Appendix I

Data Acquisition and Reduction

This appendix describes data acquisition and reduction for the different types of experiments used to generate data presented in this thesis. Listings of computer programs that have been used for this purpose are provided at the end of this Appendix. Primary gas chromatography (GC) and mass spectrometry (MS) data acquisition and reduction programs are written in 'Turbobasic' (Borland International Inc., Scotts Valley, CA). Lotus 123 (Lotus Development Corporation, Cambridge, MA) has been used for further data reduction. These have been used on an IBM-PC (XT) microcomputer.

1. Data Acquisition

Data can be recorded from the MS or from the GC alone; also, GC-MS data where both data are followed (and are linked) can be followed. Three main data acquisition programs are used. The first, called "LOCTMS", is used to locate digital/analog (D/A) mass spectrometric peak locations; these are then used in the other two programs. The second, called "MSONLY", records MS data continuously for a user-programmed duration. Up to 5 masses can be monitored; data is collected at 2 readings/s for each mass programmed. The third program is called "TSNEWGC" and records GC-MS data; due to the lengthy nature of these experiments, data is stored only if a GC or MS peak is found. Peak-sensing criteria are built-in for the GC and MS separately and can be user-modified. This

program is used for either GC data alone or GC-MS data. Data is collected on a GC channel and up to 5 MS channels at 2 readings/s. Detailed descriptions of these programs follow. All the programs are user-friendly and will prompt the operator to perform the next step at each stage.

1.1 LOCTMS

This program is used to search the mass spectrum for peak locations. The MS is first set up for computer control: 'Program' EXTernal, 'Function' MULTiplier and 'Gain' 6. The cursor keys now controls the external D/A location that sets the amu being scanned on the MS. The '6' key is coarse up, the '4' key is coarse down, the '8' key is fine up and the '2' key is fine down. An input of 'G' prompts the user to input the desired gain. Finally, the computer provides the desired D/A location after a peak is found.

1.2 MSONLY

MSONLY is used to generate on-line MS data. Initial input consists of a filename to save to on the 'B' drive, and the number of masses that are of interest, as well as the total length of the experiment (which is limited by the number of masses followed). Then on further prompting, the masses desired are programmed. A D/A location and external gain factor for each mass are required, and can be searched for, since 'LOCTMS' is built in to this program, or keyed in if already known. At this point, the input parameters are saved, and the computer display indicates its readiness to record data after a keystroke input prompt from the operator. This program is used for experiments such as temperature programmed surface reaction (TPSR; see Chapter 2 or 4) or on-line isotopic transients

(Chapter 4).

1.3 TSNEWGC

This program is used to acquire GC and GC-MS data. It is used for computer control of the 10-port GC 'load/inject' valve and the 16-sample loop Multiposition GC valve(MPV) and also to externally start up the Perkin-Elmer Sigma 3B GC and subsequently the MS in order to analyze all the samples stored in the MPV.

The first part of the input consists of MS parameters, which are entered in a manner identical to that used in 'MSONLY'. Then GC loop data is entered; the user keys in the number of loops to be filled, the load time for each of these loops and the total time required for a sample to elute through the GC column. (This depends on the GC temperature program used.) Also programmed is the delay to start recording MS data after GC start-up; this allows the big injection air wave to elute through the MS without being recorded or changing the He carrier baseline. At this point, the computer waits for a user keypad prompt to begin the experiment.

The GC-MS temperature program takes = 30 min for each loop; there are a possible maximum of 16 such loops. So a large amount of data is generated as GC and MS readings versus time. Therefore, to simplify and limit data acquisition, the computer only stores data if a peak is detected in the GC or on any of the MS traces. Peak retention times and GC or MS reading vs. time data for 20 s around a peak are recorded. This is sufficient as each peak lasts about 6s. GC or MS peak criteria can be altered by the user in the program. MS criteria, which are more critical (since MS data is in general more noisy than the GC) can be changed by changing the input text file,

'mspkpar.txt'. This has 4 parameters, {n_1, n_2, n_3 and n_4}: n_1 is the minimum step size in numbers from 0-4096. A number 1 corresponds to a 2.44 mV change in the 0-10V MS output to the A/D converter. The value used here is 2. n_2 is the minimum number of consecutive rising steps, each equal to or greater than n_1. A value of 3 is used here. n_3 is the absolute peak height above the baseline and a value of 15 is used. Finally, n_4 sets the MS delay time between sampling to allow the MS readings to settle. This is currently set at 3000. Optimization of these parameters allows picking up almost all peaks that are visible on the screen, while minimizing the pick-up of random noise.

When the operator starts up the computer to begin the experiment, the computer initially 'homes' the 16-loop MPV and sets the 10-port valve to the LOAD position. It then switches the MPV to loop 2 and proceeds to fill loop 2 for the programmed loading time. Before it switches to loop 3, the computer beeps and awaits a user-prompt. This is to allow the operator to do the isotopic transient switching. After the loading has re-commenced, the computer loads the required number of loops for the programmed loading times. The last loop to be filled is always loop 1. After all loop-loading is complete, the computer awaits a user-prompt to commence sample-injection and GC-MS data acquisition. After this, it also waits for a GC-Ready signal from the GC. (Usually, it is ensured that when loading is complete, the GC is also ready for injection.)

When the GC is ready, the computer externally starts up the GC temperature program and immediately switches the 10-port GC

valve to the 'INJECT' position. Loop1 is the first loop to elute through the column. GC-MS data is now acquired for the input time/loop. After the sample has eluted, the IBM-PC writes the peak information to disk and again waits for the GC to get ready before switching the MPV to the next loop for injection. When saving the data for loop n, the computer also checks that there is sufficient space for the next loop data, and if found to be insufficient, a message appears on the screen, prompting the user to insert another disk and hit enter, before the next loop is injected. After all the loaded loops are eluted, the computer stops data acquisition. There is an error-handling Turbobasic subroutine built into TSNEWGC that writes into an error file if the program is abruptly terminated in the middle of an experiment. This aids trouble-shooting if the program fails.

2.0 Data Reduction

There are two programs which read data acquired by 'MSONLY' and 'TSNEWGC'. This data is then stored in files that are imported into LOTUS 123 worksheets, which are used for the next step in data analysis. Finally, if model fitting is required, a FORTRAN program, 'MINIMUM' is used.

2.1 Reading MS or GC-MS data

'MSONLY' data is read by a program called 'READMS'. This program will display on the screen the mass spectrum of any of the masses recorded; these can also be plotted on a Hewlett Packard plotter to provide a hard copy. There are two little cursors that appear on the screen that can be moved using the right and left

arrow cursor keys. These can be positioned where desired and if 'B' is entered, a new baseline will be drawn between them. If they are positioned and 'return' is entered, the area between the two cursors will be integrated and printed out on the screen. The average carrier He signal is also computed. The data for any mass can also be saved to disk as time-reading x-y pairs for further analysis by LOTUS. Calibrations are then used to convert the MS areas to actual amounts of products. MS data is usually normalized to the He signal to account for changes in the MS electron multiplier gain.

'TSNEWGC' data is read by a program called 'READTS'. This allows us to look at any loop. GC or MS data can be separately scanned, peak by peak, and peak areas can be integrated using the two screen cursors. This takes a lot of time when many peaks are present and so in both cases, the data can be automatically integrated using the 'Autointegrate' option. The most used feature of this program is the 'Autoalign' option; this will autointegrate the GC and every MS spectrum recorded for a given loop, then align the GC peaks to a library of known compound based on GC peak retention times, and finally align the MS peaks to the corresponding GC peak based on input GC-MS time delay for each known GC peak. The final step is a printout of all the data; integrated GC peaks and each MS channel and finally the aligned, integrated GC/MS data. The GC-MS data is saved to disk.

The GC- MS data is first examined for accurate peak identification and GC-MS peak line-up. (Since retention times are very sensitive to GC carrier flow rate and temperature program, there can be some variations.) Also, at low concentrations,

especially in the MS, peaks are not well-separated and require manual integration using the cursors. Any such corrections are noted on the GC-MS data hard copy. The next step in data reduction uses a LOTUS 123 spread sheet. Usually there may be up to 60-70 GC peaks and MS peaks corresponding to these on 4 masses for each sample loop, and a possible 16 such samples.

2.2 LOTUS 123 Data Analysis

The large quantity of data necessitates the use of three worksheets for data analysis. First, the GC-MS data are imported, loop by loop, into a LOTUS worksheet. This is the 'raw' data, and in this worksheet, blanks are inserted for missing peaks and any corrections in peak identification or areas is incorporated. Data for each loop consists of 5 columns; peak ID, peak number, GC area, and MS areas normalized by the He signal for ^{12} and ^{13} labelled compounds.

Once the 'GCMS.RAW' spreadsheet is created, GC data is analyzed in a separate spreadsheet. Here, GC areas are copied from the .RAW worksheet for each loop; calibrations for C_{1-14} species are used to convert these areas to concentrations. For each loop, the sum of all the concentrations for each carbon number is calculated; these sums are further used to obtain an ASF chain growth parameter for each loop. Finally, a calculation is made of the ΣnN_{Cn} product to calculate overall CO conversion for each loop.

The final worksheet is used for GC-MS data analysis. Here, for each loop, GC concentrations are imported from the GC worksheet;

MS data is copied from the .RAW worksheet. Here, MS calibrations are used to obtain ^{12} and ^{13}C MS peak concentrations. These are then divided by the corresponding GC peak concentration to obtain fractional isotopic concentrations. At a given carbon number, the MS concentrations of all detected peaks are added together and divided by the total concentrations of those peaks (obtained from GC data) for ^{12}C-labelled and ^{13}-C labelled isotopic fractions. MS data is often noisy, and there is also a 1% ^{12}CO impurity in the feed ^{13}CO; hence, further normalization is necessary. The steady state loop data in either isotope is used for this purpose. (Typically, loop 2 is the steady-state in $^{12}CO/H_2$ and loop 10 the steady-state in ^{13}CO/H2.) All data is converted to isotopic rise data, by using the fact that $F_{rise} + F_{fall} = 1.0$ at any time; for each time, the average of the ^{12}C and ^{13}C rise curve for each product is calculated. Finally, 16 points of t, $F_{rise}(t)$ are obtained. This data is now fitted to the chain growth model using a FORTRAN program, 'Minimum'. This is described in Appendix 3.

