

.MLIST SEQ,LOC,BIN
.REM_

IDENTIFICATION

PRODUCT CODE: AC-E484A-MC
PRODUCT NAME: CZTUNAO TM02/TU45 DATA TAPE CREATE
DATE: 25 MAY 1978
MAINTAINER: COMPUTER SPECIAL SYSTEMS
AUTHOR: R. B. BARNES/R. J. COLLINS

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS MANUAL.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED TO THE PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND CAN BE COPIED (WITH INCLUSION OF DIGITAL'S COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DIGITAL.

DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

COPYRIGHT (©) 1975, 1976, 1977, 1978 BY DIGITAL EQUIPMENT CORPORATION

;TABLE OF CONTENTS

PARAGRAPH	SUBJECT	PAGE
1.	ABSTRACT	1
2.	REQUIREMENTS	1
3.	LOADING PROCEDURE	1
4.	STARTING PROCEDURE	1
5.	OPERATION	2
6.	EXAMPLES	3
7.	EXCEPTION	5
8.	EXERCISER USAGE	5
9.	LISTING	

(PAGE 1)

1. ABSTRACT

THIS IS NOT A TEST PROGRAM! IT IS A SUPPLEMENT TO BOTH THE TM11/TU10 AND TMO2/TU45 DATA RELIABILITY PROGRAMS. ITS PURPOSE IS TO ALLOW THE OPERATOR TO CREATE A PAPER TAPE OF WHATEVER DATA PATTERN IS DESIRED, TO BE USED BY THE DATA RELIABILITY PROGRAMS, WHEN DATA PATTERN ZERO (0) IS SELECTED.

2. REQUIREMENTS

- A. ANY PDP-11 PROCESSOR
- B. TELETYPE OR LA30 KEYBOARD
- C. HIGH SPEED OR LOW SPEED PUNCH
- D. 1K OF CORE

3. LOADING PROCEDURE

USE STANDARD BINARY LOADING PROCEDURE

4. STARTING PROCEDURE

THE PROGRAM MAY BE STARTED AT EITHER 200 (8) OR 204 (8). WHEN STARTED AT 200 (8), OPERATING INSTRUCTIONS WILL BE PRINTED BEFORE STARTING INPUT. WHEN STARTED AT 204 (8), INPUT IS IMMEDIATELY STARTED. INPUT START IS NOTED BY AN ASTERISCK (*).

SAMPLE START AT 200: (LOAD 200, PRESS START)

EXTERNAL DATA TAPE CREATE PROGRAM
MAXIMUM OF 377 OCTAL CHARACTERS.
ENTER 3 DIGITS (0-7) FOR EACH CHARACTER.
CR WILL ECHO CR-LF.
CONTROL C ENDS INPUT AT LESS THAN 377.
A SINGLE CHARACTER CORRECTION MAY BE DONE
BY TYPING A SLASH AND RETYPING THE CHARACTER.

* (THE ASTERISCK INDICATES START OF INPUT)

SAMPLE START AT 204: (LOAD 204 PRESS START)

* (START INPUT)

(PAGE 2)

5. OPERATION:

WHEN THE ASTERISCK IS PRINTED AFTER THE START AT 200 OR 204, START INPUTTING CHARACTERS. EACH GROUP OF THREE (3) DIGITS (0-7) EQUALS ONE (1) CHARACTER ON TAPE. ENTER AS MANY 3 DIGIT GROUPS PER THE NUMBER OF CHARACTERS DESIRED IN THE PATTERN. THE PROGRAM WILL ACCEPT UP TO 256 CHARACTERS (377 OCTAL). IF LESS THAN 256 ARE DESIRED, TERMINATE INPUT BY TYPING A CONTROL C. A CARRIAGE RETURN (CR) MAY BE TYPED ANY TIME AND WILL ECHO A CR-LF BUT WILL NOT BE PLACED IN THE DATA PATTERN NOR COUNTED AS AN INPUT CHARACTER. ANY INPUT OTHER THAN AN OCTAL DIGIT (0-7), A CARRIAGE RETURN (CR), OR A CONTROL C WILL BE CONSIDERED ILLEGAL AND BE FLAGGED BY A QUESTION MARK (?). THE ILLEGAL ENTRY IS NEITHER PLACED IN THE DATA PATTERN NOR COUNTED AS A CHARACTER. WHEN INPUT IS COMPLETED (CONTROL C OR 256 CHARACTERS), THE PROGRAM TYPES END OF INPUT AND REQUESTS SELECTION OF HIGH SPEED OR LOW SPEED PUNCH FOR OUTPUT. A RESPONSE OF L TO THIS REQUEST WILL CAUSE OUTPUT ON THE TTY PUNCH, A RESPONSE OF H TO THIS REQUEST WILL OUTPUT ON THE HIGH SPEED PUNCH.

