


```

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: ;      Set_crc.com
0000 9: ;
0000 10: ; This is a program to to set a crc to make sure the
0000 11: ; byte has not changed.
0000 12: ;
0000 13: ;      init  3/08/2010      sjc new design
0000 14: ;
0000 15: ;
0000 16: ; Updated the screen Print Utility so it will work with the
0000 17: ; This change will also help "Print P:"      8/1
0000 18: ;
0000 19: ;
0000 20: ; Also added code for Control+1 pause of screen to the high
0000 21: ; handling.      8/24/2012      sjc
0000 22: ;
0000 23: ;      Last updated 8/26/2012 sjc
0000 24: ;
0000 25: ; ~~~~~
0000 26: ;-----
0000 27: ; Notes: o This source code MAY NOT be placed for download
0000 28: ;      o "mea" is a macro that loads the address of the
0000 29: ;      into the pointer specified by the second field.
0000 30: ;-----
0000 31: ; Assembler: MADMAC (tm) ST Cross Assembler (Atari Corp)
0000 32: ;      XASM  st and IBM VERSIONS
0000 33: ;-----
0000 34:
0000 35: com_file_name:      .macro
0000 36: ;      dc.b  13,"filename.ext",$9b
0000 37: ;      dc.b  12,"Set_crc.com ", $9b
0000 38: ;      .endm
0000 39:
0000 40: ;
0000 41: ; ~~~~~
0000 42: ; File revision history. Version Number in hex
0000 43: ;
0000 44: file_ver:      equ  $12
0000 45: ; ~~~~~
0000 46: ; the Month in dec for the first time this revision
0000 47: ; was compiled
0000 48: ;
0000 49: c_month:      equ  8
0000 50: ; ~~~~~
0000 51: ; the day in dec for the first time this revision
0000 52: ; was compiled
0000 53: c_day:      equ  28

```

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

```

0000 107: r..language:      equ   1
0000 108: ;~~~~~
0000 109: ;
0000 110: ;
0000 111: ;
0000 112: ;
0000 113: ;
0000 114: buff_1:              equ   $9fff-copy_buff
0000 115: ;
0000 116: ;
0000 117: ;
0000 118: ;
0000 119: eat_it_up:           .macro dest
0000 120:                      lda   %1
0000 121:                      sta   tom_mccomb
0000 122:                      lda   %1+1
0000 123:                      sta   tom_mccomb+1
0000 124:                      lda   %1+2
0000 125:                      sta   tom_mccomb+2
0000 126:                      lda   %1+3
0000 127:                      sta   tom_mccomb+3
0000 128:                      lda   %1+4
0000 129:                      sta   tom_mccomb+4
0000 130:                      lda   %1+5
0000 131:                      sta   tom_mccomb+5
0000 132:                      lda   %1+6
0000 133:                      sta   tom_mccomb+6
0000 134:                      lda   %1+7
0000 135:                      sta   tom_mccomb+7
0000 136:                      lda   %1+8
0000 137:                      sta   tom_mccomb+8
0000 138:                      jsr   kiss_it
0000 139:                      .ENDM
0000 140: ;
0000 141: nip:                 .macro dest
0000 142:                      dc.b  %1-$7000+6 / 256
0000 143:                      dc.b  0
0000 144:                      dc.b  %1-$7000+6 & $ff
0000 145:                      .ENDM
0000 146: ;
0000 147: ;
0000 148: ;
0000 149: ;
0000 150:                      .include equates
0000 151:                      .include globals
0000 152:                      .include macros
0000 153:
0000 154:                      .org  $4000
0000 155:
0000 156: win_start:
0000 157: header_info:
0000 158:                      .include header
42a7 159: readcli:

```

```

42a7      160:
42a7 a200  161:          ldx  #$00
42a9 a03f  162:          ldy  #LBUF
42ab b10a  163: .a:          lda  (comtab),y
42ad c920  164:          cmp  #' '
42af f00b  165:          beq  .gotspacechar
42b1 c99b  166:          cmp  #$9b
42b3 f004  167:          beq  .noparmerror
42b5      168:
42b5 c8    169:          iny
42b6 4cab42 170:          jmp  .a
42b9      171:
42b9      172:
42b9      173:
42b9 4c2947 174: .noparmerror: jmp  top ; based on info sto
42bc      175:
42bc a200  176: .gotspacechar: ldx  #$00
42be      177:
42be c8    178: .lop:       iny
42bf e8    179:          inx
42c0 b10a  180:          lda  (comtab),y
42c2 9da343 181:          sta  parm_1,x
42c5 8ea343 182:          stx  parm_1
42c8 c99b  183:          cmp  #$9b
42ca d0f2  184:          bne  .lop
42cc      185: ;
42cc      186: ;
42cc      187: ; Debug Routeen
42cc      188: ;
42cc 20284c 189:          jsr  cr_lf
42cf 20284c 190:          jsr  cr_lf
42d2      191:          mea  parm_1,pr_ptr
42d2 a9a3  ----:          lda  #low parm_1
42d4 8588  ----:          sta  pr_ptr
42d6 a943  ----:          lda  #high parm_1
42d8 8589  ----:          sta  pr_ptr + 1
42da 20c650 192:          jsr  echos
42dd 20284c 193:          jsr  cr_lf
42e0 20284c 194:          jsr  cr_lf
42e3      195: ;          rts
42e3      196:
42e3 4c2947 197:          jmp  top
42e6      198:
42e6      199:
42e6      200: ;-----
42e6      201: ;
42e6      202: ;this is the fix code.
42e6      203: ;for each of the fucked up
42e6      204: ;commands!
42e6      205: ;
42e6      206: ;
42e6      207: ;-----
42e6      208: ;

```

```

42e6      209: ;N_S.BYTE $60
42e6 a343b7 210: n_s_buf:      dc.w  parm_1,crc_sum
42ea 0200  211:      dc.w  2
42ec      212:      nip  $7007
42ec 00    ----:      dc.b  $7007-$7000+6 / 256
42ed 00    ----:      dc.b  0
42ee 0d    ----:      dc.b  $7007-$7000+6 & $ff
42ef      213:
42ef      214: ;-----
42ef      215: ;
42ef 00    216: ov_counter:      dc.b  0
42f0      217:
42f0      218: ;
42f0      219: ;
42f0      220: ;-----
42f0      221: outbuff:
42f0 000000 222:      dc.b 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
430d 000000 223:      dc.b 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
432a      224: ;
432a      225: ;
432a 00    226: work_out:      dc.b  0
432b 000000 227:      dc.w 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4367 000000 228:      dc.w 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
43a3      229: ;
43a3      230: ;
43a3      231: ;
43a3      232: parm_1:      ds.b  80
43f3      233:
43f3      234:
43f3 50    235: from_kbd:      dc.b  80
43f4      236:
43f4      237: tom_mccomb:
43f4 0000  238:      dc.w  0
43f6 0000  239:      dc.w  0
43f8 0000  240:      dc.w  0
43fa 0000  241:      dc.w  0
43fc 00    242:      dc.b  0
43fd 000000 243:      dc.b  0,0,0,0,0
4402      244:
4402      245:
4402      246:
4402      247: ;
4402      248: kiss_it:
4402      249:
4402      250:
4402      251:
4402 a220  252:      ldx  #disk2
4404 20614e 253:      jsr  close
4407      254:      mea  parm_1,s_ptr
4407 a9a3  ----:      lda  #low parm_1
4409 8582  ----:      sta  s_ptr
440b a943  ----:      lda  #high parm_1
440d 8583  ----:      sta  s_ptr + 1

```

```

440f      255:      mea  outbuff,d_ptr
440f a9f0 ----:      lda  #low outbuff
4411 8584 ----:      sta  d_ptr
4413 a942 ----:      lda  #high outbuff
4415 8585 ----:      sta  d_ptr + 1
4417 20694e 256:      jsr  move_str
441a      257:      mea  outbuff,s_ptr
441a a9f0 ----:      lda  #low outbuff
441c 8582 ----:      sta  s_ptr
441e a942 ----:      lda  #high outbuff
4420 8583 ----:      sta  s_ptr + 1
4422      258:      mea  work_out,d_ptr
4422 a92a ----:      lda  #low work_out
4424 8584 ----:      sta  d_ptr
4426 a943 ----:      lda  #high work_out
4428 8585 ----:      sta  d_ptr + 1
442a 207b4e 259:      jsr  toascii
442d      260:      mea  work_out,fn_ptr
442d a92a ----:      lda  #low work_out
442f 8586 ----:      sta  fn_ptr
4431 a943 ----:      lda  #high work_out
4433 8587 ----:      sta  fn_ptr + 1
4435 20164e 261:      jsr  open_in
4438 3003  262:      bmi  no_good
443a 4c4544 263:      jmp  version
443d      264: no_good:
443d a220  265:      ldx  #disk2
443f 20614e 266:      jsr  close
4442 4cce45 267:      jmp  youdo
4445      268: ;
4445      269: version:
4445 a900  270:      lda  #$00
4447 8d2e46 271:      sta  f.sector
444a 8d2f46 272:      sta  f.sector+1
444d a900  273:      lda  #$00
444f 8d3046 274:      sta  offset
4452 a220  275:      ldx  #disk2
4454 200d46 276:      jsr  point
4457      277:      mea  buffer2,pr_ptr
4457 a9a7 ----:      lda  #low buffer2
4459 8588 ----:      sta  pr_ptr
445b a944 ----:      lda  #high buffer2
445d 8589 ----:      sta  pr_ptr + 1
445f ad4b45 278:      lda  buf_2
4462 8dad46 279:      sta  lnptr
4465 ad4c45 280:      lda  buf_2+1
4468 8dae46 281:      sta  lnptr+1
446b a220  282:      ldx  #disk2
446d 20af46 283:      jsr  bget
4470 a220  284:      ldx  #disk2
4472 20614e 285:      jsr  close
4475      286: ;
4475 ada744 287:      lda  buffer2

