0 THEN GOTO 11470
11440 IF I+FRO-1= CC(J) THEN GOTO 11470
11450 NEXT J
11460 NC=NC+1
11470 NEXT I
11480 NV=3
11490 RETURN
12000 '-----
12010 ' *DATAIN
12020 '
12030 DIM X(NC,NV)
12040 FOR I=1 TO UNT-(FRO-1)
12050 FOR J=1 TO NV
12060 FOR ZZ=1 TO 20
12070 IF ZZ>5 THEN GOTO 12090
12080 IF FFF(I+FRO-1,BB(ZZ))= 0 THEN GOTO 12120
12090 IF I+FRO-1 = CC(ZZ) THEN GOTO 12120
12100 NEXT ZZ
12110 GOTO 12130
12120 PP=PP+1 :GOTO 12170
12130 V=I
12140 V=V-PP
12150 X(V,J)=LOG(FFF(I+FRO-1,BB(J)))/LOG(10)
12160 NEXT J
12170 NEXT I
12180 RETURN
13000 '-----
13010 ' *BSTAT
13020 ' compute basic statistics
13030 '
13040 'DIM WM(NV)
13050 FOR IP=1 TO NV
13060 WM(IP)=X(1,IP)
13070 AVERAGE(IP)=0
13080 XLARGE(IP)=X(1,IP)
13090 XSMALL(IP)=X(1,IP)
13100 FOR JP=IP TO NV
13110 COV(IP,JP)=0
13120 NEXT JP
13130 NEXT IP
13140 '

```

```

13150 FOR IC=2 TO NC
13160   FOR IP=1 TO NV
13170     WX=X(IC,IP)
13180     WI=WX-WM(IP)
13190     AVERAGE(IP)=AVERAGE(IP)+WI
13200     IF XLARGE(IP)<WX THEN XLARGE(IP)=WX
13210     IF XSMALL(IP)>WX THEN XSMALL(IP)=WX
13220     FOR JP=IP TO NV
13230       SS(IP,JP)=SS(IP,JP)+WI*(X(IC,JP)-WM(JP))
13240     NEXT JP
13250   NEXT IP
13260 NEXT IC
13270 '
13280 FOR IP=1 TO NV
13290   AVERAGE(IP)=AVERAGE(IP)/NC+WM(IP)
13300 NEXT IP
13310 '
13320 FOR IP=1 TO NV
13330   WI=WM(IP)-AVERAGE(IP)
13340   FOR JP=IP TO NV
13350     SS(IP,JP)= SS(IP,JP)-WI*(WM(JP)-AVERAGE(JP))*NC : SS(JP,IP)=SS(IP,JP)
13360     COV(IP,JP)= SS(IP,JP)/(NC-1) : COV(JP,IP)=COV(IP,JP)
13370   NEXT JP
13380   VARIANCE(IP)=COV(IP,IP)
13390   SD(IP)=SQR(VARIANCE(IP))
13400 NEXT IP
13410 '
13420 FOR IP=1 TO NV
13430   FOR JP=IP TO NV
13440     RHO(IP,JP)=COV(IP,JP)/(SD(IP)*SD(JP))
13450     RHO(JP,IP)=RHO(IP,JP)
13460   NEXT JP
13470 NEXT IP
13480 ' ERASE WM
13490 RETURN
14000 '-----
14010 ' *BSTAT.PRM
14020 '       basic statistics print
14030 PRINT#1,
14040 PRINT#1,STRING$(18,"*");" Basic Statistics ";STRING$(18,"*")
14050 PRINT#1,
14060 PRINT#1,"Number of cases =";NC
14070 PRINT#1,
14080 PRINT#1,"VAR      MEAN    VARIANCE    SD          MIN          MAX"
14090 PRINT#1,STRING$(57,"-")
14100 FOR IP=1 TO NV
14110   PRINT#1,USING"X(##)";IP;
14120   PRINT#1,USING" #####.###";AVERAGE(IP);
14130   PRINT#1,USING" #####.###";VARIANCE(IP);
14140   PRINT#1,USING" #####.###";SD(IP);
14150   PRINT#1,USING" #####.###";XSMALL(IP);
14160   PRINT#1,USING" #####.###";XLARGE(IP)
14170 NEXT IP
14180 '
14190 ' upper=Correlation, lower=Covariance
14200 '
14210 N.PAGE=0
14220 LP=8
14230 FOR IS=1 TO NV STEP LP
14240   N.PAGE=N.PAGE+1
14250   IE=IS+LP-1
14260   IF IE>NV THEN IE=NV
14270   NA1$="SS" : FORM$="#####.###"
14280   FOR IP=1 TO NV :FOR JP=1 TO NV
14290     MATX(IP,JP)=SS(IP,JP)
14300   NEXT JP,IP
14310   GOSUB 14460
14320   NA1$="Covariance" : FORM$="#####.###"
14330   FOR IP=1 TO NV :FOR JP=1 TO NV
14340     MATX(IP,JP)=COV(IP,JP)
14350   NEXT JP,IP
14360   GOSUB 14460

```

