


```

0000 1: ;-----
0000 2: ;      Copyright 2010 Integrated Logic Systems
0000 3: ;      Source Code is Copyright Stephen J. Car
0000 4: ;
0000 5: ;
0000 6: ;      Please do not share this source!
0000 7: ;-----
0000 8: ;      Version.com
0000 9: ;
0000 10: ; This is a program to peak a byte and display it to
0000 11: ;      init          2/24/2010
0000 12: ;      added hex delimiter      2/25/2010
0000 13: ;      added new code for registered user 2/26/201
0000 14: ;      set up a lookup table to handle language pr
0000 15: ;      adding more Languages      4/15/2010
0000 16: ;      bug fix.. Looking in the wrong spot of name
0000 17: ;
0000 18: ;      Updated the screen Print Utility so it will work wi
0000 19: ; This change will also help "Print P:"      6/1
0000 20: ;
0000 21: ;Also added code for Control+1 pause of screen to the high
0000 22: ;handling.      8/24/2012      sjc
0000 23: ;
0000 24: ;
0000 25: ;-----
0000 26: ; Notes: o This source code MAY NOT be placed for download
0000 27: ;      o "mea" is a macro that loads the address of the
0000 28: ;      into the pointer specified by the second field.
0000 29: ;-----
0000 30: ; Assembler: MADMAC (tm) ST Cross Assembler (Atari Corp)
0000 31: ;      XASM  st and IBM VERSIONS
0000 32: ;-----
0000 33:
0000 34: com_file_name:      .macro
0000 35: ;      dc.b  12,"filename.ext",$9b
0000 36: ;      dc.b  12,"Version.com ", $9b
0000 37: ;      .endm
0000 38:
0000 39: ;
0000 40: ;~~~~~
0000 41: ; File revision history. Version Number in hex
0000 42: ;
0000 43: file_ver:      equ  $19
0000 44: ;~~~~~
0000 45: ; the Month in dec for the first time this revision
0000 46: ;was compiled
0000 47: ;
0000 48: c_month:      equ  8
0000 49: ;~~~~~
0000 50: ; the day in dec for the first time this revision
0000 51: ;was compiled
0000 52: c_day:      equ  27
0000 53: ;~~~~~

```

```
0000 54: ; the year in dec for the first time this revision
0000 55: ;was compiled
0000 56: c_year:          equ 2012
0000 57: ;~~~~~
0000 58: ; Type of compiler used in program
0000 59: ;
0000 60: ; 0 = unknown Compiler
0000 61: ; 1 = xasm
0000 62: ; 2 = mac_65
0000 63: ; 3 = basic
0000 64: ; 4 = compiled basic xl
0000 65: ; 5 = C65
0000 66: ; 6 = Action
0000 67: ;
0000 68: ; We can add more as time goes on!
0000 69: ;
0000 70: ;
0000 71: ;
0000 72: xasm:             equ 1
0000 73: mac_65:           equ 2
0000 74: basic:            equ 3
0000 75: basicxl:          equ 4
0000 76: c65:              equ 5
0000 77: action:           equ 6
0000 78:
0000 79: file_compiler:     equ xasm
0000 80: ;~~~~~
0000 81: ; is this relocatable code ?
0000 82: ; anyother value other than 1 or 2 would be unknown
0000 83: r..yes:            equ 1
0000 84: r..no:             equ 2
0000 85: ;
0000 86: ;~~~~~
0000 87: ; Gotta define if it can be relocated
0000 88: l_relocatable:     equ r..no
0000 89: ;
0000 90: ;
0000 91: ;~~~~~
0000 92: ; Gotta define if it can be relocated
0000 93: r..crl:            equ $7d
0000 94: r..crlf:           equ $9b
0000 95: r..space:          equ $20
0000 96:
0000 97: l_frstscreenbyte: equ r..space
0000 98: ;~~~~~
0000 99: ; The language the output file is in.
0000 100: ;
0000 101: ;
0000 102: ; 0 = Undefined
0000 103: ; 1 = English
0000 104: ; 2 = German
0000 105: ; 3 = Hungarian
0000 106: ; 4 = Welsh
```

```

0000    107: ; 5 = Dutch
0000    108: ; 6 = Polish
0000    109: ; 7 = Swedish
0000    110: ; 8 = French
0000    111: ; 9 = not defined yet
0000    112: ; 10 = not defined yet
0000    113: ;
0000    114: ;
0000    115: ;
0000    116: ;
0000    117: r..language: equ 1
0000    118: ;~~~~~
0000    119:
0000    120:
0000    121:         .include equates
0000    122:         .include globals
0000    123:         .include macros
0000    124:
4000    125:         .org $4000
4000    126:
4000    127: win_start:
4000    128: header_info:
4000    129:         .include header ; t
42a7    130:
42a7 a210 131:         ldx #disk1
42a9 20624d 132:         jsr close
42ac a220 133:         ldx #disk2
42ae 20624d 134:         jsr close
42b1    135:
42b1 a50a 136:         lda comtab
42b3 18 137:         CLC
42b4 6903 138:         adc # low ZCRNAME
42b6 8dc343 139:         sta cli.crunch+1
42b9 a50b 140:         lda comtab+1
42bb 6900 141:         adc # high ZCRNAME
42bd 8dc443 142:         sta cli.crunch+2
42c0 20c243 143:         jsr cli.crunch
42c3 d003 144:         bne .get_parm1 ; n
42c5 4cc543 145:         jmp parm_errors
42c8    146:
42c8    147: .get_parm1:
42c8 a50a 148:         lda comtab
42ca 18 149:         CLC
42cb 6921 150:         adc # low COMFNAM
42cd 8582 151:         sta s_ptr
42cf a50b 152:         lda comtab+1
42d1 6900 153:         adc # high COMFNAM
42d3 8583 154:         sta s_ptr+1
42d5    155:         mea parm_1,d_ptr
42d5 a93f ----:         lda #low parm_1
42d7 8584 ----:         sta d_ptr
42d9 a945 ----:         lda #high parm_1
42db 8585 ----:         sta d_ptr + 1

```

```

42dd 200455 156:          jsr  tostr
42e0          157:
42e0          158:
42e0 20c243 159:          jsr  cli.crunch
42e3 d003   160:          bne  .get_parm2
42e5 4c3a43 161:          jmp  .exit
42e8          162:
42e8 a50a   163: .get_parm2:      lda   comtab
42ea 18     164:          CLC
42eb 6921   165:          adc   # low COMFNAM
42ed 8582   166:          sta   s_ptr
42ef a50b   167:          lda   comtab+1
42f1 6900   168:          adc   # high COMFNAM
42f3 8583   169:          sta   s_ptr+1
42f5          170:          mea   parm_2,d_ptr
42f5 a98f   ----:      lda   #low parm_2
42f7 8584   ----:      sta   d_ptr
42f9 a945   ----:      lda   #high parm_2
42fb 8585   ----:      sta   d_ptr + 1
42fd 200455 171:          jsr  tostr
4300          172:
4300          173:
4300 20c243 174:          jsr  cli.crunch
4303 d003   175:          bne  .get_parm3
4305 4c3a43 176:          jmp  .exit
4308          177:
4308 a50a   178: .get_parm3:      lda   comtab
430a 18     179:          CLC
430b 6921   180:          adc   # low COMFNAM
430d 8582   181:          sta   s_ptr
430f a50b   182:          lda   comtab+1
4311 6900   183:          adc   # high COMFNAM
4313 8583   184:          sta   s_ptr+1
4315          185:          mea   parm_3,d_ptr
4315 a9df   ----:      lda   #low parm_3
4317 8584   ----:      sta   d_ptr
4319 a945   ----:      lda   #high parm_3
431b 8585   ----:      sta   d_ptr + 1
431d 200455 186:          jsr  tostr
4320 addf45 187:          lda   parm_3      ; let us se
4323 c906   188:          cmp   #6          ; is it lon
4325 9013   189:          bcc   .exit       ; nope we w
4327          190:
4327          191:          mea   parm_3,s_ptr
4327 a9df   ----:      lda   #low parm_3
4329 8582   ----:      sta   s_ptr
432b a945   ----:      lda   #high parm_3
432d 8583   ----:      sta   s_ptr + 1
432f          192:          mea   filename,d_ptr
432f a9de   ----:      lda   #low filename
4331 8584   ----:      sta   d_ptr
4333 a944   ----:      lda   #high filename
4335 8585   ----:      sta   d_ptr + 1

```

```

4337 204455 193:          jsr  toascii
433a      194:
433a      195: .exit:
433a ad1fd0 196:          lda  consol
433d c907  197:          cmp  #consol_any
433f f07e  198:          beq  .no_tell
4341      199:
4341 209151 200:          jsr  printsi
4344 9b2074 201:          dc.b  $9b," this is my cli to par
436f 9b5061 202:          dc.b  $9b,"Parm_1 =",-1
4379      203:          mea  parm_1,pr_ptr
4379 a93f  ----:      lda  #low parm_1
437b 8588  ----:      sta  pr_ptr
437d a945  ----:      lda  #high parm_1
437f 8589  ----:      sta  pr_ptr + 1
4381 209453 204:          jsr  echos
4384      205:
4384 209151 206:          jsr  printsi
4387 9b5061 207:          dc.b  $9b,"Parm_2 =",-1
4391      208:          mea  parm_2,pr_ptr
4391 a98f  ----:      lda  #low parm_2
4393 8588  ----:      sta  pr_ptr
4395 a945  ----:      lda  #high parm_2
4397 8589  ----:      sta  pr_ptr + 1
4399 209453 209:          jsr  echos
439c      210:
439c 209151 211:          jsr  printsi
439f 9b4f75 212:          dc.b  $9b,"Output File =",-1
43ae      213:          mea  parm_3,pr_ptr
43ae a9df  ----:      lda  #low parm_3
43b0 8588  ----:      sta  pr_ptr
43b2 a945  ----:      lda  #high parm_3
43b4 8589  ----:      sta  pr_ptr + 1
43b6 209453 214:          jsr  echos
43b9      215:
43b9 209151 216:          jsr  printsi
43bc 9b9bff 217:          dc.b  $9b,$9b,-1
43bf      218:
43bf      219: .no_tell:
43bf 4c8747 220:          jmp  VERSION
43c2      221:
43c2 4c0000 222: cli.crunch:      jmp  $0000
43c5      223:
43c5      224: parm_errors:
43c5      225:
43c5 209151 226:          jsr  printsi
43c8 9b9b  227:          dc.b  $9B,$9B
43ca 205379 228:          dc.b  " Syntax for use:",$9b
43db 205665 229:          dc.b  " Version Dn:>dir>fname.ext
4406 205061 230:          dc.b  " Parm1 = /P or /H",$9b
4418 206966 231:          dc.b  " if Parm1 = /P output to P
4439 206966 232:          dc.b  " if Parm1 = /H Hardcopy",$
4451 202f48 233:          dc.b  " /H must have Parm2",$9b