```

```

4478 c9ff 288:      cmp    #$ff
447a d011 289:      bne    v_fuck
447c ada844 290:     lda    buffer2+1
447f c9ff 291:      cmp    #$ff
4481 d00a 292:      bne    v_fuck
4483 adad44 293:     lda    buffer3
4486 c94c 294:      cmp    #$4c
4488 d003 295:      bne    v_fuck
448a 4ca444 296:     jmp    verry1
448d      297: v_fuck:
448d 20684c 298:     jsr    printsi
4490 9b2049 299:     dc.b   $9b," Invalid header",$9B,$
44a3 60 300:      rts
44a4      301: verry1:
44a4 4c6b45 302:     jmp    ok_good
44a7      303:
44a7      304: ;=====
44a7      305: ;This is the header data structor
44a7      306: ;
44a7      307:      .include    Hd-data
454d      308:
454d      309:
454d 48 310: ver_num:      pha
454e 4a 311:      lsr
454f 4a 312:      lsr
4550 4a 313:      lsr
4551 4a 314:      lsr
4552 205d45 315:     jsr    l8913
4555 a92e 316:     lda    #'.'
4557 20be4c 317:     jsr    echo
455a 68 318:     pla
455b 290f 319:     and    #$0f
455d c90a 320: l8913:      cmp    #$0a
455f b004 321:     bcs    l891b
4561 0930 322:     ora    #$30
4563 d002 323:     bne    l891d
4565 6936 324: l891b:      adc    #$36
4567 20be4c 325: l891d:      jsr    echo
456a 60 326:      rts
456b      327:
456b      328:
456b      329: ok_good:
456b      330: Tammy:
456b a220 331:     ldx    #disk2
456d 20614e 332:     jsr    close
4570      333:     mea    work_out,fn_ptr
4570 a92a ----:     lda    #low work_out
4572 8586 ----:     sta    fn_ptr
4574 a943 ----:     lda    #high work_out
4576 8587 ----:     sta    fn_ptr + 1
4578 a220 334:     ldx    #disk2
457a 20254e 335:     jsr    open_updt
457d adfa43 336:     lda    tom_mccomb+6 ; 7

```



```

4580 8d2e46 337:          sta  f.sector
4583 adfb43 338:          lda  tom_mccomb+7 ; 8
4586 8d2f46 339:          sta  f.sector+1
4589 adfc43 340:          lda  tom_mccomb+8 ; 9
458c 8d3046 341:          sta  offset
458f a220 342:          ldx  #disk2
4591      343:
4591      344:
4591 200d46 345:          jsr  point
4594 adf643 346:          lda  tom_mccomb+2 ;3
4597 8588 347:          sta  pr_ptr
4599 adf743 348:          lda  tom_mccomb+3 ;4
459c 8589 349:          sta  pr_ptr+1
459e adf843 350:          lda  tom_mccomb+4 ;5
45a1 8dad46 351:          sta  lnptr
45a4 adf943 352:          lda  tom_mccomb+5 ;6
45a7 8dae46 353:          sta  lnptr+1
45aa a220 354:          ldx  #disk2
45ac 20d846 355:          jsr  bput
45af a220 356:          ldx  #disk2
45b1 20614e 357:          jsr  close
45b4      358:
45b4 20684c 359:          jsr  printsi
45b7 9b2043 360:          dc.b  $9b," Command CRC updated",
45ce      361: youdo:
45ce 60 362:          rts
45cf      363:
45cf      364: ;
45cf      365: ;-----
45cf 0000 366: aux:    dc.b 0,0
45d1      367: ;-----
45d1      368: ;you must jsr to this vector for point, note, file lenght
45d1      369: ;
45d1 44783a 370: pnfddata:  dc.b  "Dx:",$00
45d5 0000 371: pnfddata1: dc.b  $00,$00
45d7      372:
45d7      373: pnf_work:
45d7 8ed545 374:          stx  pnfddata1
45da a586 375:          lda  fn_ptr
45dc 8dea45 376:          sta  pnf_1+1
45df a587 377:          lda  fn_ptr+1
45e1 8deb45 378:          sta  pnf_1+2
45e4 8cd645 379:          sty  pnfddata1+1
45e7 a001 380:          ldy  #$01
45e9 b9ffff 381: pnf_1:    lda  $ffff,y
45ec 8dd245 382:          sta  pnfddata+1
45ef acd645 383:          ldy  pnfddata1+1
45f2 aed545 384:          ldx  pnfddata1
45f5 a900 385:          lda  #$00
45f7 9d4a03 386:          sta  $034a,x
45fa a900 387:          lda  #$00
45fc 9d4b03 388:          sta  $034b,x
45ff a9d1 389:          lda  # low pnfddata

```

```

4601 9d4403 390:          sta  $0344,x
4604 a945 391:          lda  # high pnfdata
4606 9d4503 392:          sta  $0345,x
4609 20fd46 393:          jsr  c_iiov
460c 60 394:          rts
460d 395: ;
460d 396: ;
460d 397: point: ;      XIO 37 POINT
460d a925 398:          lda  #$25
460f 8ea546 399:          stx  xio_39_hold
4612 9d4203 400:          sta  $0342,x
4615 ad3046 401:          lda  offset
4618 9d4c03 402:          sta  $034c,x
461b ad2e46 403:          lda  f.sector
461e 9d4d03 404:          sta  $034d,x
4621 ad2f46 405:          lda  f.sector+1
4624 9d4e03 406:          sta  $034e,x
4627 20d745 407:          jsr  pnf_work
462a aea546 408:          ldx  xio_39_hold
462d 60 409:          rts
462e 0000 410: f.sector:      dc.w 0
4630 00 411: offset:      dc.b 0
4631 412: note: ;      XIO_38 NOTE
4631 8ea546 413:          stx  xio_39_hold
4634 a926 414:          lda  #$26
4636 9d4203 415:          sta  $0342,x
4639 20d745 416:          jsr  pnf_work
463c aea546 417:          ldx  xio_39_hold
463f bd4c03 418:          lda  $034c,x
4642 8d3046 419:          sta  offset
4645 bd4d03 420:          lda  $034d,x
4648 8d2e46 421:          sta  f.sector
464b bd4e03 422:          lda  $034e,x
464e 8d2f46 423:          sta  f.sector+1
4651 60 424: xio_38e:      rts
4652 000000 425: npsave1:      dc.b 0,0,0
4655 426: npsave:
4655 ad3046 427:          lda  offset
4658 8d5246 428:          sta  npsave1
465b ad2e46 429:          lda  f.sector
465e 8d5346 430:          sta  npsave1+1
4661 ad2f46 431:          lda  f.sector+1
4664 8d5446 432:          sta  npsave1+2
4667 60 433:          rts
4668 434: npload:
4668 ad5246 435:          lda  npsave1
466b 8d3046 436:          sta  offset
466e ad5346 437:          lda  npsave1+1
4671 8d2e46 438:          sta  f.sector
4674 ad5446 439:          lda  npsave1+2
4677 8d2f46 440:          sta  f.sector+1
467a 60 441:          rts
467b 442: ;

```

```

467b      443: f_length: ;      XIO#39 FILE LENGTH SPARTA ONLY
467b 8ea546 444:      stx   xio_39_hold
467e a927  445:      lda   #$27
4680 9d4203 446:      sta   $0342,x
4683 20d745 447:      jsr   pnf_work
4686 aea546 448:      ldx   xio_39_hold
4689 bd4c03 449:      lda   $034c,x
468c 8da746 450:      sta   fnlng
468f 8daa46 451:      sta   f_lng
4692 bd4d03 452:      lda   $034d,x
4695 8da846 453:      sta   fnlng+1
4698 8dab46 454:      sta   f_lng+1
469b bd4e03 455:      lda   $034e,x
469e 8da946 456:      sta   fnlng+2
46a1 8dac46 457:      sta   f_lng+2
46a4 60    458:      rts
46a5 0000   459: xio_39_hold: dc.b 0,0
46a7 000000 460: fnlng:   dc.b 0,0,0
46aa 000000 461: f_lng:   dc.b 0,0,0
46ad      462: ;FN_PTR.WORD 0
46ad 0000   463: lnptr:   dc.w 0
46af      464: ;
46af      465: ;-----;
46af      466: ; lda PR_PTR ; BUFFER
46af      467: ; sta #<BUFFER+1
46af      468: ;
46af      469: bget:
46af a907   470:      lda   #$07
46b1 9d4203 471:      sta   $0342,x
46b4 a588   472:      lda   pr_ptr
46b6 9d4403 473:      sta   $0344,x
46b9 a589   474:      lda   pr_ptr+1
46bb 9d4503 475:      sta   $0345,x
46be adad46 476:      lda   lnptr
46c1 9d4803 477:      sta   $0348,x
46c4 adae46 478:      lda   lnptr+1
46c7 9d4903 479:      sta   $0349,x
46ca a903   480:      lda   #$03
46cc 9d4a03 481:      sta   $034a,x
46cf a900   482:      lda   #$00
46d1 9d4b03 483:      sta   $034b,x
46d4 20fd46 484:      jsr   c_iiov
46d7 60    485:      rts
46d8      486: ;-----
46d8      487: ; PR_PTR=BUFFER FN_PTR=LEINGTH OF BUFFER
46d8      488: ;-----
46d8      489: bput:
46d8 a90b   490:      lda   #$0b
46da 9d4203 491:      sta   $0342,x
46dd a588   492:      lda   pr_ptr
46df 9d4403 493:      sta   $0344,x
46e2 a589   494:      lda   pr_ptr+1
46e4 9d4503 495:      sta   $0345,x

