


```

0000 1: ;-----
0000 2: ;
0000 3: ;      Copyright 2010 Intergraded Logic Systems
0000 4: ;      Source Code is Copyright Stephen J. Car
0000 5: ;
0000 6: ;
0000 7: ;      Please do not share this source!
0000 8: ;-----
0000 9: ;      (General hard disk formatter for real.dos SpartaDO
0000 10: ;
0000 11: ;-----
0000 12: ;      Real.dos chkdisk2.COM
0000 13: ;      This program is to replace
0000 14: ;      chkdisk pfrom the command line inter
0000 15: ;
0000 16: ;      Code History.  Init design, June 19, 2010
0000 17: ;
0000 18: ;Updated the screen Print Utility so it will work with the
0000 19: ; This change will also help "Print P:"      8/1
0000 20: ;
0000 21: ;
0000 22: ;Also added code for Control+1 pause of screen to the high
0000 23: ;handling.      8/28/2012      sjc
0000 24: ;
0000 25: ;
0000 26: ;      ~~~~~
0000 27: ;-----
0000 28: ; Notes: o This source code MAY NOT be placed for download
0000 29: ;      o "mea" is a macro that loads the address of the
0000 30: ;      into the pointer specified by the second field.
0000 31: ;-----
0000 32: ; Assembler: MADMAC (tm) ST Cross Assembler (Atari Corp)
0000 33: ;      XASM st and IBM VERSIONS
0000 34: ;-----
0000 35:
0000 36: com_file_name:      .macro
0000 37: ;      dc.b 12,"filename.ext",$9b
0000 38: ;      dc.b 12,"Chkdsk2.com ", $9b
0000 39: ;      .endm
0000 40:
0000 41: ;
0000 42: ;~~~~~
0000 43: ; File revision history. Version Number in hex
0000 44: ;
0000 45: file_ver:      equ $11
0000 46: ;~~~~~
0000 47: ; the Month in dec for the first time this revision
0000 48: ;was compiled
0000 49: ;
0000 50: c_month:      equ 8
0000 51: ;~~~~~
0000 52: ; the day in dec for the first time this revision
0000 53: ;was compiled

```

```
0000 54: c_day:          equ 28
0000 55: ;~~~~~
0000 56: ; the year in dec for the first time this revision
0000 57: ;was compiled
0000 58: c_year:          equ 2012
0000 59: ;~~~~~
0000 60: ; Type of compiler used in program
0000 61: ;
0000 62: ; 0 = unknown Compiler
0000 63: ; 1 = xasm
0000 64: ; 2 = mac_65
0000 65: ; 3 = basic
0000 66: ; 4 = compiled basic xl
0000 67: ; 5 = C65
0000 68: ; 6 = Action
0000 69: ;
0000 70: ; We can add more as time goes on!
0000 71: ;
0000 72: ;
0000 73: ;
0000 74: xasm:            equ 1
0000 75: mac_65:          equ 2
0000 76: basic:           equ 3
0000 77: basicxl:         equ 4
0000 78: c65:             equ 5
0000 79: action:          equ 6
0000 80:
0000 81: file_compiler:    equ xasm
0000 82: ;~~~~~
0000 83: ; is this relocatable code ?
0000 84: ; anyother value other than 1 or 2 would be unknown
0000 85: r..yes:          equ 1
0000 86: r..no:           equ 2
0000 87: ;
0000 88: ;~~~~~
0000 89: ; Gotta define if it can be relocated
0000 90: l_relocatable:    equ r..no
0000 91: ;
0000 92: ;
0000 93: ;~~~~~
0000 94: ; Gotta define our Header control Byte
0000 95: ;$ff=bypass intro
0000 96: ;$15 = Bypass Dos check
0000 97: r..crl:          equ $7d
0000 98: r..crlf:         equ $9b
0000 99: r..space:        equ $20
0000 100: r..bypassDOS:    equ $15
0000 101: r..bypass:       equ $ff
0000 102: r..supress:      equ $ff
0000 103:
0000 104: l_frstscreenbyte: equ r..space
0000 105: ;~~~~~
0000 106: ; The language the output file is in.
```

```

0000    107: ;
0000    108: ;
0000    109: ; 0 = Undefined
0000    110: ; 1 = English
0000    111: ; 2 = German
0000    112: ;
0000    113: r..language: equ 1
0000    114: ;~~~~~
0000    115: ;
0000    116: ;
0000    117:
0000    118:         .include equates
0000    119:         .include globals
0000    120:         .include macros
0000    121: ;
4000    122:         .org $4000
4000    123: ;
4000    124:
4000    125: win_start:
4000    126: header_info:
4000    127:         .include header
42a7    128: start:
42a7 a220 129:         ldx #disk2
42a9 20a144 130:        jsr close
42ac    131:
42ac a50a 132:         lda comtab
42ae 18 133:         CLC
42af 6903 134:         adc # low ZCRNAME
42b1 8ded42 135:        sta cli.crunch+1
42b4 a50b 136:         lda comtab+1
42b6 6900 137:         adc # high ZCRNAME
42b8 8dee42 138:        sta cli.crunch+2
42bb 20ec42 139:        jsr cli.crunch
42be d000 140:        bne .get_parm1
42c0    141: ;        jmp parm_errors
42c0    142:
42c0    143: .get_parm1:
42c0 a50a 144:         lda comtab
42c2 18 145:         CLC
42c3 6921 146:         adc # low COMFNAM
42c5 8582 147:         sta s_ptr
42c7 a50b 148:         lda comtab+1
42c9 6900 149:         adc # high COMFNAM
42cb 8583 150:         sta s_ptr+1
42cd    151:         mea parm_1,d_ptr
42cd a95c ----:        lda #low parm_1
42cf 8584 ----:        sta d_ptr
42d1 a943 ----:        lda #high parm_1
42d3 8585 ----:        sta d_ptr + 1
42d5 20d546 152:        jsr tostr
42d8 ad5e43 153:        lda parm_1+2
42db 8d1843 154:        sta p_drive+1
42de    155:

```

```

42de ad5e43 156:          lda  parm_1+2
42e1      157:
42e1 c931  158:          cmp  #'1'
42e3 900a  159:          bcc  show_help
42e5 c93a  160:          cmp  #'9'+1
42e7 b006  161:          bcs  show_help
42e9 4c9c43 162:          jmp  c_chkdisk
42ec      163:
42ec      164:
42ec 4c0000 165: cli.crunch:      jmp  $0000
42ef      166:
42ef      167: parm_errors:
42ef 20aa44 168: Show_help:      jsr  printsi
42f2 9b9b  169:          dc.b  $9B,$9B
42f4 20     170:          dc.b  " "
42f5 d3f9ee 171:          dc.b  <+128>,"Syntax"
42fb 20666f 172:          dc.b  " for use:","$9b
4305 204368 173:          dc.b  " ChkDsk2 "
430e dbc4ee 174:          dc.b  <+128>,"[Dn:]"
4313 9b9b  175:          dc.b  $9b,$9b
4315      176:
4315 ff     177:          dc.b  -1
4316 60     178:          RTS
4317      179:
4317      180:
4317 44783a 181: p_drive:      dc.b  "Dx:","$9b,0
431c      182: command_line: ds.b  64
435c      183: parm_1:      ds.b  64
439c      184:
439c      185: ;          Show disk information (SpartaDOS)
439c      186: ;          ~~~~~
439c      187:
439c      188: c_chkdisk:
439c a220  189:          ldx  #disk_2
439e 20a144 190:          jsr  close
43a1 a220  191:          ldx  #disk_2
43a3 a92f  192:          lda  #47
43a5 9d4203 193:          sta  iccom,x
43a8 a917  194:          lda  #p_drive & $ff
43aa 9d4403 195:          sta  icbal,x
43ad a943  196:          lda  #p_drive / 256
43af 9d4503 197:          sta  icbah,x
43b2 a91c  198:          lda  #command_line & $ff
43b4 9d4803 199:          sta  icbll,x
43b7 a943  200:          lda  #command_line / 256
43b9 9d4903 201:          sta  icblh,x
43bc 2056e4 202:          jsr  ciov
43bf 1003  203:          bpl  .OK
43c1 4c6d44 204:          jmp  system_error
43c4 20aa44 205: .OK:          jsr  printsi
43c7 202056 206:          dc.b  " Volume name: ",-1
43d7 a906  207:          lda  #6
43d9 8d6c44 208:          sta  .idx