```

```

4465 205061 234:          dc.b  " Parm2 = dx:>path>filename
4484 206170 235:          dc.b  " append text to!", $9b
4495      236:
4495 9bff 237:          dc.b  $9b,-1
4497 60 238:          RTS
4498      239:
4498      240: ;-----
4498      241: ;
4498 1a2048 242: INVALID:      dc.b  26," Has an invalid Com Hea
44b3      243: ;
44b3 262d2d 244: SPLIT:      dc.b  38,"-----
44da      245: ;
44da      246: ;
44da 503a9b 247: PRN:          dc.b  "P:", $9B,0
44de      248: ;
44de 44313a 249: FILENAME:      dc.b  "D1:>version.lst", $9B,0
44ef      250:          ds.b  80
453f      251: ;
453f      252: parm_1:          ds.b  80
458f      253: parm_2:          ds.b  80
45df      254: parm_3:          ds.b  80
462f      255: in_str:          ds.b  128
46af      256: instr:          ds.b  128      ;
472f      257:
472f      258: ;
472f      259: ;
472f 00 260: WHERE:          dc.b  0
4730      261:
4730 42 262: lng.LOW:          dc.b  lng.Undefined &$ff
4731 4c 263:          dc.b  lng.English &$ff
4732 54 264:          dc.b  lng.German &$ff
4733 5b 265:          dc.b  lng.Hungarian &$ff
4734 65 266:          dc.b  lng.Welsh &$ff
4735 6b 267:          dc.b  lng.Dutch &$ff
4736 71 268:          dc.b  lng.Polish &$ff
4737 78 269:          dc.b  lng.Swedish &$ff
4738 80 270:          dc.b  lng.French &$ff
4739      271:
4739 47 272: lng.HIGH:          dc.b  lng.Undefined /256
473a 47 273:          dc.b  lng.English /256
473b 47 274:          dc.b  lng.German /256
473c 47 275:          dc.b  lng.Hungarian /256
473d 47 276:          dc.b  lng.Welsh /256
473e 47 277:          dc.b  lng.Dutch /256
473f 47 278:          dc.b  lng.Polish /256
4740 47 279:          dc.b  lng.Swedish /256
4741 47 280:          dc.b  lng.French /256
4742      281: ;
4742      282: ;<<<< Note >>>> These are length byte strings that are hig
4742      283: ;
4742      284: ;
4742 09556e 285: lng.Undefined:      dc.b  9,"Undefined"
474c      286:

```

```

474c 07456e 287: lng.English:      dc.b  7,"English"
4754      288:
4754 064765 289: lng.German:      dc.b  6,"German"
475b      290:
475b 094875 291: lng.Hungarian:    dc.b  9,"Hungarian"
4765      292:
4765 055765 293: lng.Welsh:      dc.b  5,"Welsh"
476b      294:
476b 054475 295: lng.Dutch:      dc.b  5,"Dutch"
4771      296:
4771 05506f 297: lng.Polish:     dc.b  5,"Polish"
4778      298:
4778 075377 299: lng.Swedish:    dc.b  7,"Swedish"
4780      300:
4780 064672 301: lng.french:     dc.b  6,"French"
4787      302:
4787      303:
4787      304:
4787      305: ;
4787      306: VERSION:
4787      307:
4787 ad3f45 308:          lda  PARM_1
478a c904  309:          cmp  #$04
478c b02e  310:          bcs  OK
478e 209151 311:          jsr  printsi
4791 9b9b20 312:          dc.b  $9B,$9B," Usage: Version D1
47bb 60    313:          RTS
47bc      314: OK:
47bc ad9345 315:          lda  PARM_2+4
47bf c92f  316:          cmp  #'/'
47c1 d013  317:          bne  OKP
47c3 ad9445 318:          lda  PARM_2+5
47c6 20e854 319:          jsr  UPCASE
47c9 c950  320:          cmp  #'P'
47cb f006  321:          beq  OKO
47cd c948  322:          cmp  #'H'
47cf d005  323:          bne  OKP
47d1      324: ;
47d1      325: ;
47d1 a901  326: OKM:      lda  #$01
47d3 8d2f47 327: OKO:      sta  WHERE
47d6      328: ;
47d6      329: OKP:
47d6      330:
47d6      331:          mea  PARM_1,S_PTR
47d6 a93f  ----:          lda  #low PARM_1
47d8 8582  ----:          sta  S_PTR
47da a945  ----:          lda  #high PARM_1
47dc 8583  ----:          sta  S_PTR + 1
47de      332:          mea  BROWSE_FILE,D_PTR
47de a96e  ----:          lda  #low BROWSE_FILE
47e0 8584  ----:          sta  D_PTR
47e2 a94b  ----:          lda  #high BROWSE_FILE

```



```

47e4 8585 ----:      sta  D_PTR + 1
47e6 204455 333:      jsr  TOASCII
47e9 a220 334:      ldx  #disk2
47eb 20624d 335:      jsr  CLOSE
47ee      336:      mea  BROWSE_FILE,FN_PTR
47ee a96e ----:      lda  #low BROWSE_FILE
47f0 8586 ----:      sta  FN_PTR
47f2 a94b ----:      lda  #high BROWSE_FILE
47f4 8587 ----:      sta  FN_PTR + 1
47f6 a220 337:      ldx  #disk2
47f8 20174d 338:      jsr  OPEN_IN
47fb 3004 339:      bmi  VERRY2
47fd d005 340:      bne  OKOPEN
47ff f003 341:      beq  OKOPEN
4801 4c514b 342: VERRY2:      jmp  FILE_NOT_FOUND
4804      343: OKOPEN:
4804 a900 344:      lda  #$00
4806 8de14c 345:      sta  fil.sector
4809 8de24c 346:      sta  fil.sector+1
480c a900 347:      lda  #$00
480e 8de34c 348:      sta  OFFSET
4811 a220 349:      ldx  #disk2
4813 20c04c 350:      jsr  POINT
4816      351:      mea  BUFFER2,PR_PTR
4816 a9be ----:      lda  #low BUFFER2
4818 8588 ----:      sta  PR_PTR
481a a94b ----:      lda  #high BUFFER2
481c 8589 ----:      sta  PR_PTR + 1
481e ad624c 352:      lda  BUF_2
4821 8dec4c 353:      sta  LNPTR
4824 ad634c 354:      lda  BUF_2+1
4827 8ded4c 355:      sta  LNPTR+1
482a a220 356:      ldx  #disk2
482c 20ee4c 357:      jsr  BGET
482f a220 358:      ldx  #disk2
4831 20624d 359:      jsr  CLOSE
4834      360: ;
4834 adbe4b 361:      lda  BUFFER2
4837 c9ff 362:      cmp  #$FF
4839 d031 363:      bne  v_header.error
483b adbf4b 364:      lda  BUFFER2+1
483e c9ff 365:      cmp  #$FF
4840 d02a 366:      bne  v_header.error
4842 adc44b 367:      lda  BUFFER3
4845 c94c 368:      cmp  #$4C
4847 d023 369:      bne  v_header.error
4849 ad9345 370:      lda  PARM_2+4 ;0,dx:/P
484c c92f 371:      cmp  #'/'
484e d012 372:      bne  WERRY1
4850 ad9445 373:      lda  PARM_2+5
4853 20e854 374:      jsr  UPCASE
4856 c950 375:      cmp  #'P'
4858 f00b 376:      beq  ZERRY1

```

```

485a c948 377:          cmp  #'H'
485c f007 378:          beq  ZERRY1
485e c943 379:          cmp  #'C'
4860 f006 380:          beq  watcom
4862      381: ;
4862 4c8b48 382: WERRY1:          jmp  VERRY1
4865      383: ;
4865 4c9750 384: ZERRY1:          jmp  VER_PRINT
4868      385: ;
4868      386: watcom:
4868 60    387:          rts
4869      388: ;
4869 4c4450 389: PERRY1:          jmp  P_ERROR
486c      390: ;
486c      391: v_header.error:
486c ad2f47 392:          lda  WHERE
486f c900 393:          cmp  #$00
4871 f0f6 394:          beq  PERRY1
4873      395: ;
4873 208e53 396:          jsr  cr_lf
4876 208e53 397:          jsr  cr_lf
4879      398:          mea  INVALID,PR_PTR
4879 a998 ----:          lda  #low INVALID
487b 8588 ----:          sta  PR_PTR
487d a944 ----:          lda  #high INVALID
487f 8589 ----:          sta  PR_PTR + 1
4881 209453 399:          jsr  ECHOS
4884 208e53 400:          jsr  cr_lf
4887 208e53 401:          jsr  cr_lf
488a 60    402:          RTS
488b      403: VERRY1:
488b ade94b 404:          lda  irf.Com_name
488e c90c 405:          cmp  #12
4890 f034 406:          beq  file_ok
4892 209151 407:          jsr  printsi
4895 9b9b 408:          dc.b  $9b,$9b
4897 496e76 409:          dc.b  "Invalid Header Format Vers
48c3 9b9bff 410:          dc.b  $9b,$9b,-1
48c6      411:
48c6      412: file_ok:
48c6      413:
48c6 209151 414:          jsr  printsi
48c9 9b    415:          dc.b  $9b
48ca 496e74 416:          dc.b  "Internal Name "
48d8 ff    417:          dc.b  -1
48d9      418:          mea  irf.Com_name,pr_ptr
48d9 a9e9 ----:          lda  #low irf.Com_name
48db 8588 ----:          sta  pr_ptr
48dd a94b ----:          lda  #high irf.Com_name
48df 8589 ----:          sta  pr_ptr + 1
48e1 202154 419:          jsr  echos.nospace
48e4      420:
48e4 209151 421:          jsr  printsi