```

```

46e7 adad46 496:      lda  lnptr
46ea 9d4803 497:      sta  $0348,x
46ed adae46 498:      lda  lnptr+1
46f0 9d4903 499:      sta  $0349,x
46f3 a903 500:      lda  #$03
46f5 9d4a03 501:      sta  $034a,x
46f8 a900 502:      lda  #$00
46fa 9d4b03 503:      sta  $034b,x
46fd      504: c_iov:
46fd 2056e4 505:      jsr  $e456
4700 ad0303 506:      lda  $0303
4703 a8 507:      tay
4704 c081 508:      cpy  #$81
4706 9003 509:      bcc  c_iov_exit
4708 200d47 510:      jsr  drive_error
470b 60 511: c_iov_exit:  rts
470c 60 512:      rts
470d      513: drive_error:
470d 48 514:      pha
470e 20684c 515:      jsr  printsi
4711 9b2043 516:      dc.b  $9b," Ciov Error # ", $FF
4721 68 517:      pla
4722 20a24e 518:      jsr  pr_byte
4725 20284c 519:      jsr  cr_lf
4728 60 520:      rts
4729      521: ;
4729      522: ;-----
4729      523: ; main body of the source
4729      524: ; code.
4729      525: ; to Tom mcomcomb and
4729      526: ; @ircular Logic!
4729      527: ;
4729      528: ;-----
4729      529: ;
4729      530: top:
4729      531:
4729 ada343 532:      lda  parm_1
472c c904 533:      cmp  #$04
472e b030 534:      bcs  ok_
4730 20684c 535:      jsr  printsi
4733 9b9b9b 536:      dc.b  $9b,$9b,$9b," Missing Dx:>P
4749 204e6f 537:      dc.b  " No Filename Needed",$9B,$
475f 60 538:      rts
4760      539: ok_:
4760 ad484b 540:      lda  u_node
4763 ae494b 541:      ldx  u_node+1
4766 8d4a4b 542:      sta  d_node
4769 8e4b4b 543:      stx  d_node+1
476c      544: ;      lda  ov_counter
476c      545: ;      cmp  #$01
476c      546: ;      bcc  .aa
476c      547: ;      jmp  .not
476c      548: ;.aa:

```

```

476c 549: ;          lda  from_kbd
476c 550: ;          cmp  $80
476c 551: ;          beq  .not1
476c 552: ;          jmp  .not
476c 553: ;
476c 554: ;
476c 555: ;.not1:          jsr  printsi
476c 556: ;          dc.b  $9b,$9b," Enter Node Number
476c 557: ;          jsr  get_card
476c 558: ;          sta  d_node
476c 559: ;          stx  d_node+1
476c 560: ;.not:
476c 561: ;
476c 562: ;
476c 563: ;
476c 564: ;
476c 565: ;      Copy the file in source_path/source_file to dest_p
476c 566: ;      -----
476c 567: copy_file:
476c 568: ;
476c 569:          mea  parm_1,s_ptr
476c a9a3 ----:      lda  #low parm_1
476e 8582 ----:      sta  s_ptr
4770 a943 ----:      lda  #high parm_1
4772 8583 ----:      sta  s_ptr + 1
4774 570:          mea  _infile,d_ptr
4774 a9e5 ----:      lda  #low _infile
4776 8584 ----:      sta  d_ptr
4778 a947 ----:      lda  #high _infile
477a 8585 ----:      sta  d_ptr + 1
477c 207b4e 571:      jsr  toascii
477f 572: ; mea  PARM_2,S_PTR
477f 573: ; mea  _OUTFILE,D_PTR
477f 574: ; jsr  TOASCII
477f 575: ;
477f 576:          mea  _infile,fn_ptr
477f a9e5 ----:      lda  #low _infile
4781 8586 ----:      sta  fn_ptr
4783 a947 ----:      lda  #high _infile
4785 8587 ----:      sta  fn_ptr + 1
4787 a220 577:          ldx  #disk2
4789 20164e 578:      jsr  open_in
478c 300f 579:          bmi  _shutdown
478e 580: _srcok:
478e 581: ; mea  _OUTFILE,FN_PTR
478e 582: ; LDX #$40 ;use #4 as target of copy
478e 583: ; jsr  OPEN_OUT
478e 584: ; bmi  _SHUTDOWN
478e 585: _copyloop:
478e a220 586:          ldx  #disk2
4790 20c14a 587:      jsr  blockread
4793 0060 588:          dc.w  copy_buff
4795 179c 589:          dc.w  buff_1

```

```

4797 1014 590:          bpl  _readok
4799 c088 591:          cpy  #136
479b f010 592:          beq  _readok
479d 8ce447 593: _shutdown:      sty  _savey          ;co
47a0 a220 594:          ldx  #disk2          ; b
47a2 20614e 595:          jsr  close
47a5      596: ; LDX #$40
47a5      597: ; jsr  CLOSE
47a5 ace447 598:          ldy  _savey
47a8      599: ; jsr  ECHOSI
47a8 4cd24b 600:          jmp  next_run
47ab 60    601:          rts
47ac      602: ;
47ac 00    603: first_call:      dc.b  0
47ad      604: ;
47ad      605: _readok:
47ad adac47 606:          lda  first_call
47b0 c900 607:          cmp  #$00
47b2 d006 608:          bne  cly
47b4 203b4b 609:          jsr  set_pass
47b7 eeac47 610:          inc  first_call
47ba      611: ;
47ba      612: cly:
47ba a220 613:          ldx  #disk2
47bc bd4803 614:          lda  icbll,x
47bf 8ddc47 615:          sta  A_totbytes
47c2 bd4903 616:          lda  icblh,x
47c5 8ddd47 617:          sta  A_totbytes+1
47c8      618: ;
47c8 20504b 619:          jsr  wipe
47cb      620:
47cb      621: ; LDX #$40
47cb      622: ; jsr  BLOCKWRITE ;pretty sleazy, huh?
47cb      623: ; .WORD COPY_BUFF ;I love self-modifying
47cb      624: ;A_totbytes.WORD BUFF_L ; code__
47cb      625: ; bmi  _SHUTDOWN ;damn___write error!
47cb      626:
47cb addc47 627:          lda  A_totbytes
47ce c917 628:          cmp  #buff_l & $ff ;did we read
47d0 d00c 629:          bne  _done ;nope, must be done
47d2 addd47 630:          lda  A_totbytes+1
47d5 c99c 631:          cmp  #buff_l / 256
47d7 d005 632:          bne  _done
47d9 4c8e47 633:          jmp  _copyloop
47dc      634: ;
47dc 179c 635: A_totbytes:      dc.w  buff_l
47de      636:
47de a001 637: _done:          ldy  #1
47e0 4c9d47 638:          jmp  _shutdown
47e3 00    639: _idx:          dc.b  0
47e4 00    640: _savey:        dc.b  0
47e5      641: _infile:
47e5 000000 642:          dc.b  0,0,0,0,0,0,0,0,0

```

```

47ef 000000 643:          dc.b  0,0,0,0,0,0,0,0,0
47f9 000000 644:          dc.b  0,0,0,0,0,0,0,0,0
4803 000000 645:          dc.b  0,0,0,0,0,0,0,0,0
480d 000000 646:          dc.b  0,0,0,0,0,0,0,0,0
4817 000000 647:          dc.b  0,0,0,0,0,0,0,0,0
4821 000000 648:          dc.b  0,0,0,0,0,0,0,0,0
482b 000000 649:          dc.b  0,0,0,0,0,0,0,0,0
4835      650: _outfile:
4835 000000 651:          dc.b  0,0,0,0,0,0,0,0,0
483f 000000 652:          dc.b  0,0,0,0,0,0,0,0,0
4849 000000 653:          dc.b  0,0,0,0,0,0,0,0,0
4853 000000 654:          dc.b  0,0,0,0,0,0,0,0,0
485d 000000 655:          dc.b  0,0,0,0,0,0,0,0,0
4867 000000 656:          dc.b  0,0,0,0,0,0,0,0,0
4871 000000 657:          dc.b  0,0,0,0,0,0,0,0,0
487b 000000 658:          dc.b  0,0,0,0,0,0,0,0,0
4885      659:
4885      660: ;   Expand wildcarded name to create an output filename
4885      661: ;   Input is source_file, dest_file; output is out_name
4885      662: ;   -----
4885      663: ;
4885      664: ;
4885      665: ;
4885      666: put256_out:
4885 a000 667:          ldy  #0
4887 8cb74a 668:          sty  crc_sum
488a 8cb84a 669:          sty  crc_sum+1
488d 8cb94a 670:          sty  chk_sum
4890      671: ;
4890      672: crc_loop:
4890 b188 673:          lda  (pr_ptr),y
4892 48 674:          pha
4893 18 675:          clc
4894 6db94a 676:          adc  chk_sum ;calc standard check
4897 8db94a 677:          sta  chk_sum
489a 68 678:          pla
489b 4db84a 679:          eor  crc_sum+1 ;now do the CRC check
489e aa 680:          tax
489f bdb149 681:          lda  crc_hitable,x
48a2 4db74a 682:          eor  crc_sum
48a5 8db84a 683:          sta  crc_sum+1
48a8 bdb148 684:          lda  crc_lutable,x
48ab 4900 685:          eor  #0
48ad 8db74a 686:          sta  crc_sum
48b0 60 687:          rts
48b1      688: ;
48b1      689: ;
48b1      690: ;
48b1      691: crc_lutable:
48b1 002142 692:          dc.b  $00,$21,$42,$63,$84,$a5,$c6,$e
48bb 4a6b8c 693:          dc.b  $4a,$6b,$8c,$ad,$ce,$ef,$31,$1
48c5 b594f7 694:          dc.b  $b5,$94,$f7,$d6,$39,$18,$7b,$5
48cf ffe62 695:          dc.b  $ff,$de,$62,$43,$20,$01,$e6,$c