MSONLY.TXT

```
1 REM Turbo Basic program to run MS for Kamala ver 1.0 04-26-88
10 KEY OFF:CLS:REM turn off key line
12 PRINT "Switch valve interface to manual"
14 PRINT "Type any key to continue"
16 IF INKEY$="" THEN 16
20 PRINT "Place DATA disk into drive B:"
40 INPUT "Enter file name you want to save on ";FLNAME$
60 OPEN "b:"+FLNAME$+".cmy" FOR OUTPUT AS #2
70 PRINT "Enter comments, terminate with a blank line"
80 INPUT COMMENT$:IF COMMENT$="" THEN 92
90 PRINT#2,COMMENT$:GOTO 80
92 CLOSE 2
100 BASE1=&H300
110 HEXTABLE$="0123456789ABCDEF"
120 OUT BASE1+3,&H90:REM init Metrabyte PIO12
130 INJECTOR=BASE1+1:LOOPVALVE=BASE1+2:STATES=BASE1
140 LD=2:INJECT=4:FIDSTART=1:INJNORM=&H6
150 STP=&H80:HOME=&H40:SELPOS=&H20:LOOPNORM=&HFF
160 OUT INJECTOR,INJNORM
170 OUT LOOPVALVE,LOOPNORM
180 OPEN "com1:1200,n,7,1,rs,cs,ds,cd"AS 1:REM open channel of
Omega
200 BASE2=&HE6A0
210
ATOD=BASE2:DTOA=BASE2+4:COUNTER=BASE2+8:PLOTTER=BASE2+12
220
ADSTART=BASE2+&H1C:CTRESET=BASE2+&H1D:RDYCLR=BASE2+&H1E:
SCAN1=BASE2+&H1F
230 REM program 8255s of the GCMS interface
240 OUT ATOD+3,&H92:OUT DTOA+3,&H89:OUT COUNTER+3,&H92:OUT
PLOTTER+3,&H80
250 REM define dafault I/O condition
260 OUT COUNTER+2,16+4+1:REM select time-base 2 Hz
270 OUT DTOA+1,0:OUT DTOA,0:REM set mass program to 0
280 OUT PLOTTER+2,255:OUT PLOTTER+1,255:OUT PLOTTER,255:REM
zero plotter
```

```
290 OUT ATOD+2,0+6*32:gain%=6:REM select ch.0 and gain factor 6
370 DEF FNFREESPACE(DRIVE%) 'find free space on drive
380 REG 4,DRIVE%            '0=default,1=A:,2=B:,etc.
390 REG 1,&H3600            'AH=function number
400 CALL INTERRUPT &H21     'DOS function call
410 FNFREESPACE=CSNG(REG(2))*REG(3)*REG(1)
420 END DEF
490 REM
------------------------------------------------------------
------------
510 GOSUB 5000:REM get MS parameters
520 numm=nummass%
550 NEED=(NUMM*5000+1000)
552 IF NEED>FNFREESPACE(2) THEN PRINT "Not enough disk space
!!":END
562 PRINT
570 PRINT "Turn damping control of UTI to min."
580 PRINT "Select gain control of UTI to (-11)"
600 PRINT "Type any key when ready"
610 IF INKEY$="" THEN 610
620 REM DATA ACQ. START HERE-------------------------
622 MASSPG=MASSLOC(0):GOSUB 1000:REM move mass program to
1st mass
624 OUT ATOD+2,0+MASSGAIN%(0)*32:REM select 1st mass gain and
ch. 0
630 STARTDATE$=DATE$:STARTTIME$=TIME$:IF STARTDATE$<>DATE$
THEN 630
632 GOSUB 4000:REM save parameters
644 SCREEN 2:CLS:REM graphics screen
680 GOSUB 3000:REM data acquisition procedure
682 out atod+2,0+6*32:rem switch to ch.0 and gain 6
690 GOSUB 4200:REM data saving procedure
702 SCREEN 0:REM text screen again
710 FINISHDATE$=DATE$:FINISHTIME$=TIME$:IF FINISHDATE$<>DATE$
THEN 710
720 PRINT "experiment starts at ";STARTDATE$;" ";STARTTIME$
730 PRINT "          finishes at ";FINISHDATE$;" ";FINISHTIME$
740 OUT ATOD+2,0+6*32:gain%=6:REM select ch.0 and gain factor 6
```

```
990 END:REM end of main
program----------------------------------------------
1000  REM  output  dtoa  data----------------------------
1010 IF MASSPG<0 THEN MASSPG=0
1020 IF MASSPG>65535! THEN MASSPG=65535!
1030  MVAL=INT(MASSPG/256):LVAL=MASSPG-MVAL*256
1040 OUT DTOA+1,MVAL:OUT DTOA,LVAL
1050 RETURN
3000  REM data acquisition
procedure---------------------------------
3030  element%=0:x1=0:y1=199
3010 rdyfg=(INP(ATOD+1) AND &H20):REM WAIT UTIL COUNTER IS
READY
3013 if rdyfg=0  then goto 3010
3014 OUT RDYCLR,0:REM clear counter ready flag

3200  REM  read  masses---------------------
3202 if nummass%=0 then 3400:rem no mass
3210 FOR I=0 TO NUMMASS%-1
3220 REM mass is selected in previous loop
3260 OUT ADSTART,0:REM strobe A to D
3270 MASSREAD%=(INP(ATOD+1) AND 15)*256+INP(ATOD):REM read
mass  intensity
3280 MASSINT%(element%,I)=MASSREAD%:REM put reading into
buffer
3290 IF I=(NUMMASS%-1) THEN NEXTMASS%=0 ELSE NEXTMASS%=I+1
3300  out  atod+2,0+massgain%(nextmass%)*32:rem  select  new  gain
3310 MASSPG=MASSLOC(NEXTMASS%):GOSUB 1000:REM program
nextmass
3320  for  xxx=0  to  20:next:rem  delay  gen
3380  NEXT  I:REM  NEXT  MASS----------------------
3390  y2=199-massread%/22:x2=x1+1
3400  line  (x1,y1)-(x2,y2)
3402  x1=x2:y1=y2
3404  if  x1>600  then  x1=0:y1=199:cls
3410 ELEMENT%=ELEMENT%+1
3412 LOCATE 23,1:PRINT "At point ";ELEMENT%
```

```
3430 if element% < numpoint% then 3010
3470 RETURN:REM data acq proedure
return----------------------------
4000 REM Saving parameters routine--------------------------
4010 PRINT "Saving parameters; please wait"
4020 OPEN "b:"+FLNAME$+".paz" FOR OUTPUT AS #2
4040 PRINT#2,NUMMASS%
4042 print#2,numpoint%
4050 FOR I=0 TO NUMMASS%-1
4060 PRINT#2,MASSNUM%(I),MASSGAIN%(I)
4070 NEXT I
4140 CLOSE 2:RETURN:REM ----------
4200 REM saving data ----------
4202 LOCATE 1,1
4204 PRINT "Saving data, please wait"
4340 DEF SEG=VARSEG(MASSINT%(0,0))
4350 LENGTH=VARPTR(MASSINT%(numpoint%,NUMMASS%-1))
4360 BSAVE "b:"+FLNAME$+".mas",0,LENGTH
4590 RETURN:REM=========
5000 REM get MS parameters ----------
5010 INPUT "Enter number of masses interested ";NUMMASS%
5011 maxpoint%=int(16000/nummass%):print "Max. # of points
";maxpoint%
5012 input "Enter number of data points per mass ";numpoint%
5013 if numpoint%>maxpoint% then print "out of range":goto 5012
5014 if nummass%=0 then 5300
5016 dim massint%(numpoint%,nummass%)
5017 dim
massnum%(nummass%),massloc(nummass%),massgain%(nummass%)
5020 PRINT "Locate mass :"
5030 PRINT "     to the prompt of 'LOCATE' use keypad key"
5040 PRINT "       '8' for fine up"
5050 PRINT "       '2' for fine down"
5060 PRINT "       '6' for coarse up"
5070 PRINT "       '4' for coarse down"
5080 PRINT "     to move mass around "
5082 PRINT "        'K'    to use key entry"
5084 PRINT "        'G'    to select gain"
```

```
5090 PRINT "     and 'ENTER' to select location":PRINT
5100 MASSPG=0:GOSUB 1000:REM preset to mass 0
5110 FOR I=0 TO NUMMASS%-1
5120 PRINT I+1;"  ";
5130 INPUT "enter mass number ";MASSNUM%(I)
5140 PRINT "LOCATE    ";
5150 TEMP$=INKEY$:IF TEMP$="" THEN 5150
5160 IF ASC(TEMP$)=13 THEN 5250:REM select location of mass
5162 if temp$="K" then 5242
5164 if temp$="G" then gosub 5400:rem change gain%
5170 IF LEN(TEMP$)<>2 THEN 5150
5180 TEMP=ASC(RIGHT$(TEMP$,1))
5190 IF TEMP=72 THEN MASSPG=MASSPG+1
5200 IF TEMP=80 THEN MASSPG=MASSPG-1
5210 IF TEMP=77 THEN MASSPG=MASSPG+100
5220 IF TEMP=75 THEN MASSPG=MASSPG-100
5230 GOSUB 1000:REM program D to A
5240 GOTO 5150
5242 input "Enter D to A setting ";MASSPG:gosub 1000
5250 PRINT "at D to A setting of ";MASSPG
5260 MASSLOC(I)=MASSPG
5270 MASSGAIN%(I)=gain%
5272 print "gain control for mass";massnum%(i);" is";massgain%(i)
5280 PRINT
5290 NEXT I
5300
RETURN:REM-----------------------------------------------
5400  rem  change  gain%  --------------------
5410 input "enter mass gain (0 to 7) ";gain%
5420 if (gain%<0) or (gain%>7) then 5410:rem out of range
5450  out atod+2,gain%*32:rem  switch  gain
5460    return:rem    -------------------------
```

