INTRODUCING APPLE SOFTWARE BANK CONTRIBUTED PROGRAMS

With this issue we enclose the first catalog of user contributed software, and express our thanks to all those both inside and outside of Apple Computer whose efforts made it possible.

The purpose of the contributed section of the software bank is to allow APPLE users to share programs among themselves easily and at very low cost. By setting up a system to make this possible, we at APPLE hope to bring about a cross-fertilization of the user community, with better programs and applications ideas as the result.

Here is a brief description of the program. More details will be found in the enclosed catalog.

As APPLE receives contributed programs, they are screened. The necessary documentation is assembled, and written catalog descriptions are produced for each program accepted. The author is then sent a $25 APPLE merchandise certificate.

Periodically, the available programs will be assembled onto diskettes, bulk duplicated, and supplied to APPLE dealers at a modest cost. The dealers will have unlimited reproduction rights to the programs on each diskette. As each diskette is released, APPLE will distribute a catalog of its contents to you along with CONTACT. When you see something you like, just call or visit your local dealer for a copy.

The library is a resource. In it you will find some programs you can use as is, some you can improve upon, and some that may not interest you at all. That's as it should be. Use what you can, send in your improvements, and let's work together to build a super library!

Oh, about the Bard on the cover. His image is one of many graphics demonstrations available on the first set of Software Bank diskettes. It was produced with a modified facsimile machine that digitized the original artwork, and a graphic processing program on a PDP-11/03 that reduced the image to an 8K byte database. It can now be displayed by the Apple in the Hi-Res Graphics mode.

LOCAL USER GROUPS

...and the beat goes on

With this issue of CONTACT, we're pleased to add five more groups to the rapidly growing list of APPLE user groups:

CALIFORNIA
APPLE CORE
Scott Kamins
Box 4816
San Francisco, CA 94101

OREGON
STEMS FROM APPLE
Ken Hoggatt
9195 S.W. El Rose Court
Tigard, OR 97223
(503) 639-5505 (home)
(503) 644-0161, x6136 (work)

TEXAS
APPLE SEED
The Computer Shop
6812 San Pedro
San Antonio, TX 78216
Bill Hyde
(512) 828-0553
(No name given for this group)
c/o Computer Solutions
Suite 124A, 9200 Broadway
San Antonio, TX 78217
Philip W. Jackson
(512) 828-1455
(800) 292-7652 (toll free)

ARKANSAS
Interested in joining or forming an APPLE user group? Contact C. Johnson, c/o dataCope, 5706A W. 12th St., Little Rock, AR 72204. (501) 666-8588.
If you're interested in forming or joining an APPLE user group in your area, call or visit your local APPLE dealer; he'll be glad to help. And if you know of an APPLE group that's not yet been listed in this newsletter (see CONTACT No. 1 and No. 2 for the listings of the groups that we know about), write us here at APPLE. Direct your note to Phil Roybal, marketing manager.

If you're in the DOS you try to load a program that had been saved under RAM APPLESOFII, but you are now using the ROM card version, your program will not run correctly. To get around this problem, load your program and then type

CALL 54514
and your program will be correct.

Similarly, a program that was saved onto diskette from the ROM card version and later loaded under RAM APPLESOFII will also cause problems. Simply type

CALL 3314
and your program will be correct. You can now save it onto tape.

The earliest production DOS had a problem: when in APPLESOFII, any Read or Write statements with line numbers of 256 or higher would be ignored. To solve this problem on any disks you may INIT in the future, bring up the DOS, remove the Write Protect sticker from your system's master diskette, and then type:

>PR#n (boot your system)
>BLOAD RAWDOS
>(hit Reset)
>*25D6:4C D5 3F
>*25DC:2E
>*3FD5 :E8 FO 1 60 4C DD 25
>*3DOG
>BSAVE RAWDOS, A$1B00, L$2500

Any new masters created from this original master diskette will now work properly.

If (in the DOS) you try to load a program that had been saved from the ROM card version and later loaded under RAM APPLESOFII, but you are now using the ROM card version, your program will not run correctly. To get around this problem, load your program and then type

CALL 54514
and your program will be correct.

Similarly, a program that was saved onto diskette from the ROM card version and later loaded under RAM APPLESOFII will also cause problems. Simply type

CALL 3314
and your program will be correct. You can now save it onto tape.

