POLITICAL CAMPAIGN AND NAME AND ADDRESS FILE.

Financial Models and Chapter 5. Review texts. Construct ***Basic*** program or software model examples for organizations. See :

- Name and Address Files. Use ***Database*** or Comm Central.
- Financial Expense Reporting Statements. Use ***Spreadsheets***.

Name and Address Membership Lists Files. Name and address files are important to any organization (See *Oper. #2*). Start membership lists at any time. Use ***Word Processor*** MACROS.

Data File Name. Initially establish file *names*.

Computer Name and Address Relative File. Enter data. Establish records on computer files. Copy old data lists. Assemble lists from membership forms which have the current information about members.

A**. Member File Records**. Use one or two records for each member on file.

Name and address is on a record.
Use family name, or any other title.
Include house or apartment numeric address and street
Include city and zip code for mailing.

Enter the complete file. Keep current membership lists.

B**. Additional Updates**. Some records can be put on file and the rest added later. Add new records to ends of files. Keep separate name and address lists in different files for each city.

C. **Telephone Numbers**. Define data fields for telephone numbers on current records. Enter telephone numbers on records on files later. Area code and telephone number can fit in thirteen character data fields. Create accurate and up to date lists. Conduct activities with confidence.

D. Home and Business Name and Address Records.

Home Address. Write home address records.

Business Address. Write a second record after the first record. Use the second record for business addresses. ***Basic*** data field layouts on the second record may be redefined in display file format. Analyze the total number of code lines in each program in laboratory disk examples (See *Financial #2 Code Lines* and *#1 Bar Chart*).

Financial Cash Flow. Use monthly sequential files for income and expenses.

TI99 Programs. TI-Writer can not copy program code from the ***Basic*** compiler. This disk format is not in TI-Basic compiler code. Programs are not provided on ***Word Processor*** laboratory document files. Key in TI99 programs on screens. TI-Writer uses display data item fields of variable length records.

NEWSLETTERS.

Civic Organizations or Clubs. Use ***Word Processor*** or internet Navigator. Newsletters are always part of activities where dues paying members gather for some lofty purpose. Newsletters keep members up to date on the latest projects or activities, and financial status of the club. Newsletters contain telephone numbers of members so that committee chairmen can form task forces to accomplish goals. Drum up interest in projects with Newsletters.

Web Page. Create a web page with (*.html*) files. Members who miss meetings may wish to participate in ongoing events.

E-Mail. Send letters to *MEMBERS* attending meetings.

Financial #1 Bar Chart Program Analysis

PROCESS INPUT *BASIC* DATA FILES.

Select *Basic* compilers. Use relative files.

RECORD LAYOUTS.

TI99 Display Records. Relative files are fixed length display records. Data field are automatically established at thirteen (13) characters. Display records require at least two data fields for a total length of twenty six (26) to accommodate full street addresses.

Internal Records. Sequential file records are defined by field name. Determine the length of each data field. Specify any field length for TI99 internal records. Use len statements.

DOS *Qbasic* and *Gwbasic* Records.

- **DOS Fielded Records**. Specify lengths of fielded data items.
- **Match**. Match items in records. Test string segments lengths.
- **DOS Write**. Records lengths vary by data in fields. Write to sequential DOS files does not add spaces to data item fields. Easily compare files and data.
- **Visual Record** *TYPE*. Set field length Then use a sub procedure.

POLITICAL CAMPAIGNS FINANCIAL PROGRAMS*APPENDIX\I\.*

Financial Models in *Basic*. Copy programs from *APPENDIX* for quick computer systems (See *App. #2*). **RUN** programs (See *App. #1, Load Programs from Laboratory Disk*). See computer operational illustration in **Chapter 1**, *Oper. #2* and ***Basic*** *Chart #1*.