```

```

43dc ac6c44 209: .vol:      ldy  .idx
43df b91c43 210:      lda  command_line,y
43e2 200045 211:      jsr  echo
43e5 ee6c44 212:      inc  .idx
43e8 ad6c44 213:      lda  .idx
43eb c90e   214:      cmp  #14
43ed 90ed   215:      bcc  .vol
43ef 20aa44 216:      jsr  printsi
43f2 9b2042 217:      dc.b  "$9b," Bytes/sector: ",-1
4403 ad1d43 218:      lda  command_line+1
4406 f014   219:      beq  .double
4408 20aa44 220: .single:  jsr  printsi
440b 313238 221:      dc.b  "128 (single)", $9b,-1
4419 4c2d44 222:      jmp  .total
441c 20aa44 223: .double:  jsr  printsi
441f 323536 224:      dc.b  "256 (double)", $9b,-1
442d 20aa44 225: .total:   jsr  printsi
4430 546f74 226:      dc.b  "Total sectors: ",-1
4440 ad1e43 227:      lda  command_line+2
4443 ae1f43 228:      ldx  command_line+3
4446 206946 229:      jsr  pr_card
4449 20aa44 230:      jsr  printsi
444c 9b2046 231:      dc.b  "$9b," Free sectors: ",-1
445d ad2043 232:      lda  command_line+4
4460 ae2143 233:      ldx  command_line+5
4463 206946 234:      jsr  pr_card
4466 20aa44 235:      jsr  printsi
4469 9bff   236:      dc.b  $9b,-1
446b 60     237:      rts
446c 00     238: .idx:    dc.b  0
446d       239:
446d       240:
446d       241: ;   Display error message of flag in Y register
446d       242: ;   ~~~~~
446d       243:
446d       244: system_error:
446d 8c8e44 245:      sty  .hold
4470 20aa44 246:      jsr  printsi
4473 537973 247:      dc.b  "System Error: ",-1
4482 ad8e44 248:      lda  .hold
4485 206746 249:      jsr  pr_byte
4488 20aa44 250:      jsr  printsi
448b 9bff   251:      dc.b  $9b,-1
448d 60     252:      rts
448e       253:
448e       254:
448e 00     255: .hold:   dc.b  0
448f       256:
448f       257: ;-----
448f       258: ;wait for a vbi to happen
448f       259: ;-----
448f a514   260: wait_vbi: lda  $14
4491 c514   261: .w1:    cmp  $14

```

```

4493 f0fc 262:          beq  .w1
4495 60 263:          rts
4496 264:
4496 265: ;-----
4496 266: ;CHANGES THE CHARTER TO UPCASE
4496 267: ;-----
4496 268: upcase
4496 c961 269:          cmp  #$61
4498 9006 270:          bcc  .a
449a c97a 271:          cmp  #$7a
449c b002 272:          bcs  .a
449e e91f 273:          sbc  #$1f
44a0 60 274: .a:          rts
44a1 275:
44a1 276: ;~~~~~
44a1 277: ; Genric Close command
44a1 278: ;~~~~~
44a1 279: close:
44a1 a90c 280:          lda  #$0c
44a3 9d4203 281:          sta  $0342,x
44a6 2056e4 282:          jsr  ciov
44a9 60 283:          rts
44aa 284:
44aa 285: ;
44aa 286: ;
44aa 287: ;-----
44aa 288: ;prints Routeen macro
44aa 289: ;this is a slow print to the screen
44aa 290: ;usage jsr  prints
44aa 291: ; .byte $9b," string to be printed",$ff
44aa 292: ;-----
44aa 293: ;
44aa 294: prints:
44aa 68 295:          pla
44ab 8dbb44 296:          sta  .pstr+1
44ae 68 297:          pla
44af 8dbc44 298:          sta  .pstr+2
44b2 eebb44 299: .prsl:    inc  .pstr+1
44b5 d003 300:          bne  .pstr
44b7 eebe44 301:          inc  .pstr+2
44ba adffff 302: .pstr:    lda  $ffff
44bd c9ff 303:          cmp  #$ff
44bf f006 304:          beq  .estri
44c1 200045 305:          jsr  fast_output
44c4 4cb244 306:          jmp  .prsl
44c7 adbc44 307: .estri:   lda  .pstr+2
44ca 48 308:          pha
44cb adbb44 309:          lda  .pstr+1
44ce 48 310:          pha
44cf 60 311:          rts
44d0 312: ;
44d0 313: ;
44d0 314: ;

