Computers in Chemical Education Newsletter

Fall 1989

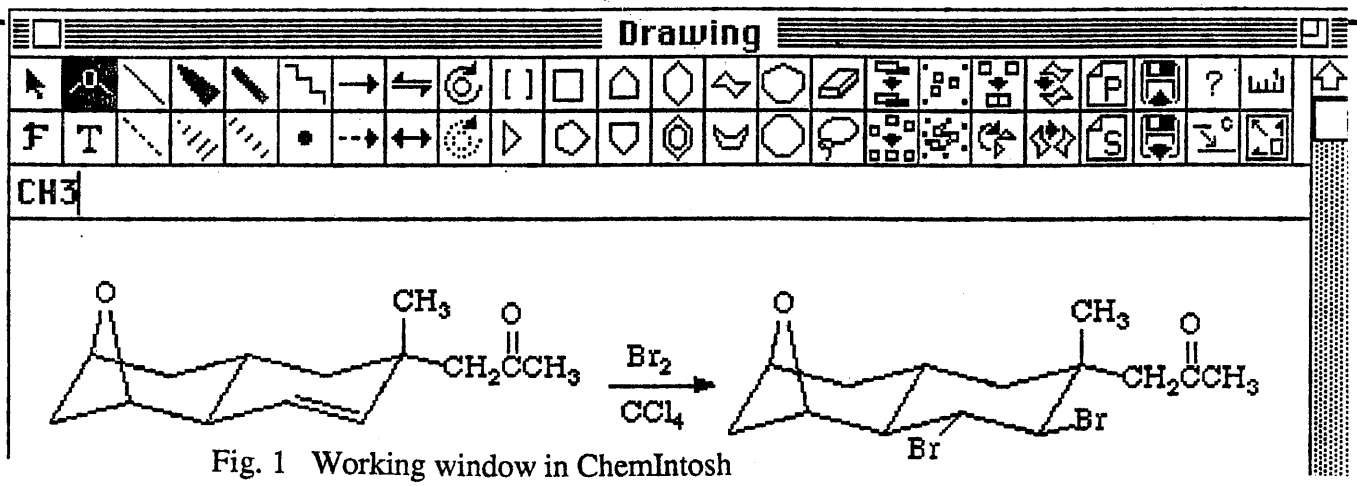

Fig. 1 Working window in ChemIntosh

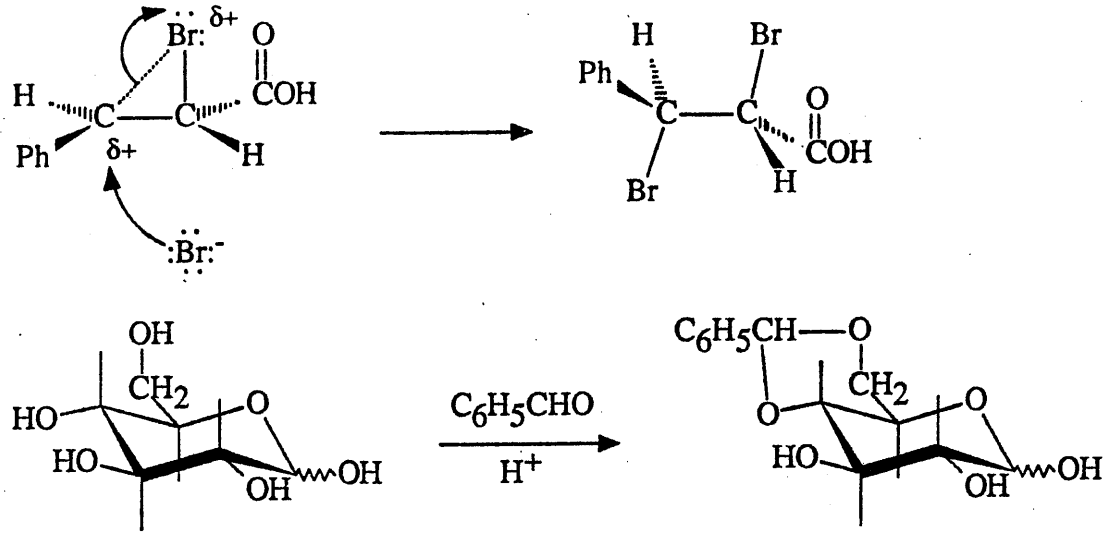

Fig. 2 Examples of output quality

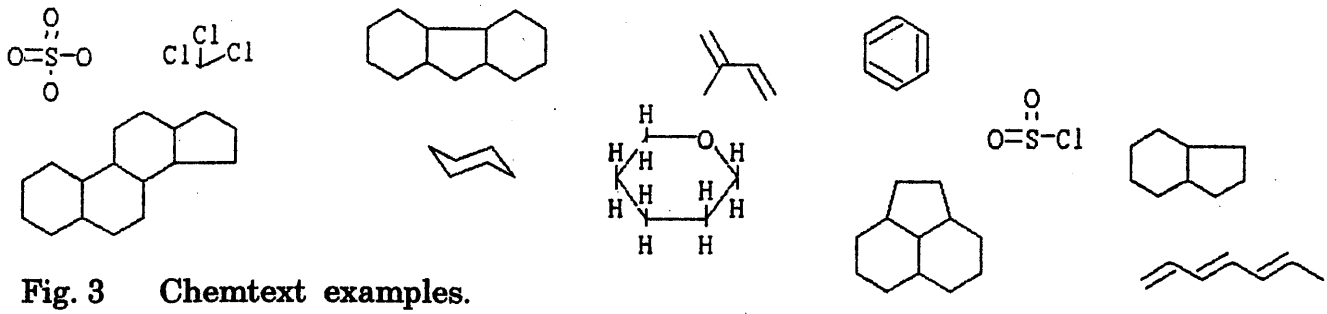

Fig. 3 Chemtext examples.

Published by the ACS Division of Chemical Education
Committee On Computers in Chemical Education
M.Lynn James, Chairman, Dept. of Chemistry, University of Northern Colorado, Greeley, CO 80639

Editor Brian J Pankuch, Department of Chemistry, Union County College, Cranford, NJ 07016

Submissions: General articles should be sent to editor Brian Pankuch at the above address. We need both 1) printed copy (hardcopy) and 2) a readable file on an IBM compatible diskette , 5 1/4 or 3 1/2. We have fewer problems with 3 1/2.
Submission deadlines: Fall issue - Sept. 25; Winter issue - Jan. 25; Spring issue - March 25.

Some Supported Word Processors	File Format
DCA files	.DCA
Microsoft Windows Write	.WRI
Microsoft Word	.DOC
MultiMate	.DOC
WordPerfect	.WP
WordStar 3.3	.WS
XyWrite III	.XYW

For example if you use MultiMate to make a file named foo send it as foo.DOC. FOR OTHER WORD PROCESSORS SEE Aldus PageMaker 3.0.
Please see the backcover for more complete list of readable wordprocessors, we can only handle IBM compatible applications. Please always send typed material in case of disk problems.