The earliest production DOS had a problem: when in APPLESOFII, any Read or Write statements with line numbers of 256 or higher would be ignored. To solve this problem on any disks you may INIT in the future, bring up the DOS, remove the Write Protect sticker from your system's master diskette, and then type:

>PR#n (boot your system)
>BLOAD RAWDOS
>(hit Reset)
>*25D6:4C D5 3F
>*25DC:2E
>*3FD5 :E8 FO 1 60 4C DD 25
>*3DOG
>BSAVE RAWDOS, A$1B00, L$2500

Any new masters created from this original master diskette will now work properly.

Two corrections to CONTACT 2, June '78

There is one correction to be made in each of two program listings in the HOW TO section of the last issue of CONTACT.

- The CONVERT program on Page 6: statement 251 should read
251 IF T<>0 THEN NEXT B:B=B+1: IF B<E THEN 40
- The HOW TO SET LOMEM WITHOUT HARDLY TRYING program on page 7: we should have tried harder. We now hasten to point out that, in statement 40, B cannot be a variable, since setting the new LOMEM will destroy the variables (as we ourselves said, in the paragraph just preceding the listing). The concept works, but make sure that you POKE in numbers in that line, not variables.
Cockpit errors and DOS

A high percentage — 75 percent, to be precise — of disk errors found so far are due to users trying to run APPLESOFT II by typing "RUN APPLESOFT." This seems reasonable, but APPLE II doesn't see it quite that way. As a result (among other things) you cannot reload programs saved on disk.

To get things to come out right — all pointers where they should be, etc. — from Integer BASIC simply type

FP (Carriage Return)

and you'll bring in APPLESOFT II, pointers and all.

To get back to Integer BASIC simply type INT.

About that Communications Card manual . . .

We've put together some additional information that will be needed by many of you using our Communications Card. Included are a routine for use with low-speed printers, information on setting the card's status register, information on setting up a 1200-baud data rate, and a number of other interesting tidbits, as well.

We will be including this addendum to the card manual with all future card shipments, and we'll also be distributing it to all our dealers. So if you already have an APPLE Communications Card, just call your APPLE dealer to get your copy of the additional card info — it'll be available about the time that you read this.

cannot be erased. Every cassette includes a write/protect feature in the form of two small plastic tabs on the edge of the cassette opposite its business end. When these tabs are pushed in, pulled out, or otherwise gotten out of the way, the cassette can no longer be written into.

Clearing the air. We make no profit from your contributions to the user portion of the APPLE software bank. We set up this portion of the bank simply to encourage and to ease the exchange of programs among APPLE users. The honorarium that we pay for contributions to this portion of the bank is merely our way of encouraging such contributions.

From time to time, of course, APPLE does purchase software, and does so at the fair market value. Such purchases are negotiated individually, based upon market conditions, applicability of the product, etc.

OUT OF THE MIST....

...ASCII character equivalents

The following table presents decimal-number equivalents and APPLE keyboard equivalents to ASCII (American Standard Code for Information Interchange) characters. That is, if you were to scan the keyboard directly in BASIC, these are the characters that you would read; or, if you were to go into memory, these characters would appear as string-variable values.
In the APPLE keyboard column, read SMC as a SHIFT/CONTROL-M BC as CONTROL-B, etc. The decimal numbers listed are those before clearing of the keyboard strobe; after clearing of the strobe, the number is the listed value minus 128 (e.g., after clearing of the strobe, 145 becomes 17). If an APPLE keyboard equivalent doesn’t exist, or if there is no decimal equivalent, then the table shows a double dash (--); if the keyboard equivalent is identical to the ASCII character itself, then the table shows “sa.”

Such information can make your programming life much easier, because it lets you get keyboard data directly into a BASIC program without the use of an INPUT statement. As you know, INPUT statements can be limiting as, for example, when you type in a line and the screen yells SYNTAX ERROR at you. Well, what is the error? It may take a long time to find.

If, on the other hand, the keystrokes are picked directly off the keyboard then you — the programmer — are in command every step of the way; you make the decisions as to what goes and what does not, and in a way that tells you exactly what’s wrong as you go wrong.