```

```

44d0      315: ;-----
44d0      316: ;actual screen handler
44d0      317: ;screen offset definitions
44d0      318: ;-----
44d0      319: scr_offset:
44d0 0000  320:          dc.w  0
44d2 2800  321:          dc.w  40
44d4 5000  322:          dc.w  80
44d6 7800  323:          dc.w 120
44d8 a000  324:          dc.w 160
44da c800  325:          dc.w 200
44dc f000  326:          dc.w 240
44de 1801  327:          dc.w 280
44e0 4001  328:          dc.w 320
44e2 6801  329:          dc.w 360
44e4 9001  330:          dc.w 400
44e6 b801  331:          dc.w 440
44e8 e001  332:          dc.w 480
44ea 0802  333:          dc.w 520
44ec 3002  334:          dc.w 560
44ee 5802  335:          dc.w 600
44f0 8002  336:          dc.w 640
44f2 a802  337:          dc.w 680
44f4 d002  338:          dc.w 720
44f6 f802  339:          dc.w 760
44f8 2003  340:          dc.w 800
44fa 4803  341:          dc.w 840
44fc 7003  342:          dc.w 880
44fe 9803  343:          dc.w 920
4500      344: ;
4500      345: ;
4500      346: echo:
4500      347:
4500      348: fast_output:
4500 855a  349:          sta  $5a
4502 ad0407 350:          lda  sc.redirect
4505 c902  351:          cmp  #print_p
4507 f01f  352:          beq  .go_cio
4509      353: .leo:
4509 adff02 354:          lda  $02ff      ; CONTROL 1
450c c900  355:          cmp  #$00
450e d0f9  356:          bne  .leo
4510      357:
4510 a55a  358:          lda  $5a
4512 c920  359:          cmp  #32
4514 9004  360:          bcc  .do_look
4516 c97d  361:          cmp  #125
4518 9013  362:          bcc  .do_here
451a a000  363: .do_look:      ldy  #0
451c b92346 364: .lookup:      lda  .char_table,y
451f f00c  365:          beq  .do_here
4521 c55a  366:          cmp  $5a
4523 f003  367:          beq  .go_cio

```



```

4525 c8 368:          iny
4526 d0f4 369:          bne .lookup
4528      370:
4528      371:
4528 a55a 372: .go_cio:      lda $5a
452a 4c3346 373:          jmp putlocal
452d      374:
452d      375:
452d a000 376: .do_here:      ldy #0
452f a55d 377:          lda $5d
4531 915e 378:          sta ($5e),y
4533 a55a 379:          lda $5a
4535 c99b 380:          cmp #$9b
4537 f044 381:          beq .docr
4539 200246 382: .notcr:      jsr .get_adr
453c a55a 383:          lda $5a
453e 297f 384:          and #$7f
4540 c920 385:          cmp #32
4542 9007 386:          bcc .add64
4544 c960 387:          cmp #96
4546 9009 388:          bcc .sub32
4548 4c5445 389:          jmp .asis
454b 18 390: .add64:      clc
454c 6940 391:          adc #64
454e 4c5445 392:          jmp .asis
4551 38 393: .sub32:      sec
4552 e920 394:          sbc #32
4554 245a 395: .asis:      bit $5a
4556 1002 396:          bpl .xxlate
4558 0980 397:          ora #$80
455a a000 398: .xxlate:      ldy #0
455c 915e 399:          sta ($5e),y
455e e655 400:          inc $55
4560 e663 401:          inc $63
4562 a555 402:          lda $55
4564 c928 403:          cmp #40
4566 b019 404:          bcs .next_row
4568 c8 405:          iny
4569 b15e 406:          lda ($5e),y
456b 855d 407:          sta $5d
456d 0980 408:          ora #$80
456f aef002 409:          ldx 752
4572 d002 410:          bne .nocurs1
4574 915e 411:          sta ($5e),y
4576 e65e 412: .nocurs1:      inc $5e
4578 d002 413:          bne .nooav
457a e65f 414:          inc $5e+1
457c 60 415: .nooav:      rts
457d a900 416: .docr:      lda #0
457f 8563 417:          sta $63
4581 a552 418: .next_row:      lda $52
4583 8555 419:          sta $55
4585 e654 420:          inc $54