READMS.TXT

```
100 REM loading data from MSONLY program ver 1.0 5-27-88 by
Henry Chan
110 base2=&he6a0
120 plotter=base2+12
130 out plotter+3,&h80
140 xplot%=0:yplot%=0:gosub 4200:rem zero plotter
310 PRINT "Place data disk in drive B:, type any key to continue"
312 IF INKEY$="" THEN 312
314 FILES "b:*.paz"
320 INPUT "Enter file name ";FLNAME$
330 OPEN "b:"+FLNAME$+".cmy" FOR INPUT AS #2
340 WHILE NOT EOF(2)
350 LINE INPUT#2,COMMENT$
360 PRINT COMMENT$
370 WEND
380 CLOSE 2
390 INPUT "Is this the right one (y/n) ";YES$
400 IF YES$="y" THEN 430
410 IF YES$="n" THEN 310
420 GOTO 390
430 PRINT
500 REM Loading parameters-------------------------
510 PRINT "Loading parameters, please wait"
520 OPEN "b:"+FLNAME$+".paz" FOR INPUT AS #2
540 INPUT#2,NUMMASS%:PRINT "Number of masses observed
=";NUMMASS%
542 input#2,numpoint%
54' dim
massint%(numpoint%,nummass%),massnum%(nummass%),massgain%(
nummass%)
546 dim ydel%(10)
550 FOR I=0 TO NUMMASS%-1
560 INPUT#2, MASSNUM%(I),MASSGAIN%(I)
570 PRINT "MASS";I+1;"is";MASSNUM%(I),"Gain factor";MASSGAIN%(I)
580 NEXT I
660 CLOSE 2:REM parametes loaded----------------
```

```
1000  REM  read  data----------------------
1100 print "reading data; please wait"
1110 def seg=varseg(massint%(0,0))
1120 bload "b:"+flname$+".mas",0
1180 screen 2
1190 rem data
loaded-----------------------------------------------
1200 rem display data
1210 for i=0 to nummass%-1
1220 print "mass";massnum%(i);" used gain factor";massgain%(i)
1230 next i
1240 input "Which mass (0 to exit)";massdisp%
1241 if massdisp%=0 then 2990
1250  msptr%=-1
1252 for i=0 to nummass%-1
1260 if massnum%(i)=massdisp% then msptr%=i
1262 if massnum%(i)=4 then refptr%=i
1270 next i
1280 if msptr%=-1 then print "mass not found; try again":goto 1200
1281 input "Save in x,y pair (y/n) ";an$
1282 if an$<>"y" then 1284  else gosub 4400
1283 input "Do you want to continue to plot (y/n) ";an$:if an$<>"y"
then 2980
1284 input "Plotter output (y/n) ";an$
1285 if an$="y" then plotyes=1 else if an$="n" then plotyes=0 else
goto 1282
1286 if plotyes then print "Plotting is selected"
1290 print "Calculating scale; please wait"
1300 ymax%=-1:ymin%=4096
1310 for i=0 to numpoint%-1 step int(1+numpoint%/600)
1320 y%=massint%(i,msptr%)
1330 if y%>ymax% then ymax%=y%:xmax%=i
1340 if y%<ymin% then ymin%=y%:xmin%=i
1350 next i
1352 print ymax%,xmax%,ymin%,xmin%
1360  yscale%=(ymax%-ymin%)/170+1
1362 print yscale%
1370  yoffset%=ymin%
```

```
1380 print yoffset%
1390 print "Resetting plotter, lift pen and type any key to continue"
1400 if inkey$="" then 1400
1410 xplot%=0:yplot%=0:gosub 4200:rem zero plotter
2000 print "Type any key to start plotting routine"
2004 if inkey$="" then 2004
2006 base1%=0:base2%=numpoint%-1:head%=base1%:tail%=base2%
2010 cls:gosub 3000:xplot=0
2090 xscale%=numpoint%/600+1
2094 intsts=0:integral=0:flag%=0
2096 x1=0:y1=190-(massint%(0,msptr%)-yoffset%)/yscale%
2098 if plotyes then inc=1 else inc=int(1+numpoint%/600)
2100 for i=0 to numpoint%-1 step inc
2110 x2=i/xscale%:y%=massint%(i,msptr%):yplot%=y%
2120 y2=190-(y%-yoffset%)/yscale%
2140 line (x1,y1)-(x2,y2)
2141 if plotyes then gosub 4200
2142 y1=y2:x1=x2:xplot=xplot+0.90:xplot%=int(xplot)
2290 if inkey$=chr$(27) then i=numpoint%:flag%=1
2300 next i
2302 if flag%=1 then 2980
2304 plotyes=0
2310 locate 2,1:print "Integration routine; use 'ESC' to return to
main menu "
2330 y1=190-(massint%(base1%,msptr%)-yoffset%)/yscale%
2340 y2=190-(massint%(base2%,msptr%)-yoffset%)/yscale%
2350 line (base1%/xscale%,y1)-(base2%/xscale%,y2):rem line base
line
2352 locate 4,1:print "BASE LINE " massint%(base1%,msptr%);" at
";base1%;" ";
2354 print massint%(base2%,msptr%);" at ";base2%;"                    ";
2360 y1=190-(massint%(head%,msptr%)-yoffset%)/yscale%
2370 y2=190-(massint%(tail%,msptr%)-yoffset%)/yscale%
2380 line (head%/xscale%,y1-10)-(head%/xscale%,y1-20)
2390 line (tail%/xscale%,y2-10)-(tail%/xscale%,y2-20)
2392 locate 5,1:print massint%(head%,msptr%); " at ";head%;"  ";
2394 print massint%(tail%,msptr%);" at ";tail%;                "
";
```

```
2400 ky$=inkey$:if ky$="" then 2400
2410 ky%=asc(ky$)
2420 if ky%=27 then 2980
2430 if ky%=13 then 2600
2440 if ky$="B" then gosub 2800:base1%=head%:base2%=tail%:goto
2330
2442 if ky$="R" then goto 2010:rem redraw
2445 if ky$="8" then if head%<numpoint%-21 then
nhead%=head%+20:goto 2510
2446 if ky$="2" then if head%>20 then nhead%=head%-20:goto 2510
2447 if ky$="6" then if tail%<numpoint%-21 then
ntail%=tail%+20:goto 2510
2448 if ky$="4" then if tail%>20 then ntail%=tail%-20:goto 2510
2450 if len(ky$)<2 then 2400
2460 temp=asc(right$(ky$,1))
2470 if temp=72 then if head%<numpoint%-1 then nhead%=head%+1
2480 if temp=80 then if head%>0 then nhead%=head%-1
2490 if temp=77 then if tail%<numpoint%-1 then ntail%=tail%+1
2500 if temp=75 then if tail%>0 then ntail%=tail%-1
2510 line (head%/xscale%,y1-10)-(head%/xscale%,y1-20),0
2520 line (tail%/xscale%,y2-10)-(tail%/xscale%,y2-20),0
2530 head%=nhead%:tail%=ntail%
2540 goto 2360
2600 rem integrate
2610 locate 5,40:print "Integrating; please wait                "
2620 sum=0:sum1=0:offs=0
2622
slope=(massint%(base2%,msptr%)-massint%(base1%,msptr%))/(base
2%-base1%)
2624 yx0=massint%(base2%,msptr%)-slope*base2%
2630 offs=((tail%+head%)/2*slope+yx0)*(tail%-head%+1)
2640 for ji=head% to tail%
2650 sum=sum+cdbl(massint%(ji,msptr%))
2652 sum1=sum1+cdbl(massint%(ji,refptr%))
2660 next ji
2670 sum=sum-abs(offs)
2680 locate 6,40:print "Mass 4 average
";int(abs(sum1/(tail%-head%+1)));
```

```
2682 locate 5,40:print "Integral= ";int(sum);"          ";
2690 goto 2400
2800 rem clear base line
2810  y1=190-(massint%(base1%,msptr%)-yoffset%)/yscale%
2820  y2=190-(massint%(base2%,msptr%)-yoffset%)/yscale%
2830 line (base1%/xscale%,y1)-(base2%/xscale%,y2),0:rem line base
line
2840  return
2980 cls:goto 1200
2990 end
3000 rem ymax% subroutine
3020 locate 1,1:print flname$;" MASS"massdisp%;"
YMAX";ymax%;"at";xmax%;
3022 print " YMIN";ymin%;"at";xmin%
3030 print "use 'ESC' key to abort display";
3031 if plotyes then print "; Plotting is selected";
3032 locate 24,1:print i;:locate 25,73:print numpoint%;
3040  return
4200  rem  output  to  plotter  ----------------
4210 if xplot%<0 then xplot%=0
4220 if xplot%>4095 then xplot%=4095
4230 if yplot%<0 then yplot%=0
4240 if yplot%>4095 then yplot%=4095
4250 plot0%=255-yplot% and 255
4260 plot2%=255-xplot% and 255
4270  plot1%=255-(int(xplot%/256)*16+int(yplot%/256))
4280 out plotter+2,plot2%:out plotter+1,plot1%:out plotter,plot0%
4290   return:rem  ---------------------------
4400  rem  saving  x,y  pair--------------------
4405 print "Make sure you have enough disk space in drive B:"
4410 input "Please enter new filename for X,Y pair ";flname1$
4412 open "b:"+flname1$ for output as #2
4420 for i=0 to numpoint%-1
4430  print#2,  i/2,massint%(i,msptr%)
4440 next i
4450 close 2
4460  return
```

