

UNITED STATES
DEPARTMENT OF THE INTERIOR
GEOLOGICAL SURVEY

A User's Guide to the Automatic Phase Picking System for
Portable 3-component Seismograph 5-day Tapes

By

Mari L. Kauffmann¹

Open-File Report 86-109

1985

This report is preliminary and has not been reviewed for conformity with U.S. Geological Survey editorial standards and stratigraphic nomenclature. Any use of trade names is for descriptive purposes only and does not constitute endorsement by the USGS.

¹Office of Earthquakes, Volcanoes and Engineering
345 Middlefield Road MS-977
Menlo Park, California 94025

CONTENTS

	Page
List of figures	ii
A. Introduction	1
B. General Description of the System	1
1. Hardware	1
2. Software	1
3. Analog tapes, field notes, and record keeping	2
C. Description of Components	2
1. Scopes	2
2. Tape Drive	2
3. Time Code Translator	2
4. The RTP	3
5. Card Chassis	3
6. Switch Box	3
D. Controls and Settings	3
1. Scopes	3
2. Tape Drive Unit	4
3. Time Correction	4
4. Time Code Translator	4
5. The Switch Box	4
6. Booting the RTP	4
E. Running the Software Programs	5
1. Running 5DAYRTP	5
2. Running DRIFT	5
a. Files created by DRIFT	6
3. Running MRG5D	6
a. Files created by MRG5D	7
4. Running MRGPL	7
a. Files created by MRGPL	7
F. Operations Procedure	7
1. Before running a tape	7
2. Running a tape	8
References Cited	9
Acknowledgements	10
Figures	11
Software Source Codes	15

LIST OF FIGURES

1. Playback unit diagram
2. 5-day tape calender
3. Graph of linear drift readings
4. Software flowchart
5. Source code for program 5DAYRTP
6. Source code for program DRIFT
7. Source code for program MRG5D
8. Source code for program MRGPL

A. Introduction

This report describes and illustrates the procedures for operating the Automatic Phase Picking System (APPS) for 5-day tapes. Since the early 1960's the USGS has deployed portable 3-component seismographs which originally recorded in analog format for 10.6 days on a 7200 ft. reel of tape. The first major use of these portable seismographs was at Parkfield, California after the 1966 earthquake. In 1970, the recording equipment was modified to operate at a tape speed of 5.3 days per reel. Originally data from these portable seismographs were retrieved by manually processing paper playbacks or by digitizing each event. In the last several years routine processing of data from the permanent network has become automated following the development of the Allen-Ellis Real Time Picker (RTP) microprocessor system (Allen, 1982). Development of the APPS in 1984 was an extension of this technology to the 5-day portable field seismographs consisting, essentially, of the addition of an RTP microprocessor system to the high-gain vertical component analog output of the 5-day tapes. The tapes are played out with a standard Bell and Howell playback unit at 20 times real-time. The phase records are collected on the UNIX 11/70 disk.

In the first use of the APPS, approximately 3600 aftershocks were picked from 114 reels of 5-day tapes recorded during the 1983 Coalinga, California aftershock sequence. The picks were later merged with those from the permanent CALNET stations and from other temporary networks. The resulting hypocenters and fault-plane solutions are reported in Eberhart-Phillips and Reasenberg (1985).

B. General Description of the System

The location of the components described below are shown in the diagram of the playback unit (Figure 1).

1. Hardware

Most of the APPS hardware is housed in the Bell and Howell Tape Drive Cabinet. Also included are the RTP boards, Time Code Translator (TCT), TCT amplifier, the WWVB and IRIG-C difference circuit, tape drive, and the necessary electronics for decoding and conditioning the WWVB and IRIG-C signals, as well as oscilloscopes for monitoring the seismic traces and time codes. A Televideo terminal allows communication with the UNIX PDP-11/70 and a switch box permits the computer to except keyboard commands or RTP phase records.

2. Software

Four interactive UNIX processing programs allow the technician to edit, merge and sort the phase data produced by the APPS. Program 5DAYRTP reads the RTP output and adds the day, month, year and station name to the records. Program DRIFT calculates the time correction between WWVB and IRIG-C. Because the playback system reverses the sign on the time correction display unit, this program makes a sign correction in the phase file. Drift also deletes false phases (that is, phases with times before the start or after the end of the tape), and readings with times more than one day later than the previous

event. Program MRG5D combines and sorts the phase files into events creating one phase file for each day. Program MRGPL merges up to five HYP071 phase lists into a single phase list, creating one summary file and one phase file for each day.

3. Analog tapes, field notes and record keeping

The 5-day tapes are 1/2-inch, 8 channel, analog tapes. Each set of tapes that returns from the field is accompanied by a set of field notes containing pertinent information about station performance. Reading the field notes throughout the playback process is important because changes in station performance may occur in the field which effect processing. For example, gaps in the phase data are often the result of battery failure. Without field notes gaps in the data may be left unexplained. Setting up a note book to record drift readings, in-house and field equipment performance, is recommended. A three letter station abbreviation for each station sight must be established before the playback process begins. Drawing a record calendar of existing tapes provides a quick reference summary (see Figure 2). Labelling the tape box once a tape has been played back is also a useful part of the record keeping process.

C. Description of Components

1. The Scopes

There are three oscilloscopes located on top of the tape transport unit. The scope on the left displays the WWVB trace. The seismic signal is displayed on the center scope and the IRIG-C trace is displayed on the right scope.

2. The Tape Drive

The reels, rollers and tape heads make up the tape drive. The power panel to the right of the tape drive controls the tape drive unit. Below the tape drive is the Time Code Translator (TCT) which consists of two parts: the top section displays the corresponding Julian day, hour, minute and second tape time. The mode and filter settings are located in the lower section of the TCT below the digital display. The TCT error light is located on the bottom left corner of the seconds reading.

3. The Time Code Translator

A visual display of the time difference between IRIG-C and WWVB is necessary for time code corrections in the RTP phases. The time code correction or drift reading is located at the bottom of the main power panel. The correction must be read at least twice, once at the beginning and once at the end of each tape. Assuming a linear change in the time delay, program DRIFT calculates the time corrections for each phase read from the tape. The drift was assumed to be linear based on analysis of 114 tapes from 12 different stations. Errors based on an assumption of linear drift were smaller than the timing accuracy of the RTP (Figure 3).

4. The RTP

Within the tape drive unit is the RTP which times p-phases with respect to a translated IRIG-C time on the tape. The 5-day RTP samples 40 points per second tape-time (800 points per second real time) from one channel on the 5-day tape. This differs from the Calnet RTP which samples 200 points per second from 8 channels. Coda length is measured from the onset time to the time at which the amplitude falls and remains below 125 mv for 4 seconds. The first-motion polarities assigned by the RTP for 0-weight observations were judged to be accurate approximately 80% of the time, based on a comparison with 10 hand-picked events recorded in the Coalinga, California, 1983 aftershock sequence.

5. Card Chassis

FM Discriminators and Filters are located in the card racks in the bottom section of the cabinet below the TCT. There are three shelves of cards. The top shelf has 9 slots housing 5 cards. The WWVB and IRIG-C Discriminators are located in slots 1 and 2. Slot 3 houses the reference discriminator and slot 5 houses the reference conditioner and divider circuit. Slot 7 is the 5-day compensation/filter circuit and slot 9 is the WWVB vs IRIG-C phase compensation circuit. Slots 4, 6, and 8 are empty.

The middle shelf contains the four RTP cards. Card 1 is a single-board microprocessor. Card 2, (labeled A 13), supports the supervisor program and controls card 1. Card 3 is also a single-board microprocessor. Card 2 associates the signals it receives from card 1. Card 3 analyzes the results from card 4 (labeled 114).