```

14370 NAI$="Correlation" : FORM$="###.####"
14380 FOR IP=1 TO NV :FOR JP=1 TO NV
14390 MATX(IP,JP)=RHO(IP,JP)
14400 NEXT JP,IP
14410 GOSUB 14460
14420 NEXT IS
14430 PRINT#1,
14440 PRINT#1,"[NOTE: Variances and Covariances are divided by n-1 ]"
14450 RETURN
14460 '
14470 '*PRNT MATRIX
14480 '
14490 PRINT#1,
14500 PRINT#1,NAI$
14510 IF NV<=LP THEN PRINT#1, ELSE PRINT#1," No. ";N.PAGE
14520 PRINT#1," ";
14530 FOR IP=IS TO IE
14540 PRINT#1,USING" X(##)";IP;
14550 NEXT IP : PRINT#1,
14560 PRINT#1,STRING$((IE-IS+1)*9+6,"-")
14570 FOR JP=1 TO NV
14580 PRINT#1,USING"X(##)";JP;
14590 FOR IP=IS TO IE
14600 PRINT#1,USING FORM$ ;MATX(JP,IP);
14610 NEXT IP : PRINT#1,
14620 NEXT JP
14630 RETURN
15000 '-----
15010 ' *MODEL
15020 ' get model
15030 IY=1 : IX(0)=IY
15040 '
15050 NX=0
15060 FOR I=1 TO NV
15070 IF I=IY THEN 15100
15080 NX=NX+1
15090 IX(NX)=I
15100 NEXT I
15110 '
15120 YVAR=VARIANCE(IY)
15130 YSD =SQR(YVAR)
15140 RETURN
16000 '-----
16010 ' *MODEL.PR
16020 ' model print
16030 PRINT#1,
16040 PRINT#1, " ***** Result *****"
16050 PRINT#1,
16060 PRINT#1," DEPENDENT VARIABLE = X(";IY;)"
16070 PRINT#1," INDEPENDENT VARIABLES =";
16080 LP=8
16090 FOR I=1 TO NX STEP LP
16100 PRINT#1," ";
16110 IE=I+LP-1
16120 IF IE>NX THEN IE=NX
16130 FOR J=I TO IE
16140 PRINT#1,USING" X(##)";IX(J);
16150 NEXT J
16160 PRINT#1,
16170 NEXT I
16180 RETURN
20000 '-----
20010 ' *STEP.PRN
20020 ' step print
20030 FOR IENTER=1 TO NX
20040 IP=IX(IENTER)
20050 GOSUB 22010 ' *BEATON
20060 NEXT IENTER
20070 '
20080 RHOY=RHO(IY,IY)
20090 VY=VARIANCE(IY): SSY=VY*(NC-1)
20100 RSQUARE=1-RHOY

```

```

20110 R=SQR(RSQUARE)
20120 AJRSQ=RSQUARE-RHOY*NX/(NC-NX-1)
20130 PRINT#1,
20140 PRINT#1, "Multiple correlation "; USING "###.#####" ; R
20150 PRINT#1, "R-Square "; USING "###.#####" ; RSQUARE,
20160 PRINT#1, " Adjusted R-Square "; USING "###.#####" ; AJRSQ
20170 PRINT#1,
20180 PRINT#1, " ***** Analysis of Variance *****"
20190 PRINT#1,
20200 PRINT#1, " Sum of Mean "
20210 PRINT#1, " D.F. Square Square"
20220 PRINT#1, STRING$(41, "-")
20230 W11=RSQUARE*SSY
20240 W12=W11/NX
20250 FO=RSQUARE*(NC-NX-1)/(NX*RHOY)
20260 PRINT#1, USING"Regression ####.#####.## ";NX;W11;
20270 PRINT#1, USING"#####.##";W12
20280 W21=RHOY*SSY
20290 W22=W21/(NC-NX-1)
20300 SDEST=SQR(W22)
20310 PRINT#1, USING"Residual ####.#####.## ";NC-NX-1;W21;
20320 PRINT#1, USING"#####.##";W22
20330 PRINT#1, STRING$(41, "-")
20340 PRINT#1, USING"Total ####.#####.##";NC-1;SSY
20350 PRINT#1,
20360 PRINT#1, USING"S.D. of Estimate Error = ###.#####";SDEST
20370 PRINT#1, USING"FO Value = #####.#####";FO
20380 PRINT#1,
20390 PRINT#1, " partial'
20400 PRINT#1, " No. BETA correlation BETA F"
20410 PRINT#1, STRING$(49, "-")
20420 W=0
20430 SEB0=0
20440 FOR I=1 TO NX
20450 II=IX(I)
20460 BETA(I)=RHO(II, IY)/SQR(VARIANCE(II)/VY)
20470 W=W+AVERAGE(II)*BETA(I)
20480 PACO(II, IY)=RHO(II, IY)/SQR(RHOY*RHO(II, II)+RHO(IY, II)^2)
20490 PRINT#1, USING"X(##) ####.#####";II;BETA(I);
20500 PRINT#1, USING"###.#####";PACO(II, IY);
20510 SES=RHO(II, II)*RHOY/(NC-NX-1); SQSES=SQR(SES)
20520 SEB=SES*VY/VARIANCE(II) : SQSEB=SQR(SEB)
20530 BF(I)=BETA(I)^2/SEB
20540 PRINT#1, USING"####.#####";SQSEB;BF(I)
20550 SEB0=SEB0+AVERAGE(II)^2*SEB
20560 NEXT I
20570 BETA(0)=AVERAGE(IY)-W
20580 SEB=0
20590 FOR K=1 TO NX-1
20600 KK=IX(K)
20610 WA=AVERAGE(KK) : WV=VARIANCE(KK)
20620 FOR J=K+1 TO NX
20630 JJ=IX(J)
20640 SEB=SEB+2*RHO(JJ, KK)*WA*AVERAGE(JJ)/SQR(WV*VARIANCE(JJ))
20650 NEXT J
20660 NEXT K
20670 SEB0=SQR(SEB0+(SEB+(NC-1)/NC)*VY*RHOY/(NC-NX-1))
20680 PRINT#1, USING"const ####.#####";BETA(0);
20690 PRINT#1, SPC(9);
20700 PRINT#1, USING"#####.#####";SEB0
20710 PRINT#1,
20720 SDD=SS(2,2) ; SDH=SS(2,3) ; SHH=SS(3,3)
20730 C22=1/(SHH-SDH^2/SDD)
20740 C21=(-1)*C22*SDH/SDD
20750 C11=1/SDD-C21*(SDH/SDD)
20760 PRINT#1, USING"C22=###.##### C21=###.##### C11=###.#####";C22;C21;C11
20770 RETURN
21000 '-----
21010 ' *RESIDUAL
21020 ' residual
21030 PRINT#1,

```