You can see one way in which this idea is applied in “Being Precise in INTEGER,” in this issue’s HOW TO section. In this Multiple Precision Arithmetic listing, statements 2500-2610 input data directly from the keyboard; they tell you exactly what’s happening, the nature of any error you may have committed, etc. — and all by making use of the equivalence in the table presented here.

### TABLE OF ASCII CHARACTER VALUES

<table>
<thead>
<tr>
<th>ASCII CHAR</th>
<th>APPLE KYBD EQUIV</th>
<th>DEC CHAR</th>
<th>APPLE KYBD EQUIV</th>
<th>DEC CHAR</th>
<th>APPLE KYBD EQUIV</th>
<th>DEC CHAR</th>
<th>APPLE KYBD EQUIV</th>
<th>DEC CHAR</th>
<th>APPLE KYBD EQUIV</th>
<th>DEC CHAR</th>
</tr>
</thead>
<tbody>
<tr>
<td>NUL</td>
<td>SpC</td>
<td>128</td>
<td>SP</td>
<td>Space Bar</td>
<td>160</td>
<td>@</td>
<td>sa.</td>
<td>192</td>
<td>--</td>
<td>224</td>
</tr>
<tr>
<td>SOH</td>
<td>A^</td>
<td>129</td>
<td>!</td>
<td>!</td>
<td>161</td>
<td>A</td>
<td>193</td>
<td>a</td>
<td>sa. 225</td>
<td></td>
</tr>
<tr>
<td>STX</td>
<td>B^</td>
<td>130</td>
<td>#</td>
<td>#</td>
<td>163</td>
<td>B</td>
<td>194</td>
<td>b</td>
<td>226</td>
<td></td>
</tr>
<tr>
<td>ETX</td>
<td>C^</td>
<td>131</td>
<td>$</td>
<td>$</td>
<td>164</td>
<td>C</td>
<td>195</td>
<td>c</td>
<td>227</td>
<td></td>
</tr>
<tr>
<td>EOT</td>
<td>D^</td>
<td>132</td>
<td>%</td>
<td>%</td>
<td>165</td>
<td>E</td>
<td>197</td>
<td>e</td>
<td>228</td>
<td></td>
</tr>
<tr>
<td>ENQ</td>
<td>E^</td>
<td>133</td>
<td>&amp;</td>
<td>&amp;</td>
<td>166</td>
<td>F</td>
<td>198</td>
<td>f</td>
<td>229</td>
<td></td>
</tr>
<tr>
<td>ACK</td>
<td>F^</td>
<td>134</td>
<td>'</td>
<td>'</td>
<td>167</td>
<td>G</td>
<td>199</td>
<td>g</td>
<td>230</td>
<td></td>
</tr>
<tr>
<td>BEL</td>
<td>C^</td>
<td>135</td>
<td>(</td>
<td>(</td>
<td>168</td>
<td>H</td>
<td>200</td>
<td>h</td>
<td>231</td>
<td></td>
</tr>
<tr>
<td>BS</td>
<td>--</td>
<td>136</td>
<td>)</td>
<td>)</td>
<td>169</td>
<td>I</td>
<td>201</td>
<td>i</td>
<td>232</td>
<td></td>
</tr>
<tr>
<td>HT</td>
<td>J^</td>
<td>137</td>
<td>*</td>
<td>*</td>
<td>170</td>
<td>J</td>
<td>202</td>
<td>j</td>
<td>233</td>
<td></td>
</tr>
<tr>
<td>LF</td>
<td>K^</td>
<td>138</td>
<td>+</td>
<td>+</td>
<td>171</td>
<td>K</td>
<td>203</td>
<td>k</td>
<td>234</td>
<td></td>
</tr>
<tr>
<td>VT</td>
<td>L^</td>
<td>139</td>
<td>,</td>
<td>,</td>
<td>172</td>
<td>L</td>
<td>204</td>
<td>l</td>
<td>235</td>
<td></td>
</tr>
<tr>
<td>FF</td>
<td>M^</td>
<td>140</td>
<td>.</td>
<td>.</td>
<td>173</td>
<td>M</td>
<td>205</td>
<td>m</td>
<td>236</td>
<td></td>
</tr>
<tr>
<td>CR</td>
<td>N^</td>
<td>141</td>
<td>/</td>
<td>/</td>
<td>174</td>
<td>N</td>
<td>206</td>
<td>n</td>
<td>237</td>
<td></td>
</tr>
<tr>
<td>SO</td>
<td>O^</td>
<td>142</td>
<td>0</td>
<td>P</td>
<td>176</td>
<td>O</td>
<td>207</td>
<td>o</td>
<td>238</td>