The bottom shelf is the Tape Transport Electronics Tray. Slots 1 and 5 house the reproduce and direct reproduce cards respectively.

6. The Switch Box

Allows the computer to accept either keyboard commands or RTP phase output records. Card one of the RTP chassis, the Televideo and UNIX 11/70 are all connected to the switch box.

D. Controls and Correct Settings

1. The Scopes

Controls: The top left toggle switch labeled "vertical" on each scope allows the operator to check the zero line of the time code traces. When switched from REP (reproduce) to GRD (ground) the zero line appears.

Setting: The zero line must be centered to cross the middle of the time code so the tops of the IRIG and WWVB time code waveforms are above zero and the bottoms below zero. Adjustment of the zero line is done with the TCT amplifier. The zero line will be different for each tape but should not vary much within each tape. It is therefore necessary to check the zero line each time a tape is played.

2. The Tape Drive Unit

The Bell and Howell Tape drive is used in the normal fashion.

- 1) the ready light is ON when the tape is threaded tautly.
- 2) the phase lock selector set on TACH or TAPE
- 3) the record test selector set on NORM
- 4) the tape speed is 3 3/4 i.p.s.

3. Time Correction

The time code correction or drift reading is located at the bottom of the power panel. The time code translator error light must be off when taking a drift reading. If the error light is on, the drift reading is not fixed and will not be accurate. The time correction reading updates itself every 3 seconds.

4. Time Code Translator

If the playback unit remains assembled and operational, these settings will not need adjusting during the tape playing process. Fred Butler assembled the playback unit and maintains the equipment on a regular basis. Once the TCT has been set by Fred it should not need further adjusting.

5. The Switch Box

Controls: A toggle switch on the switch box allows information to be fed to the computer either directly from the RTP or interactively by the keyboard operator.

Setting: If the RTP is picking phases the switch box must be switched to PP. If the operator wants to communicate with the computer and the RTP is not picking phases the toggle switch should be set on T.

6. Booting the RTP

The RTP consists of four boards located in the middle card rack toward the bottom of the tape drive cabinet. The card in slot 2 (labeled A 13) has 2 toggle switches. The card in slot 4 (labeled 114) has 3 lights, a green and red light near the top and another red light at the bottom.

The picker is UP if the green light at the top of the 114 card is on and the red light at the bottom is flashing on and off. The RTP is DOWN when the green light is off and red light located directly below it stays on. To reset the RTP flip the top toggle switch on card A 13 up and down once.

If the picker goes down while processing a tape, a second phase file should be created by re-running program 5DAYRTP and later merging the two files together. If this occurs:

- 1) back up the tape to a point before the RTP failure
- 2) reboot the RTP
- 3) start program 5DAYRTP
- 4) start the tape, creating a new phase file
- 5) merge the phase files for the tape.

E. Running the Software Programs

See Figure 4 for software flowchart.

1. Running 5DAYRTP

Program 5DAYRTP reads the RTP output from the 5-day tape and adds the date and station names to the phases.

Example:

```
._?5dayrtp
input your name
mari
input station name - 3 letters
ESU
input time of beginning of tape
month,day,hour,minute
08,11,01,13
station = ESU
beginning 8 11 1 13
Is this correct? (input Y or N)
Y
Dear mari
System is now ready, to begin reading data directly from ppicker.
Switch to 'PP' when ready.
When tape is done, switch to 'T' and type in STOP
```

5DAYRTP will create a STA.phs phase file and a STA.out information file. (STA is the 3 letter station abbreviation).

2. Running DRIFT

DRIFT calculates the time correction between WWVB and IRIG-C.

Example:

```
._?drift
input your name
mari
input name of phase file
ESU0811.phs
phase file = ESU0811.phs
Is this correct? (input Y or N)
Y
input time of beginning drift reading
year,month,day,hour,minute
85,08,11,01,20
input beginning drift, as read from playback display
0.05
input time of ending drift reading
year,month,day,hour,minute
85,08,14,22,23
input ending drift, as read from playback display
0.15
```



```

station = ESU
beginning 85 8 11 1 20 drift = 0.05
ending 85 8 14 22 23 drift = 0.15

```

Is this correct? (input Y or N)

```

Y
ESU0811.dph 7 8
xxxxx ERROR IN INPUT xxxxx ESU XPD0 83 5 412 5 4.98
      0. -22
      skipping to next line
xxxxx ERROR IN INPUT xxxxx ESU XPD0 83 5 412 721.61
      0. -13
      skipping to next line
xxxxx ERROR IN INPUT xxxxx ESU XPD0 83 711 1 036.98
      0. -15
      skipping to next line
47 phases corrected for drift for station ESU
-?

```

(The ERROR IN INPUT message will appear if an event in the phase file occurred 2 days before the beginning tape time or 3 days after the end time).

a. Files created by DRIFT

DRIFT creates a STA.drft output log file and a STA.dph file of corrected phases with time corrections. Because the RTP lists all picks as 1983, the phases in the STA.dph file will need edited if the data is not from 1983.

3. Running MRG5D

MRG5D runs with a maximum of 12 STA.dph files. This program sorts the corrected phases into files that begin and end at midnight and combines events with less than 5 second time differences between consecutive P readings.

To Run MRG5D create a file containing the names of the STA.dph files to be merged. The command `l *.dph > list.dph` will work. Then to run `mrg5d` type:

`MRG5D list.dph > MRG1.out`
 Example:

```

-?l *.dph > list.dph
-?cat list.dph
ELC0813.dph
EOM0814.dph
ESU0811.dph
ESU0814.dph
EYB0813.dph
list.dph
-?e list.dph
"list.dph" 6 lines, 69 characters
:8d
5 EYB0813.dph
:uq
"list.dph" 5 lines, 60 characters
-?mrg5d list.dph > mrg1.out&
[1] 9776
-?

```

a. Files Created by MRG5D. The files created by MRG5D will be titled month and day followed by .phs1, .phs2 etc. (Example: 0811.phs1) A third run of MRG5D containing more phases for August 11 with create a 3rd .phs file (0811.phs3) The 0811.phs1, 0811.phs2, etc., files are hypo71 phase lists.

4. Running MRGPL. MRGPL merges up to five hypo71 phase lists or five 0811.phs1, 0811.phs2 etc. files together creating a single phase list and a summary list. For example: 0811.phs1 0811.phs2, 0811.phs3 would become one phase list called 0811.PL1. At this point other phase data maybe merged in with the 5-day phases.

To run MRGPL create a file containing the names of .phs files to be merged. The command 0814.phs* > list.pl will work. Then to run MRGPL type:
mrgpl list.pl > pl.out
Example:

```
_?l 0814.phs* > list.pl
_?cat list.pl
0814.phs1
0814.phs2
0814.phs3
_?mrgpl list.pl > pl.out&
```

a. Files Created by MRGPL. The files created by MRGPL will be the month and day followed by .PL1 and .SL1. 0811.PL1 is a complete phase list for August 11th and 0811.SL1 is the summary file.

F. Operation Procedure

1. Before running a tape:

- 1) Establish three letter station abbreviations for each station sight.
- 2) Draw a calendar of existing tapes for a quick reference summary.
- 3) Label a binder and photocopy drift sheets to record necessary information.
- 4) Check the field notes for possible problems with the tapes or field equipment. Make sure the box label matches the information in the field notes and is copied correctly to the drift sheet.
- 5) Plan to label the tape boxes once the tape has been played.
- 6) Secure a jar of liquid freon and a supply of q-tips for cleaning the heads and rollers.