WHEN OUTPUT IS COMPLETE, THE PROGRAM WILL AGAIN REQUEST AN OUTPUT RESPONSE. IF EITHER H OR L IS TYPED, THE SAME DATA PATTERN IS AGAIN OUTPUT. THIS CAN BE REPEATED AS MANY TIMES AS DESIRED. IF NO MORE OUTPUT IS NEEDED, BUT A DIFFERENT PATTERN IS DESIRED, TYPE A CR TO RETURN TO START OF INPUT WHICH WILL BE INDICATED BY AN ASTERISCK (*). THE FIRST CHARACTER PUNCHED ON THE TAPE IS THE NUMBER OF CHARACTERS ON THAT TAPE AND IS NOT USED AS PART OF THE PATTERN BY THE EXERCISERS. THE DATA ON THE TAPE WILL APPEAR AS BYTES IN CORE WHEN USED BY THE EXERCISERS.

(PAGE 3)

6. EXAMPLES

THE FOLLOWING EXAMPLES SHOW THE TAPE LAYOUT RESULTING FROM ITS INPUT AND THE RESULTANT CORE MAP IN THE EXERCISER, (READ THE EXERCISER DOCUMENT TO SEE HOW TO USE THESE PATTERN TAPES)

EXAMPLE 1: LOAD AND START AT 204 (8)

*0001112223334449?377/6(CONTROL C)

END OF INPUT
ASSURE PUNCH IS ON
AND TYPE L FOR LOW SPEED
OR H FOR HIGH SPEED
OR CR FOR RESTART WITH NO PUNCH

L (OUTPUT IS NOW MADE ON TTY PUNCH)

OUTPUT TAPE BIT LAYOUT:

00000.110 (NUMBER OF CHARACTERS IN PATTERN IS 6)
00000.000
01001.001
10010.010 (THE DOT . REPRESENTS THE SPROCKET HOLE)
11011.011
00100.100
11111.110

EXERCISER CORE MAP:

WRITE BUFFER: 0100100100000000
+2: 1101101110010010 (6 CHARACTER = 3 WORDS)
+4: 1111111011011011

(PAGE 4)

EXAMPLE 2: LOAD AND START 204 (8)

*377(CR)
500(CR)
8?G?3(CR)
334266(CONTROL C)

END OF INPUT
ASSURE PUNCH IS ON
AND TYPE L FOR LOW SPEED
OR H FOR HIGH SPEED
OR CR FOR RESTART WITH NO PUNCH

H(OUTPUT IS NOW MADE OF HIGH SPEED PUNCH)

OUTPUT TAPE BIT LAYOUT:

0000.101 (NUMBER OF CHARACTERS IS 5)
1111.111
0100.000
11011.011 (THE DOT REPRESENTS THE SPROCKET HOLE)
0010.110
1000.000

EXERCISER CORE MAP:

WRITE BUFFER: 0100000011111111
+2: 0001011011011011
+4: 0000000010000000

(5 CHARACTER = 2 WORDS + 1 BYTE)

(PAGE 5)

7. EXCEPTION

NOTE THAT THE FIRST DIGIT OF THE 3 DIGITS PER CHARACTER IS LEFT JUSTIFIED. BECAUSE THE TAPE IS ONLY EIGHT (8) BITS WIDE, THE MOST SIGNIFICANT BIT OF THE FIRST DIGIT INPUT FOR EACH CHARACTER IS LOST. SEE EXAMPLE ONE (1), THE FIFTH (5) CHARACTER INPUT IS 444, BUT THE TAPE OUTPUT SHOWS 00100100 BECAUSE THE MOST SIGNIFICANT BIT IS LOST. EXAMPLE 2, THE SECOND (2) CHARACTER INPUT, ALSO SHOWS THIS. REMEMBER, THE FIRST DIGIT INPUT FOR EACH CHARACTER WILL ONLY SHOW THE TWO (2) LEAST SIGNIFICANT BITS OF THAT DIGIT. THE OTHER EXCEPTION TO KEEP IN MIND, IS THAT IF INPUT IS TERMINATED AT SOME NUMBER OF DIGITS NOT DIVISABLE BY THREE (3), THE PARTIAL CHARACTER AT THE END OF THE FIELD WILL BE FILLED TO THE RIGHT WITH ZEROS (0). EXAMPLE 2, THE FIFTH (5) CHARACTER, HAS NOT BEEN COMPLETED BY INPUT BEFORE TERMINATION. SEE THE TAPE LAYOUT, CHARACTER 5 WHICH SHOWS THAT THE TWO (2) LEAST SIGNIFICANT DIGIT POSITIONS ARE FILLED WITH ZEROES (0) TO COMPLETE THE CHARACTER FOR OUTPUT.