	Program	Membership Description	FR	TO
1	Names	Initial Name And Address	19	20
2	Update Names	Corrects Address or Add Precinct	21	22
3	Last Name	Creates A Last Name List File	22	
4	Phone List	Retrieve Phone Number by Name	23	
5	List Names	Mailing List	24	
6	Yearbook	Campaign Finance Balance	24	25
7	Correct Checks	Corrects Checks on Sequential Fil	27	28
8	Reconcile	Balances Bank Statement	28	30

App. #3, *Financial Program Example List*
Budget Lab. Show #1: B:\BUDGET\BUDGET.SHW
90CENSUS CHT 34,735 06-08-93 9:47p
BANK CHT 39,836 11-07-93 11:10p
BUDGET CBK 8,192 06-07-96 5:05p
CARDS CHT 13,672 11-02-93 4:18p
CHECKBOO CHT 8,791 11-07-93 11:08p
COMPARE CHT 10,049 11-02-93 11:18a
HOUSE CHT 2,607 07-22-92 9:48a
MAIL CHT 9,959 11-02-93 11:15a
REPORT CHT 10,151 11-07-93 11:14p
See financial model illustrations, **Chapter 5**, *Budget. #1* and *#2*

OUT	DESCRIPTIN	NU	PROGRAM	END_LINE	TOTAL
			GOLD STAR		
C.	TOTAL LINES	1	NAMES	410	40
		2	UPDATENAMES	330	32
		3	LASTNAME	170	16
		4	PHONELIST	270	26
		5	LISTNAMES	110	10
			SUBTOTAL		
			TI994A		
		6	YEARBOOK	990	90

Appendix Finance Programs

7	CORR CHECKS	510	42
8	RECONCILE	720	63
	SUBTOTAL		195

Financial #2 Appendix Total Program Code Lines

Develop Application. Modify programs input/output routines. Change one program for other applications.
RUN *Basic* programs on TI99, ***Gwbasic*** on GoldStar and **Visual *Basic*** on the Packard Bell.
RUN the same program many times.
Consider other compilers. Evaluate the number of code lines.

- ***Java***. Download from Sun Development Web site. Create (.*html*) program files. Compile.
- **Visual *Basic***. Use Microsoft Excel 7. Select MACRO.
- **Create Dialogue Boxes**. Use msgbox and inputbox dialogue boxes. Use prompts, titles and buttons. Save one or more Macro MODULES. See Yearbook converted from TI99 to **Visual *Basic***.
- **Quatra Pro**. Consider Corel ***Spreadsheets***. Document toolbars to use for models.

			PROCESS CODE			
	DESCRIPTION		PROGRAM	FR	End-Line	Tot
A.	OPEN OUTPUT		GOLD STAR (1-5)			
	FILE	1	NAMES	60	410	35
	ROUTINE	2	UPDATE NAMES	20	390	37
		3	LAST NAME	20	130	11
		4	PHONE LISTS	90	210	12
			SUBTOTAL			95
			TI994A (6-8)			
		6	YEARBOOK	870	940	7
		7	CORR CHECKS	320	500	18
		8	RECONCILE	600	690	9
			SUBTOTAL			34
			GOLD STAR (1-5)			
B.	OPEN INPUT	2	UPDATE NAMES	20	270	25
	FILE	3	LAST NAME	30	120	9
	ROUTINE	4	PHONE LISTS	20	80	6
		5	LIST NAMES	20	80	6
			SUBTOTAL			46
			TI994A (6-8)			
		6	YEARBOOK	200	320	12
		7	CORR CHECKS	160	270	11
		8	RECONCILE	340	570	23
			SUBTOTAL			46

Financial #3 Program Output Input File Code Lines

1. NAMES (**Visual Basic**).
Establish *Name* and Address files for clubs or organizations. Record dues received from members as part of club activities.