TSNEWGC.TXT

```
1 REM Turbo Basic program to run GCMS for Kamala ver 3.10
04-29-90
2 on error goto 9000
4 open "mspkpar.txt" for input as 2
5
input#2,mspkstp%:input#2,mspkre%:input#2,mspkht%:input#2,msdel
ay%
6 close 2
10 KEY OFF:CLS:REM turn off key line
12 PRINT "Switch valve interface to manual"
14 PRINT "Type any key to continue"
16 IF INKEY$="" THEN 16
20 PRINT "Place DATA disk into drive B:"
40 INPUT "Enter file name you want to save on ";FLNAME$
50 mkdir
"b:\"+flname$:flnamed$=flname$:flname$="b:\"+flname$+"\"+flname$:
drive%=1
60 OPEN FLNAME$+".cmm" FOR OUTPUT AS #2
70 PRINT "Enter comments, terminate with a blank line"
80 INPUT COMMENT$:IF COMMENT$="" THEN 92
90 PRINT#2,COMMENT$:GOTO 80
92 CLOSE 2
100 BASE1=&H300
110 HEXTABLE$="0123456789ABCDEF"
120 OUT BASE1+3,&H90:REM init Metrabyte PIO12
130 INJECTOR=BASE1+1:LOOPVALVE=BASE1+2:STATES=BASE1
140 LD=2:INJECT=4:FIDSTART=1:INJNORM=&H6
150 STP=&H80:HOME=&H40:SELPOS=&H20:LOOPNORM=&HFF
160 OUT INJECTOR,INJNORM
170 OUT LOOPVALVE,LOOPNORM
180 OPEN "com1:1200,n,7,1,rs,cs,ds,cd"AS 1:REM open channel of
Omega
200 BASE2=&HE6A0
210
ATOD=BASE2:DTOA=BASE2+4:COUNTER=BASE2+8:PLOTTER=BASE2+12
220
```

```
ADSTART=BASE2+&H1C:CTRESET=BASE2+&H1D:RDYCLR=BASE2+&H1E:
SCAN=BASE2+&H1F
230 REM program 8255s of the GCMS interface
240 OUT ATOD+3,&H92:OUT DTOA+3,&H89:OUT COUNTER+3,&H92:OUT
PLOTTER+3,&H80
250 REM define dafault I/O condition
260 OUT COUNTER+2,16+4+1:REM select time-base 2 Hz
270 OUT DTOA+1,0:OUT DTOA,0:REM set mass program to 0
280 OUT PLOTTER+2,255:OUT PLOTTER+1,255:OUT PLOTTER,255:REM
zero  plotter
290 OUT ATOD+2,0+6*32:gain%=6:REM select ch.0 and gain factor 6
300 DIM MASSINT%(103,200),GC%(60,80)
310 DIM GCPKST%(1,80),gcpara%(585)
320 DIM MASSNUM%(7),MASSLOC%(1,7),LOADTIME%(15)
340 DIM GC$(2)
350 GC$(1)="FID":GC$(2)="TCD"
370 DEF FNFREESPACE(DRIVE%) 'find free space on drive
380 REG 4,DRIVE%            '0=default,1=A:,2=B:,etc.
390 REG 1,&H3600            'AH=function number
400 CALL INTERRUPT &H21     'DOS function call
410 FNFREESPACE=CSNG(REG(2))*REG(3)*REG(1)
420 END DEF
490 REM
------------------------------------------------------------
------------
500 INPUT "Please choose which GC 1. FID, 2. TCD ";GCTYPE%
502 IF (GCTYPE%<1) OR (GCTYPE%>2) THEN PRINT "WRONG TYPE":GOTO
500
510 GOSUB 5000:REM get MS parameters
520 GOSUB 5400:REM get loop loading parameters
530 INPUT "Select GC recording loop (0 to select all loops)
";GCRECORD%
540 IF (GCRECORD%<0) OR (GCRECORD%>NUMLOOP%) THEN PRINT "OUT
OF RANGE":GOTO 530
560 INPUT "Enter time (sec.) for each loop "; DUETIME%
562 PRINT
570 PRINT "Turn damping control of UTI to min."
580 PRINT "Select gain control of UTI to (-11)"
```

```
590 PRINT "Program the Omega for ramp and soak cycle"
592 PRINT "Put Omega to remote control"
594 PRINT "Switch valve interface to computer"
596 print "Turn gas valve for loading"
600 PRINT "Type any key when ready"
610 IF INKEY$="" THEN 610
620 REM DATA ACQ. START HERE-------------------------
622 MASSPG=massloc%(0,0):GOSUB 1000:REM move mass program to
1st mass
624 OUT ATOD+2,0+MASSloc%(1,0)*32:REM select 1st mass gain and
ch. 0
630 STARTDATE$=DATE$:STARTTIME$=TIME$:IF STARTDATE$<>DATE$
THEN 630
632 GOSUB 4000:REM save parameters
640 GOSUB 4600:REM do gas loop loading
642 GOSUB 1110:REM home loop valves
644 SCREEN 2:CLS:REM graphics screen
645 Print:print "Turn manual gas valve for inject"
647 print "Type any key to continue"
648 if inkey$="" then 648
649 gcpkptr%=80:mspkptr%=200:rem init. for storage need
650 FOR ll=0 TO NUMLOOP%-1 :REM do gc loop one at time
654 LOCATE 20,1:PRINT "Waiting for GC loop ";ll+1
656 PRINT "Last loop data:"
657 gosub 1600:rem check storage and wait if not enough
658 GCPKPTR%=0:REM reset GC peak count
660 GOSUB 2000:REM wait until gc is ready
662 open flname$+".cmm" for append as #2
664  print#2,time$,;
666 close 2
668 LOOPDATE$=DATE$:LOOPTIME=TIMER:IF LOOPDATE$<>DATE$ THEN
668
670 GOSUB 2400:REM start gc temp ramp
671 GOSUB 1360:REM switch injector valve to inject
673 for ji=0 to 500:next ji:rem delay for valve to switch
674 if ll=0 then 676:rem if 1st loop skip next step
675 GOSUB 1150:REM step to next loop
676 GOSUB 2900:REM renew screen
```

```
680 GOSUB 3000:REM data acquisition procedure
682 open flname$+".cmm" for append as #2
684  print#2,time$,gcstop$
686 close 2
690 GOSUB 4200:REM data saving procedure
700 NEXT II:REM do next gc loop
702 SCREEN 0:REM text screen again
710 FINISHDATE$=DATE$:FINISHTIME$=TIME$:IF FINISHDATE$<>DATE$
THEN 710
720 PRINT "experiment starts at ";STARTDATE$;" ";STARTTIME$
730 PRINT "          finishes at ";FINISHDATE$;" ";FINISHTIME$

980 out atod+2,6*32:rem switch UTi gain back to 6
990 END:REM end of main
program----------------------------------------------
1000  REM  cutput  dtoa  data----------------------------
1010 IF MASSPG<0 THEN MASSPG=0
1020 IF MASSPG>65535! THEN MASSPG=65535!
1030 MVAL=INT(MASSPG/256):LVAL=MASSPG-MVAL*256
1040 OUT DTOA+1,MVAL:OUT DTOA,LVAL
1042 if masspg>32767 then massprog%=32767-65536 else
massprog%=masspg
1050 RETURN
1100  REM  16  loop  valves  control-----------------------
1110 REM home loop valves------
1120 OUT LOOPVALVE,LOOPNORM XOR HOME
1122 FOR JI=0 TO 300:NEXT JI
1130 OUT LOOPVALVE,LOOPNORM
1132 FOR JI=0 TO 2000:NEXT JI
1140 RETURN
1150 REM step loop valve-------
1160 OUT LOOPVALVE,LOOPNORM XOR STP
1162 FOR JI=0 TO 300:NEXT JI
1170 OUT LOOPVALVE,LOOPNORM
1180 RETURN
1200 REM position loop valve------
1210 POX=INT(LOOPPOS/10)
1220 POX=POX*16+INT(LOOPPOS-POX*10+.2)
```

```
1230 OUT LOOPVALVE,LOOPNORM XOR POX
1240 OUT LOOPVALVE,LOOPNORM XOR (POX OR SELPOS)
1250 OUT LOOPVALVE,LOOPNORM XOR POX
1260 OUT LOOPVALVE,LOOPNORM
1270    RETURN:REM----------------------------------------
1300  REM  injector  valve  control----------------------
1310  REM  load  injector-----------
1320 OUT INJECTOR,INJNORM XOR LD
1330 FOR JI=0 TO 1500:NEXT JI:REM delay
1340 OUT INJECTOR,INJNORM
1350 RETURN
1360   REM   inject-------------------
1370 OUT INJECTOR,INJNORM XOR INJECT
1380 FOR JI=0 TO 1500:NEXT JI:REM delay
1390 OUT INJECTOR,INJNORM
1400 RETURN
1410    REM    ----------------------------------------------
1500   REM   wait   fid   ready--------------------------------
1502 FOR JI=0 TO 20
1510 IF INP(STATES) AND 1 THEN JI=0:REM test until FID is ready
1512 NEXT JI
1520 RETURN
1530  REM  FID  remote  start--------
1540 OUT INJECTOR,INJNORM XOR FIDSTART
1550 FOR JI=0 TO 300:NEXT JI:REM delay
1560 OUT INJECTOR,INJNORM
1570 RETURN
1580    REM-------------------------------------------------
1600 rem check drive B: storage and wait if not enough--
1610 mspkneed%=mspkptr%+10:if mspkneed%>200 then
mspkneed%=200
1620 IF (GCRECORD%=0) OR (GCRECORD%=II+1) then
gcpkneed%=gcpkptr%+5 else gcpkneed%=0
1622 if gcpkneed%>80 then gcpkneed%=80
1630  need1=104*cdbl(mspkneed%)*2
1640  need2=61*cdbl(gcpkneed%)*2
1650 need=need1+need2
1660 if need < fnfreespace(2) then 1690
```

```
1670 print "Not enough disk space; put another formated disk in B:"
1680 print "Type 'return' key to continue"
1682 if inkey$<> chr$(13) then 1682
1684 if need >= fnfreespace(2) then 1670
1686 mkdir "b:\"+flnamed$
1690     return:rem----------------------------------------
2000 REM wait for gc to be ready ---------------------
2010 PRINT "CHECKING " GC$(GCTYPE%)
2020 ON GCTYPE% GOSUB 1500,2100:REM fid or tcd
2030 PRINT "GC IS READY"
2040 RETURN:REM appropiate gc is ready----------------
2100 REM wait until TCD is ready--------
2110 GOSUB 2200:REM read Omega PSW
2120 IF (OMEPSW AND 128) THEN 2110:REM test Omega stop flag
2130 RETURN:REM TCD is ready ----------
2200 REM read Omega PSW ----------------
2210 PRINT#1,"00:RPSW:":REM command Omega to read
2220 INPUT#1,OME$:IF MID$(OME$,2,2)<>"00" THEN PRINT
"error";OME$:STOP
2230 PSW$=MID$(OME$,5,2)
2240
OMEPSW=(INSTR(LEFT$(PSW$,1),HEXTABLE$)-1)*16+INSTR(RIGHT$(P
SW$,1),HEXTABLE$)-1
2250    RETURN:REM-------------------------
2400 REM start GC temp ramp--------------------------
2410 PRINT:PRINT "STARTING ";GC$(GCTYPE%)
2420 ON GCTYPE% GOSUB 1530,2500:REM fid or tcd start
2430 PRINT GC$(GCTYPE%);" IS STARTED"
2440 PRINT
2450     RETURN:REM----------------------------------------
2500    REM    START    TCD----------------------------------
2510 GOSUB 2200:REM read Omega PSW
2520 NEWPSW=PSW OR 128:REM set start bit for Omega
2530 PRINT #1,"00:wpsw/";HEX$(NEWPSW);":"
2540 INPUT#1,OME$:REM get reponse