ALL NEW AND RENEWAL SUBSCRIPTIONS : PLEASE SEND REMITTANCE TO LYNN JAMES USING THE SUBSCRIPTION FORM IN BACK.
RATES: USA 1 year $2.50, two years $4.50: **Other countries** 1 yr $5, two yr $9. Please make a check or money order payable in US funds to Computers in Chemical Education Newsletter.

Managing Editor Kathy Stone Bailey, The Paideia School, 1509 Ponce de Leon Ave., Atlanta, GA 30307

Consulting Editor Donald Rosenthal, Department of Chemistry, Clarkson University, Potsdam, NY 13676. Send meeting notices, etc. to Don.

Contributing Editors: Send your contribution on specific areas to the appropriate editor.

For hardware Queries and Replies. Jim Beatty,IBM and compatibles Chemistry Dept., Ripon College, Ripon, WI 54971 . **OR** Tom Oyster, Apple Products, Computer Center, Box 248, Ripon College, Ripon, WI 54971.

For software Queries and Replies. Ken Loach, Department of Chemistry, SUNY College, Platts-burgh, NY 12901. BitNet address: Loachkw@snyplaba.

For Book Reviews. Harry Pence, Department of Chemistry, SUNY-Oneonta, Oneonta, NY 13820.

COVER : Output from ChemIntosh and ChemText, see article for more details.
The newsletter is produced using Aldus PageMaker 3.0, and is printed using a AST turboLaser/PS, with the help of the Union County College Resource Center.

MESSAGE FROM THE CHAIRMAN:

The last ten years in education have seen a great deal of emphasis on acquiring and implementing the microcomputer into the K-16 curricula. Despite this emphasis, extensive and meaningful uses of the computer in education is still quite spotty and limited. What the next ten years will bring is still in question. In an article, "Infusing Computing Into the Curriculum: Challenges For the Next Decade", which appeared in the April, 1989 Academic Computing issue, David L. Smallen, Hamilton College deals with this question. In the material which follows, I have tried to summarize his more salient points.

Although there are many reasons that might be given to justify a more extensive infusion of computers, Smallen gives what he considers to be three fundamentally sound reasons: 1) the emergence of computer science as a true liberal art, 2) the need to prepare students for life in the information society, and 3) the potential that computing has for improving the instructional process. With respect to the second point, Harlan Cleveland, Dean of the Hubert Humphrey School of Public Affairs at the University of Minnesota stated "by the end of the century, approximately two-thirds of all work will be information work ... (computers) empower those who learn to use them to make complex judgments in the more mindful knowledge of alternative futures ... People who do not educate themselves-and keep reeducating themselves-to participate in the new knowledge environment will be the peasants of the information society." For the third point, the computer offers the possibility of individualizing instruction and simulating the real world, and, in doing so, the learning process can be made more interesting, challenging, and effective.

Smallen suggests seven challenges that stand in the way of revitalizing and recreating educational disciplines through the infusion of computers into the curriculum.

I. **Provide appropriate recognition and incentives for faculty to improve the process of instruction.** The lack of adequate reward structures for faculty software developers is a major road block. We see a parallel example to this problem in the traditional lack of college or university support for helping faculty become better teachers in comparison with support for research. Solutions to this lack of recognition lie in such things as having the process of evaluating software similar to the way other professional activities are evaluated and in providing national recognition for quality software and courseware that is developed.

II. **Access, pricing, and distribution mechanisms for instructional software must be improved.** Faculty must have convenient access to information about such software that includes at least a list of names of others that have used the software in a similar setting and how to contact them. Reasonable pricing is essential so that students can afford to buy it, when asked to, and so that colleges can afford to license sufficient copies for their legitimate use. Distribution mechanisms must be set up to make it possible for faculty to obtain "desk copies" of software similar to how textbooks are handled.

III. **Develop more effective planning processes for instructional use of information technology resources.** Informal planning methodologies dealing with instructional technology typically used in the past must be replaced with methods that maximize the impact of it in relationship to the individual institution's mission. The time has come for us to do more than talk about planning, we must make a commitment to do it.

IV. **Encourage the formation of consortia of universities and colleges to deal with instructional applications of information technology.** One of the great strengths of higher education in the U.S. is the diversity of institutions. Consortia need to be built on the various strengths of different institutions, such as coupling the incentives of the teaching institution with the technical expertise of the research university to develop teams of educators working on the problem of developing high quality software. Consortia can consider some of the significant hurdles present at smaller institutions including lack of technical expertise, access to information channels, and access to more favorable pricing for software.

V. **Emphasize the role of the computer as a general learning tool.** Students need to learn more software tools to help them throughout their undergraduate experience and beyond. This includes word processing, general purpose tools for problem solving, database managers, tools to improve their ability to read critically, an opportunity for creative thinking, and others.

VI. **Ethical and legal use of software must become the norm.** Institutions of higher education must make a commitment to the legal and ethical use of software, thus creating an atmosphere on campus that will instill in their students a respect for the law and a desire to act in an ethical manner. In part, this is accomplished by institutions licensing sufficient copies of software for expected usage levels thus eliminating the incentive to copy software illegally. In turn, vendors must recognize these efforts by making instructional software affordable. Further, those responsible for computer services should adopt policies of not helping people with software problems if it is clear that the software is being used illegally.

VII. **Information technology services organizations must create supportive environments.** Smallen suggests that this should include standardizing hardware and software configurations on campus to eliminate wasted time dealing with technical interfacing problems, setting up effective software libraries or network servers for public facilities to reduce the cost of instructional software implementations, setting up classrooms equipped with large screen projectors and hookups to tie into the computer systems used in instruction, and working out standards that can be used in planning new public computing facilities.

With the varied uses of computer technology that can be made in chemistry we, as chemical educators, can play an important role in this infusion. In the words of John Kemeny, a recognized pioneer in the field of using computing for instruction, "Once you succeed in integrating the computer into the classroom, you will find that your entire style of teaching changes, and I can assure you that you will never go back to the old fashioned method..."

Editor

Choosing a new computer. After 3

years of considering different computer systems and working on extra projects to get the money to buy a new system, I finally bought a new one. It wasn't easy or obvious which to choose.

My first need was for a good system to write programs on since that is my primary interest. Having transferred programs successfully between many systems I knew it could be done, but usually with much time and effort. Since I'd rather spend this effort on programming, another requirement was to develop programs on a system available to my students. So we had to have the system available on campus. It also had to be easy for very poorly prepared freshmen to use. It would be helpful to have many excellent programs available. Needed were the best in programming languages and tools, great graphics (I think the weaker the student the more important the graphics are). I needed a fast system since I don't have much patience waiting for computers to work. I'd like any new neat innovations in hardware or software to be available for my machine. It would have to be expandable.