<td></td>
</tr>
<tr>
<td>SI</td>
<td>P^</td>
<td>143</td>
<td>1</td>
<td>P</td>
<td>177</td>
<td>Q</td>
<td>208</td>
<td>p</td>
<td>239</td>
<td></td>
</tr>
<tr>
<td>DLE</td>
<td>Q^</td>
<td>144</td>
<td>2</td>
<td>R</td>
<td>178</td>
<td>Q</td>
<td>209</td>
<td>q</td>
<td>240</td>
<td></td>
</tr>
<tr>
<td>DC1</td>
<td>R^</td>
<td>145</td>
<td>3</td>
<td>S</td>
<td>179</td>
<td>R</td>
<td>210</td>
<td>r</td>
<td>241</td>
<td></td>
</tr>
<tr>
<td>DC2</td>
<td>S^</td>
<td>146</td>
<td>4</td>
<td>T</td>
<td>180</td>
<td>S</td>
<td>211</td>
<td>s</td>
<td>242</td>
<td></td>
</tr>
<tr>
<td>DC3</td>
<td>T^</td>
<td>147</td>
<td>5</td>
<td>U</td>
<td>181</td>
<td>T</td>
<td>212</td>
<td>t</td>
<td>243</td>
<td></td>
</tr>
<tr>
<td>DC4</td>
<td>U^</td>
<td>148</td>
<td>6</td>
<td>V</td>
<td>182</td>
<td>U</td>
<td>213</td>
<td>u</td>
<td>244</td>
<td></td>
</tr>
<tr>
<td>NAK</td>
<td>V^</td>
<td>149</td>
<td>7</td>
<td>W</td>
<td>183</td>
<td>V</td>
<td>214</td>
<td>v</td>
<td>245</td>
<td></td>
</tr>
<tr>
<td>SYN</td>
<td>W^</td>
<td>150</td>
<td>8</td>
<td>X</td>
<td>184</td>
<td>W</td>
<td>215</td>
<td>w</td>
<td>246</td>
<td></td>
</tr>
<tr>
<td>ETB</td>
<td>X^</td>
<td>151</td>
<td>9</td>
<td>Y</td>
<td>185</td>
<td>Y</td>
<td>216</td>
<td>x</td>
<td>247</td>
<td></td>
</tr>
<tr>
<td>CAN</td>
<td>Y^</td>
<td>152</td>
<td>10</td>
<td>Z</td>
<td>186</td>
<td>Z</td>
<td>217</td>
<td>y</td>
<td>248</td>
<td></td>
</tr>
<tr>
<td>EM</td>
<td>Z^</td>
<td>153</td>
<td>11</td>
<td>a</td>
<td>187</td>
<td>\</td>
<td>218</td>
<td>z</td>
<td>249</td>
<td></td>
</tr>
<tr>
<td>SUB</td>
<td>a</td>
<td>154</td>
<td>12</td>
<td>b</td>
<td>188</td>
<td>b</td>
<td>219</td>
<td>250</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ESC</td>
<td>--</td>
<td>155</td>
<td>13</td>
<td>c</td>
<td>189</td>
<td>c</td>
<td>220</td>
<td>251</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FS</td>
<td>--</td>
<td>156</td>
<td>14</td>
<td>d</td>
<td>190</td>
<td>m</td>
<td>221</td>
<td>252</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GS</td>
<td>S^</td>
<td>157</td>
<td>15</td>
<td>n</td>
<td>191</td>
<td>S^</td>
<td>222</td>
<td>253</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RS</td>
<td>S^</td>
<td>158</td>
<td>16</td>
<td>o</td>
<td>192</td>
<td>\</td>
<td>223</td>
<td>254</td>
<td></td>
<td></td>
</tr>
<tr>
<td>US</td>
<td>--</td>
<td>159</td>
<td>17</td>
<td>p</td>
<td>193</td>
<td>--</td>
<td>224</td>
<td>255</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

LF = Line Feed; CR = Carriage Return; SP = Space; ESC = Escape; sa. = Keyboard character same as ASCII character.
If both pages contain similar graphics figures, then switching between the pages will yield simple animation; further effects may be gleaned from an inspection of the list of POKEs on page 30 of the APPLE II Reference Manual.