```

```

48d9 6a4b28 696:      dc.b $6a,$4b,$28,$09,$ee,$cf,$ac,$8
48e3 1130d7 697:      dc.b $11,$30,$d7,$f6,$95,$b4,$5b,$7
48ed dffe9d 698:      dc.b $df,$fe,$9d,$bc,$c4,$e5,$86,$a
48f7 0223cc 699:      dc.b $02,$23,$cc,$ed,$8e,$af,$48,$6
4901 f5d4b7 700:      dc.b $f5,$d4,$b7,$96,$71,$50,$33,$1
490b bf9e79 701:      dc.b $bf,$9e,$79,$58,$3b,$1a,$a6,$8
4915 220360 702:      dc.b $22,$03,$60,$41,$ae,$8f,$ec,$c
491f 684997 703:      dc.b $68,$49,$97,$b6,$d5,$f4,$13,$3
4929 9fbedd 704:      dc.b $9f,$be,$dd,$fc,$1b,$3a,$59,$7
4933 caeb0c 705:      dc.b $ca,$eb,$0c,$2d,$4e,$6f,$80,$a
493d 042546 706:      dc.b $04,$25,$46,$67,$b9,$98,$fb,$d
4947 7f5eb1 707:      dc.b $7f,$5e,$b1,$90,$f3,$d2,$35,$1
4951 eacba8 708:      dc.b $ea,$cb,$a8,$89,$6e,$4f,$2c,$0
495b a08166 709:      dc.b $a0,$81,$66,$47,$24,$05,$db,$f
4965 5f7e1d 710:      dc.b $5f,$7e,$1d,$3c,$d3,$f2,$91,$b
496f 15344c 711:      dc.b $15,$34,$4c,$6d,$0e,$2f,$c8,$e
4979 446506 712:      dc.b $44,$65,$06,$27,$c0,$e1,$82,$a
4983 3f1ef9 713:      dc.b $3f,$1e,$f9,$d8,$bb,$9a,$75,$5
498d f1d0b3 714:      dc.b $f1,$d0,$b3,$92,$2e,$0f,$6c,$4
4997 e8c926 715:      dc.b $e8,$c9,$26,$07,$64,$45,$a2,$8
49a1 1f3e5d 716:      dc.b $1f,$3e,$5d,$7c,$9b,$ba,$d9,$f
49ab 557493 717:      dc.b $55,$74,$93,$b2,$d1,$f0
49b1      718: crc_hitable:
49b1 001020 719:      dc.b $00,$10,$20,$30,$40,$50,$60,$7
49be d1e1f1 720:      dc.b $d1,$e1,$f1,$12,$02,$32,$22,$5
49cb b3a3d3 721:      dc.b $b3,$a3,$d3,$c3,$f3,$e3,$24,$3
49d8 54a5b5 722:      dc.b $54,$a5,$b5,$85,$95,$e5,$f5,$c
49e5 766656 723:      dc.b $76,$66,$56,$46,$b7,$a7,$97,$8
49f2 586878 724:      dc.b $58,$68,$78,$08,$18,$28,$38,$c
49ff a9b95a 725:      dc.b $a9,$b9,$5a,$4a,$7a,$6a,$1a,$0
4a0c eb9b8b 726:      dc.b $eb,$9b,$8b,$bb,$ab,$6c,$7c,$4
4a19 edfdcd 727:      dc.b $ed,$fd,$cd,$dd,$ad,$bd,$8d,$9
4a26 2e1e0e 728:      dc.b $2e,$1e,$0e,$ff,$ef,$df,$cf,$b
4a33 b1a1d1 729:      dc.b $b1,$a1,$d1,$c1,$f1,$e1,$10,$0
4a40 608393 730:      dc.b $60,$83,$93,$a3,$b3,$c3,$d3,$e
4a4d 425262 731:      dc.b $42,$52,$62,$72,$b5,$a5,$95,$8
4a5a 241404 732:      dc.b $24,$14,$04,$74,$64,$54,$44,$a
4a67 c7d726 733:      dc.b $c7,$d7,$26,$36,$06,$16,$66,$7
4a74 e99989 734:      dc.b $e9,$99,$89,$b9,$a9,$58,$48,$7
4a81 cbdbeb 735:      dc.b $cb,$db,$eb,$fb,$8b,$9b,$ab,$b
4a8e 1a2a3a 736:      dc.b $1a,$2a,$3a,$fd,$ed,$dd,$cd,$b
4a9b 5c4c3c 737:      dc.b $5c,$4c,$3c,$2c,$1c,$0c,$ef,$f
4aa8 9f6e7e 738:      dc.b $9f,$6e,$7e,$4e,$5e,$2e,$3e,$0
4ab1      739: ;
4ab1      740: ;
4ab1      741: ;
4ab1      742: ;
4ab1 00    743: cksum_lo:      dc.b 0
4ab2 00    744: cksum_hi:      dc.b 0
4ab3 0000 745: crcsize:      dc.w 0
4ab5 0000 746: crctemp:      dc.w 0
4ab7 0000 747: crc_sum:      dc.w 0
4ab9 0000 748: chk_sum:      dc.w 0

```



```

4abb      749: ;
4abb 0000 750: real_sum:      dc.w 0 ; real crc check sum
4abd 0000 751: n_split:      dc.w 0 ; split of node numbe
4abf 0000 752: n_split1:     dc.w 0 ; split of node numbe
4ac1      753: ;
4ac1      754: ;
4ac1      755: ;   Blockread from disk channel in X register.
4ac1      756: ;
4ac1      757: ;   call: jsr    blockread
4ac1      758: ;           dc.w  buffer_address
4ac1      759: ;           dc.w  number_of_bytes
4ac1      760: ;   -----
4ac1      761: ;
4ac1      762: ;
4ac1 a907 763: blockread:      lda  #ccgetch
4ac3 9d4203 764:           sta  iccom,x
4ac6 4cd14a 765:           jmp  blockio
4ac9 a90b 766: blockwrite:     lda  #ccputch
4acb 9d4203 767:           sta  iccom,x
4ace 4cd14a 768:           jmp  blockio
4ad1 68 769: blockio:       pla
4ad2 8588 770:           sta  pr_ptr
4ad4 68 771:           pla
4ad5 8589 772:           sta  pr_ptr+1
4ad7 e688 773:           inc  pr_ptr
4ad9 d002 774:           bne  Anovf
4adb e689 775:           inc  pr_ptr+1
4add a000 776: Anovf:         ldy  #0
4adf b188 777:           lda  (pr_ptr),y
4ae1 9d4403 778:           sta  icbal,x
4ae4 c8 779:           iny
4ae5 b188 780:           lda  (pr_ptr),y
4ae7 9d4503 781:           sta  icbah,x
4aea c8 782:           iny
4aeb b188 783:           lda  (pr_ptr),y
4aed 9d4803 784:           sta  icbll,x
4af0 c8 785:           iny
4af1 b188 786:           lda  (pr_ptr),y
4af3 9d4903 787:           sta  icblh,x
4af6 a588 788:           lda  pr_ptr
4af8 18 789:           clc
4af9 6903 790:           adc  #3
4afb 8588 791:           sta  pr_ptr
4afd a589 792:           lda  pr_ptr+1
4aff 6900 793:           adc  #0
4b01 48 794:           pha
4b02 a588 795:           lda  pr_ptr
4b04 48 796:           pha
4b05 bd4803 797:           lda  icbll,x ;make sure not requ
4b08 1d4903 798:           ora  icblh,x ; zero bytes!
4b0b f003 799:           beq  Aoops
4b0d 4c56e4 800:           jmp  ciov
4b10 a001 801: Aoops:         ldy #1 ;signal success and exi

```

```

4b12 60    802:                rts
4b13      803: ;-----
4b13      804: ;  ERROR NUMBER
4b13      805: ;-----
4b13      806: ;
4b13 00    807: h_cardd:        dc.b  $00
4b14      808: ;
4b14      809: ; print a hex word value
4b14      810: ; low byte in a reg x is high byte
4b14      811: ;
4b14 8d134b 812: h_card:        sta  h_cardd
4b17 8a    813:                txa
4b18 20224b 814:                jsr  d_error
4b1b ad134b 815:                lda  h_cardd
4b1e 20224b 816:                jsr  d_error
4b21 60    817:                rts
4b22      818: ;
4b22      819: ;
4b22 48    820: d_error:        pha
4b23 4a    821:                lsr
4b24 4a    822:                lsr
4b25 4a    823:                lsr
4b26 4a    824:                lsr
4b27 202d4b 825:                jsr  t8913
4b2a 68    826:                pla
4b2b 290f   827:                and  #$0f
4b2d c90a   828: t8913:        cmp  #$0a
4b2f b004   829:                bcs  t891b
4b31 0930   830:                ora  #$30
4b33 d002   831:                bne  t891d
4b35 6936   832: t891b:        adc  #$36
4b37 20be4c 833: t891d:        jsr  echo
4b3a 60    834:                rts
4b3b      835: ;
4b3b      836: set_pass:
4b3b ad4a4b 837:                lda  d_node
4b3e 8d0d60 838:                sta  copy_buff+13
4b41 ad4b4b 839:                lda  d_node+1
4b44 8d0e60 840:                sta  copy_buff+14
4b47 60    841:                rts
4b48      842: ;
4b48      843: ;.byte ff ff 00 00 00 00 4c 00 00
4b48      844: ; ver month day year
4b48      845: ; .word 0
4b48      846: ;
4b48 1f00   847: u_node:        dc.w  l_dos_build
4b4a      848: ;
4b4a 0000   849: d_node:        dc.w  0
4b4c      850: ;
4b4c 0000   851: test_crc:      dc.w  0
4b4e      852: ;
4b4e 0000   853: block_w:      dc.w  0
4b50      854: ;