2550 IF MID$(OME$,2,2)<>"00" THEN PRINT "error";OME$:STOP
2560     RETURN:REM----------------------------------------
2600 REM check if gc is running return with gcstop% -------
```

```
2610 ON GCTYPE% GOSUB 2700,2800:REM fid,tcd stop ?
2612 if gcstop%=1 then sense%=sense%+1 else sense%=0
2614 if sense%<9 then gcstop%=0 else gcstop$="Stop by GC"
2620 LASPDATE$=DATE$:LASPTIME=TIMER:IF LASPDATE$<>DATE$
THEN 2620
2630 IF LASPDATE$<>LOOPDATE$ THEN LASPTIME=LASPTIME+86400!
2640 IF (LASPTIME-LOOPTIME)>DUETIME% THEN
GCSTOP%=1:gcstop$="Timer Stop"
2690   RETURN:REM--------------
2700 REM check if fid stop
2710 IF INP(STATES) AND 1 THEN GCSTOP%=0 ELSE GCSTOP%=1
2720   RETURN:REM--------------
2800 REM check if tcd stop
2810 GOSUB 2200:REM read Omega PSW
2820 IF (OMEPSW AND 128) THEN GCSTOP%=0 ELSE GCSTOP%=1
2830
RETURN:REM-----------------------------------------------
2900  REM  init  screen---------
2910 CLS:SCRNX%=0
2912 LOCATE 20,1:PRINT "GC loop ";ll+1
2920 LOCATE 21,1
2930 PRINT "GC peak found = ";GCPKPTR%
2940 print "MS peak found = ";mspkptr%
2990   RETURN:REM---------------
3000 REM data acquisition
procedure---------------------------------
3002 PASSFLAG=0:OLDCOUNT=0:ELEMENT%=0:GCPKPTR%=0:REM init.
counts & ptrs
3003 LASTCOUNT%=0:KNEE%=1:TAIL%=1:GCPKST%=0
3004 INDPTR%=0:GCSTOP%=0:SCRNX%=0:REM init. massbuf% index and
stop flag
3006 ERASE MASSINT%,GC%,GCPKST%,gcpara%
3007
gcpara%(6)=mspkstp%:gcpara%(7)=mspkre%:gcpara%(9)=mspkht%
3008  gcpara%(8)=ms4%:gcpara%(20)=100:gcpara%(26)=msdelay%
3009  gcpara%(2)=gctype%:gcpara%(5)=element:gcpara%(12)=gcpkptr%
3010 IF (INP(ATOD+1) AND &H20) =0 THEN 3010:REM WAIT UTIL
COUNTER IS READY
```

```
3012 OUT RDYCLR,0:REM clear counter ready flag
3016 if element%< (mslag%*2) then gcpara%(19)=0 else
gcpara%(19)=nummass%

3020 call readgc
(massint%(0,0),massloc%(0,0),gc%(0,0),gcpkst%(0,0),gcpara%(0))
3060 gccount%=gcpara%(3):ms%=gcpara%(156+(dispmass%-1)*61)
3062 if mspkptr% <> gcpara%(24) then mspkptr%=gcpara%(24):gosub
2920
3070 if gcpkptr% <> gcpara%(12) then gcpkptr%=gcpara%(12):gosub
2920
3075 IF SCRNX%>639 THEN GOSUB 2900:REM clear screen if full
3076 if element%=0 then 3190 :rem no display
3077 print ms%,gcpara%(27);

3080  gcscrny%=199-(gccount%-gcpara%(6))/20
3082 if gcscrny% <100 then gcscrny%=100
3084  msscrny%=99-ms%/20
3086 if msscrny% < 0 then msscrny%=0
3100 IF SCRNX%=0 THEN LINE
(0,gcscrny%)-(0,gcscrny%):oldgcy%=gcscrny%
3102 if scrnx%=0 then line
(0,msscrny%)-(0,msscrny%):oldmsy%=msscrny%:goto  3180
3104  line  (scrnx%-1,oldgcy%)-(scrnx%,gcscrny%):oldgcy%=gcscrny%
3106  line
(scrnx%-1,oldmsy%)-(scrnx%,msscrny%):oldmsy%=msscrny%
3180 SCRNX%=SCRNX%+1:REM prepare for next dot

3190  REM  end  of  gc  peak----------
3200  REM  read  masses----------


3410 ELEMENT%=ELEMENT%+1
3412 LOCATE 23,1:PRINT "At point ";ELEMENT%
3420 IF GCSTOP%=0 THEN REFDATE$=DATE$:REFTIME=TIMER
3430 IF GCSTOP%=0 THEN GOSUB 2600:REM check if gc is still
running
3450 NOWTIME=TIMER:IF REFDATE$<>DATE$ THEN
```

```
NOWTIME=NOWTIME+86400!
3460 IF NOWTIME-REFTIME<240 THEN 3010:REM continue data acq.
for 4 min.
3470 RETURN:REM data acq proedure
return----------
4000 REM Saving parameters routine----------
4010 PRINT "Saving parameters; please wait"
4020 OPEN FLNAME$+".par" FOR OUTPUT AS #2
4030 PRINT#2,GC$(GCTYPE%)
4040 PRINT#2,NUMMASS%
4050 FOR I=0 TO NUMMASS%-1
4060 PRINT#2,MASSnum%(I),MASSloc%(1,I) :rem save massnum and
gain
4070 NEXT I
4080 PRINT#2,NUMLOOP%
4090 FOR I=0 TO NUMLOOP%-1
4100 PRINT#2,LOADTIME%(I)
4110 NEXT I
4120 PRINT#2,GCRECORD%
4130 PRINT#2,DUETIME%
4140 CLOSE 2:RETURN:REM ----------
4200 REM saving data ----------
4202 LOCATE 1,1
4204 PRINT "Saving data, please wait"
4206 IF (GCRECORD%=0) OR (GCRECORD%=I!+1) THEN 4210 ELSE 4290
4210 OPEN FLNAME$+".a"+MID$(STR$(II),2) FOR OUTPUT AS #2
4212 gcpkptr%=gcpara%(12):print#2,gcpkptr%
4220 for i=0 to gcpkptr%-1
4230 print#2,gcpkst%(0,i),gcpkst%(1,i):rem save peak elem & count
4240 next i
4250 CLOSE 2
4252 if gcpkptr%=0 then 4290
4260 DEF SEG=VARSEG(GC%(0,0)):REM get seg for BASVE
4270 LENGTH=VARPTR(GC%(60,GCPKPTR%-1))+2:REM get length for
BASVE
4280 BSAVE FLNAME$+".g"+MID$(STR$(II),2),0,LENGTH
4290 if mspkptr%=0 then 4590
4310 OPEN FLNAME$+".b"+MID$(STR$(II),2) FOR OUTPUT AS #2
```

```
4320 print#2,mspkptr%:rem save # of mass peaks
4330 CLOSE 2
4340 DEF SEG=VARSEG(MASSINT%(0,0))
4350 LENGTH=VARPTR(MASSINT%(103,mspkptr%-1))+2
4360 BSAVE FLNAME$+".m"+MID$(STR$(II),2),0,LENGTH
4590 RETURN:REM=========
4600 REM GC gas loading
loop-------------------------------------------
4602 PRINT:PRINT "LOADING LOOP VALVES"
4610 GOSUB 1100:REM home loop valves
4620 GOSUB 1310:REM put injector valve to load
4622 if numloop%=1 then 4700 rem 1st loop only
4630 FOR I=1 TO NUMLOOP%-1:REM do every loop
4632 GOSUB 1150:REM step loop valve to next loop
4633 if i <> 2 then 4636  else beep:print "Type any key to continue
LOOP 3"
4634 if inkey$="" then 4634
4636 PRINT "LOADING LOOP";I+1;" FOR";LOADTIME%(I);" SEC."
4640 REFDATE$=DATE$:REFTIME=TIMER:IF REFDATE$<>DATE$ THEN
4640:REM get ref time
4650 IF REFDATE$<>DATE$ THEN DAY=1 ELSE DAY=0
4660 PASSTIME=TIMER+DAY*86400!:REM account for day boundary
4670 IF (PASSTIME-REFTIME)<LOADTIME%(I) THEN 4650:REM wait for
full load time
4690 NEXT I:REM fill all gas loops
4700 print "LOADING LOOP 1 FOR";LOADTIME%(0);" SEC."
4702 gosub 1100:rem home loop valve
4710 REFDATE$=DATE$:REFTIME=TIMER:IF REFDATE$<>DATE$ THEN
4710
4720 IF REFDATE$<>DATE$ THEN DAY=1 ELSE DAY=0
4730 PASSTIME=TIMER+DAY*86400
4740 IF (PASSTIME-REFTIME)<LOADTIME%(0) THEN 4720
4800
PRINT:RETURN:REM-------------------------------------------
-----
5000 REM get MS parameters ---------------------------
5010 INPUT "Enter number of masses interested ";NUMMASS%
5012 PRINT
```

```
5014 if nummass%=0 then 5300
5020 PRINT "Locate mass :"
5030 PRINT "     to the prompt of 'LOCATE' use keypad key"
5040 PRINT "       '8' for fine up"
5050 PRINT "       '2' for fine down"
5060 PRINT "       '6' for coarse up"
5070 PRINT "       '4' for coarse down"
5080 PRINT "     to move mass around "
5082 PRINT "        'K'     to use key entry
5084 print "        'G'     to select gain"
5090 PRINT "     and 'ENTER' to select location":PRINT
5100 MASSPG=0:GOSUB 1000:REM preset to mass 0
5110 FOR I=0 TO NUMMASS%-1
5120 PRINT I+1;"  ";
5130 INPUT "enter mass number ";MASSNUM%(I)
5132 if massnum%(i)=4 then ms4%=i+1
5140 PRINT "LOCATE    ";
5150 TEMP$=INKEY$:IF TEMP$="" THEN 5150
5160 IF ASC(TEMP$)=13 THEN 5250:REM select location of mass
5162 if temp$="K" then 5242
5164 if temp$="G" then gosub 5500:rem change gain%
5170 IF LEN(TEMP$)<>2 THEN 5150
5180 TEMP=ASC(RIGHT$(TEMP$,1))
5190 IF TEMP=72 THEN MASSPG=MASSPG+1
5200 IF TEMP=80 THEN MASSPG=MASSPG-1
5210 IF TEMP=77 THEN MASSPG=MASSPG+100
5220 IF TEMP=75 THEN MASSPG=MASSPG-100
5230 GOSUB 1000:REM program D to A
5240 GOTO 5150
5242 input "Enter D to A setting ";MASSPG
5244 goto 5230
5250 PRINT "at D to A setting of ";MASSPG
5260 MASSLOC%(0,I)=MASSProG%
5270 MASSloc%(1,I)=gain%
5272 print "gain control for mass";massnum%(i);" is";gain%
5280 PRINT
5290 NEXT I
5291 if ms4%=0 then beep:print "Mass 4 not selected" else
```

```
ms4%=ms4%-1
5292 print "Display which mass (1 to";nummass%;") ";
5294 input dispmass%
5296 input "MS delay after GC (sec.) ";mslag%
5300
RETURN:REM-----------------------------------------------
5400 REM get loop valve load data ----------------------------
5410 PRINT "GETTING LOOP VALVE LOAD DATA"
5412 print "LOOP 1 WILL BE YOUR 1ST LOOP WITH STEADY LOADING"
5414 PRINT "Load-Time for loop 1 is the min. load-time for it"
5420 INPUT "Enter number of loops (1 to 16)"; NUMLOOP%
5430 IF (NUMLOOP%<1) OR (NUMLOOP%>16) THEN 5420
5440 FOR I=0 TO NUMLOOP%-1
5450 PRINT "enter load-time (in sec.) for loop";I+1;
5460 INPUT LOADTIME%(I)
5470 NEXT I
5480 RETURN:REM
-----------------------------------------------
5500 rem change gain%----------------
5510 input "enter mass gain (0 to 7) ";gain%
5520 if (gain%<0) or (gain%>7) then 5510:rem out of range
5550 out atod+2,gain%*32:rem switch gain
5560 return:rem----------------------
9000 rem error handle routine--------------------------------
9010 open "tberr1.txt" for output as 10
9020 print#10,erl,err
9030 close 10
9032 print "ERR";err;"AT line";erl
9040
end:rem-----------------------------------------------------

10000 SUB READGC INLINE
10010 $inline "gcms.bin"
10020 end sub
```