```

21040 PRINT#1, "No.," " Y", " Est_Y", " Residual", " Elim_range"
21050 PRINT#1, STRINGS$(72, "-")
21060 FOR I=1 TO NC
21070   YHAT=BETA(0)
21080   FOR J=1 TO NX
21090     YHAT=YHAT+BETA(J)*X(I,IX(J))
21100   NEXT J
21110   YRES=X(I,IY)-YHAT
21120   ELIMR1=C11*(X(I,2)-AVERAGE(2))^2
21130   ELIMR2=2*C21*(X(I,2)-AVERAGE(2))*(X(I,3)-AVERAGE(3))
21140   ELIMR3=C22*(X(I,3)-AVERAGE(3))^2
21150   ELIMR=T01*SDEST*SQR(1-(1/NC+ELIMR1+ELIMR2+ELIMR3))
21160   IF ABS(YRES)>ELIMR THEN MARK1$="*" ELSE MARK1$=" "
21170 X(I,IY)=10^X(I,IY) : YHAT=10^YHAT : YRES=X(I,IY)-YHAT
21180 PRINT#1, USING "###";I;
21190 PRINT#1, USING "#####.#####";X(I,IY);
21200 PRINT#1, USING "#####.#####";YHAT ;
21210 PRINT#1, USING "#####.#####";YRES ;
21220 PRINT#1, USING "#####.#####";ELIMR ;
21230 PRINT#1, MARK1$
21240 NEXT I
21250 RETURN
22000 '-----
22010 ' *BEATON           'Beaton's sweep out routine
22020 '
22030 AP=RHO(IP,IP)
22040 IF ABS(AP)>=TLEVEL THEN 22060
22050 PRINT "*** Warning: pivot =";AP;"less than tolerance level =":TLEVEL
22060 FOR I=0 TO NX
22070   II=IX(I)
22080   IF II=IP THEN 22140
22090   FOR J=0 TO NX
22100     JJ=IX(J)
22110     IF JJ=IP THEN 22130
22120     RHO(II,JJ)=RHO(II,JJ)-RHO(II,IP)*RHO(IP,JJ)/AP
22130   NEXT J
22140 NEXT I
22150 FOR I=0 TO NX
22160   II=IX(I)
22170   RHO(IP,II)=RHO(IP,II)/AP
22180   RHO(II,IP)=-RHO(II,IP)/AP
22190 NEXT I
22200 RHO(IP,IP)=1/AP
22210 RETURN
23000 '-----
23010 *ATTENTION
23020 ' Calculation of multi regrestion
23030 WIDTH 80,25 : CONSOLE 0,25,0,1
23040 PRINT " "
23050 PRINT " | Preparation of volme formula. | "
23060 PRINT " "
23070 PRINT
23080 PRINT " "
23090 PRINT " | *** Enter floppy disk in Drive Number 2 *** | "
23100 PRINT " | *** if you enter floppy disk, Put Y or y ** | "
23110 PRINT " "
23120 IS = INKEY$ : IF IS = "" THEN GOTO 23120
23130 IF IS="Y" OR IS="y" THEN RETURN
23140 GOTO 10011
24000 '-----
24010 *SELECT.FILE
24020 COLOR 5
24030 PRINT " "
24040 PRINT " | The data that you have in drive 2 are as follows. | "
24050 PRINT " "
24060 PRINT "( Drive 2.)" : FILES 2
24070 COLOR 7
24080 PRINT " "
24090 PRINT " | Input data file name for volume formula. | "
24100 PRINT " | ( Data file of sum up ) | "
24110 PRINT " "
24120 INPUT FILES$
24130 XS="2:"

```