2. Running a tape:

- 1) Mount the data tape on the bottom reel with the tape label facing outward and fast reverse the tape to the top reel. REMEMBER, ALWAYS TAKE THE TAPES OFF THE HEADS WHEN USING FAST SPEEDS.
- 2) Clean the tape heads and rollers with liquid freon (NEVER USE RUBBING ALCOHOL).
- 3) Check the RTP lights. Does the system need booting up?
- 4) Login UNIX, type '5dayrtp' and answer the prompts. (The 5dayrtp program will create a STA.phs file and a STA.out file, where STA is the 3 letter station abbreviation previously determined).
- 5) Thread the tape and push 'run fwd'.
- 6) Set switch box to pp.
- 7) Once the time signals are visible and the Time Code Translator has registered, and the TCT error light is off, record the drift reading and corresponding TCT reading on the drift sheet.
- 8) Check the time trace zero line by flipping the toggle switch on the scopes from REP to GRD. Remember that the tape plays at a speed up time of 20X real time so a 5-day tape that recorded in the field for 5 days should be finished in 6 hours.
- 9) Take a second drift reading and corresponding time from the end of the tape.
- 10) When the tape is finished put the switch box to 'T' and type 'stop'. Answer the prompts.
- 11) Rewind the tape.
- 12) Remove the tape, label the box and repeat the process. A tape can be mounted in the evening and left to run overnight. To do this repeat steps 1 thru 8. The next morning rewind the tape back far enough to record the end drift reading. Now repeat steps 9 thru 12.

References

Allen, R., Automatic phase pickers: their present use and future prospects, Bull. Seism. Soc. Am., 72, S225-S242 (1982).

Eberhart-Phillips, D. and P. Reasenberg, Hypocenter locations and constrained fault-plane solutions for Coalinga aftershocks, May 2-24, 1983: Evidence for a complex rupture geometry: Mechanics of the May 2, 1983 Coalinga earthquake, U.S. Geological Survey Open-File Report 85-44, p. 202-224 (1985).

Acknowledgements

I wish to thank Fred Butler for explaining the mechanical and electrical aspects of the automatic phase picking system. Special thanks to Donna Eberhart-Phillips and Paul Reasenberg for their encouragement and suggestions while reviewing the manual.

Playback and Automatic Phase Picking System for 5-day Tapes

WWVB trace seismic signal IRIG-C trace

figure 1

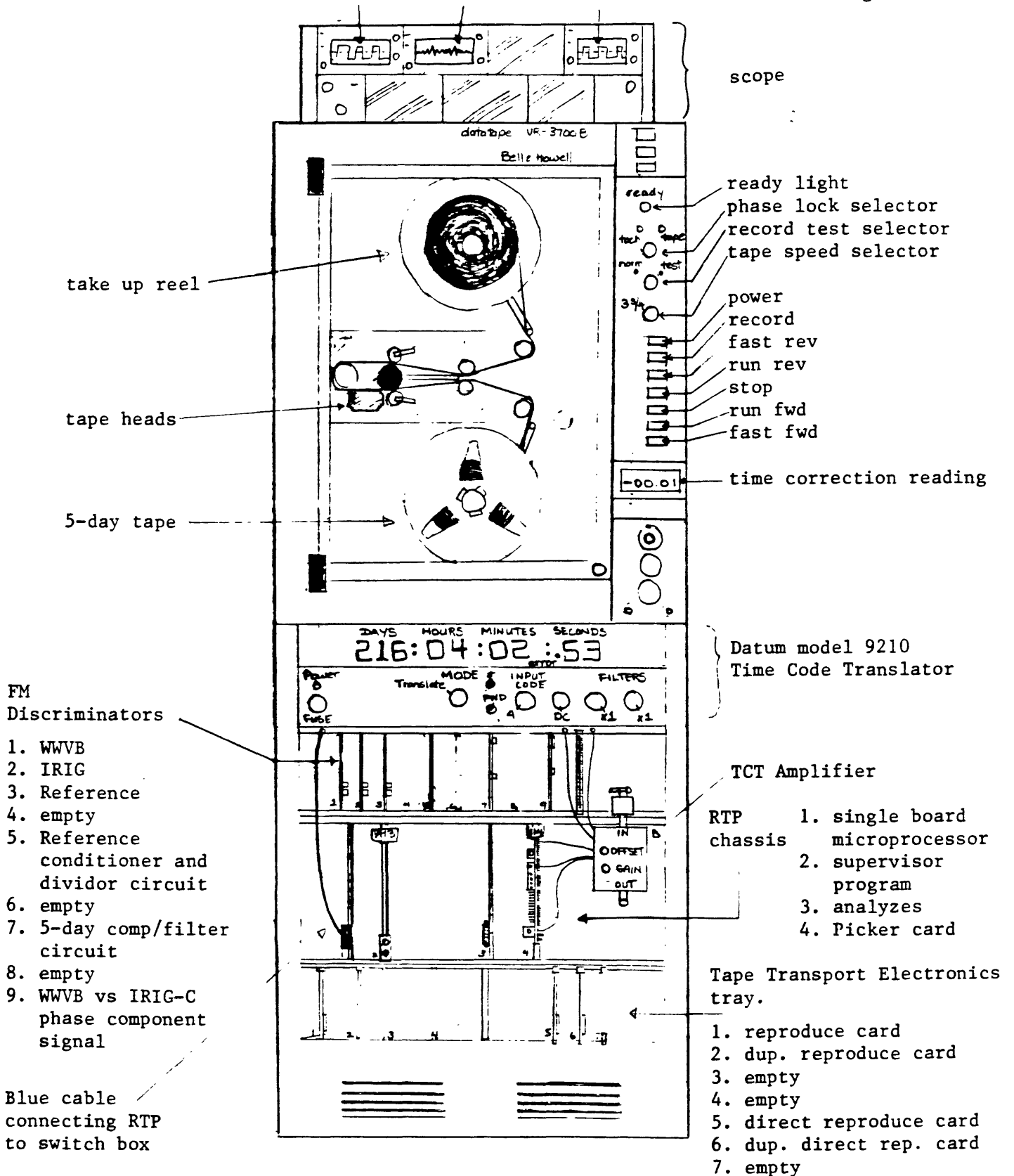
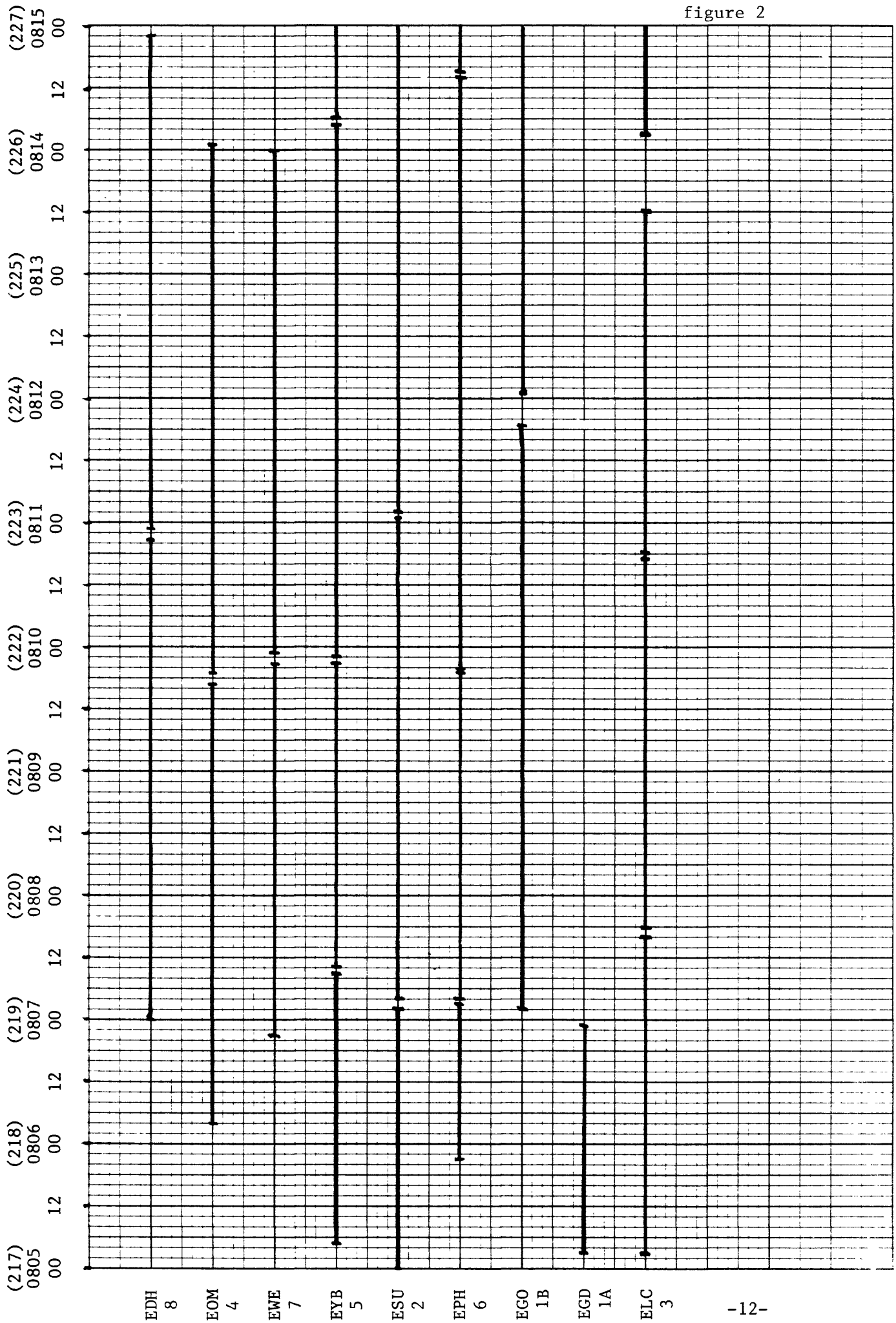