READTS.TXT

```
100 REM loading data from TSNEWGC program ver 3.10 4-27-90 by
Henry Chan
110 on error goto 9000
120  base2=&he6a0:plotter=base2+12
130  out  plotter+3,&h80
140  xplot%=0:yplot%=0:gosub  6400:rem  zero  plotter
200 DIM GCELEM%(80)
210 DIM
MASSPEAK%(7),GCPEAK%(80),pkbg%(200),pkend%(200),pkbase%(200)
220 DIM MASSNUM%(7),MASSGAIN%(7),LOADTIME%(15)
230  dim  gcret%(100),msdel%(100)
232  cls:print "Loading  reference  data"
240  open "readts1.par" for input as 1
250  for i=0 to 93
260   input#1,gcret%(i),msdel%(i)
270  next i
280  close 1
310  PRINT "Place data disk in drive B:, type any key to continue"
312 IF INKEY$="" THEN 312
314  chdir  "b:\"
316  files  "b:*."
320  INPUT "Enter file name ";FLNAME$
322  chdir  "b:"+flname$
330 OPEN "b:"+FLNAME$+".cmm" FOR INPUT AS #2
340 WHILE NOT EOF(2)
350 LINE INPUT#2,COMMENT$
360 PRINT COMMENT$
370 WEND
380 CLOSE 2
390  INPUT "Is this the right one (y/n) ";YES$
400 IF YES$="y" THEN 430
410 IF YES$="n" THEN 310
420 GOTO 390
430  PRINT
500 REM Loading
parameters----------------------------------------
```

```
510 PRINT "Loading parameters, please wait"
520 OPEN "b:"+FLNAME$+".par" FOR INPUT AS #2
530 INPUT#2,GC$:PRINT "GC type was ";GC$
540 INPUT#2,NUMMASS%:PRINT "Number of masses observed
=";NUMMASS%
550 FOR I=0 TO NUMMASS%-1
560 INPUT#2, MASSNUM%(I),MASSGAIN%(I)
570 PRINT "MASS";I+1;"is";MASSNUM%(I),"Gain factor";MASSGAIN%(I)
580 NEXT I
590 INPUT#2,NUMLOOP%:PRINT "Number of GC loop is "NUMLOOP%
600 FOR I=0 TO NUMLOOP%-1
610 INPUT#2,LOADTIME%(I)
620 PRINT "Loop";I+1;" load time is ";LOADTIME%(I);" sec."
630 NEXT I
640 INPUT#2,GCRECORD%:PRINT "GC recording loop is (0 for
all)";GCRECORD%
650 INPUT#2,DUETIME%:PRINT "Max. time for GC loop is";DUETIME%;"
sec."
660 CLOSE 2:REM parametes
loaded----------------------------------
700 REM display
parameters-------------------------------------------
710 CLS
720 GOSUB 1700:REM display parameters----------
1000 REM read
data-------------------------------------------------
1010 PRINT:
1020 INPUT "Enter which GC loop (0 to exit) ";II
1022 if ii=0 then 9900:rem exit program
1030 IF (II<1) OR (II>NUMLOOP%) THEN PRINT "Out of range":GOTO
1020
1050 IF (GCRECORD%=0) OR (GCRECORD%=II) THEN 1060 ELSE GOTO
1200
1060 REM load gc data-----
1070 OPEN "b:"+FLNAME$+".a"+MID$(STR$(I;-1),2) FOR INPUT AS#2
1080 INPUT#2,GCPKPTR%:PRINT "number of GC peak located
is";GCPKPTR%
1082 if gcpkptr%=0 then close 2:dim gc%(60,gcpkptr%+1):goto 1200
```

```
1090 FOR I=0 TO gcpkptr%-1
1100 INPUT#2,GCELEM%(I),GCPEAK%(I)
1110 NEXT I
1112 CLOSE 2
1150 PRINT "Loading GC data, please wait"
1152  gcname$="b:"+flname$+".g"+mid$(str$(ii-1),2)
1158 dim gc%(60,gcpkptr%+1)
1160 DEF SEG=VARSEG(GC%(0,0))
1170 BLOAD gcname$
1180 PRINT "Done loading GC data"
1200  REM read MS data----------
1202 if nummass%=0 then dim massint%(103,mspkptr%+1):goto
1400
1210 OPEN "b:"+FLNAME$+".b"+MID$(STR$(II-1),2) FOR INPUT AS #2
1230 INPUT#2,mspkptr%: rem get # of mass peak%
1260 CLOSE 2
1262 PRINT
1264  dim  massint%(103,mspkptr%+1)
1270 DEF SEG=VARSEG(MASSINT%(0,0))
1280 PRINT "Loading MS data, please wait"
1290 BLOAD "b:"+FLNAME$+".m"+MID$(STR$(II-1),2)
1300 PRINT "Done loading MS data"
1400   REM   Data   view----------------------------------------
1402 CLS:GOSUB 1700:GOSUB 1900:REM print peaks data
1410 PRINT "To view 1. GC peak, 2. MS peak, 3. another loop, 4. Auto
align"
1420 INPUT "Enter choice by number ";CHOICE
1430 ON CHOICE GOTO 1450,1460,1470,1462
1440 GOTO 1410:REM no choice
1450 GOSUB 2000:GOTO 1400 :REM view GC peak
1460 GOSUB 3000:GOTO 1400 :REM view MS peak
1462 gosub 7000:goto 1400 :rem auto align GC & MS data
1470 erase gc%:erase massint%:goto 700:rem clear gc% array and
repeat
1480 gosub 4000:goto 1400:rem auto-integrate GC peaks
1700 REM print parameters
1720 PRINT "GC type used was ";GC$
1730 PRINT "Number of masses observed ="NUMMASS%
```

```
1740 FOR I=0 TO NUMMASS%-1
1750 PRINT "mass";MASSNUM%(I)" gain";MASSGAIN%(I),;
1760 NEXT I
1770 PRINT
1780 PRINT "Number of GC loop is";NUMLOOP%
1790 PRINT "Load time in sec. for each loop"
1800 FOR I=0 TO NUMLOOP%-1
1810 PRINT I+1;" is";LOADTIME%(I),;
1820 NEXT I
1830 PRINT
1840 PRINT "GC recording loop (0 for all) is";GCRECORD%
1850 PRINT "Max. time for each GC loop is";DUETIME%;"sec."
1860    RETURN:REM-----------------------------------
1900 REM print GC & MS data--------
1910 PRINT "LOOP #";ii;" GC peaks located =";GCPKPTR%
1912 print "GC peak#, at element ,intensity"
1920 FOR I=0 TO GCPKPTR%-1
1930 PRINT I+1;GCELEM%(I)/2;GCPEAK%(I);
1932  tabcount=(i+1)-int((i+1)/4+.1)*4:print  tab(tabcount*20);
1940 NEXT I
1950 print:PRINT "Total mass peak ";mspkptr%
1990  RETURN:REM  ------------------
2000 REM view GC peak
2002 GCMAX=0:if gcpkptr%=0 then print "no GC peak":return
2003 FOR I=0 TO GCPKPTR%-1
2004 IF GCPEAK%(I) > GCMAX THEN GCMAX=GCPEAK%(I)
2005 next i:print:gosub 1900
2006 iNPUT "Which gc peak (e to exit,p to plot,a to
autointegrate)";GCPOINT$
2007 gcpoint=val(gcpoint$):if gcpoint$="e"  then return
2008 if gcpoint$="a" then gosub 4000:goto 2000
2009 if gcpoint$="p" then gosub 6000:goto 2000
2010 if ((gcpoint=0) or (gcpoint>gcpkptr%)) then 2000
2011 SCREEN 2:YSCALE=GCMAX/180:gosub 4500
2012  nhead=head:ntail=tail
2013 cls:line (30,0)-(30,10):rem tack peak
2014 SCRNY=199-(GC%(0,GCPOINT-1)-baseline+10)/YSCALE
2016 IF SCRNY < 0 THEN SCRNY=0
```

```
2020 LINE (0,SCRNY)-(0,SCRNY):rem 1st point
2021  locate 1,40:print "Loop #";ii
2022  locate 2,40:print "GC peak";gcpoint;"
at";gcelem%(gcpoint-1)/2;"sec."
2024  locate 3,40:print gcpeak%(gcpoint-1)
2030 FOR I=1 TO 60
2032 SCRNY=199-(GC%(I,GCPOINT-1)-baseline+10)/YSCALE
2034 IF SCRNY < 0 THEN SCRNY=0
2040 LINE -(I,SCRNY)
2050 NEXT I
2200   y1=199-(baseline-baseline+10)/yscale
2216  line  (0,y1)-(60,y1)
2222   y1=199-(gc%(head,gcpoint-1)-baseline+10)/yscale
2224   y2=199-(gc%(tail,gcpoint-1)-baseline+10)/yscale
2226  line  (head,y1)-(head,y1-5)
2228   line   (tail,y2)-(tail,y2-5)
2230 ky$=inkey$:if ky$="" then 2230
2240 if asc(ky$)=13 then 2500:rem CR
2250 if asc(ky$)=27 then 2950:rem ESC
2251 if ky$="R" then goto 2013
2252 if ky$="E" then yscale=yscale/2:goto 2013
2253 if ky$="S" then yscale=yscale*2:goto 2013
2254 if ky$="<" then baseline=baseline-1:goto 2013
2255 if ky$=">" then baseline=baseline+1:goto 2013
2258 if ky$="^" then gosub 2700:goto 2230:rem display array
2260 if len(ky$)<2 then 2230
2262  locate 5,40
2264 print "                              ";
2270   temp=asc(right$(ky$,1))
2280 if temp=72 then if head<59 then nhead=head+1:gosub 2800
2290 if temp=80 then if head>0 then nhead=head-1:gosub 2800
2300 if temp=77 then if tail<59 then ntail=tail+1:gosub 2850
2310 if temp=75 then if tail>0 then ntail=tail-1:gosub 2850
2340  head=nhead:tail=ntail:goto 2230
2500   rem  integrate----
2510 sum=0:if head=tail then 2600
2530 for ji=head to tail
2550   sum=sum+gc%(ji,gcpoint-1)-baseline
```

```
2560 next ji
2600 locate 5,40
2610 print "INTEGRAL =";sum;"      ";
2620 goto 2200
2700 rem display array--------
2710 for i=0 to 60
2715 if i=30 then print "*";
2720 print gc%(i,gcpoint-1),;
2730 next i
2740 print
2750   return:rem---------------
2800 rem tack header----------
2805 if nhead>ntail then nhead=ntail
2810 line(head,y1-1)-(head,y1-5),0:rem erase old tack
2815 y1=199-(gc%(nhead,gcpoint-1)-baseline+10)/yscale:rem
calculate new y1
2820 line(nhead,y1)-(nhead,y1-5):rem tack new leader
2830   return:rem----------------
2850 rem tack tail
2855 if ntail<nhead then ntail=nhead
2860 line(tail,y2-1)-(tail,y2-5),0:rem erase old tack
2865 y2=199-(gc%(ntail,gcpoint-1)-baseline+10)/yscale:rem
calculate new y2
2870 line(ntail,y2)-(ntail,y2-5):rem tack new tail
2880   return:rem-----------------
2950 LOCATE 22,1
2960 PRINT "Type any key to exit"
2970 IF INKEY$="" THEN 2970
2980 SCREEN 0
2990 goto 2000:rem ---------------
3000 rem view MS data----------
3002 if mspkptr%=0 then print "no mass peak":return
3006 print "View mass peaks"