NAMES (From DOS **GOLD STAR**).
Set data field sizes on record buffer layout. Update fields.

```
Type record
nout As String * 3
firstout As String * 10
lastout As String * 15
strnoout As String * 6
```

```
streetout As String * 20
cityout As String * 20
phone1out As String * 10
phone2out As String * 10
End Type
Dim addressrec As record
Sub names()
20 PHONEIN2$ = ""
30 cityin$ = "Riverview"
40 MsgBox prompt:="Names"
50 MsgBox prompt:="Write names on disk"
60 Open "e:\Address\LIST.txt" For Random As #2
Open random file on subdirectory on drive B.
80 ctr = ctr + 1
120 MsgBox prompt:="ctr " & " " & ctr
130 countr$ = Str$(ctr)
140 ok$ = InputBox(prompt:="number correct?")
Add one to a counter. Prompt for an ok.
150 If ok$ = "end" Then GoTo 430
200 firstin$ = InputBox(prompt:="First")
Locate cursers on screens. Prompt for members first name.
lastin$ = InputBox(prompt:="Last:")
Prompt for members first and last names.
240 STREETNO = InputBox(prompt:="street nu :")
250 strtno$ = Str$(STREETNO)
260 streetname$ = InputBox(prompt:="street name:")
300 phonein$ = InputBox(prompt:=" phone")

addressrec.nout = ctr
addressrec.firstout = firstin$
addressrec.lastout = lastin$
addressrec.strnoout = strtno$
addressrec.streetout = streetname$
addressrec.phone1out = phonein$
addressrec.cityout = cityin$
addressrec.phone2out = phone2$
390 Put #2, ctr, addressrec
Put data records on disk files. Clear screens.
410 GoTo 80
430 Close #2
440 MsgBox prompt:="count " & " " & ctr
450 End
End Sub
```

Name and Address Files. Analyze material with simple relative file models. Understand operations to apply to more complex models.

2. UPDATE NAMES.

Precinct Delegate Name and Addresses. The list is confidential. Candidates obtain delegate lists from county clerks for election purposes for small fees. Develop short update informational membership program models for name and *address* lists. Precinct delegates from all cities are prepared on a time schedule to meet an election deadline.

Two records for Individual.

Enter home address on the first record.

Add a field for precinct number to a second record. Enter data. Create the second record for delegate mailing. Create files. Process lists. Evaluate available information.

UPDATE NAMES.

```
10 PRINT "update names"
20 OPEN "r",#1, "b:\names\list\phone"
30 FIELD 1, 3 AS C$, 10 AS FIRST$ , 15 AS LAST$ , 6 AS STRNO$, 20 AS STREET$, 20 AS CITY$, 10 AS
PHONE1$, 10 AS PHONE2$
```

The Names program has the same data field description. Use one record for all the data fields.

```
30 PRINT "enter end to close file"
40 INPUT "enter record number";RECNO$
```

Locate record numbers of records to update.

```
50 IF RECNO$="end" THEN 380
60 RECNO=VAL(RECNO$)
50 IF EOF(1) THEN 380
60 GET #1,RECNO
70 N=VAL(C$)
80 STRNO=VAL(STRNO$)
90 PRINT N;FIRST$;LAST$,STRNO;STREET$, PHONE1$, PHONE2$,
100 RECNO$=STR$(RECNO)
110 FIRSTIN$=FIRST$
120 INPUT "enter first:";FIR$
130 IF FIRST$="" THEN 180
140 INPUT "enter last:";LA$
150 FIRSTIN$=FIR$
```

Prompt for each data item field. Update fields with specific input. Complete the entry. Make the next name and *address* record entry.

```
160 LASTIN$=LA$
170 GOTO 190
180 LAST$=LASTIN$
190 STRNOIN$=STRNO$
200 STREETIN$=STREET$
210 CITYIN$=CITYIN$
220 PHONEIN1$=PHONE1$
```

UPDATE NAMES

```
230 INPUT "enter phone:";PH$
```

Enter phone numbers. Update records in delegate lists.

```
240 IF PH$="" THEN 260
250 PHONEIN1$=PH$
260 PHONEIN2$=PHONE2$
270 PRINT TAB(44) CITY$
280 LSET C$=RECNO$
290 LSET FIRST$=FIRSTIN$
300 LSET LAST$=LASTIN$
310 LSET STRNO$=STRNOIN$
320 LSET STREET$=STREETIN$
330 LSET CITY$=CITYIN$
340 LSET PHONE1$=PHONEIN1$
350 LSET PHONE2$=PHONEIN2$
360 PUT #1,RECNO
370 GOTO 40
```

```
380 CLOSE #1
390 END
```

Keep Name and *address* **Lists Current**. Locate old files on disk. Update the second delegate record with business addresses and telephone number. Relative files are suited to update records on file, but not to insert of member records in lists on file. Easily update large files by this program. List data files.

3. LAST NAME.

Read Random Files. Open random as input.

Create Last Name Lists. Write sequential files with simple last *name* list as output to disk.