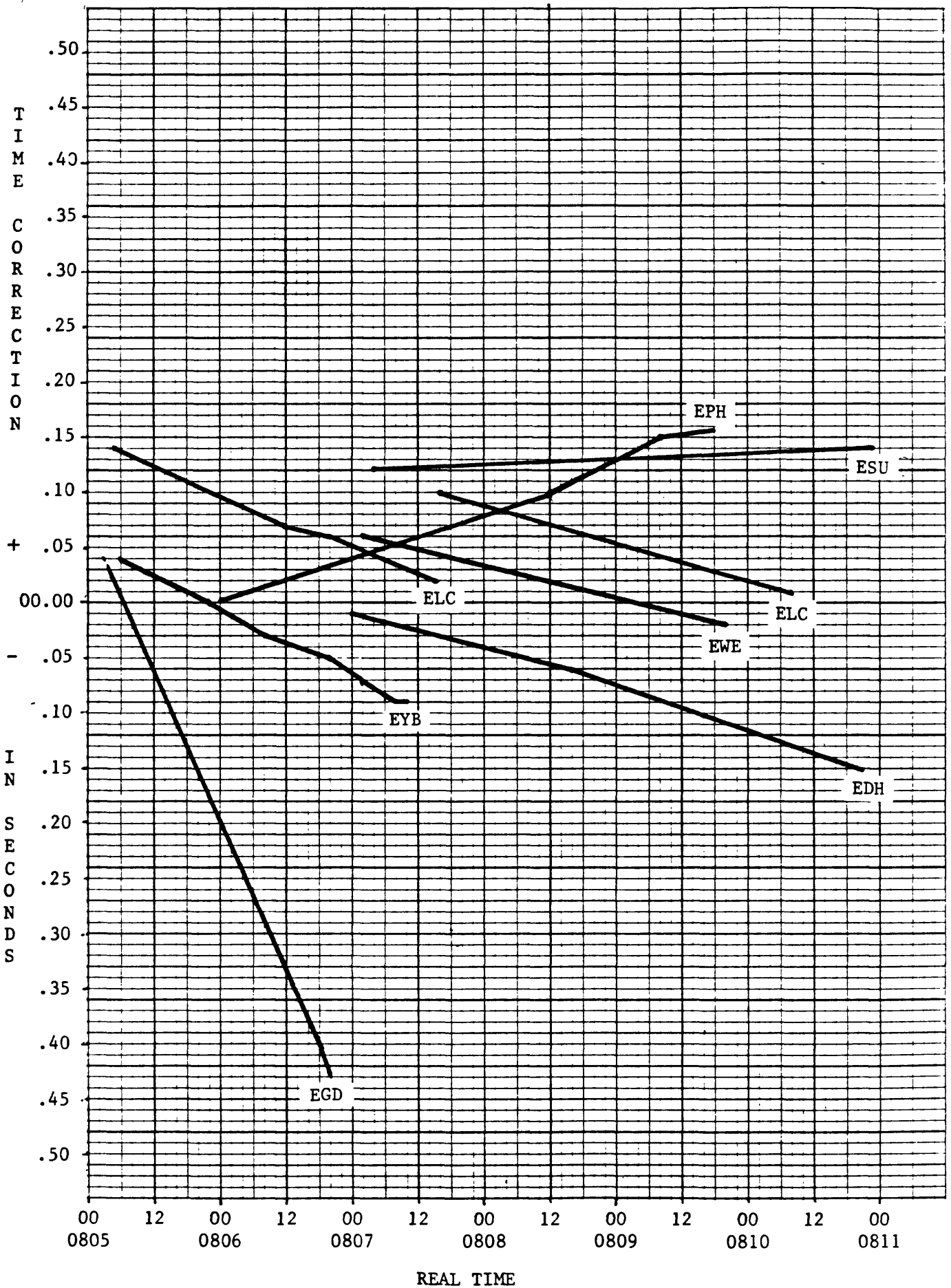
figure 2



5-DAY TAPE CALENDAR
Avenal earthquake August 4, 1985

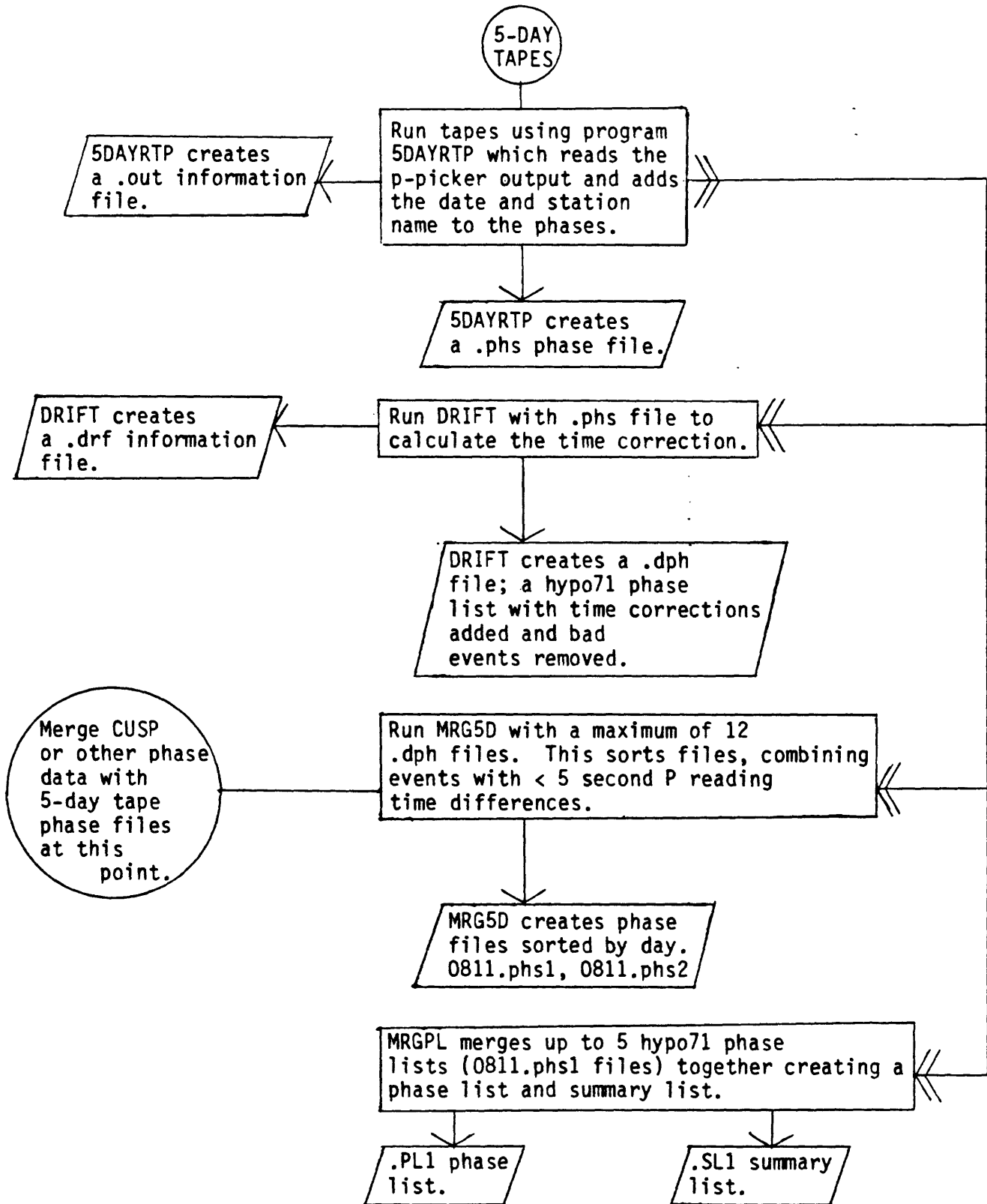
DRIFT READINGS FROM 5-DAY TAPES
 Avenal earthquake, August 4, 1985

figure 3



UNIX PROCESSING OF 5-DAY TAPES

figure 4



THIS PROGRAM READS THE P-PICKER OUTPUT FROM THE 5-DAY TAPES.
STATION ABBREVIATIONS SHOULD BE EDITED INTO THIS PROGRAM.

```

dimension xmo(12)
character*80 line
character*4 sta, strms(13)
character*4 md, check
character*14 file8, file9
character*40 opname
character*1 prmk(3)
data xmo/0, 31, 59, 90, 120, 151, 181, 212, 243, 273, 304, 334/
data strms/'ECT ', 'EWH ', 'ETM ', 'ETK ', 'EAQ ', 'EDG ',
2 'ESK ', 'EPM ', 'EVV ', 'EYU ', 'ETU ', 'EGR ', 'EDN '/
prmk(1)=lhX
prmk(2)=lhP
nphs=0
write(6, 300)
300 format(' input your name')
read(5, 310) opname
310 format(a40)
10 write(6, 320)
320 format(' input station name - 3 letters')
read(5, 330) sta
330 format(a4)
do 15 i=1, 13
if(sta.eq.strms(i)) goto 20
15 continue
write(6, 335) sta
335 format(' ***** STATION NAME ', a4, ' NOT ON',
2 ' LIST *****')
goto 10
20 write(6, 340)
340 format(' input time of beginning of tape', /,
2 'month, day, hour, minute')
read(5, *) imol, iday1, ibhr, ibmin
if((imol.gt.12).or.(imol.lt.1)) goto 21
if((iday1.gt.31).or.(iday1.lt.1)) goto 21
if((ibhr.gt.23).or.(ibhr.lt.0)) goto 21
if((ibmin.le.60).and.(ibmin.ge.0)) goto 22
21 write(6, 345)
345 format(' ***** INPUT TIME IS NOT APPROPRIATE *****')
goto 20
22 write(6, 380) sta, imol, iday1, ibhr, ibmin
380 format(' station = ', a4, /, ' beginning ', 2i3, 2x, 2i3,
3 /, ' Is this correct? (input Y or N)')
read(5, 390) check
390 format(a1)
if((check.eq.'N').or.(check.eq.'n')) goto 10
lst=len(sta)
c stations should always be three letters for coalinga
lst=3
imd=imol*100+iday1
write(line, 400) imd
400 format(i4)
read(line, 410) md