3010 for i=0 to nummass%-1
3020 print "Mass";massnum%(i);" gain";massgain%(i),;
3030 next i
3040 print
3070 input "Which Mass (e to exit)";mass$
```

```
3072 mass=val(mass$):if mass$="e" then return
3080 mspoint=0
3090 for i=0 to nummass%-1
3100 if mass=massnum%(i) then mspoint=i+1
3110 next i
3120 if mspoint=0 then print "Mass not found":goto 3070
3122 mpmax%=0:mspk%=0:for i=0 to mspkptr%-1
3123 if massint%(0,i)<>(mspoint-1) then 3127
3124 if massint%(2,i) > mpmax% then mpmax%=massint%(2,i)
3126 mspk%=mspk%+1:print mspk%;massint%(1,i)/2;massint%(2,i),;
3127 next i:print
3128 print mspk%;"peaks for mass";massnum%(mspoint-1)
3129 input "Which one (e to exit,p to plot, a to autointegrate)";mk$
3130 if mk$="e" then 3000
3131 if mk$="a" then gosub 5000:goto 3120:rem autointegrate
3132 if mk$="p" then gosub 6500:goto 3120:rem plot
3133 mk%=val(mk$):if mk%>mspk% then print "out of range":goto
3129
3134 dim vmloc%(mspk%):mspk%=0:for i=0 to mspkptr%-1
3135 if massint%(0,i)<>(mspoint-1) then 3137
3136 vmloc%(mspk%)=massint%(1,i):mspk%=mspk%+1:if mspk%=mk%
then vmspk%=i
3137 next i
3138 yscale=mpmax%/180:gosub 4700:erase vmloc%:rem find
baseline
3139  scrny1=199-(massint%(3,vmspk%)-baseline+10)/yscale
3142  scrny2=199-(massint%(54,vmspk%)-baseline+10)/yscale
3145 nhead=head:ntail=tail
3150 screen 2:cls
3152 line (25,0)-(25,10):rem tack peak
3160 line (0,scrny1)-(0,scrny1):rem point 1st pt.
3162 locate 1,40:print "Loop #";ii
3170 locate 2,40:print "Mass";mass;" peaks
at";massint%(1,vmspk%)/2;"sec."
3190 locate 3,40:print massint%(2,vmspk%)
3200 for i=4 to 53
3202  oscrny1=scrny1
3210  scrny1=199-(massint%(i,vmspk%)-baseline+10)/yscale
```

```
3220 if scrny1 < 0 then scrny1=0
3230  line  (i-4,oscrny1)-(i-3,scrny1)
3232  oscrny2=scrny2
3233  scrny2=199-(massint%(i+50,vmspk%)-baseline+10)/yscale
3234 if scrny2 < 0 then scrny2=0
3235  line  (i+100,oscrny2)-(i+101,scrny2)
3240 next i
3300  y1=199-(baseline-baseline+10)/yscale
3320 LINE (0,Y1)-(50,Y1):REM draw base line
3330  Y1=199-(massint%(HEAD,vmspk%)-baseline+10)/YSCALE
3340  Y2=199-(massint%(TAIL,vmspk%)-baseline+10)/YSCALE
3350 LINE (HEAD-3,Y1)-(HEAD-3,Y1-5)
3360  LINE  (TAIL-3,Y2)-(TAIL-3,Y2-5)
3370 KY$=INKEY$:IF KY$="" THEN 3370
3380 IF ASC(KY$)=13 THEN 3520:REM CR
3390 IF ASC(KY$)=27 THEN 3820:REM ESC
3400 IF KY$="R" THEN GOTO 3150
3410  if ky$="E" then yscale=yscale/2:goto 3150
3412  if ky$="S" then yscale=yscale*2:goto 3150
3413  if ky$="<" then baseline=baseline-1:goto 3150
3414  if ky$=">" then baseline=baseline+1:goto 3150
3420 IF KY$="^" THEN GOSUB 3630:GOTO 3370:REM display array
3430 IF LEN(KY$)<2 THEN 3370
3440 LOCATE 5,40
3450 PRINT "                    ";
3460 TEMP=ASC(RIGHT$(KY$,1))
3470 IF TEMP=72 THEN IF HEAD<53 THEN NHEAD=HEAD+1:GOSUB 3700
3480 IF TEMP=80 THEN IF HEAD>3 THEN NHEAD=HEAD-1:GOSUB 3700
3490 IF TEMP=77 THEN IF TAIL<53 THEN NTAIL=TAIL+1:GOSUB 3760
3500 IF TEMP=75 THEN IF TAIL>3 THEN NTAIL=TAIL-1:GOSUB 3760
3510 HEAD=NHEAD:TAIL=NTAIL:GOTO 3370
3520  REM  integrate----
3530 SUM=0:sum4=0:IF HEAD=TAIL THEN 3600
3560 FOR JI=HEAD TO TAIL
3580 SUM=SUM+massint%(JI,vmspk%)-BASELINE
3582  sum4=sum4+massint%(ji+50,vmspk%)
3590 NEXT JI
3600 LOCATE 5,40
```

```
3610 PRINT "INTEGRAL =";SUM;"     ";
3612 locate 6,40 :print "Average mass 4 intensity"
;sum4/(tail-head+1)
3620 GOTO 3300
3630  REM  display  array--------
3640 FOR I=0 TO 103
3650 IF I=28 THEN PRINT "";
3652 if i=base1 then print "<";
3654 if i=base2 then print ">";
3660 PRINT massint%(I,vmspk%),;
3670 NEXT I
3680 PRINT
3690   RETURN:REM---------------
3700  REM  tack  header----------
3710 IF NHEAD>NTAIL  THEN NHEAD=NTAIL
3720 LINE(HEAD-3,Y1-1)-(HEAD-3,Y1-5),0:REM erase old tack
3730 Y1=199-(massint%(NHEAD,vmspk%)-baseline+10)/YSCALE:REM
calculate new y1
3740 LINE(NHEAD-3,Y1)-(NHEAD-3,Y1-5):REM tack new leader
3750   RETURN:REM---------------
3760 REM tack tail
3770 IF NTAIL<NHEAD THEN NTAIL=NHEAD
3780  LINE(TAIL-3,Y2-1)-(TAIL-3,Y2-5),0:REM  erase  old  tack
3790  Y2=199-(massint%(NTAIL,vmspk%)-baseline+10)/YSCALE:REM
calculate new y2
3800  LINE(NTAIL-3,Y2)-(NTAIL-3,Y2-5):REM tack new tail
3810   RETURN:REM---------------
3820 LOCATE 22,1
3830 PRINT "Type any key to exit"
3840 IF INKEY$="" THEN 3840
3850 SCREEN 0
3860  goto  3120:REM  --------------
4000  rem  auto-integrate
4002  print  "GC  auto-integrate  loop  #";ii
4010 dim area(gcpkptr%):rem clear array
4020 gosub 4200:rem auto routine
4030  print
4040 input "Do you want to save integrals (y/n)"; yes$
```

```
4050 if yes$="y" then gosub 4300:goto 4080
4060 if yes$="n" then 4080
4070 goto 4040
4080 input "Do you want to print integrals (y/n)"; yes$
4090 if yes$="y" then gosub 4400:goto 4120
4100 if yes$="n" then 4120
4110 goto 4080
4120   erase   area:return:rem------------------------------
4200 for gcpk%=0 to gcpkptr%-1:rem do it for every GC peak
4210 gcpoint=gcpk%+1:gosub 4500:rem look baseline & define peak
4220 sum=0:sum4=0: if head=tail then 4290
4230 pkbase%(gcpk%)=baseline
4240 for ji=head to tail
4250 sum=sum+gc%(ji,gcpk%)-baseline
4260 next ji
4270 area(gcpk%)=fix(sum)
4280 print gcpk%+1;" ";area(gcpk%),:
4290 next gcpk%
4299    return:rem-----------------------------------------
4300 rem saving integrals
4302 sname$=left$(flname$,4)+"gc"+mid$(str$(ii),2)+".prn"
4310 open "b:"+sname$ for output as # 2
4312 print#2,"peak","at (sec.)","height","area","baseline"
4330 for gcpk%=0 to gcpkptr%-1
4340 print#2, gcpk%+1,gcelem%(gcpk%)/2,gcpeak%(gcpk%),:
4342 print#2, area(gcpk%),pkbase%(gcpk%)
4350 next gcpk%
4360 print#2,
4370 close 2
4390    return:rem-----------------------------------
4400   rem   print   integral-----------------------
4410 lprint flname$;"GC peaks for  Loop #";ii
4420 lprint
4428 lprint "peak","at (sec.)","height","area","baseline"
4430 NL%=0:for gcpk%=0 to gcpkptr%-1
4440 lprint
gcpk%+1,gcelem%(gcpk%)/2,gcpeak%(gcpk%),area(gcpk%),pkbase%(gc
pk%)
```

```
4450 NL%=NL%+1:IF NL%=5 THEN NL%=0:LPRINT
4480 next gcpk%
4482  lprint:lprint
4490    return:rem--------------------------------
4500 rem find base1 & base2
4501 gp=gcpoint
4502 if gp=1 then head=0 else
head=30-(gcelem%(gp-1)-gcelem%(gp-2))
4504 if gp=gcpkptr% then tail=60 else
tail=(gcelem%(gp)-gcelem%(gp-1))+30
4506 if head < 0 then head=0
4508 if tail >60 then tail=60
4582 a%=4:dim a(a%):for i=0 to 4:a(i)=4096:next i
4583 for i=5 to 60:if gc%(i,gcpoint-1) > a(0) then 4588
4584  a(0)=gc%(i,gcpoint-1)
4585 for i1=0 to 3
4586 if a(i1) < a(i1+1) then a1=a(i1):a(i1)=a(i1+1):a(i1+1)=a1
4587 next i1
4588 next i
4589 sum=0:for i=0 to 4:sum=sum+a(i):next i:baseline=sum/5:erase a
4590 basemin%=gc%(head,gcpoint-1):nhead=head
4600 for i=head to 30
4610 if gc%(i,gcpoint-1) < basemin% then
nhead=i:basemin%=gc%(i,gcpoint-1)
4620 next i
4630  basemin%=gc%(31,gcpoint-1):ntail=31
4640 for i=31 to tail
4650 if gc%(i,gcpoint-1) <= basemin% then
ntail=i:basemin%=gc%(i,gcpoint-1)
4660 next i
4670  head=nhead:tail=ntail
4680 if head < 15 then head=15
4682 if tail > 52 then tail=52
4690    return:rem--------------------------------
4700 rem find ms baseline
4710 if mk%=1 then head=0 else
head=27-(vmloc%(mk%-1)-vmloc%(mk%-2))
4720 if mk%=mspk% then tail=53 else
```

```
tail=(vmloc%(mk%)-vmloc%(mk%-1))+27
4730 if head < 2 then head=2
4732 if tail > 53 then tail=53
4740 a%=4:dim a(a%):for i=0 to 4:a(i)=4096:next i
4745 for i=2 to 53:if massint%(i,vmspk%) > a(0) then 4770
4750 a(0)=massint%(i,vmspk%)
4755 for i1=0 to 3
4760 if a(i1) < a(i1+1) then a1=a(i1):a(i1)=a(i1+1):a(i1+1)=a1
4765 next i1
4770 next i
4775 sum=0:for i=0 to 4:sum=sum+a(i):next i:baseline=sum/5:erase a
4780 basemin%=massint%(head,vmspk%):nhead=head
4785 for i=head to 27
4790 if massint%(i,vmspk%)< basemin% then
nhead=i:basemin%=massint%(i,vmspk%)
4795 next i
4800 basemin%=massint%(28,vmspk%):ntail=28
4805 for i=28 to tail
4810 if massint%(i,vmspk%)<= basemin% then
ntail=i:basemin%=massint%(i,vmspk%)
4815 next i
4820 head=nhead:tail=ntail