8. EXERCISER USAGE

THE EXERCISERS WILL READ THE TAPE CREATED BY DTC AND FILL THEIR ENTIRE WRITE BUFFER WITH REPITIONS OF THE DATA TAPE SO THAT ANY SIZE RECORD CAN BE WRITTEN.

9. LISTING

-

```
259          .LIST  SEQ,LOC,BIN
260          .TITLE DATA TAPE CREATE
261          ;CZTUNAO
262          ;R. BARNES/R. J. COLLINS
263          ;25 MAY 1978
264
265          .ENABLE ABS
266
267          000000          R0=%0
268          000001          R1=%1
269          000002          R2=%2
270          000003          R3=%3
271          000004          R4=%4
272          000005          R5=%5
273          000006          SP=%6
274          000007          PC=%7
275
276          000000          .=0
277          000200          .REPT 200
278          .+2
279          HALT
280          .ENDR
281
282          000200          .=200
283 000200 000167 000774      JMP      START          ;STARTING ADDRESS=200(8) FOR HELP
284
285          000204          .=204
286 000204 000167 001012      JMP      ST1          ;STARTING ADDRESS FOR NO HELP
287
288          001000          .=1000
289          ;CONSTANTS*****
290
291 001000 177560          TKS:   177560          ;LOW SPEED PUNCH
292 001002 177562          TKB:   177562
293 001004 177564          TPS:   177564
294 001006 177566          TPB:   177566
295 001010 177554          PPS:   177554          ;HIGH SPEED PUNCH
296 001012 177556          PPB:   177556
297 001014 177776          PSW:   177776          ;PROGRAM STATUS WORD
298
299          ;BUFFERS*****
300
301 001016 000000          TIB:   0          ;INPUT BUFFER
302 001020 000000          TOB:   0          ;OUTPUT BUFFER
303
```



```
304          001200          . =1200
305          ;PROGRAM START AND HOUSEKEEPING*****
306
307 001200 012777 000340 177606 START: MOV #340,@PSW ;SET TO PRIORITY LEVEL 7
308 001206 012706 000500          MOV #500,SP ;SET STACK TO 500
309 001212 012704 002176          MOV #MSG1,R4
310 001216 004767 000532          JSR PC,TTOUT ;TYPE HELP MESSAGE
311 001222 012777 000340 177564 ST1: MOV #340,@PSW
312 001230 012706 000500          MOV #500,SP
313 001234 005067 177560          CLR TOB
314 001240 005067 177552          CLR TIB ;CLEAR BUFFERS
315 001244 012700 000250          MOV #250,R0 ;SET SIZE IF DATA AREA
316 001250 012702 002754          MOV #DAM40,R2 ;SET START OF AREA TO CLEAR
317 001254 005022          ST2: CLR (R2)+ ;CLEAR DATA AREA
318 001256 005300          DEC R0 ;CLEAR R0 FOR USE AS CHARACTER COUNTER
319 001260 001375          BNE ST2 ;BR IF NOT DONE
320 001262 005001          CLR R1 ;CLEAR R1 FOR USE AS DIGIT POSITION POINTER
321 001264 012702 003017          MOV #DA+1,R2 ;SET START OF DATA AREA
322 001270 004767 000642          ST3: JSR PC,CRLF ;TYPE CR,LF AND *
323
```

```
324                                     :DATA READ FROM TTY*****
325
326 001274 004767 000552          READ: JSR      PC,TTIN      ;GO INPUT DATA
327 001300 122767 000215 1775    CMPB     #215,TIB
328 001306 001007                BNE      RD1          ;BR IF NOT CR
329 001310 012767 000212 177502    MOV      #212,TOB