```

```

48e7 205365 422:      dc.b  " Serial #"
48f0 ff 423:      dc.b  -1
48f1 ade74b 424:      lda  irf.serialnum
48f4 aee84b 425:      ldx  irf.serialnum+1
48f7 205653 426:      jsr  pr_card
48fa 208e53 427:      jsr  cr_lf
48fd 428:
48fd 429:
48fd addd4b 430: .nono:      lda  irf.rcopy
4900 c900 431:      cmp  #$00
4902 d008 432:      bne  .reg
4904 209151 433:      jsr  printsi
4907 ceef4 434:      dc.b  <+128>,"Not"
490a 20 435:      dc.b  " "
490b ff 436:      dc.b  -1
490c 437: .reg:
490c 209151 438:      jsr  printsi
490f 526567 439:      dc.b  "Registered",-1
491a 440:
491a addd4b 441:      lda  irf.rcopy
491d c900 442:      cmp  #$00
491f f03a 443:      beq  .go_auth
4921 c981 444:      cmp  #$81
4923 f010 445:      beq  .go_site
4925 c901 446:      cmp  #$01
4927 d032 447:      bne  .go_auth
4929 209151 448:      jsr  printsi
492c 20746f 449:      dc.b  " to :",-1
4932 4c4049 450:      jmp  .nom
4935 451:
4935 209151 452: .go_site:      jsr  printsi
4938 205369 453:      dc.b  " Site :"
```

```

493f ff 454:      dc.b  -1
4940 455: .nom:
4940 456:      mea  irif.register_name,pr_ptr
4940 a916 ----:      lda  #low irif.register_name
4942 8588 ----:      sta  pr_ptr
4944 a94c ----:      lda  #high irif.register_name
4946 8589 ----:      sta  pr_ptr + 1
4948 20da53 457:      jsr  echos.unify
494b 458:      mea  irif.register_name,pr_ptr
494b a916 ----:      lda  #low irif.register_name
494d 8588 ----:      sta  pr_ptr
494f a94c ----:      lda  #high irif.register_name
4951 8589 ----:      sta  pr_ptr + 1
4953 20b853 459:      jsr  echosinv
4956 460:
4956 209151 461:      jsr  printsi
4959 9bff 462:      dc.b  $9b,-1
495b 463:
495b 464: .go_auth:
495b 465:
495b 209151 466:      jsr  printsi
```

```

495e 9b4175 467:          dc.b  $9b,"Author :",-1
4969          468:
4969          469:          mea  irif.Authors_name,pr_ptr
4969 a9f7  ----:          lda  #low irif.Authors_name
496b 8588  ----:          sta  pr_ptr
496d a94b  ----:          lda  #high irif.Authors_name
496f 8589  ----:          sta  pr_ptr + 1
4971          470:
4971 a000  471:          ldy  #$00
4973 b188  472:          lda  (pr_ptr),y
4975 c900  473:          cmp  #$00
4977 f003  474:          beq  .no_aut
4979          475:
4979 20b853 476:          jsr  echosinv
497c          477:
497c          478:
497c          479:
497c 209151 480: .no_aut:      jsr  printsi
497f 9bff  481:          dc.b  $9b,-1
4981          482:
4981 209151 483:          jsr  printsi
4984 436f6d 484:          dc.b  "Company :",-1
498e          485:
498e          486:          mea  irif.company_id,pr_ptr
498e a935  ----:          lda  #low irif.company_id
4990 8588  ----:          sta  pr_ptr
4992 a94c  ----:          lda  #high irif.company_id
4994 8589  ----:          sta  pr_ptr + 1
4996          487:
4996 a000  488:          ldy  #$00
4998 b188  489:          lda  (pr_ptr),y
499a c900  490:          cmp  #$00
499c f003  491:          beq  .no_co
499e          492:
499e 20b853 493:          jsr  echosinv
49a1          494:
49a1          495: .no_co:
49a1 209151 496:          jsr  printsi
49a4 9bff  497:          dc.b  $9b,-1
49a6          498:
49a6 209151 499:          jsr  printsi
49a9 436f6d 500:          dc.b  "Compiler:",-1
49b3          501:
49b3 add74b 502:          lda  irf.c_compiler
49b6 c900  503:          cmp  #$00
49b8 d00f  504:          bne  .a
49ba 209151 505:          jsr  printsi
49bd d5eeeb 506:          dc.b  <+128>,"Unknown"
49c4 9bff  507:          dc.b  $9b,-1
49c6 4c094a 508:          jmp  .exit
49c9 c901  509: .a:          cmp  #$01
49cb d00c  510:          bne  .b
49cd 209151 511:          jsr  printsi

```

```

49d0 f8e1f3 512:      dc.b  <+128>,"xasm"
49d4 9bff 513:      dc.b  $9b,-1
49d6 4c094a 514:      jmp   .exit
49d9      515: .b:
49d9 c902 516:      cmp   #$02
49db d00e 517:      bne   .c
49dd 209151 518:      jsr   printsi
49e0 cde1e3 519:      dc.b  <+128>,"Mac/65"
49e6 9bff 520:      dc.b  $9b,-1
49e8 4c094a 521:      jmp   .exit
49eb      522:
49eb      523: .c:
49eb c905 524:      cmp   #$05
49ed d00b 525:      bne   .d
49ef 209151 526:      jsr   printsi
49f2 e3b6b5 527:      dc.b  <+128>,"c65"
49f5 9bff 528:      dc.b  $9b,-1
49f7 4c094a 529:      jmp   .exit
49fa      530:
49fa      531: .d:
49fa c906 532:      cmp   #$06
49fc d00b 533:      bne   .exit
49fe 209151 534:      jsr   printsi
4a01 c1e3f4 535:      dc.b  <+128>,"Action"
4a07 9bff 536:      dc.b  $9b,-1
4a09      537:
4a09      538: .exit:
4a09      539: .reloc:
4a09 add84b 540:      lda   irf.relocation
4a0c c901 541:      cmp   #$01
4a0e d01c 542:      bne   .a
4a10 209151 543:      jsr   printsi
4a13 d2e5ec 544:      dc.b  <+128>,"Relocation within C
4a2a 9bff 545:      dc.b  $9b,-1
4a2c      546: .a:
4a2c 209151 547:      jsr   printsi
4a2f 526561 548:      dc.b  "RealDos Support Build",-1
4a46 addb4b 549:      lda   irf.dos_build
4a49 aedc4b 550:      ldx   irf.dos_build+1
4a4c 205653 551:      jsr   pr_card
4a4f      552:
4a4f 209151 553:      jsr   printsi
4a52 9b4c61 554:      dc.b  $9b,"Language:",-1
4a5d      555:
4a5d      556: ;=====
4a5d      557: ; this reflects how many languages we have
4a5d      558: ;=====
4a5d      559:
4a5d add94b 560:      lda   irf.language
4a60 c909 561:      cmp   #lng.HIGH-lng.LOW &$ff
4a62 9020 562:      bcc   .eng
4a64 209151 563:      jsr   printsi
4a67 cce1ee 564:      dc.b  <+128>,"Language Not Set in

```

```

4a7f 9bff 565:          dc.b  $9b,-1
4a81 4c974a 566:          jmp   .dono
4a84      567:
4a84 acd94b 568: .eng:      ldy   irf.language
4a87 b93047 569:          lda   lng.LOW,y
4a8a 8588 570:          sta   pr_ptr
4a8c b93947 571:          lda   lng.HIGH,y
4a8f 8589 572:          sta   pr_ptr+1
4a91 20b853 573:          jsr   echosinv
4a94 208e53 574:          jsr   cr_lf
4a97      575:
4a97      576: .dono:
4a97 209151 577:          jsr   printsi
4a9a 9b 578:          dc.b  $9b
4a9b 4f7267 579:          dc.b  "Org's at "
4aa4 a4 580:          dc.b  <+128>,"$"
4aa5 ff 581:          dc.b  -1
4aa6 adcf4b 582:          lda   irf.c_start
4aa9 aed04b 583:          ldx   irf.c_start+1
4aac 202a53 584:          jsr   pr_hex_card
4aaf 209151 585:          jsr   printsi
4ab2 20456e 586:          dc.b  " End at "
4aba a4 587:          dc.b  <+128>,"$"
4abb ff 588:          dc.b  -1
4abc      589:
4abc add14b 590:          lda   irf.eof
4abf aed24b 591:          ldx   irf.eof+1
4ac2 202a53 592:          jsr   pr_hex_card
4ac5      593:
4ac5 adcc4b 594:          lda   irf.c_crc+1
4ac8 cddc4b 595:          cmp   irf.dos_build+1
4acb d01c 596:          bne   .crc
4acd      597:
4acd adcb4b 598:          lda   irf.c_crc
4ad0 cddb4b 599:          cmp   irf.dos_build
4ad3 d014 600:          bne   .crc
4ad5      601:
4ad5 209151 602:          jsr   printsi
4ad8 9b 603:          dc.b  $9b
4ad9 a0c3d2 604:          dc.b  <+128>," CRC Not Set"
4ae5 ff 605:          dc.b  -1
4ae6 4c054b 606:          jmp   .notcrc
4ae9      607:
4ae9      608:
4ae9 209151 609: .crc:      jsr   printsi
4aec 9b496e 610:          dc.b  $9b,"Internal CRC "
4afa a4 611:          dc.b  <+128>,"$"
4afb ff 612:          dc.b  -1
4afc adcb4b 613:          lda   irf.c_crc
4aff aecc4b 614:          ldx   irf.c_crc+1
4b02 202a53 615:          jsr   pr_hex_card
4b05      616:
4b05      617: .notcrc:

```

```

4b05 209151 618:      jsr  printsi
4b08 9b9b56 619:      dc.b  $9b,$9b,"Version ",$FF
4b13 adc74b 620:      lda  irf.c_ver
4b16 20644c 621:      jsr  VER_NUM
4b19 209151 622:      jsr  printsi
4b1c 206c61 623:      dc.b  " last updated ",$FF
4b2b adc84b 624:      lda  irf.c_month
4b2e 205453 625:      jsr  PR_BYTE
4b31 a92f 626:      lda  #'/'
4b33 20e751 627:      jsr  ECHO
4b36 adc94b 628:      lda  irf.c_day
4b39 205453 629:      jsr  PR_BYTE
4b3c a92f 630:      lda  #'/'
4b3e 20e751 631:      jsr  ECHO
4b41 adcd4b 632:      lda  irf.c_year
4b44 aece4b 633:      ldx  irf.c_year+1
4b47 205653 634:      jsr  pr_card
4b4a 208e53 635:      jsr  cr_lf
4b4d 208e53 636:      jsr  cr_lf
4b50 60 637:      RTS
4b51 638: ;
4b51 639: FILE_NOT_FOUND:
4b51 a220 640:      ldx  #disk2
4b53 20624d 641:      jsr  CLOSE
4b56 209151 642:      jsr  printsi
4b59 9b9b20 643:      dc.b  $9B,$9B," File not found!",
4b6d 60 644:      RTS
4b6e 645: ;
4b6e 646: BROWSE_FILE:      ds.b  80
4bbe 647: ;
4bbe 648: ;=====
4bbe 649: ;This is the header data structor
4bbe 650: ;
4bbe 651:      .include  Hd-data
4c64 652:
4c64 653:
4c64 654: ;
4c64 655: ; this process the version
4c64 656: ;of a .com
4c64 657: ;
4c64 658: ;-----
4c64 659: ;
4c64 660: VER_NUM:
4c64 48 661:      PHA
4c65 4a 662:      LSR
4c66 4a 663:      LSR
4c67 4a 664:      LSR
4c68 4a 665:      LSR
4c69 20744c 666:      jsr  .a
4c6c a9ae 667:      lda  #'.'+128
4c6e 20e751 668:      jsr  ECHO
4c71 68 669:      PLA
4c72 290f 670:      and  #$0F