```

```

4b50      855:
4b50      856: wipe:
4b50      857:          mea    copy_buff,pr_ptr
4b50 a900 ----:      lda    #low copy_buff
4b52 8588 ----:      sta    pr_ptr
4b54 a960 ----:      lda    #high copy_buff
4b56 8589 ----:      sta    pr_ptr + 1
4b58      858:          mea    copy_buff,fr0
4b58 a900 ----:      lda    #low copy_buff
4b5a 85d4 ----:      sta    fr0
4b5c a960 ----:      lda    #high copy_buff
4b5e 85d5 ----:      sta    fr0 + 1
4b60 20aad9 859:      jsr    ifp
4b63 20b6dd 860:      jsr    fmove
4b66 addc47 861:      lda    A_totbytes
4b69 85d4 862:      sta    fr0
4b6b addd47 863:      lda    A_totbytes+1
4b6e 85d5 864:      sta    fr0+1
4b70 20aad9 865:      jsr    ifp
4b73 2066da 866:      jsr    fadd
4b76 20d2d9 867:      jsr    fpi
4b79      868: ;
4b79 a5d4 869:      lda    fr0
4b7b 8d4e4b 870:      sta    block_w
4b7e a5d5 871:      lda    fr0+1
4b80 8d4f4b 872:      sta    block_w+1
4b83      873: ;
4b83 20684c 874:      jsr    printsi
4b86 9b2053 875:      dc.b    $9b," START $",-1
4b90 a900 876:      lda    # low copy_buff
4b92 a260 877:      ldx    # high copy_buff
4b94 20144b 878:      jsr    h_card
4b97 20684c 879:      jsr    printsi
4b9a 20454e 880:      dc.b    " END $",-1
4ba1 ad4e4b 881:      lda    block_w
4ba4 ae4f4b 882:      ldx    block_w+1
4ba7 20144b 883:      jsr    h_card
4baa      884: ; RTS
4baa 20284c 885:      jsr    cr_lf
4bad      886: ;
4bad      887: ; jsr    GET_KEY
4bad      888: ;
4bad a000 889:      ldy    #$00
4baf      890: ;
4baf      891: ; $FFFF
4baf b188 892: wp1:      lda    (pr_ptr),y
4bb1      893: ; jsr    ECHO
4bb1 209048 894:      jsr    crc_loop
4bb4 c8 895:      iny
4bb5 c000 896:      cpy    #$00
4bb7 d0f6 897:      bne    wp1
4bb9 e689 898:      inc    pr_ptr+1 ;WP1+2
4bbb a000 899:      ldy    #$00

```

```

4bbd a589 900:          lda  pr_ptr+1 ;WP1+2
4bbf cd4f4b 901:          cmp  block_w+1
4bc2 90eb 902:          bcc  wp1
4bc4 a000 903:          ldy  #$00
4bc6 b188 904: wp2:      lda  (pr_ptr),y
4bc8      905: ; jsr    ECHO
4bc8 209048 906:          jsr  crc_loop
4bcb c8 907:          iny
4bcc cc4e4b 908:          cpy  block_w
4bcf 90f5 909:          bcc  wp2
4bd1 60 910:          rts
4bd2      911: ;
4bd2      912: next_run:
4bd2 20684c 913:          jsr  printsi
4bd5 9b9b20 914:          dc.b  $9b,$9b," Crc =",-1
4bdf adb74a 915:          lda  crc_sum
4be2 aeb84a 916:          ldx  crc_sum+1
4be5 20144b 917:          jsr  h_card
4be8 20284c 918:          jsr  cr_lf
4beb 20284c 919:          jsr  cr_lf
4bee      920: ; RTS
4bee      921: ;
4bee      922: ;
4bee      923: fix_elite:
4bee      924:          eat_it_up n_s_buf
4bee ade642 ----:          lda  n_s_buf
4bf1 8df443 ----:          sta  tom_mccomb
4bf4 ade742 ----:          lda  n_s_buf+1
4bf7 8df543 ----:          sta  tom_mccomb+1
4bfa ade842 ----:          lda  n_s_buf+2
4bfd 8df643 ----:          sta  tom_mccomb+2
4c00 ade942 ----:          lda  n_s_buf+3
4c03 8df743 ----:          sta  tom_mccomb+3
4c06 adea42 ----:          lda  n_s_buf+4
4c09 8df843 ----:          sta  tom_mccomb+4
4c0c adeb42 ----:          lda  n_s_buf+5
4c0f 8df943 ----:          sta  tom_mccomb+5
4c12 adec42 ----:          lda  n_s_buf+6
4c15 8dfa43 ----:          sta  tom_mccomb+6
4c18 aded42 ----:          lda  n_s_buf+7
4c1b 8dfb43 ----:          sta  tom_mccomb+7
4c1e adee42 ----:          lda  n_s_buf+8
4c21 8dfc43 ----:          sta  tom_mccomb+8
4c24 200244 ----:          jsr  kiss_it
4c27 60 925:          rts
4c28      926: ;
4c28      927: ;-----
4c28      928: ;This does a c/r to the screen
4c28      929: ;-----
4c28      930:
4c28 a99b 931: cr_lf:      lda  #$9b
4c2a 20be4c 932:          jsr  echo
4c2d 60 933:          rts

```

```

4c2e 934:
4c2e 935:
4c2e 936:
4c2e 937: ;-----
4c2e 938: ;CHANGES THE CHARTER TO UPCASE
4c2e 939: ;-----
4c2e 940: upcase
4c2e c961 941:      cmp  #$61
4c30 9006 942:      bcc  .a
4c32 c97a 943:      cmp  #$7a
4c34 b002 944:      bcs  .a
4c36 e91f 945:      sbc  #$1f
4c38 60   946: .a:      rts
4c39 947:
4c39 948: ;
4c39 949: ;-----
4c39 950: ;this one make's a nice tone
4c39 951: ;-----
4c39 952: bell
4c39 8d654c 953:      sta  .data
4c3c 8c664c 954:      sty  .data+1
4c3f 8e674c 955:      stx  .data+2
4c42 a0af 956:      ldy  #$af
4c44 a914 957:      lda  #$14
4c46 8d00d2 958:      sta  $d200
4c49 8c01d2 959: .a:      sty  $d201
4c4c a900 960:      lda  #$00
4c4e 8514 961:      sta  $14
4c50 a514 962: .b:      lda  $14
4c52 c902 963:      cmp  #$02
4c54 90fa 964:      bcc  .b
4c56 88   965:      dey
4c57 c0a0 966:      cpy  #$a0
4c59 b0ee 967:      bcs  .a
4c5b ad654c 968:      lda  .data
4c5e ac664c 969:      ldy  .data+1
4c61 ae674c 970:      ldx  .data+2
4c64 60   971:      rts
4c65 000000 972: .data:      dc.b  $00,$00,$00
4c68 973: ;
4c68 974: ;
4c68 975: ;
4c68 976: ;-----
4c68 977: ; jsr prints
4c68 978: ;this prints an inline string
4c68 979: ; terminated by $ff
4c68 980: ;-----
4c68 981: ; jsr prints
4c68 982: ; dc.b "print this",$9B,$FF
4c68 983: echosi:
4c68 68   984: prints:      pla
4c69 8d794c 985:      sta  pstr+1
4c6c 68   986:      pla

```

```

4c6d 8d7a4c 987:          sta  pstr+2
4c70 ee794c 988: prsl:      inc  pstr+1
4c73 d003  989:          bne  pstr
4c75 ee7a4c 990:          inc  pstr+2
4c78 adffff 991: pstr:      lda  $ffff
4c7b c9ff  992:          cmp  #$ff
4c7d f006  993:          beq  estri
4c7f 20be4c 994:          jsr  fast_output
4c82 4c704c 995:          jmp  prsl
4c85 ad7a4c 996: estri:     lda  pstr+2
4c88 48  997:          pha
4c89 ad794c 998:          lda  pstr+1
4c8c 48  999:          pha
4c8d 60  1000:         rts
4c8e      1001: ;
4c8e      1002: ;-----
4c8e      1003: ;actual screen handler
4c8e      1004: ;screen offset definitions
4c8e      1005: ;-----
4c8e      1006: scr_offset:
4c8e 0000 1007:          dc.w  0
4c90 2800 1008:          dc.w  40
4c92 5000 1009:          dc.w  80
4c94 7800 1010:          dc.w 120
4c96 a000 1011:          dc.w 160
4c98 c800 1012:          dc.w 200
4c9a f000 1013:          dc.w 240
4c9c 1801 1014:          dc.w 280
4c9e 4001 1015:          dc.w 320
4ca0 6801 1016:          dc.w 360
4ca2 9001 1017:          dc.w 400
4ca4 b801 1018:          dc.w 440
4ca6 e001 1019:          dc.w 480
4ca8 0802 1020:          dc.w 520
4caa 3002 1021:          dc.w 560
4cac 5802 1022:          dc.w 600
4cae 8002 1023:          dc.w 640
4cb0 a802 1024:          dc.w 680
4cb2 d002 1025:          dc.w 720
4cb4 f802 1026:          dc.w 760
4cb6 2003 1027:          dc.w 800
4cb8 4803 1028:          dc.w 840
4cba 7003 1029:          dc.w 880
4cbc 9803 1030:          dc.w 920
4cbe      1031: ;
4cbe      1032: echo:
4cbe      1033: fast_output:
4cbe 855a 1034:          sta  $5a
4cc0 ad0407 1035:          lda  sc.redirect
4cc3 c902  1036:          cmp  #print_p
4cc5 f01f  1037:          beq  .go_cio
4cc7      1038: .leo:
4cc7 adff02 1039:          lda  $02ff      ; CONTROL 1