My first thought was a Sun. I worked with one for quite awhile learning C and UNIX. Quite impressive in most departments, but too complicated for my freshmen. Also we don't have any available for students.

The Apple II systems don't seem powerful enough, but extra hardware keeps coming out and we have many around campus. Many students also own their own. But they are pretty slow and the programming tools are not the best.

IBM and especially clones were appealing. You can get enormous amounts of great software, add on parts, and if a clone would serve your purposes a hefty discount from IBM prices. Almost any new idea in software or hardware would have to be available eventually because of the large number of IBM PC's around.

Macintosh also looked good because large amounts of great software, add ons, etc. But no Mac clones so the prices looked quite formidable.

At this point I was thinking about an IBM 70 or 80, and Mac II,. but leaning pretty heavily toward an IBM clone because of the price. Just to be sure I called some colleagues at Princeton to see what they thought. One physcist somewhat, to my surprise, was heavily in favor of the Mac. His responsibility is purchasing systems for a micro lab used by students and faculty and he travels to all the computer shows. Many vendors love to say Princeton is using their stuff so he gets many previews of programs and hardware long before the rest of us mortals.

He said the Mac interface is so consistent across applications that once you have learned the Mac interface you have about 80% of each new program. In his opinion Mac software is 3-4 years ahead of similar IBM software. This probably isn't true for programs written by the same company for both machines, but it does seem to take more than windows and a mouse to make a good useable interface.

Further research with colleagues in chemical industry brought similar comments. Even graphs showing significant decreases in time required to learn new programs on the Mac. Time to learn new applications on IBM systems supposedly doesn't decrease as much because each is quite different from the others. This was a big point for me since I've seen my family struggle getting work done on my old system when they had been off for a while. I just couldn't imagine my average student putting in the time and effort to learn a new set of commands for each program. In fact since I've been using computers with my students for over 12 years I know they won't learn new programs without a lot of effort on my part.

So I started looking seriously at the Mac II. Prices still bothered me, but I found that list prices have little to do with what you pay. Discounts range from 25-50% from list! Since prices are changing constantly and the discounts you can get depend very much on your personal situation, I won't even try to quote prices. First decide on the equipment then shop around for the best prices. It really makes a difference. Your best discounts are usually available thru your college, state purchasing plans, a relative who works for a company that gets sizeable discounts.

I decided on a Mac IIx with a 80 meg hard disk, the new high density diskette drive (1.4 meg) which can also read and write to IBM formatted disks! I also got A/UX the UNIX like operating system which is new to the Mac. Working with it for about 6 months now, I'm getting pretty comfortable. So far it has pretty much lived up to its billing. It does take awhile to learn and longer to program, but my learning time on each new well designed (follows the Mac guidelines) program is shorter and shorter. I find it very intuitive and friendly. You can experiment instead of reading manuals and you can usually do what you want without crashing the system.

My experience with my students has been excellent. I asked my students how many had experience with the Mac - it was less than 5%. I arranged to bring them into the Mac lab and showed them how to start the Mac and bring up and use a program I wrote. Within 5-10 minutes pairs of students at each Mac were successfully doing and learning about problems. I was of course very pleased. If new users can be up and running with this little, effort the future is looking more promising. Comments from students ranged from 'it's a lot of fun' to 'it was almost too easy to learn' (these are students talking about chemistry!!!). Not a single negative comment. Yet.

My own experiences are similar. My programming tools are the most advanced available. They are much more complicated than the average wordprocessor or spreadsheet. The Mac interface is so intuitive that I can lay off for 6-8 weeks and start right back as if I worked yesterday. My family also loves it.

Problem: I have A/UX 1.01 (Apple Unix) up on my machine. It takes 65 meg of my hard disk and I cannot run a single program I have under it. This leaves under 15 meg for my stuff on a 80 meg hard disk. Although I'm probably developing some good habits in throwing old stuff away I'd really like more of my disk storage available. Version 1.1 A/UX showed up recently, on something over 30 diskettes. I looked at the rather large amounts of documentation and decided I may need a lot of time to set it up. UNIX is powerful but it is difficult to work with. If I can get any info on the new version I may try it, but I'm seriously considering erasing A/UX and starting over without it. I've not found any good sources on using A/UX , and this makes the lack of understandable documentation a major problem. Most Mac ,documentation is very good but not for A/UX.

Recommendations: 1) Chemists generally need complicated word processing so the enhanced keyboard is very worthwhile.

2) Once I filled a 650 meg disk in 3 weeks so I feel a big disk for storage is useful.

3) I got used to a large screen monitor while using the Sun, so I have a 20 inch Moniterm Viking 2/72. At the moment I have no pressing need for color. The large size screen is much more useful for almost everything. I'd get a no glare coating for the screen. Very helpful. I use a standard IBM screen for this newsletter and it probably takes 8-10 times longer because of the constraints of the screen. All you need is to win a lottery to pay for the equipment.

NOTE:

Chemtext is a chemist's wordprocessor from Molecular Design (415-895-1313). The third figure down on the cover was done with ChemText. I find it difficult to use, nonintuitive, crashes, loses material, and has some of the worst documentation I have seen. I suppose if you used it a lot you'd get used to it. Personally I can do everything I need in far more pleasant ways. If anyone has used it extensively and likes it feel free to send a positive review.

```
SOURCE FILE: FLAM1
------ NEXT OBJECT FILE NAME IS FLAME.OBJ0
7C00:             1          ORG  $7C00
0000:             2          OBJ  $7C00
7C00:             3 *
7C00:             4 ********************
7C00:             5 * FLAME
7C00:             6 ********************
7C00:             7 *
0025:             8 VTAB     EQU  $25
0024:             9 HTAB     EQU  $24
6200:            10 SET1     EQU  $6200     ;REGULAR CHARACTER SET
6200:            11 SET2     EQU  $6200     ;ALTERNATL CHARACTER SET
7C00:            12 FLIKLOC  EQU  $7C00     ;VTAB AND HTAB OF FLAME LOCATION IN
7C00:            13 *                            THESE TWO BYTES
7C02:            14 FLAG     EQU  $7C02     ;STOP FLAG
7C03:            15 TOTCNT   EQU  $7C03     ;A COUNTER - WHEN TOTCNT IS ZERO
7C00:            16 *                       THE ROUTINE EXITS
7C05:            17 BURNCT   EQU  $7C05     ;FLAME PICTURE TO BE DISPLAYED
7C00:            18 *
7C00:            19 *
7C00:            20          DS   16        ;SAVE 16 BYTES FOR CONSTANTS
7C10:            21 *
7C10:A9 0B       22          LDA  #11       ;VTAB OF FLAME LOCATION
7C12:8D 00 7C    23          STA  FLIKLOC
7C15:A9 09       24          LDA  #9        ;HTAB OF FLAME LOCATION
7C17:8D 01 7C    25          STA  FLIKLOC+1
7C1A:A9 00       26          LDA  #$00      ;ZERO THE STOP FLAG
7C1C:8D 02 7C    27          STA  FLAG
7C1F:8D 03 7C    28          STA  TOTCNT
7C22:A9 E2       29          LDA  #$E2      ;THE SECOND FLAME PICTURE
7C24:8D 05 7C    30          STA  BURNCT
7C27:A5 25       31          LDA  VTAB      ;SAVE LOCATION FROM WHICH
7C29:48          32          PHA            ;THE ROUTINE WAS CALLED
7C2A:A5 24       33          LDA  HTAB
7C2C:48          34          PHA
7C2D:            35 *
7C2D:            36 *
7C2D:20 83 7C    37 ROUND    JSR  LOCFL
7C30:            38 *
7C30:20 06 62    39 FLIKER   JSR  SET2
7C33:AD 05 7C    40          LDA  BURNCT    ;LOAD CURRENT REPRESENTATION OF FLAME
7C36:C9 EA       41          CMP  #$EA      ;THIS IS THE LAST FLAME PICTURE
7C38:D0 05       42          BNE  NO
7C3A:A9 E1       43          LDA  #$E1      ; LOAD FIRST FLAME PICTURE
7C3C:8D 05 7C    44          STA  BURNCT
7C3F:20 ED FD    45 NO       JSR  $FDED     ;PRINT FLAME PICTURE
7C42:20 6C 7C    46          JSR  DELAY
7C45:EE 05 7C    47          INC  BURNCT    ;GOTO NEXT FLAME PICTURE
7C48:EE 03 7C    48          INC  TOTCNT
7C4B:D0 12       49          BNE  ROO
7C4D:            50 *
7C4D:20 83 7C    51 EXIT     JSR  LOCFL
7C50:A9 A0       52          LDA  #$A0      ;BLANK FLAME
7C52:20 ED FD    53          JSR  $FDED
7C55:68          54          PLA            ; RETURN TO CALLING LOCATION
7C56:85 24       55          STA  HTAB
7C58:68          56          PLA
7C59:85 25       57          STA  VTAB
7C5B:20 00 62    58          JSR  SET1      ;REVERT TO NORMAL CHARACTER SET
7C5E:60          59          RTS
7C5F:            60 *
7C5F:20 72 7C    61 ROO      JSR  KYBD      ;CHECK KEYBOARD
7C62:AD 02 7C    62          LDA  FLAG
7C65:C9 10       63          CMP  #$10      ;IS FLAG SET?
7C67:F0 E4       64          BEQ  EXIT
7C69:4C 2D 7C    65          JMP  ROUND
7C6C:            66 *
7C6C:            67 *
7C6C:A9 20       68 DELAY    LDA  #$20      ;STIRRER AND DROPS WILL GO HERE
7C6E:20 A8 FC    69          JSR  $FCA8
7C71:60          70          RTS
7C72:            71 *
7C72:AD 00 C0    72 KYBD     LDA  $C000     ;WAS A KEY PRESSED
7C75:2C 10 C0    73          BIT  $C010     ;CLEAR KEYBOARD STROBE
7C78:C9 D3       74          CMP  #$D3      ;LOOK FOR S KEY
7C7A:F0 01       75          BEQ  STOP
7C7C:60          76          RTS
7C7D:            77 *
7C7D:A9 10       78 STOP     LDA  #$10      ;SET FLAG IF S WAS PRESSED
7C7F:8D 02 7C    79          STA  FLAG
7C82:60          80          RTS
7C83:            81 *
7C83:AD 00 7C    82 LOCFL    LDA  FLIKLOC   ;SET TABS TO FLAME LOCATION
7C86:85 25       83          STA  VTAB
7C88:AD 01 7C    84          LDA  FLIKLOC+1
7C8B:85 24       85          STA  HTAB
7C8D:80          86          RTS
7C8E:            87 *

*** SUCCESSFUL ASSEMBLY: NO ERRORS
```

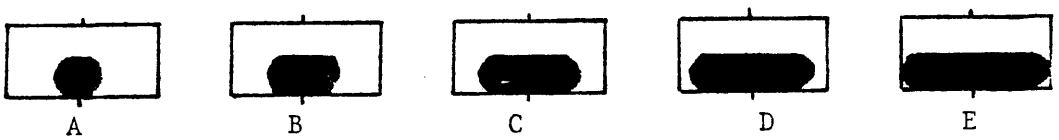

Figure 1

An Animation of Distillation
Part II
The Stirrer

This is the second of a series of articles which describe how to program an animated graphic for distillation. The complete graphic shows a flickering flame, a rotating stirrer bar, falling liquid, and a slowly increasing liquid distillate. In Part I the production of the flame was described. In this part, we explore the illusion of a spinning bar. It is assumed that you have a copy of CHEMUTIL-2(1) available for your Apple II series microcomputer.

A BASIC listing of a program which shows a spinning bar is given on page 20 of the CHEMUTIL-2 documentation. It works by printing a series of images of the bar. It uses the same principle that was used to simulate the flame but with a significant difference in application. The flame printed characters directly over one another, and it did not matter in which order they were chosen. The bar uses two characters printed side by side and the order is important.

To show a half turn of the bar requires the definition of ten new characters. These are then printed in pairs rapidly. The pairs are shown in Figure 1.

A full turn of the bar requires rapid printing in place of each pair in the order A B C D E D C B A. A continuing loop like that gives the illusion of a rotating stirrer bar. The CHEMUTIL-2 program is not quite as sophisticated as this in that it actually loops through A B C D E A. You can see that if you add a delay to that program so that each image stays on the screen for a longer time(2). The result of that economy of code is that the bar has a small hitch in its rotation which actually helps the illusion in that rotating magnetic bars often have a small kick in an otherwise smooth movement.

We need to rewrite the routine so that it can be successfully integrated with the flame routine and the yet-to-be developed falling drop code. The same logistical problems must be solved as were encountered with the flame. An appropriate program loop would be:-

Locate bar position : Print bar pair(A) : Delay : Locate bar position : Print bar pair(B) : Delay : etc.

However, we must arrange to have the full nine pair sequence for a full turn rather than the truncated five pair sequence of the CHEMUTIL-2 example. The reason is that we cannot be sure that the bar will rotate fast enough so that the hitch is acceptable when the flame and falling drop animation are integrated into it. We can rotate the bar slower and maintain the illusion when the nine pair sequence is in place. One way to code this would be to have two loops. The first loop prints images A to E and the second from D to B. However execution time can be reduced by defining six more characters which are duplicates of pairs D, C and B respectively. Then successive full sweeps through the eight pairs of characters will constitute a series of full turns of the stirrer bar. The additional six characters lowing the ten characters already available in CHEMUTIL-2 starting at $7390.

Listing 1 will add the duplicated characters to CHEMUTIL-2 and Listing 2 is the BASIC code for the rotating bar. Notice the similarity of Listing 2 to the program used to show the flickering flame. The ability to stop the animation by pressing the S-key has been included.