```

```

4c74 c90a 671: .a:      cmp  #$0A
4c76 b004 672:          bcs  .b
4c78 0930 673:          ORA  #$30
4c7a d002 674:          bne  .c
4c7c 6936 675: .b:      adc  #$36
4c7e      676: .c:
4c7e 4980 677:          eor  #$80
4c80 20e751 678:        jsr  ECHO
4c83 60 679:          RTS
4c84      680: ;
4c84      681: ;
4c84      682: ;-----
4c84      683: ;you must jsr to this vector for point, note, file lenght
4c84      684: ;
4c84 44783a 685: PNFDATA:      dc.b  "Dx:",$00
4c88 0000 686: PNFDATA1:     dc.b  $00,$00
4c8a      687: PNF_WORK:
4c8a 8e884c 688:          stx  PNFDATA1
4c8d a586 689:          lda  FN_PTR
4c8f 8d9d4c 690:          sta  PNF_1+1
4c92 a587 691:          lda  FN_PTR+1
4c94 8d9e4c 692:          sta  PNF_1+2
4c97 8c894c 693:          STY  PNFDATA1+1
4c9a a001 694:          LDY  #$01
4c9c b9ffff 695: PNF_1:      lda  $FFFF,Y
4c9f 8d854c 696:          sta  PNFDATA+1
4ca2 ac894c 697:          LDY  PNFDATA1+1
4ca5 ae884c 698:          ldx  PNFDATA1
4ca8 a900 699:          lda  #$00
4caa 9d4a03 700:          sta  $034A,X
4cad a900 701:          lda  #$00
4caf 9d4b03 702:          sta  $034B,X
4cb2 a984 703:          lda  # low PNFDATA
4cb4 9d4403 704:          sta  $0344,X
4cb7 a94c 705:          lda  # high PNFDATA
4cb9 9d4503 706:          sta  $0345,X
4cbc 2056e4 707:          jsr  $E456
4cbf 60 708:          RTS
4cc0      709: ;
4cc0      710: ;
4cc0      711: ;      XIO 37 POINT
4cc0      712: POINT:
4cc0 a925 713:          lda  #$25
4cc2 8ee44c 714:          stx  XIO_39_HOLD
4cc5 9d4203 715:          sta  $0342,X
4cc8 ade34c 716:          lda  OFFSET
4ccb 9d4c03 717:          sta  $034C,X
4cce ade14c 718:          lda  fil.sector
4cd1 9d4d03 719:          sta  $034D,X
4cd4 ade24c 720:          lda  fil.sector+1
4cd7 9d4e03 721:          sta  $034E,X
4cda 208a4c 722:          jsr  PNF_WORK
4cdd aee44c 723:          ldx  XIO_39_HOLD

```



```

4ce0 60    724:          RTS
4ce1      725:
4ce1 0000   726: fil.sector:      dc.w  0
4ce3 00     727: OFFSET:          dc.b  0
4ce4 0000   728: XIO_39_HOLD:     dc.b  $00,$00
4ce6 000000 729: FNLNG:          dc.b  0,0,0
4ce9 000000 730: F_LNG:          dc.b  0,0,0
4cec 0000   731: LNPTR:          dc.w  0
4cee      732: ;
4cee      733: ;
4cee      734: ;
4cee      735: ;-----;
4cee      736: ; LDA PR_PTR ; BUFFER
4cee      737: ; STA #<BUFFER+1
4cee      738: ;
4cee      739: BGET:
4cee a907   740:          lda  #$07
4cf0 9d4203 741:          sta  $0342,X
4cf3 a588   742:          lda  PR_PTR
4cf5 9d4403 743:          sta  $0344,X
4cf8 a589   744:          lda  PR_PTR+1
4cfa 9d4503 745:          sta  $0345,X
4cfd adec4c 746:          lda  LNPTR
4d00 9d4803 747:          sta  $0348,X
4d03 aded4c 748:          lda  LNPTR+1
4d06 9d4903 749:          sta  $0349,X
4d09 a903   750:          lda  #$03
4d0b 9d4a03 751:          sta  $034A,X
4d0e a900   752:          lda  #$00
4d10 9d4b03 753:          sta  $034B,X
4d13 2056e4 754:          jsr  $E456
4d16 60     755:          RTS
4d17      756: ;
4d17      757:
4d17      758:
4d17 a904   759: open_in:      lda  #4
4d19 4c354d 760:              jmp  do_open
4d1c      761:
4d1c a908   762: open_out:     lda  #8
4d1e 4c354d 763:              jmp  do_open
4d21      764:
4d21 a909   765: open_app:     lda  #9
4d23 4c354d 766:              jmp  do_open
4d26      767:
4d26 a90c   768: open_updt:    lda  #12
4d28 4c354d 769:              jmp  do_open
4d2b      770:
4d2b a906   771: open_dir:     lda  #6
4d2d 4c354d 772:              jmp  do_open
4d30      773:
4d30 a914   774: open_raw:     lda  #20      ;open raw d
4d32 4c354d 775:              jmp  do_open
4d35      776:

```

```

4d35      777: do_open
4d35 48   778:      pha      ;open files
4d36 8a   779:      txa      ;
4d37 48   780:      pha
4d38 20624d 781:      jsr  close      ;make sure
4d3b 68   782:      pla
4d3c aa   783:      tax
4d3d 68   784:      pla
4d3e 9d4a03 785:      sta  icaux1,x
4d41 a903   786:      lda  #ccopen
4d43 9d4203 787:      sta  iccom,x
4d46 a586   788:      lda  fn_ptr
4d48 9d4403 789:      sta  icbal,x
4d4b a587   790:      lda  fn_ptr+1
4d4d 9d4503 791:      sta  icbah,x
4d50 a950   792:      lda  #80
4d52 9d4803 793:      sta  icbll,x
4d55 a900   794:      lda  #0
4d57 9d4903 795:      sta  icblh,x
4d5a a980   796:      lda  #128      ;for long d
4d5c 9d4b03 797:      sta  icaux2,x
4d5f 4c56e4 798:      jmp   ciov
4d62      799:
4d62      800:
4d62      801:
4d62      802:
4d62      803: ;
4d62      804: ; generic close
4d62      805: close:
4d62 a90c   806:      lda  #$0c
4d64 9d4203 807:      sta  $0342,x
4d67 2056e4 808:      jsr  $e456
4d6a 60    809:      rts
4d6b      810:
4d6b      811: ;
4d6b      812: ;
4d6b      813: ;
4d6b 000000 814: FR_HOLD:      dc.b  0,0,0,0,0,0,0,0,0,0,0,0
4d80      815: ;
4d80      816: ;
4d80      817: ;
4d80 a200   818: FR_BYTE:      ldx  #0
4d82 86d5   819: FR_CARD:      stx  FR0+1
4d84 85d4   820: FR_NUM:       sta  FR0
4d86 20aad9 821:      jsr  IFP      ;float it
4d89 20e6d8 822:      jsr  FASC      ;to ascii
4d8c a000   823:      LDY  #0
4d8e b1f3   824: .a:          lda  (INBUFF),Y
4d90 08     825:      PHP
4d91 297f   826:      and  #$7F      ;last byte is negat
4d93 996c4d 827:      sta  FR_HOLD+1,Y ;JSR ECHO
4d96 c8     828:      INY
4d97 8c6b4d 829:      STY  FR_HOLD

```

```

4d9a      830: ;
4d9a 28   831:      PLP
4d9b 10f1 832:      BPL .a
4d9d 60   833:      RTS
4d9e      834: ;-----
4d9e      835: ;
4d9e      836: ;
4d9e      837: ;
4d9e      838: ;-----
4d9e      839: ;
4d9e      840: OUT_BUFF:
4d9e 000000 841:      dc.b 0,0,0,0,0,0,0,0,0
4da8 000000 842:      dc.b 0,0,0,0,0,0,0,0,0
4db2 000000 843:      dc.b 0,0,0,0,0,0,0,0,0
4dbc 000000 844:      dc.b 0,0,0,0,0,0,0,0,0
4dc6 000000 845:      dc.b 0,0,0,0,0,0,0,0,0
4dd0 000000 846:      dc.b 0,0,0,0,0,0,0,0,0
4dda 000000 847:      dc.b 0,0,0,0,0,0,0,0,0
4de4 000000 848:      dc.b 0,0,0,0,0,0,0,0,0
4dee 000000 849:      dc.b 0,0,0,0,0,0,0,0,0
4df8 000000 850:      dc.b 0,0,0,0,0,0,0,0,0
4e02 000000 851:      dc.b 0,0,0,0,0,0,0,0,0
4e0c 000000 852:      dc.b 0,0,0,0,0,0,0,0,0
4e16 000000 853:      dc.b 0,0,0,0,0,0,0,0,0
4e20 000000 854:      dc.b 0,0,0,0,0,0,0,0,0
4e2a 000000 855:      dc.b 0,0,0,0,0,0,0,0,0
4e34 000000 856:      dc.b 0,0,0,0,0,0,0,0,0
4e3e 000000 857:      dc.b 0,0,0,0,0,0,0,0,0
4e48 000000 858:      dc.b 0,0,0,0,0,0,0,0,0
4e52 000000 859:      dc.b 0,0,0,0,0,0,0,0,0
4e5c 000000 860:      dc.b 0,0,0,0,0,0,0,0,0
4e66 000000 861:      dc.b 0,0,0,0,0,0,0,0,0
4e70 000000 862:      dc.b 0,0,0,0,0,0,0,0,0
4e7a 000000 863:      dc.b 0,0,0,0,0,0,0,0,0
4e84 000000 864:      dc.b 0,0,0,0,0,0,0,0,0
4e8e 000000 865:      dc.b 0,0,0,0,0,0,0,0,0
4e98 000000 866:      dc.b 0,0,0,0,0,0,0,0,0
4ea2 000000 867:      dc.b 0,0,0,0,0,0,0,0,0
4eac      868: ;
4eac 012f 869: SLASH:      dc.b 1, "/"
4eae      870: ;
4eae 0e204c 871: file.dater:      dc.b 14, " Last Updated "
4ebd      872: ;
4ebd 0d2056 873: VER_POP_D:      dc.b 13, " Version 1.0 "
4ecb      874: ;
4ecb      875: C_LOW:
4ecb 57d9ee 876:      dc.b UNK &$ff, MAC &$ff, IASM &
4ed2      877: C_HIGH:
4ed2 4f4e4e 878:      dc.b UNK /256, MAC /256, IASM /
4ed9      879: ;
4ed9      880: ;
4ed9      881: MAC:
4ed9 142043 882:      dc.b 20, " Compiler = Mac5 "

```

```

4eee      883: IASM:
4eee 142043 884:          dc.b  20," Compiler = Xasm IBM"
4f03      885: I65:
4f03 142043 886:          dc.b  20," Compiler = C5 ibm"
4f18      887: ACT:
4f18 142043 888:          dc.b  20," Compiler = Action  "
4f2d      889: SASM:
4f2d 142043 890:          dc.b  20," Compiler = Xasm St "
4f42      891: S65:
4f42 142043 892:          dc.b  20," Compiler = C5 St "
4f57      893: UNK:
4f57 142043 894:          dc.b  20," Compiler = Unknown "
4f6c      895: ;
4f6c      896: ;
4f6c      897: ;
4f6c      898: ;
4f6c      899: ;
4f6c     900: VER_POP:
4f6c 48    901:          PHA
4f6d 4a    902:          LSR
4f6e 4a    903:          LSR
4f6f 4a    904:          LSR
4f70 4a    905:          LSR
4f71 20854f 906:          jsr  .a
4f74      907:
4f74 68    908:          PLA
4f75 290f   909:          and  #$0F
4f77 c90a   910:          cmp  #$0A
4f79 b004   911:          bcs  .b
4f7b 0930   912:          ORA  #$30
4f7d d002   913:          bne  .c
4f7f 6936   914: .b:      adc  #$36
4f81 8dc94e 915: .c:      sta  VER_POP_D+12
4f84 60     916:          RTS
4f85      917: ;
4f85 c90a   918: .a:      cmp  #$0A
4f87 b004   919:          bcs  .d
4f89 0930   920:          ORA  #$30
4f8b d002   921:          bne  .e
4f8d 6936   922: .d:      adc  #$36
4f8f 8dc74e 923: .e:      sta  VER_POP_D+10
4f92 60     924:          RTS
4f93      925: ;
4f93      926: ;-----
4f93      927: ;
4f93      928: ;
4f93      929: ;
4f93      930: ;
4f93      931: ;
4f93      932: ;-----
4f93      933: ;
4f93     934: PRINT_V:
4f93     935: ;