```

24140  FILENAME$=X$+FILES
24150  IF ( FILES = "" ) THEN GOTO 24080
24160  OPEN FILENAME$ AS #2
24170  IF LOF(2)<>0 THEN RETURN
24180  FOR I=1 TO 500 : BEEP 1 : BEEP 0 : NEXT I
24190  PRINT " "
24200  PRINT " | You have not file yet. Return to first!!! "
24210  PRINT " "
24220  CLOSE
24230  KILL FILENAME$
24240  PRINT "Back to menu!!!"
24250  FOR I=1 TO 6000
24260  NEXT I
24270  RUN "menuuu.PRO"

```

(主要 6 樹種 の 計算 結果)

***** Basic Statistics *****

Number of cases = 105

VAR	MEAN	VARIANCE	SD	MIN	MAX
X(1)	0.434	0.362	0.601	-1.741	1.657
X(2)	-0.249	0.063	0.252	-0.987	0.256
X(3)	1.219	0.026	0.161	0.322	1.525

SS

	X(1)	X(2)	X(3)
X(1)	37.609	15.480	7.178
X(2)	15.480	6.586	2.552
X(3)	7.178	2.552	2.688

Covariance

	X(1)	X(2)	X(3)
X(1)	0.362	0.149	0.069
X(2)	0.149	0.063	0.025
X(3)	0.069	0.025	0.026

Correlation

	X(1)	X(2)	X(3)
X(1)	1.00000	0.98358	0.71396
X(2)	0.98358	1.00000	0.60648
X(3)	0.71396	0.60648	1.00000

[NOTE: Variances and Covariances are divided by n-1]

***** Result *****

DEPENDENT VARIABLE = X(1)
 INDEPENDENT VARIABLES = X(2) X(3)

Multiple correlation 0.994605
 R-Square 0.989239 Adjusted R-Square 0.989028

***** Analysis of Variance *****

	D.F.	Sum of Square	Mean Square
Regression	2	37.20	18.60
Residual	102	0.40	0.00
Total	104	37.61	

S.D. of Estimate Error = 0.0630
 FO Value = 4688.1822

No.	BETA	partial correlation	BETA	F
X(2)	2.08111	0.98896	0.03087	4544.86000
X(3)	0.69494	0.81835	0.04832	206.80000
const	0.10606		0.06417	
C22=	0.588548	C21= -0.228015	C11=	0.240167

No.	Y	Est. Y	Residual	Elim. range
1	0.959999	0.810629	0.149370	0.161433
2	2.631685	2.451330	-0.180355	0.162152
3	3.323008	3.585960	-0.262952	0.162673
4	4.362420	3.697195	0.665225	0.162506
5	6.686665	5.965767	0.720898	0.162485
6	5.034818	5.911068	-0.876251	0.162212
7	11.506198	10.306969	1.199229	0.161848
8	10.234719	9.664776	0.569943	0.162072
9	8.940191	11.018343	-2.078152	0.161911
10	10.276010	11.767739	-1.491730	0.161818
11	9.108521	11.821308	-2.712788	0.161836
12	11.595506	10.146521	1.448985	0.160979
13	20.310917	18.974178	1.336739	0.161021
14	16.859808	18.888575	-2.028767	0.160689
15	21.921183	23.068832	-1.147650	0.160827
16	45.367889	50.075462	-4.707573	0.158924
17	0.094433	0.111380	-0.016947	0.158207
18	0.195218	0.172709	0.022509	0.159425
19	0.404037	0.406807	-0.002770	0.159483
20	4.724673	4.027995	0.696679	0.162673
21	0.623288	0.564731	0.058557	0.160634
22	1.495420	1.322702	0.172717	0.161042
23	2.303845	1.627853	0.675992	0.162634
24	3.414721	2.905757	0.508964	0.162374
25	3.159019	2.690243	0.468775	0.162100
26	2.944156	2.549259	0.394896	0.161332
27	5.418644	4.718157	0.700487	0.161934
28	6.250897	5.433855	0.817042	0.162438
29	6.671609	5.084509	1.587100	0.162564
30	5.070383	4.464104	0.606279	0.161812
31	5.665378	5.785040	-0.119662	0.162502
32	7.335485	6.937963	0.397521	0.161784
33	4.799361	5.986195	-1.186834	0.162011
34	8.993811	7.404425	1.589386	0.162325
35	10.171718	8.212953	1.958765	0.162228
36	7.284828	6.970229	0.314599	0.158659
37	10.395267	9.161595	1.233671	0.161415
38	15.814299	12.457643	3.356656	0.161705
39	9.544355	9.716488	-0.172132	0.159107
40	13.001108	13.855004	-0.853896	0.160640
41	0.478689	0.532697	-0.054009	0.161076
42	1.194301	1.157598	0.036703	0.162238
43	1.658151	1.672627	-0.014476	0.160055
44	1.939678	1.734488	0.205190	0.162581
45	3.715282	3.370643	0.344639	0.161481
46	7.237664	6.180156	1.057508	0.162322
47	4.043740	5.300273	-1.256533	0.161290
48	5.267836	5.958434	-0.690598	0.160705
49	6.828947	6.666017	0.162930	0.161725
50	6.790267	8.221477	-1.431210	0.162231
51	8.726336	9.869445	-1.143109	0.161651
52	10.675808	11.710508	-1.034700	0.161567
53	10.049156	8.555577	1.493579	0.158634
54	13.687737	16.709335	-3.021599	0.160100
55	0.160557	0.140096	0.010461	0.158514
56	0.139160	0.139666	-0.009506	0.158923
57	0.160564	0.188088	-0.037524	0.159651
58	0.247182	0.255286	-0.008104	0.159991

59	0.378002	0.355947	0.022055	0.160588
60	0.366920	0.386506	-0.019586	0.160224
61	0.224016	0.243122	-0.019106	0.159299
62	0.350105	0.302611	0.047494	0.160619
63	1.090598	1.315263	-0.224666	0.159588
64	1.926769	2.015569	-0.088800	0.161385
65	1.898081	2.297245	-0.399164	0.161467
66	1.634549	1.512874	0.121675	0.161846
67	2.799082	3.239150	-0.440067	0.160647
68	2.400972	2.752849	-0.351876	0.162331
69	3.135305	3.260749	-0.125444	0.162565
70	3.602448	3.494830	0.107618	0.162292
71	2.737750	2.597214	0.140536	0.161442
72	4.606974	4.708949	-0.101975	0.162139
73	6.262088	4.781613	1.480475	0.162301
74	6.892822	7.312679	-0.419857	0.160385
75	0.447477	0.470950	-0.023473	0.160944
76	0.684456	0.791322	-0.106866	0.161854
77	0.731660	0.684943	0.046716	0.161834
78	0.963563	1.333264	-0.369702	0.162521
79	1.566900	1.747716	-0.180816	0.161185
80	3.401532	2.986283	0.415249	0.162577
81	4.592953	4.492269	0.100684	0.160939
82	4.842368	4.208803	0.633565	0.162320
83	4.127731	4.106365	0.021367	0.159585
84	7.439904	7.439841	0.000062	0.161670
85	6.534128	6.798058	-0.263930	0.162382
86	0.018168	0.018862	-0.000693	0.135645
87	0.093376	0.109640	-0.016264	0.156465
88	0.540241	0.540079	0.000162	0.161003
89	0.843916	0.710212	0.133704	0.160720
90	0.913967	0.711432	0.202536	0.161580
91	2.048712	1.777532	0.271180	0.162558
92	1.841532	2.044866	-0.203334	0.162608
93	2.317856	2.559273	-0.241417	0.162058
94	2.045267	2.315114	-0.269847	0.162724
95	2.525964	2.707176	-0.181212	0.162726
96	4.430632	5.797951	-1.367320	0.161902
97	4.278069	4.048600	0.229469	0.162609
98	4.473837	4.240311	0.233526	0.162655
99	3.547527	4.462159	-0.914633	0.162640
100	4.260858	4.649537	-0.388680	0.162574
101	4.327755	4.796313	-0.468558	0.162553
102	6.512979	5.865314	0.647665	0.161950
103	3.927899	5.518593	-1.590693	0.162465
104	5.935189	5.719804	0.215385	0.162489
105	6.305274	5.891321	0.413953	0.162491

(9) fittit.REG