330 001316 004767 000512          JSR      PC,TOG      ;DO LF
331 001322 000167 177746          JMP      READ        ;GET NEXT DATA
332 001326 122767 000203 177462    RD1:    CMPB     #203,TIB
333 001334 001004                BNE      RD2          ;BR IF NOT CONTROL C
334 001336 005700                TST      RO
335 001340 001753                BEQ      ST3         ;BR IF FIRST INPUT
336 001342 000167 000234          JMP      PUNCH       ;GO TO PUNCH ROUTINE
337 001346 122767 000257 177442    RD2:    CMPB     #257,TIB ;SEE IF RUBOUT
338 001354 001002                BNE      RD2A        ;IF NOT: BR
339 001356 000167 000042          JMP      RUBOUT      ;ELSE RUBOUT LAST ENTRY
340 001362 122767 000260 177426    RD2A:   CMPB     #260,TIB
341 001370 101407                BLOS     RD3         ;BR IF NOT TOO LOW
342 001372 012767 000277 177420    RD2B:   MOV      #277,TOB
343 001400 004767 000430          JSR      PC,TOG      ;TYPE?
344 001404 000167 177664          JMP      READ
345 001410 122767 000267 177400    RD3:    CMPB     #267,TIB
346 001416 103031                BHIS     RD4         ;BR IF NOT TOO HIGH
347 001420 000167 177746          JMP      RD2B
348
```

```

349                                     ;LAST ENTRY RUBOUT ROUTINE*****
350
351 001424 000240 RUBOUT: NOP
352 001426 022701 000001 CMP #1,R1 ;SEE WHERE LAST ENTRY WAS
353 001432 101006 BHI RB0 ;IF POSITION 0: BR
354 001434 103414 BLO RB1 ;IF POSITION 1: BR
355 001436 142712 000300 BICB #300,(R2)
356 001442 005001 CLR R1 ;RESET POSITION POINTER
357 001444 000167 177624 JMP READ ;REENTER
358 001450 142742 000007 RB0: BICB #7,-(R2)
359 001454 005300 DEC R0 ;RESET CHAR POINTER
360 001456 012701 000002 MOV #2,R1 ;RESET POSITION POINTER
361 001462 000167 177606 JMP READ ;REENTER
362 001466 142712 000070 RB1: BICB #70,(R2)
363 001472 012701 000001 MOV #1,R1 ;RESET POSITION POINTER
364 001476 000167 177572 JMP READ ;REENTER
365
366                                     ;POSITION DIGITS TO FORM CHARACTER AND LOAD DATA AREA*****
367
368 001502 016703 177310 RD4: MOV TIB,R3
369 001506 142703 000370 BICB #370,R3 ;R3=STRIPPED DIGIT(0-7)
370 001512 022701 000001 CMP #1,R1 ;TEST POSITION POINTER
371 001516 101016 BHI RD6 ;DO POSITION 2
372 001520 103410 BLO RD5 ;DO POSITION 0
373 001522 000241 CLC
374 001524 106103 ROLB R3
375 001526 106103 ROLB R3 ;POSITION DIGIT 1
376 001530 106103 ROLB R3
377 001532 150312 BISB R3,(R2) ;LOAD DIGIT 1
378 001534 005201 INC R1 ;BUMP POINTER
379 001536 000167 000026 JMP RDEX ;CHECK FOR END
380 001542 150322 RD5: BISB R3,(R2)+ ;LOAD DIGIT 0
381 001544 005001 CLR R1 ;CLEAR POSITION POINTER
382 001546 005200 INC R0 ;BUMP CHARACTER COUNTER
383 001550 000167 000014 JMP RDEX ;LOAD DIGIT
384 001554 000303 RD6: SWAB R3
385 001556 000241 CLC
386 001560 006003 ROR R3 ;POSITION DIGIT 2
387 001562 006003 ROR R3
388 001564 150312 BISB R3,(R2) ;LOAD DIGIT 2 AND BUMP CHARACTER ADDRESS
389 001566 005201 INC R1 ;BUMP POINTER
390 001570 022700 000377 RDEX: CMP #377,R0
391 001574 001402 BEQ PUNCH ;BR IF FILLED DATA AREA
392 001576 000167 177472 JMP READ
393