Listing 1

```
10 FOR I = 1 to 48
20 Read X : POKE (29599 + I),X
30 NEXT I
40 DATA 0, 0, 0, 120, 124, 126, 124, 120
50 DATA 0, 0, 0, 15, 31, 63, 31, 15
60 DATA 0, 0, 0, 112, 120, 126, 120, 112
70 DATA 0, 0, 0, 7, 15, 31, 15, 7,
80 DATA O, O, O, 96, 112, 120, 112, 96,
90 DATA O, O, O, 3, 7, 15, 7, 3,
```

Listing 2

```
10 CALL 25042 : PRINT "&" : REM
ENABLE CHEMUTIL-2 SET 2
20 FOR I = 106 TO 121 STEP 2
30 VTAB 10 : HTAB 10
40 PRINT CHR$ (I) + CHR$ (I + 1)
50 X = PEEK (-16368), O : IF X = 211
THEN I = 122 : GOTO 80
60 NEXT I
70 GOTO 20
80 VTAB 20 : END
```

The source of the code is given for the machine language version. Enter it into your Apple with an assembler as via the monitor. Then, after BLOADing CHEMUTIL-2 and running Listing 1,

```
RUN  10 CALL 25042
       20 CALL 31760
       30 IF PEEK (31746) = 16 THEN
END
          40 GOTO 20
```

Notice that the resulting stirrer is very jerky. That is because the delay, at Line 71 of the source code, is too short so that several bar images are drawn while the monitor screenis being refreshed. Try poking other values at 31861 ($7C75) and seeing the effect on the bar.

The machine language version was written so that the program terminates when the S-key is pressed or after 256 bar images have been shown. The latter is controlled by the TOTSPN counter. Its effect can be negated by removing Line 54 of the source code or by poking 31832, 234 : 31833, 234 and 31834, 234. It is useful when your overall program requires that the animation cease when the S-key is pressed or after a set time by a loop like FOR I = 1 TO 200 : CALL 31760 : NEXT I.

The next part of this series will discuss how the two separate routines for stirrer and flame can be integrated so that they appear to be executing simultaneously.Only machine language source code will be given since BASIC coding would execute too slowly for a realistic illusion.

An Animation of Distillation
Part III
Combined Heating and Stirring :

The two earlier parts of this series showed how to write routines for displaying a flickering flame and a spinning stirrer bar on the Apple graphics screen using CHEMUTIL-2 (1) as the character generator. In this article, I shall describe how the routine can be combined so that the flame and stirrer appear to be activated simultaneously. Since a BASIC program would execute too slowly for effective animation only the machine language source code is given.

The overall algorithm is simple in principle.Show flame (1), show stirrer (1), delay, show flame (2), show stirrer (2), delay, etc. As we have previously indicated, the delay is present because it will allow us to control the overall speed of the illusion and also leaves room for the introduction of the falling drop routine later.

This algorithm would work but is too limited because the bar would rotate at the same speed as the flame flickered. the simple alternation of flame and stirrer means that whatever affects the speed of one affects the speed of the other. A rapidly flickering flame with a slow stirrer would be impossible. Fortunately, we have already allowed for the latter possibility when we wrote the separate routines because they have separate delay routines incorporated into each. More importantly, we included the counter TOTCNT and TOTSPN which allow an exit from either routine when the counters are zeroed by the executing programs. this is the true significance of Line 48 and. Line 54 in the FLAME and SPINNER

```
SOURCE FILE: SPINNER
------ NEXT OBJECT FILE NAME IS SPINNER.OBJ0
7C00:            1          ORG  $7C00
0000:            2          OBJ  $7C00
7C00:            3 *
7C00:            4 ********************
7C00:            5 * SPINNER
7C00:            6 ********************
7C00:            7 *
0025:            8 VTAB     EQU  $25
0024:            9 HTAB     EQU  $24
6200:           10 SET1     EQU  $6200      ;REGULAR CHARACTER SET
6206:           11 SET2     EQU  $6206      ;ALTERNATE CHARACTER SET
7C02:           12 FLAG     EQU  $7C02      ;STOP FLAG
7C06:           13 SPINLOC  EQU  $7C06      ;VTAB AND HTAB LOCATION IN
7C00:           14 *           THESE TWO BYTES
7C08:           15 TOTSPN   EQU  $7C08      ;A COUNTER -WHEN TOTSPN IS ZERO
7C00:           16 *                           THE ROUTINE EXITS
7C0A:           17 SPNCNT   EQU  $7C0A      ;BAR IMAGE TO BE DISPLAYED
7C00:           18 *
7C00:           19 *
7C00:           20          DS   16
7C10:           21 *
7C10:A9 0B      22          LDA  #11
7C12:8D 06 7C   23          STA  SPINLOC    ;VTAB OF BAR LOCATION
7C15:A9 09      24          LDA  #9
7C17:8D 07 7C   25          STA  SPINLOC+1  ;HTAB OF BAR LOCATION
7C1A:A9 00      26          LDA  #$00       ;ZERO THE STOP FLAG
7C1C:8D 02 7C   27          STA  FLAG
7C1F:8D 08 7C   28          STA  TOTSPN
7C22:A9 EA      29          LDA  #$EA
7C24:8D 0A 7C   30          STA  SPNCNT     ;LOAD FIRST BAR IMAGE
7C27:A5 25      31          LDA  VTAB       ;SAVE LOCATION FROM WHICH
7C29:48         32          PHA             ;THE ROUTINE WAS CALLED
7C2A:A5 24      33          LDA  HTAB
7C2C:48         34          PHA
7C2D:           35 *
7C2D:           36 *
7C2D:AD 06 7C   37 SPINST   LDA  SPINLOC    ;SET TABS TO BAR LOCATION
7C30:85 25      38          STA  VTAB
7C32:AD 07 7C   39          LDA  SPINLOC+1
7C35:85 24      40          STA  HTAB
7C37:           41 *
7C37:20 06 62   42 SPINIT   JSR  SET2
7C3A:AC 0A 7C   43          LDA  SPNCNT     ;LOAD CURRENT FIRST BAR IMAGE
7C3D:C9 FA      44          CMP  #$FA       ;THIS IS THE LAST BAR IMAGE
7C3F:D0 05      45          BNE  RO
7C41:A9 EA      46          LDA  #$EA       ;RESTART BAR IMAGES
7C43:8D 0A 7C   47          STA  SPNCNT
7C46:20 ED FD   48 RO       JSR  $FDED      ;PRINT LEFT BAR IMAGE
7C49:EE 0A 7C   49          INC  SPNCNT
7C4C:AD 0A 7C   50          LDA  SPNCNT
7C4F:20 ED FD   51          JSR  $FDED      ;PRINT RIGHT BAR IMAGE
7C52:20 74 7C   52          JSR  DELAY
7C55:EE 0A 7C   53          INC  SPNCNT     ;GOTO NEXT BAR PICTURE
7C58:EE 08 7C   54          INC  TOTSPN
7C5B:D0 0A      55          BNE  RCO
7C5D:           56 *
7C5D:68         57 EXIT     PLA             ;RETURN TO CALLING LOCATION
7C5E:85 24      58          STA  HTAB
7C60:68         59          PLA
7C61:85 25      60          STA  VTAB
7C63:20 00 62   61          JSR  SET1       ;REVERT TO NORMAL CHARACTER SET
7C66:60         62          RTS
7C67:           63 *
7C67:20 7A 7C   64 RCO      JSR  KYBD       ;CHECK KEYBOARD
7C6A:AD 02 7C   65          LDA  FLAG
7C6D:C9 10      66          CMP  #$10       ;IS FLAG SET
7C6F:F0 EC      67          BEQ  EXIT
7C71:4C 2D 7C   68          JMP  SPINST
7C74:           69 *
7C74:           70 *
7C74:A9 01      71 DELAY    LDA  #$01       ;LEAVE ROOM FOR MORE
7C76:20 A8 FC   72          JSR  $FCA8
7C79:60         73          RTS
7C7A:           74 *
7C7A:AD 00 C0   75 KYBD     LDA  $C000      ;CHECK FOR KEY PRESS
7C7D:2C 10 C0   76          BIT  $C010      ;CLEAR KEYBOARD STROBE
7C80:C9 D3      77          CMP  #$D3       ;LOOK FOR S KEY
7C82:F0 01      78          BEQ  STOP
7C84:60         79          RTS
7C85:           80 *
7C85:A9 10      81 STOP     LDA  #$10       ;SET FLAG IF S WAS PRESSED
7C87:8D 02 7C   82          STA  FLAG
7C8A:60         83          RTS
7C8B:           84 *

*** SUCCESSFUL ASSEMBLY: NO ERRORS
```