```

10000 '==== fittit.REG =====
10010 '
10020 '   PROGRAM   save "fittit.REG"
10030 '
10040 '   Fitting of volume fomula
10050 '
10060 '=====
10070 '
10080 '   DEFDBL F,V,D,H,X,Y,Z
10090 '   BETA0=0 : BETA1=0 : BETA2=0
10100 '   GOSUB *ATTENTION
10110 '   GOSUB *SELECT.FILE
10120 '   GOSUB *READ.FILE
10130 '   GOSUB *INIT.JOB
10140 '
10150 '   DIM V(NC),D(NC),H(NC), X(NC),Y(NC),Z(NC)
10160 '   OPEN "lpt1:" FOR OUTPUT AS #1
10170 '
10180 '   GOSUB 12000   '*DATAIN .

```

```

10190 GOSUB 12220      '*REG.TEST
10200 GOSUB 12630      '*RESID
10210 GOSUB 12870      '*REG.PRI
10220 RUN "menuuu.PRO"
11000 '-----
11010 *READ.FILE
11020   CLS
11030   MAXITEM=10 : LLEN=58 : PLEN=256 : BF=PLEN\LLEN
11040   DIM FS(MAXITEM)
11050   DIM FFF(200,MAXITEM)
11060   PREC = 0 : LREC = 0
11070   CLS : PRINT "Wait !!!   Reading data !!!"
11080   FOR PREC = 1 TO LOF(2)
11090     GET #2 ,PREC
11100     FOR K=0 TO BF-1
11110       FIELD #2,K*LLEN AS DUMMYS, 2 AS FPS ,2 AS F$(0),2 AS F$(1),4 AS F$(2)
         ,4 AS F$(3),4 AS F$(4),4 AS F$(5),4 AS F$(6),8 AS F$(7),8
         AS F$(8),8 AS F$(9),8 AS F$(10)
11120       IF (CVI(FPS)<> -1) THEN GOTO 11210
11130       LREC = LREC + 1
11140       FOR I=0 TO MAXITEM
11150         ON I+1 GOTO 11160,11160,11170,11170,11180,11180,11180,11190,11190,
         11190,11190
11160         FFF(LREC ,I)=CVI(F$(I))      : GOTO 11200
11170         FFF(LREC ,I)=CVS(F$(I))/100 : GOTO 11200
11180         FFF(LREC ,I)=CVS(F$(I))      : GOTO 11200
11190         FFF(LREC ,I)=CVD(F$(I))      : GOTO 11200
11200       NEXT I
11210     NEXT K
11220   NEXT PREC
11230   INPUT "Input constant term          " ; A
11240   INPUT "Input regression coefficient X1" ; B
11250   INPUT "Input regression coefficient X2" ; C
11260 RETURN
11270 '-----
11280 *INIT.JOB
11290 CLS : WIDTH 80,25 : CONSOLE 0,25,0,1
11300 DIM BB(10) ,CC(20)
11310 LOCATE 10, 1: PRINT"-----"
11320 LOCATE 10, 2: PRINT"| Diameter at breatht height ***** 1 |"
11330 LOCATE 10, 3: PRINT"| Diameter at commercial height ***** 2 |"
11340 LOCATE 10, 4: PRINT"| Total height ***** 3 |"
11350 LOCATE 10, 5: PRINT"| Commercial height ***** 4 |"
11360 LOCATE 10, 6: PRINT"| Battress height ***** 5 |"
11370 LOCATE 10, 7: PRINT"| Volume with bark ***** 6 |"
11380 LOCATE 10, 8: PRINT"| Volume without bark ***** 7 |"
11390 LOCATE 10, 9: PRINT"| Branch volume ***** 8 |"
11400 LOCATE 10,10: PRINT"| Volume with bark + Branch volume ***** 9 |"
11410 LOCATE 10,11: PRINT"-----"
11420   COLOR 5
11430 LOCATE 5,12   : PRINT "**** Select criterion variable (Y) ****"
11440 LOCATE 50,12  : INPUT Y      :BB(1)=Y+1
11450   FOR I= 1 TO 2
11460 LOCATE 5,14   : PRINT "**** Select explanatry variables X1, X2 ****"
11470 LOCATE 1*5+45,15 : INPUT QQ   :BB(I+1)=QQ+1
11480   NEXT I
11490 CLS
11500 '-----
11510 PRINT "Selection of data "
11520 PRINT "From what No. until what No. do you want to caluculate ?"
11530   INPUT "From";FRO
11540   INPUT "Until";UNT
11550 '-----
11560 CLS
11570 LOCATE 10, 1: PRINT"-----"
11580 LOCATE 10, 2: PRINT"| Input exclusive data No. |"
11590 LOCATE 10, 3: PRINT"| If end , input 0 |"
11600 LOCATE 10, 4: PRINT"-----"
11610   FOR I=1 TO 20
11620 LOCATE 1*5,5: INPUT CC
11630   IF CC = 0 THEN GOTO 11670
11640     CC(I)=CC

```