Open new input files and *name and address* file. Add telephone number. Locate and update telephone numbers quickly.

LAST NAME

```
Type record
nout As String * 3
firstout As String * 10
lastout As String * 15
strnoout As String * 6
streetout As String * 20
cityout As String * 20
phone1out As String * 10
phone2out As String * 10
End Type
Dim addressrec As record

Sub lastname()
10 MsgBox prompt:="last names file"
20 Open "e:\address\last" For Output As 2
30 Open "e:\address\list.txt" For Random As #1
```

LAST NAME

```
50 If EOF(1) Then GoTo 150
60 recno = recno + 1
70 Get #1, recno, addressrec
n = addressrec.nout
first$ = addressrec.firstout
last$ = addressrec.lastout
strno$ = addressrec.strnoout
street$ = addressrec.streetout
phone1$ = addressrec.phone1out
city$ = addressrec.cityout
phone2$ = addressrec.phone2out
90 strnumb = Val(strno$)
100 MsgBox recno & Chr(13) & first$ & " " & last$ & Chr(13) & strnumb & " " & street$ & Chr(13) & phone1$ &
" " & phone2$
110 MsgBox " " & city$
120 Print #2, recno & " " & last$
130 GoTo 50
140 Rem
150 Close #1
160 MsgBox prompt:="count " & recno
170 End
End Sub
```

Enter simple criteria to search through lists for last *names*. Obtain necessary information from this file. No special function (F) key or menu selection is necessary.

4. DOS PHONE LISTS.
RUN Phone List. Load last name file lists into memory. Open relative files. Enter last names in telephone number program prompts. Retrieve telephone numbers of delegates in models. Use additional Web site data.

PHONE LISTS

```
Type record
nout As String * 3
firstout As String * 10
lastout As String * 15
strnoout As String * 6
streetout As String * 20
cityout As String * 20
phone1out As String * 10
phone2out As String * 10
strtno As Integer
End Type

Dim addressrec As record
Dim holdname$(10)
```

PHONE LISTS

```
Sub phonlist()
10 MsgBox prompt:="Phone list"
20 Open "e:\address\last" For Input As #2
30 If EOF(2) Then GoTo 90
40 Input #2, n, last$
50 MsgBox n & " " & last$
60 X = X + 1
70 holdname$(X) = last$
80 GoTo 30
90 Open "e:\address\list.txt" For Random As #1
110 If EOF(1) Then GoTo 260
120 LNAME$ = InputBox(prompt:="enter name")
121 If LNAME$ = "end" Then GoTo 260
122 COMPLEN = Len(LNAME$)
123 For Ln = 1 To X
125 COMPARENAME$ = Left$(holdname$(Ln), COMPLEN)
```

Compare. String segments match name input.

```
140 If COMPARENAME$ = LNAME$ Then GoTo 180
150 Next Ln
160 MsgBox prompt:="try again"
170 GoTo 120
180 recno = Ln
190 Get #1, recno, addressrec
nout$ = addressrec.nout
first$ = addressrec.firstout
last$ = addressrec.lastout
strno$ = addressrec.strnoout
street$ = addressrec.streetout
city$ = addressrec.cityout
phone1$ = addressrec.phone1out
```

```
phone2$ = addressrec.phone2out
200 n = Val(ctr$)
210 strnumb = Val(strno$)
220 MsgBox nout$ & " " & first$ & " " & last$ & " " & strnumb & " " & street$ & " " & phone1$ & " " & phone2$
230 MsgBox city$
240 GoTo 110
260 Close #1
270 End

End Sub
```

Use this *phone* list at a moments notice if telephone numbers are required.

5. LIST NAMES.

List name and *addresses* on files. List all records on screens. Use lprint. Print lists on printers. Use *mailing* lists.