410 format(a4)
if(md(1:1).eq.' ') md(1:1)='0'
file8(1:lst)=sta

```

```

. file8(1st+1:1st+4)=md
. file8(1st+5:1st+8)='.phs'
file9(1:1st+5)=file8(1:1st+5)
file9(1st+6:1st+8)='out'
open(unit=08,file=file8,status='new',form='formatted',
2 err=23)
open(unit=09,file=file9,status='new',form='formatted',
2 err=23)
goto 24
23 open(unit=08,file=file8,status='old',form='formatted')
open(unit=09,file=file9,status='old',form='formatted')
24 write(9,310) opname
write(9,385) sta,imol,iday1,ibhr,ibmin
385 format(' station = ',a4,/, ' beginning ',2i2,2x,2i2)
c read in phase card from ppicker
lo=len(opname)
lol=lo+1
opname(lol:lol)=', '
write(6,630) opname
630 format(' Dear ',a40,/, ' System is now ready, '
2 ' to begin reading data directly from ppicker.',
3 /, ' Switch to "PP" when ready.',/, ' When'
4 ' tape is done, switch to "T" and type in'
5 ' STOP')
25 read(5,190,end=99) line
190 format(a80)
check=line(1:4)
if((check.eq.'STOP').or.(check.eq.'stop')) goto 99
c check for blank line
if(line(5:6).eq.' ') goto 25
read(line,200,err=90) prm(3),iwt,kyr,imo,kday,khr,
2 kmin,sec,ifmp
c write(6,200) prm(3),iwt,kyr,imo,kday,khr,
c 2 kmin,sec,ifmp
200 format(6x,a1,i1,1x,3i2,i2,i2,f5.2,t71,i5)
c include a blank card so it can be treated as an
c event by merge program
kyr=80
write(8,620) sta,(prm(j),j=1,3),iwt,kyr,imo,kday,khr,kmin,
2 sec,timcor,ifmp
620 format(a4,3a1,i1,1x,3i2,2i2,f5.2,41x,f5.2,i5)
if(nphs.gt.0) goto 60
60 nphs=nphs+1
goto 25
90 write(9,800) line
write(6,800) line
800 format(' ***** ERROR IN INPUT ***** ',a80,
2 /,10x,'skipping to next line')
goto 25
99 continue
ihr2=khr
imo2=imo
imn2=kmin
iday2=kday
write(9,810) nphs,sta,imol,iday1,ibhr,ibmin,imo2,iday2,ihr2,imn2
810 format(5x,i6,' phases read for station ',a4,'from tape ',
2 /,9x,'beginning ',2i2,1x,2i2,' and ending ',
3 2i2,1x,2i2)
write(6,300)
read(5,310) opname