```

```

4f93      936: ;          LDY  PARM_1
4f93      937: ;          DEY
4f93      938: ;          STY  PARM_1
4f93      939:          mea  PARM_1,S_PTR
4f93 a93f ----:          lda  #low PARM_1
4f95 8582 ----:          sta  S_PTR
4f97 a945 ----:          lda  #high PARM_1
4f99 8583 ----:          sta  S_PTR + 1
4f9b      940:          mea  OUT_BUFF,D_PTR
4f9b a99e ----:          lda  #low OUT_BUFF
4f9d 8584 ----:          sta  D_PTR
4f9f a94d ----:          lda  #high OUT_BUFF
4fa1 8585 ----:          sta  D_PTR + 1
4fa3 202d55 941:          jsr  MOVE_STR
4fa6      942:
4fa6 adc74b 943:          lda  irf.c_ver
4fa9 206c4f 944:          jsr  VER_POP
4fac      945:          mea  VER_POP_D,S_PTR
4fac a9bd ----:          lda  #low VER_POP_D
4fae 8582 ----:          sta  S_PTR
4fb0 a94e ----:          lda  #high VER_POP_D
4fb2 8583 ----:          sta  S_PTR + 1
4fb4      946:          mea  OUT_BUFF,D_PTR
4fb4 a99e ----:          lda  #low OUT_BUFF
4fb6 8584 ----:          sta  D_PTR
4fb8 a94d ----:          lda  #high OUT_BUFF
4fba 8585 ----:          sta  D_PTR + 1
4fbc 206b55 947:          jsr  CONCAT
4fbf      948: ;
4fbf      949:          mea  file.dater,S_PTR
4fbf a9ae ----:          lda  #low file.dater
4fc1 8582 ----:          sta  S_PTR
4fc3 a94e ----:          lda  #high file.dater
4fc5 8583 ----:          sta  S_PTR + 1
4fc7      950:          mea  OUT_BUFF,D_PTR
4fc7 a99e ----:          lda  #low OUT_BUFF
4fc9 8584 ----:          sta  D_PTR
4fcb a94d ----:          lda  #high OUT_BUFF
4fcd 8585 ----:          sta  D_PTR + 1
4fcf 206b55 951:          jsr  CONCAT
4fd2      952: ;
4fd2 adc84b 953:          lda  irf.c_month
4fd5 20804d 954:          jsr  FR_BYTE
4fd8      955:          mea  FR_HOLD,S_PTR
4fd8 a96b ----:          lda  #low FR_HOLD
4fda 8582 ----:          sta  S_PTR
4fdc a94d ----:          lda  #high FR_HOLD
4fde 8583 ----:          sta  S_PTR + 1
4fe0      956:          mea  OUT_BUFF,D_PTR
4fe0 a99e ----:          lda  #low OUT_BUFF
4fe2 8584 ----:          sta  D_PTR
4fe4 a94d ----:          lda  #high OUT_BUFF
4fe6 8585 ----:          sta  D_PTR + 1

```

```

4fe8 206b55 957:      jsr  CONCAT
4feb      958: ;
4feb      959:      mea  SLASH,S_PTR
4feb a9ac ----:      lda  #low SLASH
4fed 8582 ----:      sta  S_PTR
4fef a94e ----:      lda  #high SLASH
4ff1 8583 ----:      sta  S_PTR + 1
4ff3      960:      mea  OUT_BUFF,D_PTR
4ff3 a99e ----:      lda  #low OUT_BUFF
4ff5 8584 ----:      sta  D_PTR
4ff7 a94d ----:      lda  #high OUT_BUFF
4ff9 8585 ----:      sta  D_PTR + 1
4ffb 206b55 961:      jsr  CONCAT
4ffe      962: ;
4ffe adc94b 963:      lda  irf.c_day
5001 20804d 964:      jsr  FR_BYTE
5004      965:      mea  FR_HOLD,S_PTR
5004 a96b ----:      lda  #low FR_HOLD
5006 8582 ----:      sta  S_PTR
5008 a94d ----:      lda  #high FR_HOLD
500a 8583 ----:      sta  S_PTR + 1
500c      966:      mea  OUT_BUFF,D_PTR
500c a99e ----:      lda  #low OUT_BUFF
500e 8584 ----:      sta  D_PTR
5010 a94d ----:      lda  #high OUT_BUFF
5012 8585 ----:      sta  D_PTR + 1
5014 206b55 967:      jsr  CONCAT
5017      968: ;
5017      969:      mea  SLASH,S_PTR
5017 a9ac ----:      lda  #low SLASH
5019 8582 ----:      sta  S_PTR
501b a94e ----:      lda  #high SLASH
501d 8583 ----:      sta  S_PTR + 1
501f      970:      mea  OUT_BUFF,D_PTR
501f a99e ----:      lda  #low OUT_BUFF
5021 8584 ----:      sta  D_PTR
5023 a94d ----:      lda  #high OUT_BUFF
5025 8585 ----:      sta  D_PTR + 1
5027 206b55 971:      jsr  CONCAT
502a      972: ;
502a adca4b 973:      lda  irf.c_year_short
502d 20804d 974:      jsr  FR_BYTE
5030      975:      mea  FR_HOLD,S_PTR
5030 a96b ----:      lda  #low FR_HOLD
5032 8582 ----:      sta  S_PTR
5034 a94d ----:      lda  #high FR_HOLD
5036 8583 ----:      sta  S_PTR + 1
5038      976:      mea  OUT_BUFF,D_PTR
5038 a99e ----:      lda  #low OUT_BUFF
503a 8584 ----:      sta  D_PTR
503c a94d ----:      lda  #high OUT_BUFF
503e 8585 ----:      sta  D_PTR + 1
5040 206b55 977:      jsr  CONCAT

```

```
5043 60 978:          RTS
5044      979:
5044      980: P_ERROR:
5044 a220 981:          ldx  #disk2
5046 20624d 982:         jsr  CLOSE
5049 ad2f47 983:         lda  WHERE
504c c950 984:         cmp  #'P'
504e d00b 985:         bne  P_ERR
5050      986:         mea  PRN,FN_PTR
5050 a9da ----:         lda  #low PRN
5052 8586 ----:         sta  FN_PTR
5054 a944 ----:         lda  #high PRN
5056 8587 ----:         sta  FN_PTR + 1
5058 4c6350 987:         jmp  OPYT
505b      988: P_ERR:
505b      989:         mea  FILENAME,FN_PTR
505b a9de ----:         lda  #low FILENAME
505d 8586 ----:         sta  FN_PTR
505f a944 ----:         lda  #high FILENAME
5061 8587 ----:         sta  FN_PTR + 1
5063 a220 990: OPYT:          ldx  #disk2
5065 20214d 991:         jsr  OPEN_APP
5068 3027 992:         bmi  F_ERROR
506a      993:
506a      994:         mea  SPLIT,PR_PTR
506a a9b3 ----:         lda  #low SPLIT
506c 8588 ----:         sta  PR_PTR
506e a944 ----:         lda  #high SPLIT
5070 8589 ----:         sta  PR_PTR + 1
5072 a220 995:          ldx  #disk2
5074 204754 996:         jsr  PRINTSD
5077      997:
5077      998:         mea  PARM_1,PR_PTR
5077 a93f ----:         lda  #low PARM_1
5079 8588 ----:         sta  PR_PTR
507b a945 ----:         lda  #high PARM_1
507d 8589 ----:         sta  PR_PTR + 1
507f a220 999:          ldx  #disk2
5081 204754 1000:         jsr  PRINTSD
5084      1001:         mea  INVALID,PR_PTR
5084 a998 ----:         lda  #low INVALID
5086 8588 ----:         sta  PR_PTR
5088 a944 ----:         lda  #high INVALID
508a 8589 ----:         sta  PR_PTR + 1
508c a220 1002:         ldx  #disk2
508e 204754 1003:         jsr  PRINTSD
5091      1004:
5091 a220 1005: F_ERROR:        ldx  #disk2
5093 20624d 1006:         jsr  CLOSE
5096 60 1007:          RTS
5097      1008: ;
5097      1009: VER_PRINT:
5097 20934f 1010:         jsr  PRINT_V
```

```

509a a220 1011:      ldx  #disk2
509c 20624d 1012:    jsr  CLOSE
509f ad2f47 1013:    lda  WHERE
50a2 c950 1014:      cmp  #'P'
50a4 d00b 1015:      bne  P_E
50a6      1016:      mea  PRN,FN_PTR
50a6 a9da ----:      lda  #low PRN
50a8 8586 ----:      sta  FN_PTR
50aa a944 ----:      lda  #high PRN
50ac 8587 ----:      sta  FN_PTR + 1
50ae 4cb950 1017:    jmp  OYPT
50b1      1018: P_E:
50b1      1019:      mea  FILENAME,FN_PTR
50b1 a9de ----:      lda  #low FILENAME
50b3 8586 ----:      sta  FN_PTR
50b5 a944 ----:      lda  #high FILENAME
50b7 8587 ----:      sta  FN_PTR + 1
50b9 a220 1020: OYPT: ldx  #disk2
50bb 20214d 1021:    jsr  OPEN_APP
50be 30d1 1022:      bmi  F_ERROR
50c0      1023:      mea  SPLIT,PR_PTR
50c0 a9b3 ----:      lda  #low SPLIT
50c2 8588 ----:      sta  PR_PTR
50c4 a944 ----:      lda  #high SPLIT
50c6 8589 ----:      sta  PR_PTR + 1
50c8 a220 1024:      ldx  #disk2
50ca 204754 1025:    jsr  PRINTSD
50cd      1026:      mea  OUT_BUFF,PR_PTR
50cd a99e ----:      lda  #low OUT_BUFF
50cf 8588 ----:      sta  PR_PTR
50d1 a94d ----:      lda  #high OUT_BUFF
50d3 8589 ----:      sta  PR_PTR + 1
50d5 a220 1027:      ldx  #disk2
50d7 204754 1028:    jsr  PRINTSD
50da a220 1029:      ldx  #disk2
50dc 20624d 1030:    jsr  CLOSE
50df 60 1031:      RTS
50e0      1032:
50e0      1033: PRINT_csv:
50e0      1034:      mea  PARM_1,S_PTR
50e0 a93f ----:      lda  #low PARM_1
50e2 8582 ----:      sta  S_PTR
50e4 a945 ----:      lda  #high PARM_1
50e6 8583 ----:      sta  S_PTR + 1
50e8      1035:      mea  OUT_BUFF,D_PTR
50e8 a99e ----:      lda  #low OUT_BUFF
50ea 8584 ----:      sta  D_PTR
50ec a94d ----:      lda  #high OUT_BUFF
50ee 8585 ----:      sta  D_PTR + 1
50f0 202d55 1036:    jsr  MOVE_STR
50f3      1037:
50f3 adc74b 1038:    lda  irf.c_ver
50f6 206c4f 1039:    jsr  VER_POP