DOS LIST NAMES

```
Type record
nout As String * 3
firstout As String * 10
lastout As String * 15
strnoout As String * 6
streetout As String * 20
cityout As String * 20
phone1out As String * 10
phone2out As String * 10
End Type
Dim addressrec As record

Sub lastname()
10 MsgBox prompt:="last names file"
20 Open "e:\address\last" For Output As 2
30 Open "e:\address\list.txt" For Random As #1
50 If EOF(1) Then GoTo 150
60 recno = recno + 1
Add one to the record number for the record key of
each record to be printed.
70 Get #1, recno, addressrec
n = addressrec.nout
first$ = addressrec.firstout
last$ = addressrec.lastout
strno$ = addressrec.strnoout
street$ = addressrec.streetout
phone1$ = addressrec.phone1out
city$ = addressrec.cityout
phone2$ = addressrec.phone2out
90 strnumb = Val(strno$)
100 MsgBox recno & Chr(13) & first$ & " " & last$ & Chr(13) & strnumb & " " & street$ & Chr(13) & phone1$ &
" " & phone2$
110 MsgBox " " & city$
120 Print #2, recno & " " & last$
130 GoTo 50
140 Rem
150 Close #1
```

```
160 MsgBox prompt:="count " & recno
170 End
End Sub
```

CHECK RECONCILIATION AND BANK BALANCE.
TI99 to Visual *Basic*.
6. YEARBOOK.
Conduct Fund Raising Activities (See **Chapter 5**, *Computer Prompt #1* and *#2* for a Michigan Secretary of State Campaign and Financial Reporting Ledger). Obtain forms prior to using computer disk systems for reports (See *Budget #1* and *#2*). **RUN** simple programs. Record small number of checks and deposits made during the year. Follow a series of simple program prompt input statements. Enter data. Calculate running cash balances. Maintain expense reporting category subtotal. Append *checks* or *deposits* for new periods to ends of the check balancing files. (See line codes 200-320 for Listcheck program examples).

YEARBOOK

```
Sub checkbook()
100 MsgBox Title:="Rep yearbook checks", prompt:="", Buttons:=vbExclamation
memo$ = "45.45"
110 MsgBox prompt:=memo$, Buttons:=vbInformation, Title:="Bank Balance"
120 Dim CMEMO$(50), DEPNAME$(50)
130 Dim CAMT(50), CNUM(50), DPAMt(50)
160 Mon = InputBox(prompt:="enter election period mmm for month")
```

Report primary or general election activities. Create two files for yearly cash flow. Use inputbox for **Visual *Basic***.

```
170 file$ = "c:\test\" + Mon + ".txt"
180 INIT$ = InputBox(prompt:="If initial monthly file then enter yes ")
190 If INIT$ = "yes" Then GoTo 330
```

If no file exist enter:
YES
Otherwise the program will terminate.
Change the input # and print statement for **Visual *Basic***.

```
200 Open file$ For Input As #2
230 If EOF(2) Then GoTo 320
240 Input #2, A, B, C$, D$
250 Debug.Print A, B, C$, D$
```

The ledger file has two numeric data fields, (A and B) and two alphabetic data fields (C$ and D$)

```
260 If C$ = "dep" Then GoTo 290
270 BALANCE = BALANCE - B
```

Checks written reduce the balance.

```
280 GoTo 300
290 BALANCE = BALANCE + B
300 ENTRY = ENTRY + 1
310 GoTo 230
320 Close #2
330 MsgBox prompt:=memo$
340 BALANCE = InputBox(prompt:="bank balance? ")
350 MsgBox Title:="Enter each outstanding check number and amount", prompt:="Enter a zero for the check
number when finished"
```

Enter the beginning balance.

```
420 cnumber = InputBox(prompt:="check number: ")
430 If cnumber = 0 Then GoTo 600
```

YEARBOOK

```
440 If cnumber = 999 Then GoTo 460
450 GoTo 470
460 COUNT999 = COUNT999 + 1
```

```
470 N = N + 1
480 CNUM(N) = cnumber
memo$ = InputBox(prompt:=cnumber, Title:="enter memo: ")
CAMOUNT = InputBox(prompt:="check amount: ", Title:=memo$)
510 CAMT(N) = CAMOUNT
520 CTOTAL = CTOTAL + CAMT(N)
530 CMEMO$(N) = memo$
540 CMEMO$(N) = memo$
550 GoTo 420
600 MsgBox prompt:=COUNT99, Title:="Service charge:"
610 MsgBox prompt:="Enter each outstanding deposit amount "
640 MsgBox prompt:="Enter a zero amount when finished "
670 M = M + 1
680 depmemo$ = InputBox(prompt:="enter dues memo: ")
690 DEPOSIT = InputBox(prompt:="deposit amount: ", Title:=depmemo$)
700 DPAMt(M) = DEPOSIT
710 DEPNAME$(M) = depmemo$
720 If DPAMt(M) = 0 Then GoTo 790
730 If depmemo$ = "int" Then GoTo 750
740 GoTo 770
750 DEP999 = DEP999 + 1
770 DTOTAL = DTOTAL + DPAMt(M)
780 GoTo 670
790 NBAL = BALANCE - CTOTAL + DTOTAL
800 If COUNT999 = DEP999 Then GoTo 830
MsgBox prompt:=COUNT999 = DEP999, Title:="bank count error: "
820 GoTo 670
830 M = M - 1
840 MsgBox prompt:=NBAL, Title:="new balance"
860 MsgBox prompt:=NBAL - CBAL, Title:="correction "
```