```

11650             NEXT I
11660     CLS : PRINT "Wait !!!  Calculatting !!!"
11670     FOR I= 1 TO UNT-(FRO-1)
11680         FOR J=0 TO 20
11690             IF J>5 THEN GOTO 11710
11700             IF FFF(I+FRO-1,BB(J))=0 THEN GOTO 11740
11710             IF I+FRO-1 =CC(J) THEN GOTO 11740
11720         NEXT J
11730         NC=NC+1
11740     NEXT I
11750     NV=3
11760 RETURN
12000 '-----
12010 *DATAIN
12020 DIM XX(NC,NV)
12030 FOR I=1 TO UNT-(FRO-1)
12040     FOR J=1 TO NV
12050         FOR QQ=1 TO 20
12060             IF QQ>5 THEN GOTO 12080
12070             IF FFF(I+FRO-1,BB(QQ))= 0 THEN GOTO 12110
12080             IF I+FRO-1 =CC(QQ) THEN GOTO 12110
12090         NEXT QQ
12100     GOTO 12120
12110     PP=PP+1 :GOTO 12160
12120     V=I
12130     V=V-PP
12140     XX(V,J)=LOG(FFF(I+FRO-1 ,BB(J)))/LOG(10)
12150     NEXT J
12160 NEXT I
12170 FOR I=1 TO NC
12180     X(I) = XX(I,1)
12190     Y(I) = A+B*(XX(I,2))+ C*(XX(I,3))
12200 NEXT I
12210 RETURN
12220 '-----
12230 ' *REG.TEST
12240 '
12250 SX=0 : SY=0
12260 SXX=0 : SYY=0 : SXY=0
12270 XL=X(1) : YL=Y(1)
12280 XS=X(1) : YS=Y(1)
12290 FOR I=1 TO NC
12300     SX=SX+X(I)
12310     SY=SY+Y(I)
12320     SXX=SXX+X(I)^2
12330     SYY=SYY+Y(I)^2
12340     SXY=SXY+X(I)*Y(I)
12350     IF XL<X(I) THEN XL=X(I)
12360     IF YL<Y(I) THEN YL=Y(I)
12370     IF XS>X(I) THEN XS=X(I)
12380     IF YS>Y(I) THEN YS=Y(I)
12390 NEXT I
12400 '
12410 XBAR=SX/NC
12420 YBAR=SY/NC
12430 XVAR=(SXX-SX^2/NC)/NC
12440 YVAR=(SYY-SY^2/NC)/NC
12450 COV=(SXY-SX*SY/NC)/NC
12460 XHENSA=SQR(XVAR)
12470 YHENSA=SQR(YVAR)
12480 RHO=COV/(XHENSA*YHENSA)
12490 '
12500 BHAT=COV/XVAR
12510 AHAT=YBAR-BHAT*XBAR
12520 '
12530 SEST =0
12540 FOR I=1 TO NC
12550     SEST=SEST+(Y(I)-AHAT-BHAT*X(I))^2
12560 NEXT I
12570 A=0 : B=1
12580 FD1=NC*(AHAT-A)^2
12590 FD2=2*SX*(AHAT-A)*(BHAT-B)
12600 FD3=SXX*(BHAT-B)^2

```