```

```
write(9,310) opname  
close(8)  
close(9)  
stop  
end
```

DRIFT

THIS PROGRAM CALCULATES THE TIME CORRECTION IN THE PHASE LIST.

```

integer kmo(13),lmo(13)
character*80 line
character*4 sta,stmns(13)
character*14 file8,file9,file7
character*40 opname
character*1 prmk(3),check
data stmns/'EPM ','ESK ','ETU ','EYU ','EAQ ','EDG ',
2 'EGR ','EVV ','ETM ','ETK ','EWH ','ECT ','EDN '/
c This data is for NON-leap years
data kmo/0,31,59,90,120,151,181,212,243,273,304,334,365/
c This data is for LEAP years
data lmo/0,31,60,91,121,152,182,213,244,274,305,335,366/
prmk(1)=lhX
prmk(2)=lhP
nphs=0
c input drift and day,hr,min at beginning and end
c of tape, +=fast, -=slow.
write(6,300)
300 format(' input your name')
read(5,310) opname
310 format(a40)
8 write(6,325)
325 format(' input name of phase file')
read(5,326) file7
326 format(a14)
write(6,250) file7
250 format(' phase file = ',a14,/,
2 ' Is this correct? (input Y or N)')
read(5,390) check
if((check.eq.'N').or.(check.eq.'n')) goto 8
sta=file7(1:3)
10 write(6,340)
340 format(' input time of beginning drift reading',/,
2 'year,month,day,hour,minute')
read(5,*) iyrl,imol,iday1,ibhr,ibmin
if((imol.gt.12).or.(imol.lt.1)) goto 21
if((iday1.gt.31).or.(iday1.lt.1)) goto 21
if((ibhr.gt.23).or.(ibhr.lt.0)) goto 21
if((ibmin.le.60).and.(ibmin.ge.0)) goto 22
21 write(6,345)
345 format(' ***** INPUT TIME IS NOT APPROPRIATE *****')
goto 10
22 write(6,350)
350 format(' input beginning drift, as read from playback display')
read(5,*) bdrift
30 write(6,360)
360 format(' input time of ending drift reading',/,
2 'year,month,day,hour,minute')
read(5,*) iyrl2,imo2,iday2,iehr,iemin
if((imo2.gt.12).or.(imo2.lt.1)) goto 32
if((iday2.gt.31).or.(iday2.lt.1)) goto 32
if((iehr.gt.23).or.(iehr.lt.0)) goto 32
if((iemin.le.60).and.(iemin.ge.0)) goto 35
32 write(6,345)
goto 30
35 write(6,370)

```

```

370 format(' input ending drift, as read from playback display')
    read(5,*) edrift
    write(6,380) sta,iyr1,imol,iday1,ibhr,ibmin,bdrift,
    2 iyr2,imo2,iday2,iehr,iemin,edrft
380 format(' station = ',a4,/, ' beginning ',3i3,2x,2i3,
    2 2x,'drift =',f6.2,/, ' ending ',3i3,2x,2i3,
    3 2x,'drift =',f6.2,/, ' Is this correct? (input Y or N)')
    read(5,390) check
390 format(a1)
    if((check.eq.'N').or.(check.eq.'n')) goto 10
c
c Change the sign of bdrift and edrift to be correct
    bdrift= -bdrift
    edrift= -edrft
    lenf=lenstr(file7)
    lenfp=lenf-3
    if(file7(lenfp:lenf).eq.'.phs') lenf=lenf-4
    lenf1=lenf+1
    lenf4=lenf+4
    file8(1:lenf)=file7(1:lenf)
    file8(lenf1:lenf4)='.dph'
    write(6,611) file8, lenf,lenf1,lenf4
611 format(a15,2i3)
    file9(1:lenf)=file7(1:lenf)
    file9(lenf1:lenf4)='.drf'
    open(unit=08, file=file8, status='new', form='formatted')
    open(unit=09, file=file9, form='formatted')
    write(9,310) opname
    write(9,385) sta,iyr1,imol,iday1,ibhr,ibmin,bdrift,
    2 iyr2,imo2,iday2,iehr,iemin,edrft
385 format(' station = ',a4,/, ' beginning ',i4,2i3,2x,2i3,
    2 2x,'drift =',f6.2,/, ' ending ',i4,2i3,2x,2i3,
    3 2x,'drift =',f6.2)
    if(mod(iyr1,4).eq.0) then
        bday=float(lmo(imol)+iday1)
        yr1=366.0
    else
        bday=float(kmo(imol)+iday1)
        yr1=365.0
    end if
    pday1=bday
    if(mod(iyr2,4).eq.0) then
        eday=float(lmo(imo2)+iday2)
        if(iyr2.gt.iyr1) eday=eday+yr1
    else
        eday=float(kmo(imo2)+iday2)
        if(iyr2.gt.iyr1) eday=eday+yr1
    end if
    drift=edrft-bdrift
    emn=(eday-bday)*1440.0 + (iehr-ibhr)*60.0 + (iemin-ibmin)
c read in phase card from ppicker
    open(unit=07, file=file7, status='old', form='formatted',
    2 blank='zero', access='sequential')
    rewind (7)
    25 read(7,190,end=99) line
190 format(a80)
    read(line,200,err=90) sta,(prmk(j),j=1,3),iwt,kyr,imo,kday,khr,
    2 kmin,sec,ifmp
c write(6,200) prmk(3),iwt,kyr,imo,kday,khr,
c 2 kmin,sec,ifmp

```

```

200 format(a4,3a1,i1,1x,5i2,f5.2,t71,i5)
c check for blank line or bad month
  if(kyr.eq.0) goto 25
  if((imo.gt.12).or.(imo.lt.1)) goto 25
  if(mod(kyr,4).eq.0) then
    pday=float(lmo(imo)+kday)
    if(kyr.gt.iyrl) pday=pday+yrl
  else
    pday=float(kmo(imo)+kday)
    if(kyr.gt.iyrl) pday=pday+yrl
  end if
c check for bad times that are before start or after
c end of tape
c write(6,603) pday,bday,eday
c 603 format(' pday,bday,eday ',3f5.0)
  if((pday-bday).lt.-2) goto 90
  if((pday-eday).gt.3) goto 90
c check for times more than 1 day later than last time
  if((nphs.gt.1).and.(abs(pday-pday1).gt.1)) goto 90
c find time correction
  dmn=(pday-bday)*1440.0+(khr-ibhr)*60.0+(kmin-ibmin)+(sec/60.0)
  timcor=(dmn/emn)*drift + bdrift
c write(6,832) pday,dmn,timcor
832 format(' pday,dmn,timcor',i5,f15.0,f20.3)
c write out hypo71 phase card with time correction
c include a blank card so it can be treated as an
c event by merge program
  write(8,620) sta,(prmk(j),j=1,3),iwt,kyr,imo,kday,khr,kmin,
  2 sec,timcor,ifmp
620 format(a4,3a1,i1,1x,3i2,2i2,f5.2,41x,f5.2,i5)
  if(nphs.gt.0) goto 60
  jd1=pday
  ihr1=khr
  imn1=kmin
60 nphs=nphs+1
  pday1=pday
  goto 25
90 write(6,800) line
800 format(' ***** ERROR IN INPUT ***** ',a80,
  2 /,10x,'skipping to next line')
  goto 25
99 continue
  jd2=pday
  ihr2=khr
  imn2=kmin
  write(9,810) nphs,sta,jd1,ihr1,imn1,jd2,ihr2,imn2
810 format(5x,i6,' phases corrected for drift for station ',a4,
  2 /,9x,'beginning ',i4,2i3,' and ending ',
  3 i4,2i3)
  write(6,425) nphs,sta
425 format(i5,' phases corrected for drift for station ',a4)
  close(8)
  close(9)
  close(7)
  stop
  end