(NOTE: Don’t try this with APPLESOFT in RAM. It starts at hex 800 – the second page of graphic space. A block move into that area will send your APPLESOFT BASIC into the bit bucket!)

The name of the game
is the saving of the name

If you’re an APPLESOFT II user working with, say, an inventory list with names, then you’re in a bit of trouble if you want to SAVE the complete list to cassette tape, names and all: APPLESOFT II will save the numbers but not the names (strings). (Of course, the nicest way to SAVE such a list is to disk.)

If you need to save strings to tape however, the following program will do the job very nicely. Note that statement 10 creates space for the strings; 1010 gives you information about free memory space and how far up the variables are; 1050 writes out the length of the tape’s string area; and 1070 writes all the desired information to tape.

**Syntax Error**

**LIST**

1 REM
2 REM PROGRAM TO SAVE STRINGS
3 REM TO CASSETTE TAPE.
4 REM BY R. WIGGINTON (6/78)
5 REM
10 DIM A$(10)
20 PRINT “TYPE IN NINE STRINGS, SEPARATED BY”: PRINT “CARRIAGE RETURNS.”
30 FOR K = 1 TO 9: INPUT A$(K): NEXT K
40 REM NOW SAVE AS TO TAPE.
50 GOSUB 1000
55 PRINT “STRINGS ARE NOW ON TAPE. TO RECALL, TYPE ‘GOTO 100’, REMIND AND START TAPE, AND PRESS ‘RTN.’.”
57 PRINT “LET TAPE RUN UNTIL CURSOR RETURNS.”
60 END
100 REM THIS PART RECALLS THE
101 REM STRINGS FROM TAPE.
102 REM
110 DIM B$(10)
120 GOSUB 2000
130 FOR K = 1 TO 9: PRINT B$(K): NEXT K
140 END
1000 REM STORE A$ TO TAPE.
1003 PRINT “INSERT CLEAR TAPE, START RECORDING.”
1005 PRINT “HIT ANY KEY WHEN READY”: GET Z
1010 X = Z: FRE (0): STORE A$: REM STORE A$ REALLY STORES POINTERS
1020 REM IN ORDER FOR THIS PROGRAM TO WORK, HUMB MUST BE AT THE SAME
1021 REM VALUE WHEN THE STRINGS ARE RECALLED AS WHEN THEY ARE STORED.
1030 X = PEEK (115) + PEEK (116) * 256 + PEEK (111) - PEEK (112) * 256
1040 GOSUB 2100
1050 PEEK 30, X - INT (X / 256) * 256: PEEK 31, X / 256: CALL -307: REM X = X INTO LOGS 30531, AND WRITE IT TO TAPE.
1050 REM (X) IS THE LENGTH OF THE STRING AREA.
1030 REM HAVE NOW WRITTEN EVERYTHING.
1090 PRINT “D.K.”: RETURN
2000 RECALL B$: REM OBT POINTERS BACK.
2010 GOSUB 2100: CALL -259: REM OBT LENGTH OF STRING AREA
2020 X = PEEK (30) + PEEK (31) * 256: REM X IS LENGTH OF AREA TO READ IN
2030 X = PEEK (115) + PEEK (116) * 256 - X
2040 PEEK 60, X - INT (X / 256) * 256: PEEK 61, X / 256
2050 PEEK 62, PEEK (115): PEEK 63, PEEK (116): CALL -259
2060 RETURN
2100 PEEK 60,30: PEEK 61,0: PEEK 62,31: PEEK 63,0: RETURN: REM SET CASS
ETE ROUTINE POINTERS.

**POKE**

1
Lining things up, point by point

Since most BASICs justify (i.e., line up) to the left-most column, a display of multi-digit, decimal-pointed numbers can be awkward to read and somewhat unattractive.

A solution to this problem would be to use a tabulation routine that "justifies (or lines up) on the decimal point." Such a routine would position the numbers in a column so that the decimal points are vertically aligned. The short program listed below does exactly that.