```

```

4cca c900 1040:      cmp    #$00
4ccc d0f9 1041:      bne    .leo
4cce      1042:
4cce a55a 1043:      lda    $5a
4cd0 c920 1044:      cmp    #32
4cd2 9004 1045:      bcc    .do_look
4cd4 c97d 1046:      cmp    #125
4cd6 9013 1047:      bcc    .do_here
4cd8 a000 1048: .do_look:      ldy    #0
4cda b9e14d 1049: .lookup:      lda    .char_table,y
4cdd f00c 1050:      beq    .do_here
4cdf c55a 1051:      cmp    $5a
4ce1 f003 1052:      beq    .go_cio
4ce3 c8 1053:      iny
4ce4 d0f4 1054:      bne    .lookup
4ce6      1055:
4ce6      1056:
4ce6 a55a 1057: .go_cio:      lda    $5a
4ce8 4cf14d 1058:      jmp    putlocal
4ceb      1059:
4ceb      1060:
4ceb a000 1061: .do_here:      ldy    #0
4ced a55d 1062:      lda    $5d
4cef 915e 1063:      sta    ($5e),y
4cf1 a55a 1064:      lda    $5a
4cf3 c99b 1065:      cmp    #$9b
4cf5 f044 1066:      beq    .docr
4cf7 20c04d 1067: .notcr:      jsr    .get_adr
4cfa a55a 1068:      lda    $5a
4cfc 297f 1069:      and    #$7f
4cfe c920 1070:      cmp    #32
4d00 9007 1071:      bcc    .add64
4d02 c960 1072:      cmp    #96
4d04 9009 1073:      bcc    .sub32
4d06 4c124d 1074:      jmp    .asis
4d09 18 1075: .add64:      clc
4d0a 6940 1076:      adc    #64
4d0c 4c124d 1077:      jmp    .asis
4d0f 38 1078: .sub32:      sec
4d10 e920 1079:      sbc    #32
4d12 245a 1080: .asis:      bit    $5a
4d14 1002 1081:      bpl    .xxlate
4d16 0980 1082:      ora    #$80
4d18 a000 1083: .xxlate:      ldy    #0
4d1a 915e 1084:      sta    ($5e),y
4d1c e655 1085:      inc    $55
4d1e e663 1086:      inc    $63
4d20 a555 1087:      lda    $55
4d22 c928 1088:      cmp    #40
4d24 b019 1089:      bcs    .next_row
4d26 c8 1090:      iny
4d27 b15e 1091:      lda    ($5e),y
4d29 855d 1092:      sta    $5d

```

```

4d2b 0980 1093:      ora  #$80
4d2d aef002 1094:     ldx  752
4d30 d002 1095:      bne  .nocurs1
4d32 915e 1096:      sta  ($5e),y
4d34 e65e 1097: .nocurs1:      inc  $5e
4d36 d002 1098:      bne  .nooav
4d38 e65f 1099:      inc  $5e+1
4d3a 60 1100: .nooav:      rts
4d3b a900 1101: .docr:      lda  #0
4d3d 8563 1102:      sta  $63
4d3f a552 1103: .next_row:     lda  $52
4d41 8555 1104:      sta  $55
4d43 e654 1105:      inc  $54
4d45 a454 1106:      ldy  $54
4d47 c018 1107:      cpy  #24
4d49 b013 1108:      bcs  .scroll
4d4b 20c04d 1109:     jsr  .get_adr
4d4e a000 1110:      ldy  #0
4d50 b15e 1111:      lda  ($5e),y
4d52 855d 1112:      sta  $5d
4d54 aef002 1113:     ldx  752
4d57 d004 1114:      bne  .nocurs2
4d59 0980 1115:      ora  #$80
4d5b 915e 1116:      sta  ($5e),y
4d5d 60 1117: .nocurs2:     rts
4d5e 1118:
4d5e c654 1119: .scroll:      dec  $54
4d60 a558 1120:      lda  $58
4d62 8568 1121:      sta  $68
4d64 18 1122:      clc
4d65 6928 1123:      adc  #40
4d67 855e 1124:      sta  $5e
4d69 a559 1125:      lda  $58+1
4d6b 8569 1126:      sta  $68+1
4d6d 6900 1127:      adc  #0
4d6f 855f 1128:      sta  $5e+1
4d71 a000 1129:      ldy  #0
4d73 a203 1130:      ldx  #3
4d75 b15e 1131: .sclloop:     lda  ($5e),y
4d77 9168 1132:      sta  ($68),y
4d79 c8 1133:      iny
4d7a d0f9 1134:      bne  .sclloop
4d7c e65f 1135:      inc  $5e+1
4d7e e669 1136:      inc  $68+1
4d80 ca 1137:      dex
4d81 d0f2 1138:      bne  .sclloop
4d83 b15e 1139: .sclloop2:    lda  ($5e),y
4d85 9168 1140:      sta  ($68),y
4d87 c8 1141:      iny
4d88 c098 1142:      cpy  #152
4d8a 90f7 1143:      bcc  .sclloop2
4d8c a558 1144:      lda  $58
4d8e 18 1145:      clc

```



```

4d8f 6998 1146:      adc  #920&$ff
4d91 855e 1147:      sta  $5e
4d93 a559 1148:      lda  $58+1
4d95 6903 1149:      adc  #920/256
4d97 855f 1150:      sta  $5e+1
4d99 a000 1151:      ldy  #0
4d9b 98 1152:      tya
4d9c 915e 1153: .clear:      sta  ($5e),y
4d9e c8 1154:      iny
4d9f c028 1155:      cpy  #40
4da1 90f9 1156:      bcc  .clear
4da3 aef002 1157:      ldx  752
4da6 d006 1158:      bne  .nocurs3
4da8 a980 1159:      lda  #$80
4daa a455 1160:      ldy  $55
4dac 915e 1161:      sta  ($5e),y
4dae a900 1162: .nocurs3:      lda  #0
4db0 855d 1163:      sta  $5d
4db2 a55e 1164:      lda  $5e
4db4 18 1165:      clc
4db5 6555 1166:      adc  $55
4db7 855e 1167:      sta  $5e
4db9 a55f 1168:      lda  $5e+1
4dbb 6900 1169:      adc  #0
4dbd 855f 1170:      sta  $5e+1
4dbf 60 1171:      rts
4dc0 a554 1172: .get_adr:      lda  $54
4dc2 0a 1173:      asl
4dc3 a8 1174:      tay
4dc4 a558 1175:      lda  $58
4dc6 18 1176:      clc
4dc7 798e4c 1177:      adc  scr_offset,y
4dca 855e 1178:      sta  $5e
4dcc a559 1179:      lda  $58+1
4dce 798f4c 1180:      adc  scr_offset+1,y
4dd1 855f 1181:      sta  $5e+1
4dd3 a55e 1182:      lda  $5e
4dd5 18 1183:      clc
4dd6 6555 1184:      adc  $55
4dd8 855e 1185:      sta  $5e
4dda a55f 1186:      lda  $5e+1
4ddc 6900 1187:      adc  #0
4dde 855f 1188:      sta  $5e+1
4de0 60 1189:      rts
4de1 1190: .char_table:
4de1 1b1c1d 1191:      dc.b  27,28,29,30,31,125,126,127
4de9 9c9d9e 1192:      dc.b  156,157,158,159,253,254,255
4df1 1193:
4df1 1194:
4df1 a20b 1195: putlocal:      ldx  #11
4df3 8e4203 1196:      stx  $0342
4df6 a200 1197:      ldx  #0
4df8 8e4803 1198:      stx  $0348

```

```

4dfb 8e4903 1199:          stx  $0349
4dfe 4c56e4 1200:          jmp  $e456
4e01      1201: ;
4e01      1202:
4e01      1203:
4e01      1204:
4e01      1205: ;-----
4e01      1206: ; GET KEY ROUTEEN
4e01      1207: ;this routeen get's one key and
4e01      1208: ;return it to the accumulator
4e01      1209: ;-----
4e01      1210: ;
4e01 86b6 1211: getkey:          stx  $b6
4e03 84b7 1212:          sty  $b7
4e05 200d4e 1213:          jsr  .a
4e08 a6b6 1214:          ldx  $b6
4e0a a4b7 1215:          ldy  $b7
4e0c 60 1216:          rts
4e0d ad25e4 1217: .a:          lda  $e420+5
4e10 48 1218:          pha
4e11 ad24e4 1219:          lda  $e420+4
4e14 48 1220:          pha
4e15 60 1221:          rts
4e16      1222:
4e16 a904 1223: open_in:      lda  #4
4e18 4c344e 1224:          jmp  do_open
4e1b      1225:
4e1b a908 1226: open_out:     lda  #8
4e1d 4c344e 1227:          jmp  do_open
4e20      1228:
4e20 a909 1229: open_app:     lda  #9
4e22 4c344e 1230:          jmp  do_open
4e25      1231:
4e25 a90c 1232: open_updt:    lda  #12
4e27 4c344e 1233:          jmp  do_open
4e2a      1234:
4e2a a906 1235: open_dir:     lda  #6
4e2c 4c344e 1236:          jmp  do_open
4e2f      1237:
4e2f a914 1238: open_raw:     lda  #20          ;open raw d
4e31 4c344e 1239:          jmp  do_open
4e34      1240:
4e34      1241: do_open:
4e34 48 1242:          pha          ;open files
4e35 8a 1243:          txa          ;
4e36 48 1244:          pha
4e37 20614e 1245:          jsr  close      ;make sure
4e3a 68 1246:          pla
4e3b aa 1247:          tax
4e3c 68 1248:          pla
4e3d 9d4a03 1249:          sta  icaux1,x
4e40 a903 1250:          lda  #ccopen
4e42 9d4203 1251:          sta  iccom,x