```

MRG5D

THIS PROGRAM MERGES PHASE FILES INTO EVENTS CREATING ONE PHASE FILE FOR EACH DAY.

```
character*80 filnam(12),line(12)
character*40 phsfil
character*4 ksta(1000)
character*4 prmk(1000),blank
real sec(1000),tcr(1000),tsc(1000),fmp(1000)
integer ipf(12),kdate(12),kyr(1000),kmo(1000),
2 kdy(1000),khr(1000),kmn(1000),
3 iptr(1000),imo(12),lmo(12)
c This data is for NON-leap years
  data imo/0,31,59,90,120,151,181,212,243,273,304,334/
c This data is for LEAP years
  data lmo/0,31,60,91,121,152,182,213,244,274,305,335/
  blank='
  nev=0
  tlast=0
  iptot=1
  kday=0
  open(1)
c sort 1 hour of arrival times at a time
  5 srtlim=3600.0
c limit possible time in seconds
  tmax=31700000.0
c read in filenames
  do 10 nf=1,12
    read(5,400,end=20) filnam(nf)
c    write(6,400) filnam(nf)
  10 continue
400 format(a80)
  20 nf=nf-1
    open(unit=2,file='mrg.out')
c open 5day phase files for each station
  do 25 i=1,nf
    ku=6+i
    open(unit=ku,file=filnam(i),blank='zero',status='old',
  2 form='formatted')
    write(2,401) filnam(i),ku
    write(6,401) filnam(i),ku
  401 format(' file= ',a30,' Unit=',i4)
    rewind(ku)
c read first line
  22 read(ku,400) line(i)
c check for blank line
  if(line(i)(1:4).eq.blank) goto 22
  25 continue
c
c
c read first phase for each file and sort
  30 do 35 i=1,nf
    read(line(i),430) jyr,jmo,jdy,jhr,jmn,sc
    kdate(i)=jmo*100+jdy
c    write(6,431) jyr,jmo,jdy,jhr,jmn,sc
  431 format(9x,5i2,f5.2)
    if(mod(jyr,4).eq.0) then
      dayjul=float(lmo(jmo)+jdy)
    else
```



```

        dayjul=float(imo(jmo)+jdy)
    . end if
    , tsc(i)=(dayjul*1440.0+float(jhr*60+jmn))*60.0+sc
c   write(6,425) i,kdate(i),dayjul,tsc(i)
425 format(' i=',i4,', kdate=',i5,', dayjul=',
    2 f5.0,', tsc=',f20.2)
    35 continue
430 format(9x,5i2,f5.2)
    call sort(tsc,ipf,nf)
    scmax=tsc(ipf(1))+srtlim
    if(tsc(ipf(1)).ge.tmax) goto 200
    do 40 i=2,nf
    if(tsc(ipf(i)).ge.tmax) goto 45
    if(kdate(ipf(i)).gt.kdate(ipf(1))) goto 45
    if(tsc(ipf(i)).gt.scmax) goto 45
    40 continue
    45 nfs=i-1
c   write(6,426) scmax,nfs
426 format(' scmax=',f20.2,', nfs=',i5)
    if(nfs.le.0) goto 200
    if(kdate(ipf(1)).gt.kday1) call opnphs(phsfil,kdate(ipf(1)),kday1)
c
c
c   read in all 5day phases for one time period
50 ip=1
    do 70 i=1,nfs
    ku=6+ipf(i)
55 read(line(ipf(i)),440) ksta(ip),prmk(ip),kyr(ip),kmo(ip),
    2 kdy(ip),khr(ip),kmn(ip),sec(ip),tcr(ip),fmp(ip)
c   check for blank line
    if(ksta(ip).eq.blank) goto 60
    if(kyr(ip).eq.0) goto 60
c   check for next day
c   write(6,607) kdy(ip),kdy(1)
607 format(' kdy(i)',i5,'kdy(1)',i5)
    if(kdy(ip).gt.kdy(1)) goto 70
440 format(2a4,1x,3i2,2i2,f5.2,41x,f5.2,f5.0)
c   convert arrival time to seconds
    if(mod(kyr(ip),4).eq.0) then
        dayjul=float(lmo(kmo(ip))+kdy(ip))
    else
        dayjul=float(imo(kmo(ip))+kdy(ip))
    end if
    tsc(ip)=(dayjul*1440.0+float(khr(ip)*60+kmn(ip)))*60.0+sec(ip)
    if(tsc(ip).gt.scmax) goto 70
    ip=ip+1
    iptot=iptot+1
60 read(ku,400,end=68) line(ipf(i))
    goto 55
68 write(line(ipf(i))(12:15),445)
445 format('1299')
70 continue
c
c
    ncs=ip-1
    if(ncs.eq.0) goto 200
    do 80 i=1,ncs
80 iptr(i)=i
    call sort(tsc,iptr,ncs)
c   write to phsfil in sorted order

```

```

      ir=iptr(1)
      if((tsc(ir)-tlast).lt.5.0) goto 85
      write(1,460) blank
      nev=nev+1
85  write(1,440) ksta(ir),prmk(ir),kyr(ir),kmo(ir),
      2 kdy(ir),khr(ir),kmn(ir),sec(ir),tcr(ir),fmp(ir)
      ir1=ir
      do 100 i=2,ncs
      ir=iptr(i)
c  if time difference more than 5 sec, start new event
      if((tsc(ir)-tsc(ir1)).lt.5.0) goto 90
      write(1,460) blank
      nev=nev+1
90  write(1,440) ksta(ir),prmk(ir),kyr(ir),kmo(ir),
      2 kdy(ir),khr(ir),kmn(ir),sec(ir),tcr(ir),fmp(ir)
      ir1=ir
100 continue
460 format(a4)
      tlast=tsc(iptr(ncs))
      goto 30
200 write(2,470) iptot,nev
      write(6,470) iptot,nev
470 format(' end merge program, ',i6,' arrival times,',i5,' events')
      do 210 i=1,nf
      ku=i+6
      close(ku)
210 continue
      close(1)
      close(2)
      stop
      end

```

c
c
c

```

      subroutine sort(x,iptr,n)
c  indirect sort routine from Meissner&Organick p352
      dimension x(n),iptr(n)
      do 5 i=1,n
      iptr(i)=i
5  continue
      n1=n-1
      do 10 j=1,n1
      next=iptr(j+1)
      do 20 i=j,1,-1
c  write(6,400) j,i,iptr(i)
400  format(' j=',i4,', i=',i4,', iptr(i)=',i4)
      if(x(next).ge.x(iptr(i))) goto 9
      iptr(i+1)=iptr(i)
20  continue
9  iptr(i+1)=next
10  continue
      return
      end

```

c

```

      subroutine opnphs(phsfil,kday,kday1)
c  create file with proper name for day and open
      character*40 phsfil
      close(1)
      kday1=kday
      n=1

```

```
    phsfil(5:9)='.phs1'  
52 write(phsfil(1:4),420) kday  
420 format(i4)  
    if(phsfil(1:1).eq.' ') phsfil(1:1)='0'  
    open(unit=1,file=phsfil,status='new',err=100)  
    rewind(1)  
    return  
100 continue  
    n=n+1  
    write(phsfil(9:9),430) n  
430 format(i1)  
    open(unit=1,file=phsfil,status='new',err=100)  
    rewind(1)  
    return  
end
```