```

```
394                                     ;TAPE PUNCH ROUTINE*****
395
396 001602 110067 001210      PUNCH:  MOVB   RO,DA           ;LOAD DATA AREA SIZE
397 001606 062700 000100      ADD     #100,RO          ;EXPAND FOR LEADER/TRAILER
398 001612 012701 002754      MOV     #DAM40,R1       ;LOAD PUNCH START ADDRESS
399 001616 012704 002574      PG:    MOV     #MSG2,R4
400 001622 004767 000126      JSR    PC,TTOUT        ;TYPE PUNCH REQUEST(H OR L)
401 001626 004767 000220      PO:    JSR    PC,TTIN    ;GET RESPONSE
402 001632 122767 000314 177156  CMPB   #314,TIB
403 001640 001421              BEQ    P1               ;BR IF LS PUNCH
404 001642 122767 000310 177146  CMPB   #310,TIB
405 001650 001427              BEQ    P2               ;BR IF HS PUNCH
406 001652 122767 000215 177136  CMPB   #215,TIB        ;SEE IF CR
407 001660 001002              BNE   PE               ;IF NOT: BR
408 001662 000167 177334      JMP    ST1              ;ELSE RESTART
409 001666 012767 000277 177124  PE:    MOV     #277,TOB
410 001674 004767 000134      JSR    PC,TOG           ;TYPE?
411 001700 000167 177722      JMP    P0
412
413                                     ;PUNCH TAPE ON LOW SPEED*****
414
415 001704 112167 177110      P1:    MOVB   (R1)+,TOB
416 001710 004767 000120      JSR    PC,TOG           ;PUNCH CHARACTER
417 001714 005300              DEC    RO
418 001716 001372              BNE   P1               ;BR IF NOT DONE
419 001720 116700 001072      MOVB   DA,RO
420 001724 000167 177652      JMP    PUNCH           ;RESTART
421
422                                     ;PUNCH TAPE ON HIGH SPEED*****
423
424 001730 112167 177064      P2:    MOVB   (R1)+,TOB
425 001734 004767 000160      JSR    PC,THG           ;PUNCH CHARACTER
426 001740 005300              DEC    RO
427 001742 001372              BNE   P2               ;BR IF NOT DONE
428 001744 116700 001046      MOVB   DA,RO
429 001750 000167 177626      JMP    PUNCH
430
```

431 ;TTY OUTPUT SUBROUTINE*****

```

432
433 001754 112467 177040      TTOUT:  MOVB  (R4)+,TOB
434 001760 122767 000043 177032  CMPB  #43,TOB
435 001766 001430          BEQ   TEX
436 001770 122767 000045 177022  CMPB  #45,TOB
437 001776 001403          BEQ   TCRLF
438 002000 004767 000030          JSR   PC,TOG
439 002004 000763          BR    TTOUT
440 002006 112767 000015 177004  TCRLF: MOVB  #15,TOB
441 002014 004767 000014          JSR   PC,TOG
442 002020 112767 000012 176772  MOVB  #12,TOB
443 002026 004767 000002          JSR   PC,TOG
444 002032 000750          BR    TTOUT
445 002034 105777 176744          TOG:  TSTB  @TPS
446 002040 100375          BPL   TOG
447 002042 116777 176752 176736  MOVB  TOB,@TPB
448 002050 000207          TEX:  RTS   PC
    
```

449 ;TTY READ SUBROUTINE*****

```

450
451
452 002052 005077 176722      TTIN:  CLR   @TKS
453 002056 005077 176720      CLR   @TKB
454 002062 005067 176730      CLR   TIB
455 002066 105777 176706      TTIN1: TSTB  @TKS
456 002072 100375          BPL   TTIN1
457 002074 017767 176702 176714  MOV   @TKB,TIB
458 002102 105777 176676      TTIN2: TSTB  @TPS
459 002106 100375          BPL   TTIN2
460 002110 116777 176702 176670  MOVB  TIB,@TPB
461 002116 000207          RTS   PC
    
```

462 ;HIGH SPEED PUNCH SUBROUTINE*****

```

463
464
465 002120 105777 176664      THG:  TSTB  @PPS
466 002124 100375          BPL   THG
467 002126 116777 176666 176656  MOVB  TOB,@PPB
468 002134 000207          RTS   PC
    
```