```



```

50f9      1040:          mea   VER_POP_D,S_PTR
50f9 a9bd ----:      lda   #low VER_POP_D
50fb 8582 ----:      sta   S_PTR
50fd a94e ----:      lda   #high VER_POP_D
50ff 8583 ----:      sta   S_PTR + 1
5101      1041:          mea   OUT_BUFF,D_PTR
5101 a99e ----:      lda   #low OUT_BUFF
5103 8584 ----:      sta   D_PTR
5105 a94d ----:      lda   #high OUT_BUFF
5107 8585 ----:      sta   D_PTR + 1
5109 206b55 1042:      jsr   CONCAT
510c      1043: ;
510c      1044:          mea   file.dater,S_PTR
510c a9ae ----:      lda   #low file.dater
510e 8582 ----:      sta   S_PTR
5110 a94e ----:      lda   #high file.dater
5112 8583 ----:      sta   S_PTR + 1
5114      1045:          mea   OUT_BUFF,D_PTR
5114 a99e ----:      lda   #low OUT_BUFF
5116 8584 ----:      sta   D_PTR
5118 a94d ----:      lda   #high OUT_BUFF
511a 8585 ----:      sta   D_PTR + 1
511c 206b55 1046:      jsr   CONCAT
511f      1047: ;
511f adc84b 1048:      lda   irf.c_month
5122 20804d 1049:      jsr   FR_BYTE
5125      1050:          mea   FR_HOLD,S_PTR
5125 a96b ----:      lda   #low FR_HOLD
5127 8582 ----:      sta   S_PTR
5129 a94d ----:      lda   #high FR_HOLD
512b 8583 ----:      sta   S_PTR + 1
512d      1051:          mea   OUT_BUFF,D_PTR
512d a99e ----:      lda   #low OUT_BUFF
512f 8584 ----:      sta   D_PTR
5131 a94d ----:      lda   #high OUT_BUFF
5133 8585 ----:      sta   D_PTR + 1
5135 206b55 1052:      jsr   CONCAT
5138      1053: ;
5138      1054:          mea   SLASH,S_PTR
5138 a9ac ----:      lda   #low SLASH
513a 8582 ----:      sta   S_PTR
513c a94e ----:      lda   #high SLASH
513e 8583 ----:      sta   S_PTR + 1
5140      1055:          mea   OUT_BUFF,D_PTR
5140 a99e ----:      lda   #low OUT_BUFF
5142 8584 ----:      sta   D_PTR
5144 a94d ----:      lda   #high OUT_BUFF
5146 8585 ----:      sta   D_PTR + 1
5148 206b55 1056:      jsr   CONCAT
514b      1057: ;
514b adc94b 1058:      lda   irf.c_day
514e 20804d 1059:      jsr   FR_BYTE
5151      1060:          mea   FR_HOLD,S_PTR

```

```

5151 a96b ----:      lda  #low FR_HOLD
5153 8582 ----:      sta  S_PTR
5155 a94d ----:      lda  #high FR_HOLD
5157 8583 ----:      sta  S_PTR + 1
5159 1061:          mea  OUT_BUFF,D_PTR
5159 a99e ----:      lda  #low OUT_BUFF
515b 8584 ----:      sta  D_PTR
515d a94d ----:      lda  #high OUT_BUFF
515f 8585 ----:      sta  D_PTR + 1
5161 206b55 1062:      jsr  CONCAT
5164 1063: ;
5164 1064:          mea  SLASH,S_PTR
5164 a9ac ----:      lda  #low SLASH
5166 8582 ----:      sta  S_PTR
5168 a94e ----:      lda  #high SLASH
516a 8583 ----:      sta  S_PTR + 1
516c 1065:          mea  OUT_BUFF,D_PTR
516c a99e ----:      lda  #low OUT_BUFF
516e 8584 ----:      sta  D_PTR
5170 a94d ----:      lda  #high OUT_BUFF
5172 8585 ----:      sta  D_PTR + 1
5174 206b55 1066:      jsr  CONCAT
5177 1067:
5177 adca4b 1068:      lda  irf.c_year_short
517a 20804d 1069:      jsr  FR_BYTE
517d 1070:          mea  FR_HOLD,S_PTR
517d a96b ----:      lda  #low FR_HOLD
517f 8582 ----:      sta  S_PTR
5181 a94d ----:      lda  #high FR_HOLD
5183 8583 ----:      sta  S_PTR + 1
5185 1071:          mea  OUT_BUFF,D_PTR
5185 a99e ----:      lda  #low OUT_BUFF
5187 8584 ----:      sta  D_PTR
5189 a94d ----:      lda  #high OUT_BUFF
518b 8585 ----:      sta  D_PTR + 1
518d 206b55 1072:      jsr  CONCAT
5190 60 1073:          RTS
5191 1074:
5191 1075: ;
5191 1076: ;
5191 1077: ;-----
5191 1078: ;printsi Routeen macro
5191 1079: ;this is a slow print to the screen
5191 1080: ;usage jsr  printsi
5191 1081: ; .byte $9b," string to be printed",$ff
5191 1082: ;-----
5191 1083: ;
5191 68 1084: printsi:          pla
5192 8da251 1085:          sta  pstr+1
5195 68 1086:          pla
5196 8da351 1087:          sta  pstr+2
5199 eea251 1088: prsl:          inc  pstr+1
519c d003 1089:          bne  pstr

```

```

519e eea351 1090:      inc    pstr+2
51a1 adffff 1091: pstr:    lda    $ffff
51a4 c9ff  1092:      cmp    #$ff
51a6 f006  1093:      beq    estri
51a8 20e751 1094:      jsr    fast_output
51ab 4c9951 1095:      jmp    prsl
51ae ada351 1096: estri:  lda    pstr+2
51b1 48    1097:      pha
51b2 ada251 1098:      lda    pstr+1
51b5 48    1099:      pha
51b6 60    1100:      rts
51b7      1101: ;
51b7      1102: ;
51b7      1103: ;
51b7      1104: ;
51b7      1105: ;-----
51b7      1106: ;actual screen handler
51b7      1107: ;screen offset definitions
51b7      1108: ;-----
51b7 0000  1109: scr_offset: dc.w  0
51b9 2800  1110:      dc.w  40
51bb 5000  1111:      dc.w  80
51bd 7800  1112:      dc.w 120
51bf a000  1113:      dc.w 160
51c1 c800  1114:      dc.w 200
51c3 f000  1115:      dc.w 240
51c5 1801  1116:      dc.w 280
51c7 4001  1117:      dc.w 320
51c9 6801  1118:      dc.w 360
51cb 9001  1119:      dc.w 400
51cd b801  1120:      dc.w 440
51cf e001  1121:      dc.w 480
51d1 0802  1122:      dc.w 520
51d3 3002  1123:      dc.w 560
51d5 5802  1124:      dc.w 600
51d7 8002  1125:      dc.w 640
51d9 a802  1126:      dc.w 680
51db d002  1127:      dc.w 720
51dd f802  1128:      dc.w 760
51df 2003  1129:      dc.w 800
51e1 4803  1130:      dc.w 840
51e3 7003  1131:      dc.w 880
51e5 9803  1132:      dc.w 920
51e7      1133: ;
51e7      1134: echo:
51e7      1135: fast_output:
51e7 855a  1136:      sta    $5a
51e9 ad0407 1137:      lda    sc.redirect
51ec c902  1138:      cmp    #print_p
51ee f01f  1139:      beq    .go_cio
51f0      1140: .leo:
51f0 adff02 1141:      lda    $02ff      ; CONTROL 1
51f3 c900  1142:      cmp    #$00

```

```

51f5 d0f9 1143:      bne  .leo
51f7      1144:
51f7 a55a 1145:      lda  $5a
51f9 c920 1146:      cmp  #32
51fb 9004 1147:      bcc  .do_look
51fd c97d 1148:      cmp  #125
51ff 9013 1149:      bcc  .do_here
5201 a000 1150: .do_look:      ldy  #0
5203 b90a53 1151: .lookup:      lda  .char_table,y
5206 f00c 1152:      beq  .do_here
5208 c55a 1153:      cmp  $5a
520a f003 1154:      beq  .go_cio
520c c8 1155:      iny
520d d0f4 1156:      bne  .lookup
520f      1157:
520f      1158:
520f a55a 1159: .go_cio:      lda  $5a
5211 4c1a53 1160:      jmp  putlocal
5214      1161:
5214      1162:
5214 a000 1163: .do_here:      ldy  #0
5216 a55d 1164:      lda  $5d
5218 915e 1165:      sta  ($5e),y
521a a55a 1166:      lda  $5a
521c c99b 1167:      cmp  #$9b
521e f044 1168:      beq  .docr
5220 20e952 1169: .notcr:      jsr  .get_adr
5223 a55a 1170:      lda  $5a
5225 297f 1171:      and  #$7f
5227 c920 1172:      cmp  #32
5229 9007 1173:      bcc  .add64
522b c960 1174:      cmp  #96
522d 9009 1175:      bcc  .sub32
522f 4c3b52 1176:      jmp  .asis
5232 18 1177: .add64:      clc
5233 6940 1178:      adc  #64
5235 4c3b52 1179:      jmp  .asis
5238 38 1180: .sub32:      sec
5239 e920 1181:      sbc  #32
523b 245a 1182: .asis:      bit  $5a
523d 1002 1183:      bpl  .xxlate
523f 0980 1184:      ora  #$80
5241 a000 1185: .xxlate:      ldy  #0
5243 915e 1186:      sta  ($5e),y
5245 e655 1187:      inc  $55
5247 e663 1188:      inc  $63
5249 a555 1189:      lda  $55
524b c928 1190:      cmp  #40
524d b019 1191:      bcs  .next_row
524f c8 1192:      iny
5250 b15e 1193:      lda  ($5e),y
5252 855d 1194:      sta  $5d
5254 0980 1195:      ora  #$80

```

```

5256 aef002 1196:      ldx  752
5259 d002  1197:      bne  .nocurs1
525b 915e  1198:      sta  ($5e),y
525d e65e  1199: .nocurs1:      inc  $5e
525f d002  1200:      bne  .nooav
5261 e65f  1201:      inc  $5e+1
5263 60    1202: .nooav:      rts
5264 a900  1203: .docr:      lda  #0
5266 8563  1204:      sta  $63
5268 a552  1205: .next_row:   lda  $52
526a 8555  1206:      sta  $55
526c e654  1207:      inc  $54
526e a454  1208:      ldy  $54
5270 c018  1209:      cpy  #24
5272 b013  1210:      bcs  .scroll
5274 20e952 1211:      jsr  .get_adr
5277 a000  1212:      ldy  #0
5279 b15e  1213:      lda  ($5e),y
527b 855d  1214:      sta  $5d
527d aef002 1215:      ldx  752
5280 d004  1216:      bne  .nocurs2
5282 0980  1217:      ora  #$80
5284 915e  1218:      sta  ($5e),y
5286 60    1219: .nocurs2:   rts
5287      1220:
5287 c654  1221: .scroll:    dec  $54
5289 a558  1222:      lda  $58
528b 8568  1223:      sta  $68
528d 18    1224:      clc
528e 6928  1225:      adc  #40
5290 855e  1226:      sta  $5e
5292 a559  1227:      lda  $58+1
5294 8569  1228:      sta  $68+1
5296 6900  1229:      adc  #0
5298 855f  1230:      sta  $5e+1
529a a000  1231:      ldy  #0
529c a203  1232:      ldx  #3
529e b15e  1233: .sclloop:   lda  ($5e),y
52a0 9168  1234:      sta  ($68),y
52a2 c8    1235:      iny
52a3 d0f9  1236:      bne  .sclloop
52a5 e65f  1237:      inc  $5e+1
52a7 e669  1238:      inc  $68+1
52a9 ca    1239:      dex
52aa d0f2  1240:      bne  .sclloop
52ac b15e  1241: .sclloop2:  lda  ($5e),y
52ae 9168  1242:      sta  ($68),y
52b0 c8    1243:      iny
52b1 c098  1244:      cpy  #152
52b3 90f7  1245:      bcc  .sclloop2
52b5 a558  1246:      lda  $58
52b7 18    1247:      clc
52b8 6998  1248:      adc  #920&$ff