source codes.

The improved algorithm becomes:
(A) Show a series of flame images with a delay between them until the TOTCNT counter becomes zero then ?
(B) Show a series of stirrer images with a delay between them until the TOTSPN counter becomes zero then
(C) Loop back to (A).

The flame delay and the stirrer delay need not be the same length and one (or both) of them can be used to check the keyboard periodically. There are now four locations which can be used to adjust the speed of the overall animation or the flame and stirrer separately. They correspond to the lengths of the two delay loops and the initial values chosen for TOTCNT and TOTSPN.

The source code for the combined routines shows how simple it ?was to combine them. Since the separate routines wee debugged separately, we can be certain that if the combination does not function the interface is at fault. In my experience the most likely errors at this point are made by not initializing variables and counters as the switch is made from one working routine to the other.

To execute the code it must be entered at $7C10 and CHEMUTIL-2 BLOADed. Listing 1 from part II must have been run to supply the extra stirrer images. Then the following brief BASIC program is run:

```
10 CALL 25042 : PRINT "&"
20 CALL 31760 C
```

You will now see a flickering flame with a rotating bar over it. The bar has been placed over the flame by Lines 33-36 of the source code. You can easily alter its relative position if you desire but this program puts it over the flame because it is the correct relative position for a distillation which is our ultimate goal.

In this code as written there are 32 flame images between every stirrer image. i.e. There are 256 flame images for every full turn of the stirrer bar. It is instructive to see the effects of introducing other values from 1-255 into TOTSPN, TOTCNT and the two delay intervals. Try poking values into locations 31850 (TOTSPN), 131855 (TOTCNT), 31891 (DELAY) and 31963 (DELSPN).

An advantage to writing the separate routines and then interfacing them is that they can still be used separately when needed. The equivalent combination in BASIC is a loop like GOSUB IFLAME : GOSUB STIRRER. A loop like this will show only the flame if the first line of the stirrer routine is RETURN. So by poking the first byte of the SPINST routine in the machine language code with the equivalent of RETURN only the flame image will appear. Try Poke 31913, 96 and see the result.

The disadvantage of interfacing separate routines in this way is that the combination will execute slower than a specially designed combination program. This is seldom of real consequence for machine language because

the length of the delay loops can be adjusted to compensate, but in BASIC programs it can be devastating. Remember that when you do animation in machine language you usually must deliberately slow it down but in BASIC the animation will be too slow from the first.

The fourth part of this series will discuss the falling drop and accumulating liquid animation.

(1) Bendall, V. "CHEMUTIL-2, A Chemistry Programming Utility"; Project SERAPHIM, NSF Science Education; Department of Chemistry, Eastern Michigan University, Ypsilanti, MI 48197, 1985.