MRCPL*

THIS PROGRAM MERGES HYPO71 PHASE LISTS TOGETHER CREATING
ONE SUMMARY FILE AND ONE PHASE FILE FOR EACH DAY.

```
character*80 filnam(5),line(5)
character*4 ksta(1000)
character*1 rfmp(1000)
character*4 prmk(1000),blank
double precision tsc(1000)
real sec(1000),tcr(1000),fmp(1000)
integer ipf(12),kdate(12),kyr(1000),kmo(1000),
2 kdy(1000),khr(1000),kmn(1000),
3 iptr(1000),imo(12),lmo(12)
c This data is for NON-leap years
  data imo/0,31,59,90,120,151,181,212,243,273,304,334/
c This data is for LEAP years
  data lmo/0,31,60,91,121,152,182,213,244,274,305,335/
  blank='
  nev=0
  tlast=0
  iptot=1
  kday=0
  open(1)
  open(3)
c allow 5 sec interval between first readings for an
c event
  5 srtlim=5.0
c limit possible time in seconds
  tmax=31700000.0
c read in filenames
  do 10 nf=1,5
    read(5,400,end=20) filnam(nf)
  c    write(6,400) filnam(nf)
  10 continue
400 format(a80)
  20 nf=nf-1
    open(unit=2,file='mrgpl.out')
c open 5day phase files for each station
  do 25 i=1,nf
    ku=6+i
    open(unit=ku,file=filnam(i),blank='zero',status='old',
2 form='formatted')
    write(2,401) filnam(i),ku
    write(6,401) filnam(i),ku
401 format(' file= ',a40,' Unit=',i4)
    rewind(ku)
c read first line
  22 read(ku,400) line(i)
c check for blank line
  if(line(i)(1:4).eq.blank) goto 22
  25 continue
c
c
c read first phase for each file and sort
  30 do 35 i=1,nf
    read(line(i),430) jyr,jmo,jdy,jhr,jmn,sc
    kdate(i)=jmo*100+jdy
  c    write(6,431) jyr,jmo,jdy,jhr,jmn,sc
431 format(9x,5i2,f5.2)
```

```

      , if(mod(jyr,4).eq.0) then
      ,   dayjul=float(lmo(jmo)+jdy)
      else
      ,   dayjul=float(imo(jmo)+jdy)
      end if
      tsc(i)=(dayjul*1440.0+float(jhr*60+jmn))*60.0+sc
c   write(6,425) i,kdate(i),dayjul,tsc(i)
425 format(' i=',i4,', kdate=',i5,', dayjul=',
      2 f5.0,', tsc=',f20.2)
      35 continue
430 format(9x,5i2,f5.2)
      call sort(tsc,ipf,nf)
c
c
      if(kdate(ipf(1)).gt.kday1) call opnphs(kdate(ipf(1)),kday1)
c check for bad readings at the beginning of an event
      read(line(ipf(1)),435) iwt
c   write(6,611) iwt
611 format(i5)
435 format(7x,i1)
      if(iwt.lt.4) goto 37
      ku=6+ipf(1)
36   write(1,400) line(ipf(1))
      read(ku,400) line(ipf(1))
      read(line(ipf(1)),435) iwt
      if(iwt.ge.4) goto 36
      read(line(1),430) jyr,jmo,jdy,jhr,jmn,sc
      if(mod(jyr,4).eq.0) then
      ,   dayjul=float(lmo(jmo)+jdy)
      else
      ,   dayjul=float(imo(jmo)+jdy)
      end if
      tsc(1)=(dayjul*1440.0+float(jhr*60+jmn))*60.0+sc
37 scmax=tsc(ipf(1))+srtlim
c check for end of data
      if(tsc(ipf(1)).ge.tmax) goto 200
c write 1st reading to .sl list
      nev=nev+1
      write(3,400) line(ipf(1))
c
c
      do 40 i=2,nf
c check for same event on next phase list
      if(tsc(ipf(i)).ge.tmax) goto 45
      if(kdate(ipf(i)).gt.kdate(ipf(1))) goto 45
      if(tsc(ipf(i)).gt.scmax) goto 45
40 continue
45 nfs=i-1
c   write(6,426) scmax,nfs
426 format(' scmax=',f20.2,', nfs=',i5)
      if(nfs.le.0) goto 200
c
c
c read in all phases for one event
50 ip=1
      do 70 i=1,nfs
      ku=6+ipf(i)
55 read(line(ipf(i)),440) ksta(ip),prmk(ip),kyr(ip),kmo(ip),
      2 kdy(ip),khr(ip),kmn(ip),sec(ip),tcr(ip),fmp(ip),rfmp(ip)
c check for blank line to end event

```

```

      if(ksta(ip).eq.blank) goto 69
      if(kyr(ip).eq.0) goto 69
440 format(2a4,1x,3i2,2i2,f5.2,4lx,f5.2,f5.0,a1)
c convert arrival time to seconds
      if(mod(kyr(ip),4).eq.0) then
          dayjul=float(lmo(kmo(ip))+kdy(ip))
      else
          dayjul=float(imo(kmo(ip))+kdy(ip))
      end if
      tsc(ip)=(dayjul*1440.0+float(khr(ip)*60+kmn(ip)))*60.0+sec(ip)
      ip=ip+1
      iptot=iptot+1
60 read(ku,400,end=68) line(ipf(i))
      goto 55
68 write(line(ipf(i))(12:15),445)
445 format('1299')
      goto 70
c read first line of next event
69 read(ku,400,end=68) line(ipf(i))
      if(line(ipf(i))(1:4).eq.blank) goto 69
70 continue
c
c
c sort arrivals and write to phase file
      ncs=ip-1
      if(ncs.eq.0) goto 30
      do 80 i=1,ncs
80 iptr(i)=i
      call sort(tsc,iptr,ncs)
c write to phsfil in sorted order
      ir=iptr(1)
85 write(1,440) ksta(ir),prmk(ir),kyr(ir),kmo(ir),
      2 kdy(ir),khr(ir),kmn(ir),sec(ir),tcr(ir),fmp(ir),rfmp(ir)
      irl=ir
      do 100 i=2,ncs
      ir=iptr(i)
90 write(1,440) ksta(ir),prmk(ir),kyr(ir),kmo(ir),
      2 kdy(ir),khr(ir),kmn(ir),sec(ir),tcr(ir),fmp(ir),rfmp(ir)
      irl=ir
100 continue
      write(1,460) blank
460 format(a4)
      tlast=tsc(iptr(ncs))
      goto 30
200 write(2,470) iptot,nev
      write(6,470) iptot,nev
470 format(' completed merging phase lists, ',i6,' arrival times,',
      2 i5,' events')
      do 210 i=1,nf
      ku=i+6
      close(ku)
210 continue
      close(1)
      close(3)
      close(2)
      stop
      end

```

c
c
c

```

, subroutine sort(x,iptr,n)
c indirect sort routine from Meissner&Organick p352
  double precision x(n)
  dimension iptr(n)
  do 5 i=1,n
  iptr(i)=i
5 continue
  n1=n-1
  do 10 j=1,n1
    next=iptr(j+1)
    do 20 i=j,1,-1
c      write(6,400) j,i,iptr(i)
400    format(' j=',i4,' i=',i4,' iptr(i)=' ,i4)
        if (x(next).gt.x(iptr(i))) goto 9
        iptr(i+1)=iptr(i)
    20    continue
    9    iptr(i+1)=next
    10    continue
  return
  end

c
  subroutine opnphs(kday,kday1)
c create file with proper name for day and open
  character*40 phsfil,slfil
  close(1)
  close(3)
  kday1=kday
  n=1
  phsfil(5:8)='.p11'
  slfil(5:8)='.s11'
52 write(phsfil(1:4),420) kday
420 format(i4)
  if(phsfil(1:1).eq.' ') phsfil(1:1)='0'
  slfil(1:4)=phsfil(1:4)
  open(unit=1,file=phsfil,status='new',err=100)
  rewind(1)
  open(unit=3,file=slfil,status='new',err=100)
  rewind(3)
  return
100 continue
  n=n+1
  write(phsfil(8:8),430) n
  write(slfil(8:8),430) n
430 format(i1)
  open(unit=1,file=phsfil,status='new',err=100)
  rewind(1)
  open(unit=3,file=slfil,status='new',err=100)
  rewind(3)
  return
  end

```