```

```

4587 a454 421:      ldy  $54
4589 c018 422:      cpy  #24
458b b013 423:      bcs  .scroll
458d 200246 424:     jsr  .get_adr
4590 a000 425:      ldy  #0
4592 b15e 426:      lda  ($5e),y
4594 855d 427:      sta  $5d
4596 aef002 428:     ldx  752
4599 d004 429:      bne  .nocurs2
459b 0980 430:      ora  #$80
459d 915e 431:      sta  ($5e),y
459f 60 432: .nocurs2:      rts
45a0      433:
45a0 c654 434: .scroll:      dec  $54
45a2 a558 435:      lda  $58
45a4 8568 436:      sta  $68
45a6 18 437:      clc
45a7 6928 438:      adc  #40
45a9 855e 439:      sta  $5e
45ab a559 440:      lda  $58+1
45ad 8569 441:      sta  $68+1
45af 6900 442:      adc  #0
45b1 855f 443:      sta  $5e+1
45b3 a000 444:      ldy  #0
45b5 a203 445:      ldx  #3
45b7 b15e 446: .sloop:      lda  ($5e),y
45b9 9168 447:      sta  ($68),y
45bb c8 448:      iny
45bc d0f9 449:      bne  .sloop
45be e65f 450:      inc  $5e+1
45c0 e669 451:      inc  $68+1
45c2 ca 452:      dex
45c3 d0f2 453:      bne  .sloop
45c5 b15e 454: .sloop2:     lda  ($5e),y
45c7 9168 455:      sta  ($68),y
45c9 c8 456:      iny
45ca c098 457:      cpy  #152
45cc 90f7 458:      bcc  .sloop2
45ce a558 459:      lda  $58
45d0 18 460:      clc
45d1 6998 461:      adc  #920&$ff
45d3 855e 462:      sta  $5e
45d5 a559 463:      lda  $58+1
45d7 6903 464:      adc  #920/256
45d9 855f 465:      sta  $5e+1
45db a000 466:      ldy  #0
45dd 98 467:      tya
45de 915e 468: .clear:      sta  ($5e),y
45e0 c8 469:      iny
45e1 c028 470:      cpy  #40
45e3 90f9 471:      bcc  .clear
45e5 aef002 472:     ldx  752
45e8 d006 473:      bne  .nocurs3

```

```

45ea a980 474:      lda  #$80
45ec a455 475:      ldy  $55
45ee 915e 476:      sta  ($5e),y
45f0 a900 477: .nocurs3:  lda  #0
45f2 855d 478:      sta  $5d
45f4 a55e 479:      lda  $5e
45f6 18 480:      clc
45f7 6555 481:      adc  $55
45f9 855e 482:      sta  $5e
45fb a55f 483:      lda  $5e+1
45fd 6900 484:      adc  #0
45ff 855f 485:      sta  $5e+1
4601 60 486:      rts
4602 a554 487: .get_adr:  lda  $54
4604 0a 488:      asl
4605 a8 489:      tay
4606 a558 490:      lda  $58
4608 18 491:      clc
4609 79d044 492:      adc  scr_offset,y
460c 855e 493:      sta  $5e
460e a559 494:      lda  $58+1
4610 79d144 495:      adc  scr_offset+1,y
4613 855f 496:      sta  $5e+1
4615 a55e 497:      lda  $5e
4617 18 498:      clc
4618 6555 499:      adc  $55
461a 855e 500:      sta  $5e
461c a55f 501:      lda  $5e+1
461e 6900 502:      adc  #0
4620 855f 503:      sta  $5e+1
4622 60 504:      rts
4623 505: .char_table:
4623 1b1c1d 506:      dc.b 27,28,29,30,31,125,126,127
462b 9c9d9e 507:      dc.b 156,157,158,159,253,254,255
4633 508:
4633 509:
4633 510: putlocal:
4633 a20b 511:      ldx  #11
4635 8e4203 512:      stx  $0342
4638 a200 513:      ldx  #0
463a 8e4803 514:      stx  $0348
463d 8e4903 515:      stx  $0349
4640 4c56e4 516:      jmp  $e456
4643 517: ;
4643 518:
4643 519:
4643 520:
4643 521:
4643 522: ;
4643 523: ; prints: prints string with address in PR_PTR, length by
4643 524: ; ~~~~~
4643 525:
4643 526: echos:

```

```

4643 a000 527:          ldy  #0
4645 b188 528:          lda  (pr_ptr),y      ;ge
4647 f01b 529:          beq  .xpri          ;ex
4649 8d6646 530:         sta  .maxlth
464c c8 531:          iny
464d 8c6546 532:         sty  .tempy
4650 533:
4650 ac6546 534: .pril:          ldy  .tempy
4653 b188 535:          lda  (pr_ptr),y
4655 c900 536:          cmp  #$00
4657 f00b 537:          beq  .xpri
4659 200045 538:         jsr  echo
465c ee6546 539: .prnext:        inc  .tempy
465f ce6646 540:         dec  .maxlth
4662 d0ec 541:          bne  .pril
4664 60 542: .xpri:         rts
4665 543:
4665 01 544: .tempy:         dc.b 1
4666 01 545: .maxlth:        dc.b 1
4667 546:
4667 547:
4667 548:
4667 549:
4667 550: ;~~~~~
4667 551: ;Prints a byte or a card to the screen
4667 552:
4667 553:
4667 554: pr_byte:
4667 a200 555:          ldx  #$00
4669 556: pr_card:
4669 85d4 557:          sta  fr0
466b 86d5 558:          stx  fr0+1
466d a900 559:          lda  #$00
466f 85f2 560:          sta  cix
4671 8d9746 561:         sta  .bytecard
4674 20aad9 562:         jsr  ifp
4677 20e6d8 563:         jsr  fasc
467a a000 564:         ldy  #$00
467c 565: .aprbloop:
467c ee9746 566:         inc  .bytecard
467f b1f3 567:         lda  ($f3),y
4681 08 568:         php
4682 297f 569:         and  #$7f
4684 999846 570:         sta  .bytecard+1,y
4687 c8 571:         iny
4688 572: ;         jsr  echo
4688 28 573:         plp
4689 10f1 574:         bpl  .aprbloop
468b 575:         mea  .bytecard,pr_ptr
468b a997 ----:         lda  #low .bytecard
468d 8588 ----:         sta  pr_ptr
468f a946 ----:         lda  #high .bytecard
4691 8589 ----:         sta  pr_ptr + 1

```

```

4693 204346 576:          jsr  echos
4696 60      577:          rts
4697        578:
4697        579: .bytecard:      ds.b  10
46a1        580:
46a1        581:
46a1        582:
46a1        583:
46a1        584: cr_lf:
46a1 20aa44 585:          jsr  printsi
46a4 9bff   586:          dc.b  $9b,-1
46a6 60     587:          rts
46a7        588:
46a7        589:
46a7        590: ;
46a7        591: ;-----
46a7        592: ; PRINT BYTE
46a7        593: ;-----
46a7 85b8   594: sprint:          sta  $b8
46a9 a904   595:          lda  #$04
46ab 8d4a03 596:          sta  $034a
46ae 86b6   597:          stx  $b6
46b0 84b7   598:          sty  $b7
46b2 a206   599:          ldx  #$06
46b4 bd01e4 600:          lda  $e400+1,x
46b7 85b3   601:          sta  $b3
46b9 bd00e4 602:          lda  $e400,x
46bc 85b2   603:          sta  $b2
46be a5b8   604:          lda  $b8
46c0 20cf46 605:          jsr  .a
46c3 a6b6   606:          ldx  $b6
46c5 a4b7   607:          ldy  $b7
46c7 a90c   608:          lda  #$0c
46c9 8d4a03 609:          sta  $034a
46cc a5b8   610:          lda  $b8
46ce 60     611:          rts
46cf        612: ;
46cf e6b2   613: .a:          inc  $b2
46d1 6cb200 614:          jmp  ($b2)
46d4 60     615:          rts
46d5        616:
46d5        617: ;   Make a length-byte string from an ASCII ($9b termin
46d5        618: ;   ~~~~~
46d5        619:
46d5 a000   620: tostr:      ldy  #0
46d7 b182   621: .tfe:      lda  (s_ptr),y
46d9 f007   622:          beq  .gotend
46db c99b   623:          cmp  #$9b
46dd f003   624:          beq  .gotend
46df c8     625:          iny
46e0 d0f5   626:          bne  .tfe
46e2 8cfd46 627: .gotend:    sty  .lth
46e5 c000   628:          cpy  #0

```

```
46e7 f00c 629:      beq  .rtn
46e9 a000 630:      ldy  #0
46eb b182 631: .loop:   lda  (s_ptr),y
46ed c8   632:      iny
46ee 9184 633:      sta  (d_ptr),y
46f0 ccf4 634:      cpy  .lth
46f3 90f6 635:      bcc  .loop
46f5 a000 636: .rtn:    ldy  #0
46f7 adfd 637:      lda  .lth
46fa 9184 638:      sta  (d_ptr),y
46fc 60   639:      rts
46fd     640:
46fd 01   641: .lth:    dc.b  1
46fe     642:
46fe     643: ;
46fe     644:
46fe     645:
46fe     646: Win_end: ds.b  0
46fe     647: ;
46fe     648: ;
```

End assembly: no errors