```

```

4e45 a586 1252:      lda  fn_ptr
4e47 9d4403 1253:      sta  icbal,x
4e4a a587 1254:      lda  fn_ptr+1
4e4c 9d4503 1255:      sta  icbah,x
4e4f a950 1256:      lda  #80
4e51 9d4803 1257:      sta  icbll,x
4e54 a900 1258:      lda  #0
4e56 9d4903 1259:      sta  icblh,x
4e59 a980 1260:      lda  #128      ;for long d
4e5b 9d4b03 1261:      sta  icaux2,x
4e5e 4c56e4 1262:      jmp  ciov
4e61      1263:
4e61      1264:
4e61 a90c 1265: close:      lda  #ccclose
4e63 9d4203 1266:      sta  iccom,x
4e66 4c56e4 1267:      jmp  ciov
4e69      1268:
4e69      1269: ;   Move string pointed to by s_ptr to string pointed
4e69      1270: ;   ~~~~~
4e69      1271:
4e69      1272:
4e69 a000 1273: move_str:      ldy  #0
4e6b b182 1274:      lda  (s_ptr),y      ;ge
4e6d a8 1275:      tay      ;go
4e6e b182 1276: .loop:      lda  (s_ptr),y
4e70 9184 1277:      sta  (d_ptr),y
4e72 c000 1278:      cpy  #0
4e74 f004 1279:      beq  .done
4e76 88 1280:      dey
4e77 4c6e4e 1281:      jmp  .loop
4e7a 60 1282: .done:      rts
4e7b      1283:
4e7b      1284: ;   Make an ASCII ($9b terminated) string out of a lth-b
4e7b      1285: ;   ~~~~~
4e7b      1286:
4e7b a000 1287: toascii:      ldy  #0
4e7d b182 1288:      lda  (s_ptr),y      ;ge
4e7f 8da14e 1289:      sta  .lth
4e82 f019 1290:      beq  .null
4e84 c8 1291:      iny
4e85 eea14e 1292:      inc  .lth
4e88 b182 1293: .loop:      lda  (s_ptr),y
4e8a 88 1294:      dey
4e8b 9184 1295:      sta  (d_ptr),y
4e8d c8 1296:      iny
4e8e c8 1297:      iny
4e8f cca14e 1298:      cpy  .lth
4e92 d0f4 1299:      bne  .loop
4e94 aca14e 1300: .done:      ldy  .lth
4e97 88 1301:      dey
4e98 a99b 1302: .rtn:      lda  #$9b
4e9a 9184 1303:      sta  (d_ptr),y
4e9c 60 1304:      rts

```

```

4e9d a000 1305: .null:      ldy  #0
4e9f f0f7 1306:            beq  .rtn
4ea1      1307:
4ea1 01   1308: .lth:      dc.b  1
4ea2      1309:
4ea2      1310: ;~~~~~
4ea2      1311: ;Prints a byte or a card to the screen
4ea2      1312: ;
4ea2      1313: ;
4ea2      1314: pr_byte:
4ea2 a200 1315:      ldx  #$00
4ea4      1316: pr_card:
4ea4 85d4 1317:      sta  fr0
4ea6 86d5 1318:      stx  fr0+1
4ea8 a900 1319:      lda  #$00
4eaa 85f2 1320:      sta  cix
4eac 20aad9 1321:      jsr  ifp
4eaf 20e6d8 1322:      jsr  fasc
4eb2 a000 1323:      ldy  #$00
4eb4      1324: .aprbloop:
4eb4 b1f3 1325:      lda  (inbuff),y
4eb6 08   1326:      php
4eb7 297f 1327:      and  #$7f
4eb9 c8   1328:      iny
4eba 20be4c 1329:      jsr  echo
4ebd 28   1330:      plp
4ebe 10f4 1331:      bpl  .aprbloop
4ec0 60   1332:      rts
4ec1      1333:
4ec1      1334: ;   gets a CARD value - returned in A/X
4ec1      1335: ;   ~~~~~
4ec1      1336:
4ec1 20f94e 1337: get_card:      jsr  get_str
4ec4 ef4e 1338:      dc.w  .work
4ec6 050100 1339:      dc.b  5,1,0
4ec9 a8   1340:      tay
4eca c8   1341:      iny
4ecb a99b 1342:      lda  #$9b      ;in
4ecd 99ef4e 1343:      sta  .work,y
4ed0      1344:      mea  .work,inbuff
4ed0 a9ef ----:      lda  #low .work
4ed2 85f3 ----:      sta  inbuff
4ed4 a94e ----:      lda  #high .work
4ed6 85f4 ----:      sta  inbuff + 1
4ed8 a901 1345:      lda  #1
4eda 85f2 1346:      sta  cix
4edc 2000d8 1347:      jsr  afp      ;as
4edf b00a 1348:      bcs  .default
4ee1 20d2d9 1349:      jsr  fpi      ;fl
4ee4 b005 1350:      bcs  .default
4ee6 a5d4 1351:      lda  fr0
4ee8 a6d5 1352:      ldx  fr0+1
4eea 60   1353:      rts

```

```

4eeb a900 1354: .default:      lda  #0
4eed aa   1355:                tax
4eee 60   1356:                rts
4eef      1357:
4eef      1358: .work:        ds.b  10
4ef9      1359: ;
4ef9      1360: ;
4ef9      1361: ;
4ef9      1362: ; ~~~~~
4ef9      1363: ;
4ef9      1364: ;
4ef9      1365: ;  get_str  gets an input line into the specified s
4ef9      1366: ;
4ef9      1367: ;  input:  address of string in in_ptr
4ef9      1368: ;          A - max length
4ef9      1369: ;          X - mask (0=alpha/numeric, 1=numeric)
4ef9      1370: ;          Y - allow return key only (1=yes)
4ef9      1371: ;
4ef9      1372: ;  length of inputted string is returned in the 0th pos
4ef9      1373: ; ~~~~~
4ef9      1374:
4ef9 68   1375: get_str:      pla                ;en
4efa 8588 1376:                sta  pr_ptr
4efc 68   1377:      pla
4efd 8589 1378:                sta  pr_ptr+1
4eff a001 1379:                ldy  #1
4f01 b188 1380:                lda  (pr_ptr),y      ;ge
4f03 8580 1381:                sta  in_ptr
4f05 c8   1382:      iny
4f06 b188 1383:                lda  (pr_ptr),y      ;
4f08 8581 1384:                sta  in_ptr+1
4f0a c8   1385:      iny
4f0b b188 1386:                lda  (pr_ptr),y      ;
4f0d 8d374f 1387:                sta  .gsa
4f10 c8   1388:      iny
4f11 b188 1389:                lda  (pr_ptr),y      ;
4f13 8d384f 1390:                sta  .gsx
4f16 c8   1391:      iny
4f17 b188 1392:                lda  (pr_ptr),y      ;
4f19 8d394f 1393:                sta  .gsy
4f1c a588 1394:                lda  pr_ptr          ;no
4f1e 18   1395:      clc
4f1f 6905 1396:                adc  #5
4f21 8588 1397:                sta  pr_ptr
4f23 a589 1398:                lda  pr_ptr+1
4f25 6900 1399:                adc  #0
4f27 48   1400:      pha
4f28 a588 1401:                lda  pr_ptr
4f2a 48   1402:      pha
4f2b ad374f 1403:                lda  .gsa
4f2e ae384f 1404:                ldx  .gsx
4f31 ac394f 1405:                ldy  .gsy
4f34 4c3a4f 1406:                jmp  get_string