469 ;CR, LF, * TYPE SUBROUTINE*****

```

470
471
472 002136 012767 000215 176654  CRLF:  MOV   #215,TOB
473 002144 004767 177664          JSR   PC,TOG
474 002150 012767 000212 176642  MOV   #212,TOB
475 002156 004767 177652          JSR   PC,TOG
476 002162 012767 000252 176630  MOV   #252,TOB
477 002170 004767 177640          JSR   PC,TOG
478 002174 000207          RTS   PC
479
    
```

```
480 .EVEN
481 ;MESSAGES*****
482
483 002176 022445 054105 042524 MSG1: .ASCII /%EXTERNAL DATA TAPE CREATE PROGRAM%/
484 002204 047122 046101 042040
485 002212 052101 020101 040524
486 002220 042520 041440 042522
487 002226 052101 020105 051120
488 002234 043517 040522 022515
489 002242 040515 044530 052515 .ASCII /MAXIMUM OF 377 OCTAL CHARACTERS%/
490 002250 020115 043117 031440
491 002256 033467 047440 052103
492 002264 046101 041440 040510
493 002272 040522 052103 051105
494 002300 022523
495 002302 047105 042524 020122 .ASCII /ENTER 3 DIGITS(0-7)FOR EACH CHARACTER%/
496 002310 020063 044504 044507
497 002316 051524 030050 033455
498 002324 043051 051117 042440
499 002332 041501 020110 044103
500 002340 051101 041501 042524
501 002346 022522
502 002350 051103 053440 046111 .ASCII /CR WILL ECHO CR-LF%/
503 002356 020114 041505 047510
504 002364 041440 026522 043114
505 002372 045
506 002373 103 047117 051124 .ASCII /CONTROL C ENDS INPUT AT LESS THAN 377.%/
507 002400 046117 041440 042440
508 002406 042116 020123 047111
509 002414 052520 020124 052101
510 002422 046040 051505 020123
511 002430 044124 047101 031440
512 002436 033467 022456
513 002442 020101 044523 043516 .ASCII /A SINGLE CHARACTER CORRECTION MAY BE DONE%/
514 002450 042514 041440 040510
515 002456 040522 052103 051105
516 002464 041440 051117 042522
517 002472 052103 047511 020116
518 002500 040515 020131 042502
519 002506 042040 047117 022505
520 002514 054502 052040 050131 .ASCII /BY TYPING A SLASH AND RETYPING THE CHARACTER.%#/
521 002522 047111 020107 020101
522 002530 046123 051501 020110
523 002536 047101 020104 042522
524 002544 054524 044520 043516
525 002552 052040 042510 041440
526 002560 040510 040522 052103
527 002566 051105 022456 043
528
529 002574 022445 047105 020104 .EVEN
530 002602 043117 044440 050116 MSG2: .ASCII /%END OF INPUT%/
531 002610 052125 045
532 002613 101 051523 051125 .ASCII /ASSURE PUNCH IS ON%/
533 002620 020105 052520 041516
534 002626 020110 051511 047440
535
```

```
536 002634 022516
537 002636 047101 020104 054524 .ASCII /AND TYPE L FOR LOW SPEED%/
538 002644 042520 046040 043040
539 002652 051117 046040 053517
540 002660 051440 042520 042105
541 002666 045
542 002667 117 020122 020110 .ASCII /OR H FOR HIGH SPEED%/
543 002674 047506 020122 044510
544 002702 044107 051440 042520
545 002710 042105 045
546 002713 117 020122 051103 .ASCII /OR CR FOR RESTART WITH NO PUNCH%/
547 002720 043040 051117 051040
548 002726 051505 040524 052122
549 002734 053440 052111 020110
550 002742 047516 050040 047125
551 002750 044103 021445
552
553 .EVEN
554 ;DATA AREA*****
555
556 002754 000000 DAM40: 0
557 003016 003016 .=.+40
558 003016 000000 DA: 0
559
560
561 000001 .END
```

CRLF	002136	PUNCH	001602	RD2B	001372	ST3	001270	TPS	001004
DA	003016	P0	001626	RD3	001410	TCRLF	002006	TTIN	002052
DAM40	002754	P1	001704	RD4	001502	TEX	002050	TTIN1	002066
MSG1	002176	P2	001730	RD5	001542	THG	002120	TTIN2	002102
MSG2	002574	RBO	001450	RD6	001554	TIB	001016	TTOUT	001754
PE	001666	RB1	001466	READ	001274	TKB	001002	.	= 003020
PG	001616	RDEX	001570	RUBOUT	001424	TKS	001000		
PPB	001012	RD1	001326	START	001200	TOB	001020		
PPS	001010	RD2	001346	ST1	001222	TOG	002034		
PSW	001014	RD2A	001362	ST2	001254	TPB	001006		

. ABS. 003020 000

ERRORS DETECTED: 0

,CZTUNA.SEQ/SOL,CZTUNA.P11
RUN-TIME: 2 5 .2 SECONDS
RUN-TIME RATIO: 12/9=1.3
CORE USED: 5K (10 PAGES)