(2) For example, add the following line to the CHEMUTIL-2 listing. 75 FOR K = 1 to 100 : NEXT K and RUN the program again.

```
SOURCE FILE  SPINBURN
------ NEXT OBJECT FILE NAME IS SPINBURN.OBJ0
7C00:             1         ORG  $7C00
0000.             2         OBJ  $7C00
7C00:             3 *
7C00:             4 ********************
7C00:             5 * SPINBURN
7C00:             6 ********************
7C00:             7 *
0025:             8 VTAB    EQU  $25
0024:             9 HTAB    EQU  $24
6200:            10 SET1    EQU  $6200
6206:            11 SET2    EQU  $6206
7C00:            12 FLIKLOC EQU  $7C00    ;VTAB, HTAB OF BURNER FLAME
7C02:            13 FLAG    EQU  $7C02    ;SETS TO $10 WHEN S IS PRESSED
7C03:            14 TOTCNT  EQU  $7C03    ;NUMBER OF TIMES THROUGH FLAME CYCLE
7C05:            15 BURNCT  EQU  $7C05    ;FLAME SYMBOL BEING DRAWN $E1-$E9
7C06:            16 SPINLOC EQU  $7C06    ;VTAB, HTAB OF STIRRER BAR
7C08:            17 TOTSPN  EQU  $7C08    ;NUMBER OF TIMES THROUGH SPIN CYCLE
7C0A:            18 SPNCNT  EQU  $7C0A    ;SPIN SYMBOL BEING DRAWN  $EA-$F3
7C00:            19 *
7C00:            20         DS   16
7C10:            21 *
7C10:A9 0E       22 START   LDA  #11      ;SET VTAB OF FLAME
7C12:8D 00 7C    23         STA  FLIKLOC
7C15:A9 09       24         LDA  #9       ;SET HTAB OF FLAME
7C17:8D 01 7C    25         STA  FLIKLOC+1
7C1A:A9 00       26         LDA  #$00     ;ZERO KYBD FLAG
7C1C:8D 02 7C    27         STA  FLAG
7C1F:A9 FF       28         LDA  #$FF
7C21:8D 03 7C    29         STA  TOTCNT
7C24:8D 08 7C    30         STA  TOTSPN
7C27:A9 EA       31         LDA  #$EA
7C29:8D 0A 7C    32         STA  SPNCNT
7C2C:AD 00 7C    33         LDA  FLIKLOC   ;STIRRER BAR TWO SPACES ABOVE FLAME
7C2F:38          34         SEC
7C30:E9 02       35         SBC  #2
7C32:8D 06 7C    36         STA  SPINLOC
7C35:AD 01 7C    37         LDA  FLIKLOC+1
7C38:8D 07 7C    38         STA  SPINLOC+1
7C3B:A9 E2       39         LDA  #$E2
7C3D:8D 05 7C    40         STA  BURNCT
7C40:A5 25       41         LDA  VTAB     ;SAVE CURRENT CURSOR LOCATION
7C42:48          42         PHA
7C43:A5 24       43         LDA  HTAB
7C45:48          44         PHA
7C46:            45 *
7C46:20 E8 7C    46 ROUND   JSR  LOCFL
7C49:            47 *
7C49:20 36 62    48 FLIKER  JSR  SET2
7C4C:AD 05 7C    49         LDA  BURNCT
7C4F:C9 EA       50         CMP  #$EA
7C51:D0 05       51         BNE  RO

7C53:A9 E1       52         LDA  #$E1
7C55:8D 05 7C    53         STA  BURNCT
7C58:20 ED FD    54 RO      JSR  $FDED
7C5B:20 92 7C    55         JSR  DELAY
7C5E:EE 05 7C    56         INC  BURNCT
7C61:EE 03 7C    57         INC  TOTCNT
7C64:D0 0D       58         BNE  ROO
7C66:20 A9 7C    59         JSR  SPINST
7C69:A9 FF       60         LDA  #$FF     ;SPINS BETWEEN FLICKER = 256-TOTSPN
7C6B:8D 08 7C    61         STA  TOTSPN
```

```
7C6E:A9 FF       62         LDA  #$FF     ;FLICKERS BETWEEN SPIN = 256-TOTSPN
7C70:8D 03 7C    63         STA  TOTCNT
7C73:20 98 7C    64 ROO     JSR  DELAY
7C76:AD 02 7C    65         LDA  FLAG
7C79:C9 10       66         CMP  #$10
7C7B:F0 03       67         BEQ  EXIT
7C7D:4C 46 7C    68         JMP  ROUND
7C80:            69 *
7C80:20 E8 7C    70 EXIT    JSR  LOCFL
7C83:A9 A0       71         LDA  #$A0     ;PUT OUT FLAME OF BURNER
7C85:20 ED FD    72         JSR  $FDED
7C88:68          73         PLA           ;GET ORIGINAL CURSOR LOCATION
7C89:85 24       74         STA  HTAB
7C8B:68          75         PLA
7C8C:85 25       76         STA  VTAB
7C8E:20 00 62    77         JSR  SET1     ;RETURN TO UPPER CASE
7C91:60          78         RTS
7C92:            79 *
7C92:A9 01       80 DELAY   LDA  #$01     ;DELAY BETWEEN FLAME IMAGES
7C94:20 A8 FC    81         JSR  $FCA8
7C97:60          82         RTS
7C98:            83 *
7C98:AD 00 C0    84 KYBD    LDA  $C000
7C9B:2C 10 C0    85         BIT  $C010
7C9E:C9 D3       86         CMP  #$D3     ;CHECK FOR S
7CA0:F0 01       87         BEQ  STOP
7CA2:60          88         RTS
7CA3:A9 10       89 STOP    LDA  #$10
7CA5:8D 02 7C    90         STA  FLAG
7CA8:60          91         RTS
7CA9:            92 *
7CA9:AD 06 7C    93 SPINST  LDA  SPINLOC  ;SET TABS TO STIRRER LOCATION
7CAC:85 25       94         STA  VTAB
7CAE:AD 07 7C    95         LDA  SPINLOC+1
7CB1:85 24       96         STA  HTAB
7CB3:            97 *
7CB3:20 06 62    98 SPINIT  JSR  SET2
7CB6:AD 0A 7C    99         LDA  SPNCNT
7CB9:C9 FA      100         CMP  #$FA
7CBB:D0 05      101         BNE  RO1
7CBD:A9 EA      102         LDA  #$EA
7CBF:8D 0A 7C   103         STA  SPNCNT
7CC2:20 ED FD   104 RO1     JSR  $FDED

7CC5:EE 0A 7C   105         INC  SPNCNT
7CC8:AD 0A 7C   106         LDA  SPNCNT
7CCB:20 ED FD   107         JSR  $FDED
7CCE:20 DA 7C   108         JSR  DELSPN
7CD1:EE 0A 7C   109         INC  SPNCNT
7CD4:EE 08 7C   110         INC  TOTSPN
7CD7:D0 D0      111         BNE  SPINST
7CD9:60         112         RTS
7CDA:           113 *
7CDA:A9 01      114 DELSPN  LDA  #$01     ;DELAY BETWEEN STIRRER IMAGES
7CDC:20 A8 FC   115         JSR  $FCA8
7CDF:60         116         RTS
7CE0:           117 *
7CE0:AD 00 7C   118 LOCFL   LDA  FLIKLOC  ;SET TABS TO FLAME LOCATION
7CE3:85 25      119         STA  VTAB
7CE5:AD 01 7C   120         LDA  FLIKLOC+1
7CE8:85 24      121         STA  HTAB
7CEA:60         122         RTS
7CEB:           123 *

*** SUCCESSFUL ASSEMBLY: NO ERRORS
```

Jim Beatty , Chemistry Department, Ripon College.
Uninterruptible Power Systems for Your PC

Byte, April 1989, 14(4) 162-176. If you live in fear that the power to your PC will go out and you will lose your report or important data see this article. Twelve power systems are reviewed. Costs of a system range from $600 to about $2000. I have found my battery powered laptop a life saver during thunderstorms.

Monitor Life

The New York Times, Sunday, April 2, 1989, page F-19. Peter Lewis reviews screen dimmers for IBM compatible and Apple Macintosh computers. The screen dimmers are resident software programs which dim or blank tahe screen after a selected period of non-use. The screens are reactivated by hitting any key. I have one copy on order to prevent burn-in on a VGA monitor on an IBM compatible which is left on. An evaluation is forthcoming.

Linking Between the Mac and IBM Worlds

The New York Times, Sunday, June 19, 1989, page F-12. John Markoff offers an interesting and short review of the options in moving data from the two microcomputer worlds. The options are a mixture of software and hardware. We have used the Apple Talk option he discusses. It is low in cost and easy to use. Also see Byte, January 1989, 14(1) 317-325. These two articles complement each other and are worth filing.

Digitizing Tablets

Byte, January 1989, 14(1) 162-174. Twenty one IBM-PC digitizing tablets are reviewed. These tablets are for those who need high quality input of graphic data.

Full Spectrum Scanners

Byte, April 1989, 14(4) 189-194. Two full color scanners for the Mac and IBM worlds are evaluated. Cost is the order of $8000. This article offers a good introduction to available options in scanning.