```

12610 FO=((NC-2)*(FD1+FD2+FD3))/(2*SEST)
12620 RETURN
12630 '-----
12640 *RESID
12650 '
12660 PRINT
12670 PRINT "Do you want to print out residuals between Y(RealV) & X(Est.V)"
12680 LOCATE 60 : INPUT "Y or N ? ",Q$
12690 IF Q$="Y" OR Q$="y" THEN 12700 ELSE 12720
12700 PRINT#1, "No.," Y", "Est_Y", "Residual"
12710 PRINT#1, STRING$(50, "-")
12720 SRES=0 : SRES2=0
12730 FOR I=1 TO NC
12740 RES=X(I)-Y(I)
12750 SRES=SRES+RES
12760 SRES2=SRES2+RES^2
12770 X(I)=10^X(I) : Y(I)=10^Y(I) : RES=X(I)-Y(I)
12780 IF Q$="Y" OR Q$="y" THEN 12790 ELSE 12830
12790 PRINT#1, USING "###";I ;
12800 PRINT#1, USING"#####.#####";X(I) ;
12810 PRINT#1, USING"#####.#####";Y(I) ;
12820 PRINT#1, USING"#####.#####";RES
12830 NEXT I
12840 SDRES=SQR(SRES2/NC)
12850 ESTERR=(SDRES/XBAR)*100
12860 RETURN
12870 '-----
12880 '*REG.PRI
12890 PRINT#1,
12900 PRINT#1, " x( Real V )" , "y( Est. V )"
12910 PRINT#1, "-----"
12920 PRINT#1, "Mean ";USING"###.#####" ##.#####"; XBAR ; YBAR
12930 PRINT#1, "S.D. ";USING"###.#####" ##.#####";XHENSA;YHENSA
12940 PRINT#1, "Max ";USING"###.#####" ##.#####"; XL ; YL
12950 PRINT#1, "Min ";USING"###.#####" ##.#####"; XS ; YS
12960 PRINT#1, "-----"
12970 PRINT#1, "-----"
12980 PRINT#1, "Number of cases ";NC
12990 PRINT#1, "Covariance ";COV
13000 PRINT#1, "Correlation ";RHO
13010 PRINT#1, "-----"
13020 PRINT#1, "-----"
13030 PRINT#1, "Regression of Y on X : Y =";AHAT;"+";BHAT;"* X"
13040 PRINT#1, "-----"
13050 PRINT#1, "FO - Value =";FO
13060 PRINT#1, "-----"
13070 PRINT#1, "Error of estimate = ";ESTERR;"%"
13080 PRINT#1, "-----"
13090 RETURN
14000 '-----
14010 *ATTENTION
14020 '
14030 Fitting test of volume fomula
14040 WIDTH 80,25 : CONSOLE 0,25,0,1
14050 PRINT "-----"
14060 PRINT " Fitting test of volume formula. "
14070 PRINT "-----"
14080 PRINT
14090 PRINT "-----"
14100 PRINT "| *** Enter floppy disk in Drive Number 2 *** |"
14110 PRINT "| *** If you enter floppy disk, Put Y or y ** |"
14120 PRINT "-----"
14130 IS = INKEY$ : IF IS = "" THEN GOTO 14130
14140 IF IS="Y" OR IS="y" THEN RETURN
14150 GOTO 14090
14160 '-----
14170 *SELECT.FILE
14180 PRINT "-----"
14190 PRINT " The data that you have in drive 2 are as follows. "
14200 PRINT "-----"
14210 PRINT "( Drive 2 ) : FILES 2
14220 COLOR 7

```

```

14230 PRINT "
14240 PRINT " | Input file name for fitting of volume formula. | "
14250 PRINT "
14260 INPUT FILE$
14270 X$="2:"
14280 FILENAME$=X$+FILE$
14290 IF ( FILE$ = "" ) THEN GOTO 14230
14300 OPEN FILENAME$ AS #2
14310 IF LOF(2)<>0 THEN RETURN
14320 FOR I=1 TO 500 : BEEP 1 : BEEP 0 : NEXT I
14330 PRINT "
14340 PRINT " | You have not file yet. Return to first!!! | "
14350 PRINT "
14360 CLOSE
14370 KILL FILENAME$
14380 PRINT "Back to menu!!!"
14390 FOR I=1 TO 6000
14400 NEXT I
14410 RUN "menuuu.PRO"

```

(主要 6 樹種の材積式に主要 6 樹種以外のデータを入力した計算結果)

No.	Y	Est_Y	Residual
1	1.98699	1.74746	0.23953
2	3.53874	2.67200	0.86674
3	5.26031	5.26535	-0.00504
4	23.46198	14.62945	8.83253
5	17.84739	15.80834	2.03905
6	27.78981	22.95716	4.83264
7	19.94125	19.56425	0.37699
8	25.48383	21.42724	4.05659
9	45.77258	37.36130	8.41129
10	0.14807	0.13176	0.01631
11	0.56234	0.51636	0.04598
12	2.77799	2.68376	0.09423
13	0.94560	0.88924	0.05635
14	3.65960	4.96759	-1.30800
15	0.59181	0.62419	-0.03238
16	1.31826	1.14583	0.17243
17	0.52997	0.55783	-0.02786
18	0.85710	0.87448	-0.01738
19	7.62658	7.00085	0.62574
20	0.47543	0.35681	0.11863
21	1.12256	1.16918	-0.04662
22	1.73453	1.75817	-0.02364
23	1.27445	1.07604	0.19842
24	0.93446	1.02821	-0.09374
25	0.18779	0.19944	-0.01164
26	2.89627	2.11633	0.77994
27	1.76690	1.43515	0.33175
28	0.14389	0.13168	0.01221
29	10.90326	9.04055	1.86271
30	5.60101	5.63486	-0.03385
31	19.56997	23.17365	-3.60368
32	0.31949	0.26872	0.05078
33	0.26968	0.31941	-0.04972
34	14.21109	11.84500	2.36609
35	9.51549	10.15086	-0.63536
36	0.41732	0.45619	-0.03887
37	1.60852	1.60598	0.00254
38	1.09846	0.91122	0.18724
39	1.39543	1.20883	0.18659
40	0.51390	0.54100	-0.02710
41	3.61417	3.63434	-0.02016
42	0.48077	0.55400	-0.07323
43	0.27443	0.27758	-0.00315
44	1.03551	1.01812	0.01739
45	0.40880	0.34171	0.06708
46	0.04149	0.04340	-0.00191
47	0.28505	0.20853	0.07653
48	0.10258	0.12315	-0.02057

	x(Real V)	y(Est. V)
Mean	0.203641	0.17662
S.D.	0.725547	0.71022
Max	1.660605	1.57242
Min	-1.382106	-1.36254

Number of cases 48
Covariance .513266
Correlation .996055

Regression of Y on X : $Y = -.0219328 + .975016 * X$

F0 - Value = 6.130654002984931

Error of estimate = 34.8292 %

(0) voltab. PRO

