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
```

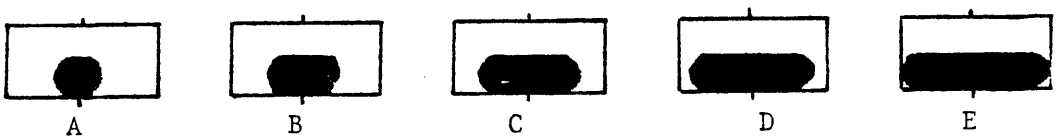

Figure 1

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