Hand-Held Scanners

Byte, June 1989, 14(6) 187-191. Six scanners costing about $300 are evaluated. The cost is low enough that we are considering one for use in entering graphs, sketches, etc. into WordPerfect documents.

COVER STORY

Please see the cover for the figures from this article.If you need more information ,query professor Currie directly.

ChemIntosh

By: James Currie
Chemistry Department
Pacific University
Forest Grove, OR 97116

ChemIntosh is a structure drawing aid for Macintosh computers. For chemists, this tool is so incredibly useful that it could be worth buying a Macintosh just to use it. While having some functional similarity to ChemDraw, the fact that ChemIntosh is a desk accessory, greatly adds to its versatility. With it, you can generate typeset quality structures of all sorts for inclusion in written articles, databases and presentation materials.

As with nearly all Macintosh programs, ChemIntosh is easy to use. A very complete palette of tools is used to create the structural parts. All the common rings, bond types, and stereochemical notations are included. Tools for rotation, inversion, resizing, alignment, and grouping are also present. Quite frankly in the one and one-half years of use, I have not encountered a structure or equation that could not be adequately and easily drawn with it. Most of the tool characteristics such as default bond lengths, double bond spacing, wedge lengths or widths are easily customizable. And of course all installed fonts are available. The resulting graphic is easily copied to the clipboard and pasted into the currently active application. All the common applications, such as word processors or databases accept these structures and allow manipulation as needed. The structures are object graphics and take full advantage of the resolution of a postscript laser printer.

ChemIntosh is large for a desk accessory (36K) and uses a very large resource file, 360K. The DA itself must be either installed in the system file or accessed through a utility such as Suitcase II. The resource file may be anywhere but must be online during use. I use ChemIntosh conveniently on an Appletalk network with both components stored on the network server. Suitcase II is used to access the DA and a shareware utility, SetPaths, is used to transparently tell the system where to look for the resource file.

A second version of this program is known as ChemConnection. This desk accessory contains a full implementation of ChemIntosh but adds a structure translation tool for converting drawings to a text equivalent for on-line searches in Chemical Abstracts. Although I have not yet tried this feature on-line it's implementation is very convenient. One merely selects the drawing and clicks on the phone icon; the text equivalent is pasted on the clipboard ready to be sent by a communications program. The feature may be customized to give the desired translation into normalized or exact bond types.

ChemIntosh retails for $295 and ChemConnection for $395. Academic discounts are available. This is one of those "can't live without" programs that while not cheap is worth every penny of it.

TENTH C.C.C.E. NATIONAL COMPUTER WORKSHOPS

Saturday, June 30 through Tuesday, July 3, 1990

Clarkson University, Potsdam, NY

(1) **WORKSHOPS**

Each participant will register for one four day workshop.
All workshops provide lectures and hands-on experience.

(A) **Getting Started**
Paul Cauchon
Canterbury School, New Milford, CT 06776 (203-354-8683)

(B) **Microcomputer Graphics**
Victor Bendall
Eastern Kentucky University, Richmond, KY 40475 (606-622-1459)

(C) **Beginning Computer Interfacing using the APPLE II**
Robert Megargle
Cleveland State University, Cleveland, OH 44115 (216-687-2466)

(D) **Laboratory Techniques for Interfacing the IBM PC**
Kenneth Ratzlaff
University of Kansas, Lawrence, KS (913-864-3754)

(E) **Productivity Software in Chemistry - Spreadsheets and Databases**
Patricia C. Flath
Paul Smith's College, Paul Smiths, NY 12970 (518-327-6247)

(F) **Introduction to C**
Kenneth Loach
SUNY at Plattsburgh, Plattsburgh, NY 12901 (518-564-4116)

(G) **HyperCard on the Macintosh**
Robert C. Best, Jr.
Route 1 Box 91B, Russell, NY 13684 (315-347-3401)

(H) **Chemistry and the NeXT Computer**
Alfred J. Lata
University of Kansas, Lawrence, KS 66045 (913-864-4054)

(I) **Chemometrics: Processing Chemical Measurements with Personal Computers**
Thomas C. O'Haver
University of Maryland, College Park, MD (301-454-6128)

(J) **Childrens' Workshop - Beyond Drills and Games** (for 8 to 12 year olds)
Estella Bray
RD #1, Box 364, Potsdam, NY 13676 (315-265-3491)

(K) **Young Adults' Workshop - Lotus 123 for the Student** (13 to 17 year olds)
Estella Bray
RD #1, Box 364, Potsdam, NY 13676 (315-265-3491)

For additional information contact Professor Donald Rosenthal, Department of Chemistry, Clarkson University, Potsdam, NY 13676 (315-268-2352).

OVER FOR ADDITIONAL INFORMATION

(2) **Other Workshop Activities**

Book exhibit
Software exhibit and exchange
Daily general meeting
Banquet included in registration fee

(3) **Participant Costs**

Registration Fee - early - by May 15	$125
- late - after May 15	$155
$105 for each additional family member	
Meals	$16.50/day
Room rates - Suite with shared bathroom	$13.80/day single
	$ 9.50/day with roommate

NERM

The Northeast Regional Meeting of the American Chemical Society will be held at Clarkson from Tuesday, June 26 through Friday, June 29, 1990. Papers are being solicited for the NERM meeting. Symposia and general sessions on Computers in Education and on Chemical Education are planned. Papers on these topics should be submitted to Professor Donald Rosenthal, Department of Chemistry, Clarkson University, Potsdam, NY 13676 (315-268-2352). General papers should be submitted to Professor Petr Zuman - NERM, Department of Chemistry, Clarkson University, Potsdam, NY 13676 (315-268-2340).

Additional Information on Workshops

A final announcement containing a fuller description of the workshops and additional details will appear in the fall. Please fill in and return the form below if you wish to receive the additional information or write to Professor Donald Rosenthal, Chairman of the Organizing Committee, Clarkson University, Potsdam, NY 13676 (315-268-2352).

TENTH C.C.C.E. NATIONAL COMPUTER WORKSHOPS

Saturday, June 30 through Tuesday, July 3, 1990

Clarkson University, Potsdam, NY

___ Please send more information on the **National Computer Workshops**
___ Please send more information on the **NERM** Meeting
___ I am particularly interested in the Workshop(s) circled below:
A B C D E F G H I J K
Please print the information requested below:

Name _____
Address _____
_____
_____
Phone: _____ (work) _____ (home)

Please return to: Center for Continuing Education,
Snell Hall, Clarkson University, Potsdam, NY 13676
Phone: (315)268-6647

COMPUTERS IN CHEMICAL
EDUCATION NEWSLETTER
ISSN 8756-8829

Brian J. Pankuch, Editor
Department of Chemistry
Union County College
Cranford, New Jersey 07016

Donald Rosenthal, Consulting Editor
Department of Chemistry
Clarkson University
Potsdam, New York 13676