```

10 '==== voltab.PRO =====
20 '
30 '   PROGRAM       save"voltab.PRO"
40 '
50 '   Preparation of volume table
60 '
70 '=====
80 '
90   CLS
100  A$="          6 especies principales (con corteza)      "
110  B$="          Resto de especies (con corteza)          "
120  C$="          6 especies principales (sin corteza)     "
130  D$="          Resto de especies (sin corteza)         "
140  E$="          Volumen de ramas (con corteza)         "
150  F$=" Chuncho, Guarngo, Higueron, Guapa, Sande, Zapote"
160  PRINT A$ : LOCATE 55,0 : PRINT "----- 1" :PRINT
170  PRINT B$ : LOCATE 55,2 : PRINT "----- 2" :PRINT
180  PRINT C$ : LOCATE 55,4 : PRINT "----- 3" :PRINT
190  PRINT D$ : LOCATE 55,6 : PRINT "----- 4" :PRINT
200  PRINT E$ : LOCATE 55,8 : PRINT "----- 5" :PRINT
210  INPUT "Select No." ; S
220  CLS
230  IF S=1 THEN Z$=A$
240  IF S=2 THEN Z$=B$
250  IF S=3 THEN Z$=C$
260  IF S=4 THEN Z$=D$
270  IF S=5 THEN Z$=E$
280  PRINT Z$
290  PRINT : PRINT
300  INPUT "Input Constant term (a) " ; A :PRINT
310  INPUT "Input Regression coefficient (b) " ; B :PRINT
320  INPUT "Input Regression coefficient (c) " ; C :PRINT
330  CLS : PRINT "Wait !!! Calculating !!!"
340  DIM VOLUM(200,40)
350  IST=2 : IED=80 : PIST=1 : PIED=2
360  JST=1 : JED=20 : PJST=1 : PJED=2
370  A$="###.##" : B$="#####" : C$="#####"
380  GOSUB *CALC
390  FOR PI=PIST TO PIED
400  FOR PJ=PJST TO PJED
410  GOSUB *TITLE
420  GOSUB *PRITAB
430  NEXT PJ,PI
440  RUN "menuuu.PRO"
450 *CALC
460  FOR I=IST TO PIED*IED STEP 2
470  FOR J=JST TO PJED*JED

```

```

480     X=I/100
490     VOLUM(I,J) = A+B*(LOG(X)/LOG(10))+C*(LOG(J)/LOG(10))
500     VOLUM(I,J)=10^VOLUM(I,J)
510     NEXT J
520     NEXT I
530     RETURN
540 *TITLE
550     LPRINT SPC(40);" "
560     LPRINT SPC(40);" | ***** TABLA DE VOLUMEN ***** |"
570     LPRINT SPC(40);" | ***** Para región noreste del ECUADOR ***** |"
580     LPRINT SPC(40);" "
590     IF S=2 THEN GOTO 630
600     IF S=4 THEN GOTO 630
610     IF S=5 THEN GOTO 630
620     LPRINT SPC(15); Z$ ;F$ : GOTO 640
630     LPRINT SPC(40); Z$
640     LPRINT
650     LPRINT SPC(33);"Regresion ajustada";
660     LPRINT SPC(3) ; "Log V = ";A;" + ";B;" * Log D + ";C;" * Log H"
670     LPRINT
680     LPRINT SPC(10) ;"Altura Comercial ( m )"
690     LPRINT " DAP " ;
700     FOR J=(PJ-1)*JED+JST TO (PJ-1)*JED+JED
710     LPRINT USING B$ ; J ;
720     NEXT J
730     LPRINT
740     LPRINT " ( cm ) "
750     RETURN
760 *PRITAB
770     X=-1
780     FOR I=(PI-1)*IED+IST TO (PI-1)*IED+IED STEP 2
790     Y=X ¥5
800     X= X+1
810     Z=X ¥5
820     IF Y<>Z THEN LPRINT
830     LPRINT USING C$ ;I;
840     LPRINT SPC(3);
850     FOR J=(PJ-1)*JED+JST TO (PJ-1)*JED+JED
860     LPRINT USING A$ ;VOLUM(I,J);
870     NEXT J
880     LPRINT
890     NEXT I
900 FOR I=1 TO 8 :LPRINT : NEXT I
910 RETURN

```

4-3 使用データ

材積式計算に使用したデータを掲げる。これは、Sumade.DAT の出力結果である。