Verify data input. If data is correct, appended input to ledger output files. Create and save check and deposit records on files.

```
860 PRINT "correction = "; NBAL - CBAL
870 OPEN #1: FILE$, SEQUENTIAL, VARIABLE, INTERNAL, APPEND
```

This open statement is changed to an
O and then a DOS A

```
870 Open file$ For Append As #1
880 For p = 1 To N
890 Write #1, p, DPAMt(p), "dep", DEPNAME$(p)
900 Next p
910 For Dd = 1 To M
```



```
Write #1, Dd, DPAMt(Dd), "dep", DEPNAME$(Dd)
930 Next Dd
940 Close #1
970 MsgBox prompt = "Print text file to list checks"
990 End
End Sub
```

TI99 Programs Convert to DOS.

Remove all colons and change opens.

Select ***Basic*** SEARCH.

Use Excel ***Spreadsheet*** Visual ***Basic*** Dialogue boxes.

A. Msgbox
B. Inputbox

7. CORRECT CHECKS.

Load *checks* from data files into computer arrays. Display checks on screens to review. Correct monthly data files. If no errors are detected, then close monthly files. Save files on disks. Consider FTP to transfer files. User ID may be needed to log on computer systems.

CORRECT CHECKS

```
90 PRINT TAB(10) "corr checks"
100 PRINT "correct checks"
110 DIM MCNUM(50), MCAMT(50), MMON$(50),MMEMO$(50)
120 PRINT "list checkbook"
130 PRINT "enter month mmm"
140 INPUT MONTH$
150 FILE$="dsk1."+MONTH$
160 OPEN #1: FILE$, SEQUENTIAL, FIXED, INPUT, INTERNAL
170 IF EOF(1) THEN 270
```

Use three character abbreviations for *month* as file names for checks. Read all records into memory.

When the
end of file EOF
is reached, begin the next program step.

```
180 COUNT=COUNT+1
190 INPUT #1: CNUM, CAMT, MON$, MEMO$
200 M=M+1
210 MCNUM(M)=CNUM
220 MCAMT(M)=CAMT
230 MMON$(M)=MON$
240 MMEMO$(M)=MEMO$
```

Alphabetic data field input names are followed by a
dollar sign $
Note that file name is alphabetic and is followed by a dollar sign $ in programs. File names are also enclosed in double quotes ""
in programs (See line number 150).

```
250 PRINT MON$;CNUM;CAMT;MEMO$
260 GOTO 170
270 CLOSE #1
280 PRINT "count ";COUNT
290 INPUT "enter corr to correct: ": CORR$
300 IF CORR$="CORR" then 320
```

Match. Match console *input* exactly with letters in double quotes. In this case, the character string to match is CORR
The program ends if no corrections are made.

```
310 END
320 OPEN #2:FILE$,OUTPUT, SEQUENTIAL,FIXED,INTERNAL
```

List all output statement parameters. Do not use default parameters because of computer to computer considerations.

```
330 FOR P=1 TO M
340 PRINT MCNUM(P), MCAMT(P), MMON$(P); MMEMO$(P)
350 INPUT "enter c to correct: " : CHANGE$
360 IF CHANGE$='c" then 390
370 PRINT #2: MCNUM(P), MCAMT(P), MMON$(P),MMEMO$(P)
```