```

```

52ba 855e 1249:      sta  $5e
52bc a559 1250:      lda  $58+1
52be 6903 1251:      adc  #920/256
52c0 855f 1252:      sta  $5e+1
52c2 a000 1253:      ldy  #0
52c4 98 1254:      tya
52c5 915e 1255: .clear:      sta  ($5e),y
52c7 c8 1256:      iny
52c8 c028 1257:      cpy  #40
52ca 90f9 1258:      bcc  .clear
52cc aef002 1259:      ldx  752
52cf d006 1260:      bne  .nocurs3
52d1 a980 1261:      lda  #$80
52d3 a455 1262:      ldy  $55
52d5 915e 1263:      sta  ($5e),y
52d7 a900 1264: .nocurs3:      lda  #0
52d9 855d 1265:      sta  $5d
52db a55e 1266:      lda  $5e
52dd 18 1267:      clc
52de 6555 1268:      adc  $55
52e0 855e 1269:      sta  $5e
52e2 a55f 1270:      lda  $5e+1
52e4 6900 1271:      adc  #0
52e6 855f 1272:      sta  $5e+1
52e8 60 1273:      rts
52e9 a554 1274: .get_adr:      lda  $54
52eb 0a 1275:      asl
52ec a8 1276:      tay
52ed a558 1277:      lda  $58
52ef 18 1278:      clc
52f0 79b751 1279:      adc  scr_offset,y
52f3 855e 1280:      sta  $5e
52f5 a559 1281:      lda  $58+1
52f7 79b851 1282:      adc  scr_offset+1,y
52fa 855f 1283:      sta  $5e+1
52fc a55e 1284:      lda  $5e
52fe 18 1285:      clc
52ff 6555 1286:      adc  $55
5301 855e 1287:      sta  $5e
5303 a55f 1288:      lda  $5e+1
5305 6900 1289:      adc  #0
5307 855f 1290:      sta  $5e+1
5309 60 1291:      rts
530a 1292: .char_table:
530a 1b1c1d 1293:      dc.b  27,28,29,30,31,125,126,127
5312 9c9d9e 1294:      dc.b  156,157,158,159,253,254,255
531a 1295:
531a 1296:
531a a20b 1297: putlocal:      ldx  #11
531c 8e4203 1298:      stx  $0342
531f a200 1299:      ldx  #0
5321 8e4803 1300:      stx  $0348
5324 8e4903 1301:      stx  $0349

```

```

5327 4c56e4 1302:          jmp   $e456
532a      1303: ;
532a      1304:
532a      1305:
532a      1306:
532a      1307:
532a      1308:
532a      1309: ;-----
532a      1310: ; Print Hex Numbers to screen byte or card
532a      1311: ;-----
532a      1312: ;
532a      1313: ;
532a      1314: ; print a hex word value
532a      1315: ; low byte in a reg x is high byte
532a      1316: ;
532a 8d3853 1317: pr_hex_card:      sta   .h_cardd
532d 8a      1318:          txa
532e 203953 1319:          jsr   pr_hex_byte
5331 ad3853 1320:          lda   .h_cardd
5334 203953 1321:          jsr   pr_hex_byte
5337 60      1322:          rts
5338      1323: ;
5338 00      1324: .h_cardd:      dc.b   $00
5339      1325: ;
5339 48      1326: pr_hex_byte:    pha
533a 4a      1327:          lsr
533b 4a      1328:          lsr
533c 4a      1329:          lsr
533d 4a      1330:          lsr
533e 204453 1331:          jsr   .a
5341 68      1332:          pla
5342 290f     1333:          and   #$0f
5344 c90a     1334: .a:          cmp   #$0a
5346 b004     1335:          bcs   .b
5348 0930     1336:          ora   #$30
534a d002     1337:          bne   .c
534c 6936     1338: .b:          adc   #$36
534e      1339: .c:
534e 4980     1340:          eor   #$80
5350 20e751 1341:          jsr   echo
5353 60      1342:          rts
5354      1343:
5354      1344: ;~~~~~
5354      1345: ; Prints a byte or a card to the screen
5354      1346: ;
5354      1347: pr_byte:
5354 a200     1348:          ldx   #$00
5356      1349: pr_card:
5356 85d4     1350:          sta   fr0
5358 86d5     1351:          stx   fr0+1
535a a900     1352:          lda   #$00
535c 85f2     1353:          sta   cix
535e 8d8453 1354:          sta   .bytecard

```

```

5361 20aad9 1355:      jsr  ifp
5364 20e6d8 1356:      jsr  fasc
5367 a000 1357:      ldy  #$00
5369      1358: .aprbloop:
5369 ee8453 1359:      inc  .bytecard
536c b1f3 1360:      lda  ($f3),y
536e 08 1361:      php
536f 297f 1362:      and  #$7f
5371 998553 1363:      sta  .bytecard+1,y
5374 c8 1364:      iny
5375 28 1365:      plp
5376 10f1 1366:      bpl  .aprbloop
5378      1367:      mea  .bytecard,pr_ptr
5378 a984 ----:      lda  #low .bytecard
537a 8588 ----:      sta  pr_ptr
537c a953 ----:      lda  #high .bytecard
537e 8589 ----:      sta  pr_ptr + 1
5380 209453 1368:      jsr  echos
5383 60 1369:      rts
5384      1370:
5384      1371: .bytecard:      ds.b 10
538e      1372:
538e      1373:
538e      1374:
538e      1375: cr_lf:
538e 209151 1376:      jsr  printsi
5391 9bff 1377:      dc.b $9b,-1
5393 60 1378:      rts
5394      1379:
5394      1380: ; prints: prints string with address in PR_PTR, length by
5394      1381: ; ~~~~~
5394      1382:
5394      1383: echos:
5394 a000 1384:      ldy  #0
5396 b188 1385:      lda  (pr_ptr),y      ;ge
5398 f01b 1386:      beq  .xpri      ;ex
539a 8db753 1387:      sta  .maxlth
539d c8 1388:      iny
539e 8cb653 1389:      sty  .tempy
53a1      1390:
53a1 acb653 1391: .pril:      ldy  .tempy
53a4 b188 1392:      lda  (pr_ptr),y
53a6 c900 1393:      cmp  #$00
53a8 f00b 1394:      beq  .xpri
53aa 20e751 1395:      jsr  echo
53ad eeb653 1396: .prnext:      inc  .tempy
53b0 ceb753 1397:      dec  .maxlth
53b3 d0ec 1398:      bne  .pril
53b5 60 1399: .xpri:      rts
53b6      1400:
53b6 01 1401: .tempy:      dc.b 1
53b7 01 1402: .maxlth:      dc.b 1
53b8      1403:

```



```

53b8      1404: ; prints: prints string with address in PR_PTR, length by
53b8      1405: ; in inverse
53b8      1406: ; ~~~~~
53b8      1407:
53b8      1408: echosinv:
53b8 a000 1409:          ldy    #0
53ba b188 1410:          lda    (pr_ptr),y      ;ge
53bc f019 1411:          beq    .xpri          ;ex
53be 8dd953 1412:          sta    .maxlth
53c1 c8 1413:          iny
53c2 8cd853 1414:          sty    .tempy
53c5      1415:
53c5 acd853 1416: .pril:          ldy    .tempy
53c8 b188 1417:          lda    (pr_ptr),y
53ca 4980 1418:          eor    #$80           ; t
53cc 20e751 1419:          jsr    echo
53cf eed853 1420: .prnext:        inc    .tempy
53d2 ced953 1421:          dec    .maxlth
53d5 d0ee 1422:          bne    .pril
53d7 60 1423: .xpri:          rts
53d8      1424:
53d8 01 1425: .tempy:          dc.b    1
53d9 01 1426: .maxlth:         dc.b    1
53da      1427:
53da      1428:
53da      1429: ; prints: prints string with address in PR_PTR, length by
53da      1430: ; in inverse
53da      1431: ; ~~~~~
53da      1432:
53da      1433: echos.unify:
53da a000 1434:          ldy    #0
53dc b188 1435:          lda    (pr_ptr),y      ;ge
53de f03e 1436:          beq    .xpri          ;ex
53e0 8d2054 1437:          sta    .maxlth
53e3 c8 1438:          iny
53e4 8c1f54 1439:          sty    .tempy
53e7      1440:
53e7 ac1f54 1441:          ldy    .tempy
53ea b188 1442:          lda    (pr_ptr),y
53ec 20e854 1443:          jsr    upcase
53ef 9188 1444:          sta    (pr_ptr),y
53f1 4c1654 1445:          jmp    .prnext
53f4      1446:
53f4 ac1f54 1447: .pril:          ldy    .tempy
53f7 c002 1448:          cpy    #$02
53f9 901b 1449:          bcc    .prnext
53fb b188 1450:          lda    (pr_ptr),y      ; c
53fd 20f654 1451:          jsr    locase
5400 9188 1452:          sta    (pr_ptr),y
5402      1453:
5402 c920 1454:          cmp    #' '           ; l
5404 d010 1455:          bne    .prnext
5406 ee1f54 1456:          inc    .tempy