4830 if head < 12 then head=12
4840 if tail > 50 then tail=50
4850    return:rem--------------------------------
5000 REM auto-integrate ms
5010 PRINT "MS auto-integrate loop #";II;" for
mass";massnum%(mspoint-1)
5020 DIM
AREA1(msPKPTR%),height%(mspkptr%),area4(mspkptr%),mselem%(m
spkptr%)
5022 dim vmloc%(mspk%),msbase%(mspkptr%)
5024 gosub 5200
5026 print
5030 if pkcount%=0 then print "no peak":goto 5120
5040 INPUT "Do you want to save integrals (y/n)"; YES$
5050 IF YES$="y" THEN GOSUB 5510:GOTO 5080
5060 IF YES$="n" THEN 5080
```

```
5070 GOTO 5040
5080 INPUT "Do you want to print integrals (y/n)"; YES$
5090 IF YES$="y" THEN GOSUB 5610:GOTO 5120
5100 IF YES$="n" THEN 5120
5110 GOTO 5080
5120 ERASE
AREA1,height%,area4,mselem%,vmioc%,msbase%:RETURN:REM-------
- - - - -
5200 rem ms auto-integrate
5210 mspk%=0:for i=0 to mspkptr%-1
5220 if massint%(0,i) <> (mspoint-1) then 5240
5230 vmioc%(mspk%)=massint%(1,i):mspk%=mspk%+1
5240 next i
5250 mk%=1:pkcount%=0
5260 FOR msPKi%=0 TO msPKPTR%-1:REM do it for every peak of this
mass
5270 if massint%(0,mspki%) <> (mspoint-1) then 5430
5280 mselem%(pkcount%)=massint%(1,mspki%)
5290 height%(pkcount%)=massint%(2,mspki%)
5300 vmspk%=mspki%
5310 gosub 4700
5320 mk%=mk%+1
5330 sum=0:sum4=0:area1(pkcount%)=0:area4(pkcount%)=0
5340 for i=head to tail
5350
AREA1(pkcount%)=AREA1(pkcount%)+(massint%(I,msPKi%)-baseline)
5360 area4(pkcount%)=area4(pkcount%)+(massint%(I+50,mspki%))
5370 NEXT I
5380 area4(pkcount%)=area4(pkcount%)/(tail-head+1)
5390 msBASE%(pkcount%)=baseline
5400 AREA1(pkcount%)=FIX(AREA1(pkcount%))
5410 PRINT pkcount%+1;" ";int
(AREA1(pkcount%)),int(area4(pkcount%))
5420 pkcount%=pkcount%+1
5430 NEXT msPKi%
5440     return:rem-----------------------------
5510 REM saving integrals
5512
```

```
sname$=left$(finame$,4)+mid$(str$(massnum%(mspoint-1)+100),3)
5514  sname$=sname$+mid$(str$(ii),2)+".prn"
5520 OPEN "b:"+sname$ FOR OUTPUT AS # 2
5540 FOR msPKi%=0 TO pkcount%-1
5550  PRINT#2,  msPKi%+1,mselem%(mspki%)/2,height%(mspki%),;
5560  print#2,
int(AREA1(msPKi%)),msbase%(mspki%),int(area4(mspki%))
5570 NEXT msPKi%
5580  PRINT#2,
5590 CLOSE 2
5600    RETURN:REM--------------------------------
5610   REM   print   integral-----------------------
5620 LPRINT FLNAME$;"    Loop #";ll ;"autointegrate
mass";massnum%(mspoint-1)
5630 LPRINT
5640 LPRINT "peak"; TAB(12);"at (sec.)";TAB(24);"height";
5642 LPRINT  TAB(36);"area";TAB(48);"baseline"TAB(58);"mass  4
avg.";
5643  lprint tab(70);"pk.  ht."
5650 NL%=0:FOR msPK%=0 TO pkcount%-1
5660 LPRINT
msPK%+1;TAB(12);msELEM%(msPK%)/2;TAB(24);height%(mspk%);
5670 LPRINT
TAB(36);int(AREA1(msPK%));TAB(48);msBASE%(msPK%);
5671  lprint  TAB(58);int(area4(mspk%));
5672  lprint  tab(70);height%(mspk%)-msbase%(mspk%)
5674 NL%=NL%+1:IF NL%=5 THEN NL%=0:LPRINT
5680 NEXT msPK%
5682   lprint:lprint
5690    RETURN:REM--------------------------------
6000   rem   GC   plot------------------------------
6002 print "Plot GC; Pen up ; type any key to continue"
6004  if inkey$="" then 6004
6010 xplot%=0:pbase%=gc%(0,0):yplot%=pbase%:gosub 6400
6012 print "Pen down ;type any key to continue"
6014  if inkey$="" then 6014
6020 for i=0 to gcpkptr%-1 :rem for every gcpeaks
6030  if xplot% >= (gcelem%(i)-30) then 6100
```

```
6040 yplot%=pbase%:gosub 6400
6050 xplot%=xplot%+1
6060 goto 6030
6100 selem%=xplot%-gcelem%(i)+30
6110 if selem% > 60 then 6200
6120 for i1=selem% to 60
6130 yplot%=gc%(i1,i):gosub 6400
6140 xplot%=xplot%+1
6150 next i1
6160 pbase%=gc%(0,i)
6170 for i1=0 to 60
6180 if pbase% > gc%(i1,i) then pbase%=gc%(i1,i)
6190 next i1
6200 next i
6390    return:rem--------------------------------
6400  rem  plot-------
6410 if xplot%<0 then xplot%=0
6420 if xplot%>4095 then xplot%=4095
6430 if yplot%<0 then yplot%=0
6440 if yplot%>4095 then yplot%=4095
6450 plot0%=255-yplot% and 255
6460 plot2%=255-xplot% and 255
6470  plot1%=255-(int(xplot%/256)*16+int(yplot%/256))
6480 out plotter+2,plot2%:out plotter+1,plot1%:out plotter,plot0%
6490 return:rem ----
6500   rem   MS   plot------------------------------
6502 print "Plot MS"massnum%(mspoint-1);" Pen up ; type any key to
continue"
6504 if inkey$="" then 6504
6510 xplot%=0:pbase%=0:yplot%=pbase%:gosub 6400
6512 print "Pen down ;type any key to continue"
6514 if inkey$="" then 6514
6520 for i=0 to mspkptr%-1 :rem for every mspeaks
6522 if massint%(0,i) <> (mspoint-1) then 6700
6524 if xplot%=0 then pbase%=massint%(3,i)
6530 if xplot% >= (massint%(1,i)-28+3) then 6600
6540 yplot%=pbase%:gosub 6400
6550 xplot%=xplot%+1
```

```
6560 goto 6530
6600  selem%=xplot%-massint%(1,i)+28
6610 if selem% > 52 then 6700
6620 for i1=selem% to 52
6630 yplot%=MASSINT%(i1,i):gosub 6400
6640 xplot%=xplot%+1
6650 next i1
6660 pbase%=massint%(3,i)
6670 for i1=3 to 52
6680 if pbase% > massint%(i1,i) then pbase%=massint%(i1,i)
6690 next i1
6700 next i
6990     return:rem--------------------------------
7000 rem auto align GC & MS data
7002 print "Check printer is ready, and there is enough paper"
7004 print "Type any key to continue"
7006 if inkey$="" then 7006
7008 Lprint "Auto align Loop";ii;" of file ";flname$:lprint
7010 dim
area(gcpkptr%),area1(mspkptr%),height%(mspkptr%),area4(mspkptr-
%)
7012 dim
mselem%(mspkptr%),vmloc%(mspkptr%),msbase%(mspkptr%),pkid%(g
cpkptr%)
7014 dim form(gcpkptr%-1,nummass%+1)
7022 print:print:print "Auto align routine"
7024 print "Now doing auto integrating GC"
7030 gosub 4200:rem auto gc integrate
7032 gosub 4400:rem print gc integrals
7040 print "Now aligning GC data with peak ID"
7050  idptr%=0
7060 for igc%=0 to gcpkptr%-1
7070  oskew%=30000
7072 if idptr%=94 then 7094
7080  skew%=abs(gcret%(idptr%)-gcelem%(igc%)/2)
7090 if skew% < oskew% then oskew%=skew%:idptr%=idptr%+1:goto
7072
7094 pkid%(igc%)=idptr%
```

```
7096  form(igc%,0)=idptr%:form(igc%,1)=area(igc%)
7098  rem  idptr%=idptr%-1
7110 next igc%
7120 for ims%=0 to nummass%-1
7130  mspoint=ims%+1
7140 print "auto integrating mass";massnum%(ims%)
7150 gosub 5200:rem ms integrate
7152 gosub 5610:rem print ms integrate
7160 if pkcount%=0 then print "no peak for this mass":goto 7700
7170 print "Aligning MS with GC data"
7180 igc%=0:for imspk%=0 to pkcount%-1
7190  oskew%=30000
7192 if igc% = gcpkptr% then 7240
7200
skew%=abs((mselem%(imspk%)-gcelem%(igc%))/2-msdel%(pkid%(igc
%)-1))
7210 if skew% < oskew% then oskew%=skew%:igc%=igc%+1:goto
7192
7220  form(igc%-1,ims%+2)=area1(imspk%)/area4(imspk%)
7230  igc%=igc%-1
7240 next imspk%
7700 next ims%
7701  lprint chr$(12);"Align  Data  for  Loop";ii;"  of  file
";flname$:lprint
7702  lprint  "pkid";tab(15);"gcarea";
7703 for i=0 to nummass%-1
7704  lprint  using  "########";massnum%(i);
7705  next  i:lprint
7706 for i=0 to gcpkptr%-1
7707  lprint  form(i,0):tab(5);gcelem%(i)/2;tab(15);form(i,1);tab(25)
7708 for iform=2 to nummass%+1:lprint using
"###.####";form(i,iform);:next
7709  lprint:next  i:lprint  chr$(12);
7710  sname$=left$(flname$,4)+"ag"+mid$(str$(ii),2)+".prn"
7720 open "b:"+sname$ for output as #2
7730  print#2,"pkid","gcarea",;
7740 for i=0 to nummass%-1
7750  print#2,using  "########";massnum%(i);
```

```
7760 next i
7762  print#2,
7770 for i=0 to gcpkptr%-1
7772  print#2,form(i,0),form(i,1),;
7780 for iform=2 to nummass%+1
7800  print#2,using  "###.####";form(i,iform);
7810 next iform
7820  print#2,
7830 next i
7840 close 2
7990 erase
area,area1,height%,area4,mselem%,vmloc%,msbase%,pkid%,form
7992     return:rem----------------------------------
9000  rem  error  handling  routine----------------
9010 if (err=76) and (erl=322) then resume next:rem not a dir
9020 if (err=53) and (erl=314) then resume next:rem no *.par
9030 if (err=53) and (erl=316) then resume next:rem no <DIR>
9040 if (err=53) and (erl=1210) then nummass%=0:goto 1200:rem no
"".b"files
9090 open "tberr.txt" for output as 10
9100  print#10,erl,err
9110 close 10
9200 end:rem error end
9900 chdir "b:\":rem reset B drive working dir
9910 END:rem end program
```