```

```

4f37      1407:
4f37 00    1408: .gsa:          dc.b  0
4f38 00    1409: .gsx:          dc.b  0
4f39 00    1410: .gsy:          dc.b  0
4f3a      1411:
4f3a      1412:
4f3a c953  1413: get_string:    cmp    #83          ;ma
4f3c 9002  1414:                bcc    .gsentry    ;ye
4f3e a950  1415:                lda    #80
4f40 8d5350 1416: .gsentry:      sta    .maxlth      ;sa
4f43 8e5450 1417:                stx    .mask
4f46 8c5550 1418:                sty    .return
4f49      1419:
4f49      1420:
4f49 a580  1421:                lda    in_ptr
4f4b 8d4d50 1422:                sta    .address      ;ho
4f4e a581  1423:                lda    in_ptr+1      ; t
4f50 8d4e50 1424:                sta    .address+1
4f53 a900  1425:                lda    #0
4f55 8d5750 1426:                sta    .instr        ;ze
4f58      1427:
4f58 20ad50 1428: .gsloop:      jsr    get_key        ;ge
4f5b c95c  1429:                cmp    #"          ;can
4f5d f0f9  1430:                beq    .gsloop      ;ye
4f5f c99b  1431:                cmp    #$9b        ;ca
4f61 d003  1432:                bne    .nogs        ;no
4f63 4cff4f 1433:                jmp    .gscr        ;ye
4f66 c90e  1434: .nogs:        cmp    #14          ;tr
4f68 d003  1435:                bne    .not14
4f6a 4cff4f 1436:                jmp    .gscr
4f6d c97e  1437: .not14:      cmp    #126
4f6f f066  1438:                beq    .bspace
4f71 c91e  1439:                cmp    #30
4f73 f062  1440:                beq    .bspace
4f75 c918  1441:                cmp    #24          ;^X
4f77 d003  1442:                bne    .nowipe
4f79 4cec4f 1443:                jmp    .wipeout
4f7c 297f  1444: .nowipe:      and    #$7f          ;st
4f7e 8d5650 1445:                sta    .key          ;ho
4f81      1446:
4f81      1447: ;    now, see if it falls within requested mask
4f81      1448: ;    ~~~~~
4f81      1449:
4f81      1450:
4f81 ae5450 1451:                ldx    .mask          ;ge
4f84 f004  1452:                beq    .alphanum
4f86 e001  1453:                cpx    #1
4f88 f00b  1454:                beq    .numeric
4f8a      1455:
4f8a c920  1456: .alphanum:    cmp    #32          ;<
4f8c 90ca  1457:                bcc    .gsloop      ;ye
4f8e c97b  1458:                cmp    #'z'+1        ;>
4f90 b0c6  1459:                bcs    .gsloop      ;ye

```

```

4f92 4ca04f 1460:      jmp    .gsstore
4f95      1461:
4f95 c930 1462: .numeric:      cmp    #'0'      ;<
4f97 90bf 1463:      bcc    .gsloop      ;ye
4f99 c93a 1464:      cmp    #'9'+1      ;>
4f9b b0bb 1465:      bcs    .gsloop      ;ye
4f9d 4ca04f 1466:      jmp    .gsstore
4fa0      1467:
4fa0 ad5750 1468: .gsstore:      lda    .instr      ;ge
4fa3 cd5350 1469:      cmp    .maxlth
4fa6 f0b0 1470:      beq    .gsloop      ;ig
4fa8 b0ae 1471:      bcs    .gsloop      ;m
4faa a8 1472:      tay
4fab c8 1473:      iny
4fac ad5650 1474:      lda    .key
4faf c920 1475:      cmp    #32      ;is
4fb1 d007 1476:      bne    .notspc      ;no
4fb3 c001 1477:      cpy    #1      ;ls
4fb5 d003 1478:      bne    .notspc
4fb7 4c584f 1479:      jmp    .gsloop      ;ye
4fba 995750 1480: .notspc:      sta    .instr,y      ;sa
4fbd ee5750 1481:      inc    .instr      ;in
4fc0      1482: ;      bit    hide      ;hi
4fc0      1483: ;      bpl    .nohide
4fc0      1484: ;      lda    #'*'
4fc0      1485: ;      bne    .showit
4fc0      1486: ;.nohide:      lda    .key
4fc0 20be4c 1487: .showit:      jsr    echo
4fc3 ad5350 1488:      lda    .maxlth      ;on
4fc6 c901 1489:      cmp    #1
4fc8 d00a 1490:      bne    .nohot      ;no
4fca ad5750 1491:      lda    .instr
4fcd c901 1492:      cmp    #1      ;l
4fcf d003 1493:      bne    .nohot
4fd1      1494: ;      lda    u_flags
4fd1      1495: ;      and    #f_hotkey      ;us
4fd1      1496: ;      beq    .nohot      ;no
4fd1 4cff4f 1497:      jmp    .gscr      ;ye
4fd4      1498:
4fd4 4c584f 1499: .nohot:      jmp    .gsloop
4fd7      1500:
4fd7 ad5750 1501: .bspace:      lda    .instr      ;ca
4fda c900 1502:      cmp    #0      ;i
4fdc d003 1503:      bne    .nt0
4fde 4c584f 1504:      jmp    .gsloop
4fe1 ce5750 1505: .nt0:      dec    .instr      ;-1
4fe4 a97e 1506:      lda    #126      ;ou
4fe6 20be4c 1507:      jsr    echo
4fe9 4c584f 1508:      jmp    .gsloop
4fec      1509:
4fec ad5750 1510: .wipeout:      lda    .instr      ;ca
4fef d003 1511:      bne    .wt0
4ff1 4c584f 1512:      jmp    .gsloop

```

```

4ff4 ce5750 1513: .wt0:      dec  .instr      ;-1
4ff7 a97e 1514:      lda  #126      ;ou
4ff9 20be4c 1515:      jsr  echo
4ffc 4cec4f 1516:      jmp  .wipeout   ;go
4fff      1517:
4fff ad5750 1518: .gscr:      lda  .instr      ;ge
5002 d008 1519:      bne  .nonnull   ;it
5004 ae5550 1520:      ldx  .return     ;al
5007 d003 1521:      bne  .nonnull   ;ye
5009 4c584f 1522:      jmp  .gsloop    ;no
500c      1523:
500c      1524: ;   done, so move our string into where they want it.
500c      1525: ;   ~~~~~
500c      1526:
500c a582 1527: .nonnull:    lda  s_ptr
500e 8d4f50 1528:      sta  .holds
5011 a583 1529:      lda  s_ptr+1
5013 8d5050 1530:      sta  .holds+1
5016 a584 1531:      lda  d_ptr
5018 8d5150 1532:      sta  .holdd
501b a585 1533:      lda  d_ptr+1
501d 8d5250 1534:      sta  .holdd+1
5020      1535:      mea  .instr,s_ptr
5020 a957 ----:      lda  #low .instr
5022 8582 ----:      sta  s_ptr
5024 a950 ----:      lda  #high .instr
5026 8583 ----:      sta  s_ptr + 1
5028 ad4d50 1536:      lda  .address
502b 8584 1537:      sta  d_ptr
502d ad4e50 1538:      lda  .address+1
5030 8585 1539:      sta  d_ptr+1
5032 20694e 1540:      jsr  move_str    ;mo
5035 ad4f50 1541:      lda  .holds
5038 8582 1542:      sta  s_ptr
503a ad5050 1543:      lda  .holds+1
503d 8583 1544:      sta  s_ptr+1
503f ad5150 1545:      lda  .holdd
5042 8584 1546:      sta  d_ptr
5044 ad5250 1547:      lda  .holdd+1
5047 8585 1548:      sta  d_ptr+1
5049      1549: ;      lda  #0
5049      1550: ;      sta  hide
5049 ad5750 1551:      lda  .instr      ;re
504c 60 1552:      rts
504d      1553:
504d      1554:
504d      1555: .address:      ds.b  2
504f      1556: .holds:        ds.b  2
5051      1557: .holdd:        ds.b  2
5053      1558: .maxlth:      ds.b  1
5054      1559: .mask:        ds.b  1
5055      1560: .return:      ds.b  1
5056      1561: .key:        ds.b  1

```



```

5057      1562: .instr:          ds.b  86
50ad      1563:
50ad      1564: ;-----
50ad      1565: ; GET KEY ROUTEEN
50ad      1566: ;this routeen get's one key and
50ad      1567: ;return it to the accumulator
50ad      1568: ;-----
50ad      1569: ;
50ad 86b6 1570: get_key:          stx  $b6
50af 84b7 1571:                  sty  $b7
50b1 20b950 1572:                  jsr  g_fff
50b4 a6b6 1573:                  ldx  $b6
50b6 a4b7 1574:                  ldy  $b7
50b8 60 1575:                  rts
50b9 ad25e4 1576: g_fff:          lda  $e420+5
50bc 48 1577:                  pha
50bd ad24e4 1578:                  lda  $e420+4
50c0 48 1579:                  pha
50c1 60 1580:                  rts
50c2      1581: ;
50c2      1582: ; prints: prints string with address in PR_PTR, length by
50c2      1583: ; ~~~~~
50c2      1584:
50c2 a080 1585: prints:         ldy  #$80
50c4 d002 1586:                 bne  pri2
50c6 a000 1587: echos:         ldy  #0
50c8      1588:
50c8 8cf650 1589: pri2:          sty  pr_mode
50cb a000 1590:                 ldy  #0
50cd b188 1591:                 lda  (pr_ptr),y      ;ge
50cf f022 1592:                 beq  .xpri          ;ex
50d1 8df550 1593:                 sta  .maxlth
50d4 c8 1594:                 iny
50d5 8cf450 1595:                 sty  .tempy
50d8      1596:
50d8 acf450 1597: .pril:         ldy  .tempy
50db b188 1598:                 lda  (pr_ptr),y
50dd 2cf650 1599:                 bit  pr_mode
50e0 3006 1600:                 bmi  .prloc
50e2      1601: ; eor  #$80
50e2 20be4c 1602:                 jsr  echo
50e5 4ceb50 1603:                 jmp  .prnext
50e8      1604: .prloc:
50e8      1605: ; eor  #$80
50e8 20be4c 1606:                 jsr  echo
50eb eef450 1607: .prnext:       inc  .tempy
50ee cef550 1608:                 dec  .maxlth
50f1 d0e5 1609:                 bne  .pril
50f3 60 1610: .xpri:         rts
50f4      1611:
50f4 01 1612: .tempy:        dc.b  1
50f5 01 1613: .maxlth:       dc.b  1
50f6      1614:

```

```
50f6 00 1615: pr_mode:      dc.b  0
50f7    1616:
50f7    1617:
50f7    1618:
50f7    1619: ;
50f7    1620: win_end:      ds.b  0
50f7    1621: ;
6000    1622:                .org $6000
6000    1623: copy_buff:     ds.b  0
6000    1624: ;
6000    1625: ;
```

End assembly: no errors