```

```

5409 ce2054 1457:      dec  .maxlth
540c ac1f54 1458:      ldy  .tempy
540f b188  1459:      lda  (pr_ptr),y      ; m
5411 20e854 1460:      jsr  upcase
5414 9188  1461:      sta  (pr_ptr),y
5416      1462:
5416      1463:
5416 ee1f54 1464: .prnext:      inc  .tempy
5419 ce2054 1465:      dec  .maxlth
541c d0d6  1466:      bne  .pril
541e 60    1467: .xpri:      rts
541f      1468:
541f 01    1469: .tempy:      dc.b  1
5420 01    1470: .maxlth:     dc.b  1
5421      1471:
5421      1472:
5421      1473:
5421      1474: ; prints: prints string with address in PR_PTR, length by
5421      1475: ; and supress the space at the end.
5421      1476: ; ~~~~~
5421      1477:
5421      1478: echos.nospace:
5421 a000  1479:      ldy  #0
5423 b188  1480:      lda  (pr_ptr),y      ;ge
5425 f01d  1481:      beq  .xpri           ;ex
5427 8d4654 1482:      sta  .maxlth
542a c8    1483:      iny
542b 8c4554 1484:      sty  .tempy
542e      1485:
542e ac4554 1486: .pril:      ldy  .tempy
5431 b188  1487:      lda  (pr_ptr),y
5433 c920  1488:      cmp  #' '
5435 f005  1489:      beq  .prnext
5437 4980  1490:      eor  #$80           ; t
5439 20e751 1491:      jsr  echo
543c ee4554 1492: .prnext:      inc  .tempy
543f ce4654 1493:      dec  .maxlth
5442 d0ea  1494:      bne  .pril
5444 60    1495: .xpri:      rts
5445      1496:
5445 01    1497: .tempy:      dc.b  1
5446 01    1498: .maxlth:     dc.b  1
5447      1499:
5447      1500:
5447      1501:
5447      1502:
5447      1503:
5447      1504:
5447      1505:
5447      1506: ;      print a string to device in X register
5447      1507: ;      from the string pointed to by pr_ptr.
5447      1508: ;      ~~~~~
5447      1509:

```

```

5447 a000 1510: printsd:      ldy  #0
5449 b188 1511:      lda  (pr_ptr),y      ;install EO
544b a8   1512:      tay
544c c8   1513:      iny
544d a99b 1514:      lda  #$9b
544f 9188 1515:      sta  (pr_ptr),y
5451 a909 1516:      lda  #ccputrec
5453 9d4203 1517:      sta  iccom,x
5456 a588 1518:      lda  pr_ptr
5458 18   1519:      clc
5459 6901 1520:      adc  #1              ;offset pas
545b 9d4403 1521:      sta  icbal,x
545e a589 1522:      lda  pr_ptr+1
5460 6900 1523:      adc  #0
5462 9d4503 1524:      sta  icbah,x
5465 a950 1525:      lda  #80
5467 9d4803 1526:      sta  icbll,x
546a a900 1527:      lda  #0
546c 9d4903 1528:      sta  icblh,x        ;zero hi by
546f 4c56e4 1529:      jmp  ciov           ;get the li
5472      1530:
5472      1531: ;      Search the string pointed to by S_PTR for the subs
5472      1532: ;      pointed to by D_PTR. Returns a 0 if substring not
5472      1533: ;      ~~~~~
5472      1534:
5472 a000 1535: substr:      ldy  #0
5474 b182 1536:      lda  (s_ptr),y
5476 8de654 1537:      sta  .s_lth
5479 b184 1538:      lda  (d_ptr),y
547b 8de754 1539:      sta  .d_lth
547e a582 1540:      lda  s_ptr
5480 8da254 1541:      sta  .comp+1
5483 a583 1542:      lda  s_ptr+1
5485 8da354 1543:      sta  .comp+2
5488 a584 1544:      lda  d_ptr
548a 8dab54 1545:      sta  .get+1
548d a585 1546:      lda  d_ptr+1
548f 8dac54 1547:      sta  .get+2
5492 a901 1548:      lda  #1
5494 8de554 1549:      sta  .loc
5497 a000 1550:      ldy  #0
5499 a200 1551:      ldx  #0
549b 20c954 1552:      jsr  .up_str
549e 20d654 1553:      jsr  .up_sub
54a1      1554:
54a1 b9ffff 1555: .comp:      lda  $ffff,y
54a4 20e854 1556:      jsr  upcase
54a7 8de454 1557:      sta  .temp
54aa bdffff 1558: .get:      lda  $ffff,x
54ad 20e854 1559:      jsr  upcase
54b0 cde454 1560:      cmp  .temp
54b3 f00b 1561:      beq  .match
54b5 a201 1562: .nomatch:   ldx  #1

```

```

54b7 eee554 1563:      inc    .loc
54ba 20c954 1564:      jsr    .up_str
54bd 4ca154 1565:      jmp    .comp
54c0 20d654 1566: .match:      jsr    .up_sub
54c3 20c954 1567:      jsr    .up_str
54c6 4ca154 1568:      jmp    .comp
54c9      1569:
54c9 c8      1570: .up_str:      iny
54ca cce654 1571:      cpy    .s_lth
54cd 9006     1572:      bcc    .xup
54cf f004     1573:      beq    .xup
54d1 68      1574:      pla
54d2 68      1575:      pla
54d3 a900     1576:      lda    #0
54d5 60      1577: .xup:         rts
54d6      1578:
54d6 e8      1579: .up_sub:      inx
54d7 ece754 1580:      cpx    .d_lth
54da 9007     1581:      bcc    .xsub
54dc f005     1582:      beq    .xsub
54de 68      1583:      pla
54df 68      1584:      pla
54e0 ade554 1585:      lda    .loc
54e3 60      1586: .xsub:        rts
54e4      1587:
54e4 00      1588: .temp:        dc.b  0
54e5 00      1589: .loc:         dc.b  0
54e6 00      1590: .s_lth:       dc.b  0
54e7 00      1591: .d_lth:       dc.b  0
54e8      1592: ;
54e8      1593: ;
54e8      1594: ; Upcase
54e8      1595: ;
54e8      1596: ;
54e8 c961     1597: upcase:      cmp    #'a'
54ea 9009     1598:      bcc    .xupcase
54ec c97b     1599:      cmp    #'z'+1
54ee f005     1600:      beq    .xupcase
54f0 b003     1601:      bcs    .xupcase
54f2 38      1602:      sec
54f3 e920     1603:      sbc    #32
54f5 60      1604: .xupcase:    rts
54f6      1605:
54f6      1606:
54f6 c941     1607: locase:      cmp    #'A'
54f8 9009     1608:      bcc    .xlocase
54fa c95b     1609:      cmp    #'Z'+1
54fc f005     1610:      beq    .xlocase
54fe b003     1611:      bcs    .xlocase
5500 18      1612:      clc
5501 6920     1613:      adc    #32
5503 60      1614: .xlocase:    rts
5504      1615:

```

```

5504      1616:
5504      1617:
5504      1618: ;   Make a length-byte string from an ASCII ($9b termina
5504      1619: ;   ~~~~~
5504      1620:
5504 a000 1621: tostr:      ldy    #0
5506 b182 1622: .tfe:      lda    (s_ptr),y    ;find end o
5508 f007 1623:          beq    .gotend
550a c99b 1624:          cmp    #$9b
550c f003 1625:          beq    .gotend
550e c8   1626:          iny
550f d0f5 1627:          bne    .tfe
5511 8c2c55 1628: .gotend:   sty    .lth        ;save lengt
5514 c000 1629:          cpy    #0
5516 f00c 1630:          beq    .rtn
5518 a000 1631:          ldy    #0
551a b182 1632: .loop:     lda    (s_ptr),y
551c c8   1633:          iny
551d 9184 1634:          sta    (d_ptr),y
551f cc2c55 1635:          cpy    .lth
5522 90f6 1636:          bcc    .loop
5524 a000 1637: .rtn:      ldy    #0
5526 ad2c55 1638:          lda    .lth
5529 9184 1639:          sta    (d_ptr),y
552b 60   1640:          rts
552c      1641:
552c 01   1642: .lth:      dc.b    1
552d      1643:
552d      1644: ;
552d      1645: ;   Move string pointed to by s_ptr to string pointed
552d      1646: ;   ~~~~~
552d      1647:
552d      1648:
552d a000 1649: move_str:  ldy    #0
552f b182 1650:          lda    (s_ptr),y    ;ge
5531 a8   1651:          tay          ;go
5532 b182 1652: .loop:     lda    (s_ptr),y
5534 9184 1653:          sta    (d_ptr),y
5536 c000 1654:          cpy    #0
5538 f004 1655:          beq    .done
553a 88   1656:          dey
553b 4c3255 1657:          jmp    .loop
553e 60   1658: .done:     rts
553f      1659:
553f      1660:
553f      1661:
553f      1662: ;   Return the length of string at s_ptr in accumulato
553f      1663: ;   ~~~~~
553f      1664:
553f a000 1665: length:   ldy    #0
5541 b182 1666:          lda    (s_ptr),y
5543 60   1667:          rts
5544      1668:

```

```

5544      1669:
5544      1670: ;   Make an ASCII ($9b terminated) string out of a lth-b
5544      1671: ;   ~~~~~
5544      1672:
5544 a000 1673: toascii:      ldy  #0
5546 b182 1674:      lda  (s_ptr),y      ;ge
5548 8d6a55 1675:      sta  .lth
554b f019 1676:      beq  .null
554d c8 1677:      iny
554e ee6a55 1678:      inc  .lth
5551 b182 1679: .loop:      lda  (s_ptr),y
5553 88 1680:      dey
5554 9184 1681:      sta  (d_ptr),y
5556 c8 1682:      iny
5557 c8 1683:      iny
5558 cc6a55 1684:      cpy  .lth
555b d0f4 1685:      bne  .loop
555d ac6a55 1686: .done:      ldy  .lth
5560 88 1687:      dey
5561 a99b 1688: .rtn:      lda  #$9b
5563 9184 1689:      sta  (d_ptr),y
5565 60 1690:      rts
5566 a000 1691: .null:      ldy  #0
5568 f0f7 1692:      beq  .rtn
556a      1693:
556a 01 1694: .lth:      dc.b 1
556b      1695:
556b      1696:
556b      1697:
556b      1698: ;   Concatenate the string at s_ptr onto the end of st
556b      1699: ;   ~~~~~
556b      1700:
556b a000 1701: concat:      ldy  #0
556d b182 1702:      lda  (s_ptr),y      ;ge
556f 8dab55 1703:      sta  .srclth
5572 f02a 1704:      beq  .end          ;nu
5574 b184 1705:      lda  (d_ptr),y      ;ge
5576 8daa55 1706:      sta  .desty
5579 eea955 1707:      inc  .srcy
557c eeaa55 1708:      inc  .desty
557f a001 1709:      ldy  #1          ;al
5581 8ca955 1710:      sty  .srcy          ; t
5584 aca955 1711: .loop:      ldy  .srcy
5587 b182 1712:      lda  (s_ptr),y      ;ge
5589 acaa55 1713:      ldy  .desty
558c 9184 1714:      sta  (d_ptr),y      ;st
558e ada955 1715:      lda  .srcy
5591 cdab55 1716:      cmp  .srclth      ;en
5594 f008 1717:      beq  .end          ;ye
5596 eea955 1718:      inc  .srcy
5599 eeaa55 1719:      inc  .desty      ;up
559c d0e6 1720:      bne  .loop      ;an
559e a000 1721: .end:      ldy  #0

```

```
55a0 b184 1722:      lda  (d_ptr),y      ;up
55a2 18 1723:      clc              ; o
55a3 6dab55 1724:      adc  .srclth
55a6 9184 1725:      sta  (d_ptr),y
55a8 60 1726:      rts
55a9      1727:
55a9 01 1728: .srey:      dc.b 1
55aa 01 1729: .desty:      dc.b 1
55ab 01 1730: .srclth:     dc.b 1
55ac      1731:
55ac      1732:
55ac      1733:
55ac      1734: win_end:      ds.b 0
55ac      1735: ;
55ac      1736: ;
```

End assembly: no errors