Copy correct records to file number two *#2* in this step. Copy records as they appear on *old* files.
380 GOTO 470
390 INPUT "check no:"CNUM
400 INPUT "check amount:":CAMT
410 INPUT "month :mon$
420 INPUT "memo: "MEMO$
430 PRINT #2: CNUM, CAMT, MON$, MEMO$
Use the same file layout to write records. Write out only records corrected through keyboardw on file number two *#2*.
450 INPUT "enter a to add checks:": ADD$
460 IF ADD$="a" THEN 390
Enter an
a
in response to computer prompts in input statements. Add overlooked check to files.
470 NEXT P
480 PRINT "changes ":CHG
490 DISPLAY EOF(2)
Display end of file status on screens. Close files.
500 CLOSE #2
510 END

8. RECONCILE.

RECONCILE.

90 PRINT "reconcile"
100 DIM DPAM(10)
110 PRINT "enter beginning balance"
120 INPUT BEGBALAN
130 PRINT "enter each outstanding"
140 PRINT "deposit amount"
150 PRINT "enter a zero amount "
160 PRINT "when finished "
170 PRINT
180 M=M+1
190 INPUT "deposit amount?": DEPOSIT
200 DPAM(M)=DEPOSIT
210 IF DPAM(M)=0 THEN 240
220 DTOTAL=DTOTAL+DPAM(M)
230 GOTO 180
240 DIM OCNUM(50),OCAMT(50)
250 DIM CCNUM(50),CCAMT(50)
260 DIM OMEMO$(50)
270 BEGBALAN=BEGBALAN+DTOTAL
280 PRINT "list checbbook"
290 PRINT "enter month MMM"
300 INPUT MONTH$
310 FILE$="*dsk1.*"&*MONTH$*
320 PRINT "enter c for cleared checks"
330 PRINT "enter o for outstanding checks"
340 OPEN #1: *FILE$*, SEQUENTIAL, FIXED, INPUT, INTERNAL
350 IF EOF(1) THEN 570
360 COUNT=COUNT +1
370 M=COUNT

```
380 INPUT #1:CNUM, CAMT, MON$, MEMO$
390 PRINT MON$;CNUM;CAMT;MEMO$
400 INPUT C$
410 IF C$="c" THEN 490
420 IF C$="o" THEN 440
```

Look for
c cleared
o outstanding
check. Make another entry if the wrong letter is entered.

```
430 GOTO 400
```

RECONCILE

```
440 OCNUM(M)=CNUM
450 OCAMT(M)=CAMT
460 OBAL=OBAL+CAMT
470 OCOUNT=OCOUNT + 1
480 GOTO 350
490 CBAL=CBAL+CAMT
500 CCNUM(M)=CNUM
510 CCAMT(M)=CAMT
520 CCOUNT=CCOUNT +1
530 GOTO 350
540 IF MONTH$="outstand" THEN 570
550 MONTH$="outstanding"
560 GOTO 310
570 CLOSE #1
580 PRINT "COUNT ":COUNT
590 PRINT "OUTSTANDING:" ;OCOUNT; OBAL
600 OPEN #2: "DSK1.OUTSTAND", OUTPUT, SEQUENTIAL, INTERNAL, FIXED 610 FORP=1 TO M
620 IF OCAMT(P)=0 THEN 650
630 PRINT OCNUM(P);OCAMT(P)
640 PRINT #2: OCNUM(P), OCAMT(P), MONTH$, OMEMO$(P)
650 NEXT P
660 FOR C=1 TO M
670 PRINT CCNUM(C); CAMT(C); MONTH$
680 NEXT C
690 CLOSE #2
700 ENDBALAN=BEGINBAL-CBAL
710 PRINT "check cleared"; CCOUNT;CBAL
720 PRINT "end bal";ENDBALAN
```

Basic programs need several code lines to process data.

Compare Files. Copy program routines to compare files. Open and close files. Program statement code routines are similar on both TI99 and machines with WINDOWS and DOS.

Total Numbers of Code Lines.

OPEN. Analyze lines of code in all program routines to open files in laboratory examples.

Calculate the total number of coding lines required to compare files. Subtract beginning program code line number from ending line number for input file output file or update routines for each program (See *Financial #3 Routine Code Lines*).

Appendix Finance Programs