Statements 10 through 50 are merely a demonstration routine that yields the sample run shown at the end of the listing. Statements 2000 through 2140 contain the routine that actually does the work. In effect, the routine aligns the numbers by right-justifying to the digit left of the decimal point, then tacks on the decimal point and the remaining digits to the right of the decimal point.

```
10 F=10: B=9: D=9
20 A=F
30 IF A#0 THEN 1000
40 F=1000:F=F
50 GOTO 15
1000 E=0
2000 REM RIGHT-JUSTIFICATION ROUTINE
2100 REM INPUT IS ASSIGNED TO BE IN VARIABLE "A"
2020 REM THE RIGHT-HAND CHARACTER WILL APPEAR IN COLUMN CONTAINED IN VARIABLE "F"
2110 A(1)=*: IF A# THEN A(1)=: REM SET SIGN OF NUMBER
2120 C=(A(1)+1)2000: REM DETERMINE HOW FAR TO LEFT-SHIFT PRINTOUT
2130 TAB (B+1): PRINT A(1): A(1)=: REM RIGHT-JUSTIFY PRINTOUT
2140 RETURN

RUN 10.9
  800, 789
  3456789

Being precise in INTEGER

The use of INTEGER BASIC limits you to the range of numbers between -32767 and +32767. Such a limitation is, at its best, frustrating, and, at its worst, infuriating. Consider, for example, the businessman who daily deals with foreign currencies, for which the basic monetary unit may be very, very small. What's a fella' to do?

Well, what he has to do is to go to multiple-precision arithmetic by means of a routine such as we present here. While this example is for addition only, it is readily adaptable to subtraction, multiplication, and division by changes in statements 3000 through 3080. The program does its job on large numbers in the same way as we do it by long-hand arithmetic; that is, it operates on one digit at a time, then carries to the next, and so on.

This particular listing is long and slow, because we wanted to make it clear and easy to read so that you could see what's happening. You may modify it to run much faster.

Incidentally, you can get a better understanding of the program's operation by relating certain of its statements to the ASCII conversion table that is in this issue's OUT OF THE MIST section. Statements 2500 through 2520, for example, result in the keyboard being read directly. 2540 refers to CHAR = 141; reference to the table tells you that decimal 141 is actually the Carriage Return. Similarly, 2545 excludes all characters except for the digits 0 through 9 (176 through 185). Again, statement 2550 converts the ASCII characters to the numbers themselves (i.e., if CHAR = 181, then 181 - 176 = 5).
**outside the orchard**

...personal software

The G-2 line of personal-computer software is a series of program packages, with each package consisting of a cassette and full documentation. The first programs offered are "Beat the House", a Las Vegas-like game set that includes Blackjack, Craps, Slot Machines, and Roulette; "Dollars and Sense", a personal finance program; and "Clinic", a medical program that offers information on longevity, biorhythms, and dieting. Standard and Extended BASIC packages are also available. GRT Corporation, 1286 Lawrence Station Rd., Sunnyvale, CA 94086. (408) 734-2910.

**Clock interface**

Designed specifically for APPLE II, this real-time clock interface features on/off control via software; a time display anywhere on the screen as desired; time setting and testing from BASIC or machine language; relocatable software; and a slot-independent plug-in PC board. $32.95. System Design Engineering, Suite 40, 2460 W. 239th St., Torrance, CA 90505. (213) 539-2194.

**6502 software**

An organization for the exchange of tested software for 6502-based systems has been set up. The company does not buy copyrights, but pays royalties for and sells the software. Because APPLE is a 6502-based system, the programs may be of interest to APPLE users. For information, write to The 6502 Program Exchange, 2920 Moana, Reno, NV 89509.

**Business programs**

Here is a group of programs oriented to the small businessman and independent professional. The programs offer mailing list and secretarial services, time and data management, tax planning, daily scheduling, and even an exercise program. Each program runs in 16K or less of memory, and includes documentation and a security function that allows identification of unauthorized copies of the software tape. Retail prices range from $19.95 (the exercise program) to $89.95 (the tax planning and professional secretary packages). Charles Mann & Associates, 1926 South Veteran Ave., Los Angeles, CA 90025. (213) 473-0244.