KJ11-A

STACK LIMIT TEST

MD-11-DCKBF-B

EP-DCKBF-B-DL-B

FICHE 1 OF 1

MAR 1977

digital

MADE IN USA

B01

ECF:CB:TADSEQ	00010000	770224	PDP10 411	3 HDR1DCKBFBSEQ	00010000	770224

```
.REM %
```

IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DCKBF-B-D
PRODUCT NAME:	STACK LIMIT TEST STANDARD ON 11/45 KJ11-A OPTION ON 11/40
DATE:	JANUARY 1977
MAINTAINER:	DIAGNOSTIC GROUP
AUTHORS:	;JOHN ADAMS

1.0 ABSTRACT

THIS PROGRAM INCREMENTLY TESTS THE STACK LIMIT FUNCTION.

THE PROGRAM USES THE CONTENTS OF LOCATION 176 AS RHE VALUE
OF THE SWITCHES IF NO HARDWARE SWITCH REGISTER IS FOUND.
THE OPERATOR IS RESPONSIBLE FOR LOADING THE DESIRED
VALUE BEFORE STARTING THE PROGRAM. LOCATION 174 WILL
BE USED AS THE SOFTWARE DISPLAY REGISTER.

2.0 REQUIREMENTS

2.1 EQUIPMENT

BASIC 11/45 SYSTEM OR
11/40 WITH STACK LIMIT OPTION

2.2 STORAGE

THIS PROGRAM USES 0 THRU 17500

2.3 PRELIMINARY PROGRAMS

DOAA THRU DOMA

3.0 LOADING PROCEDURE

LOAD PROGRAM USING ABS LOADER.

4.0 STARTING PROCEDURE

LOAD ADDRESS 200, PRESS START, THE PROGRAM WILL LOOP AND RING
BELL ON PASS COMPLETION.

5.0 OPERATING PROCEDURE

5.1 SWITCH SETTINGS

NONE

5.2 SUBROUTINE ABSTRACTS

5.2.1 SCOPE

SCOPE IS A MOVE PC,R1 AND STORES THE PC+2 IN R1.

5.2.2 HLT

HLT IS A HALT INSTRUCTION.

6.0 ERRORS

ALL ERRORS WILL CAUSE A HALT. TRAP AND INTERRUPT ERRORS WILL
CAUSE A HALT AT VECTOR+2.

```
                                6.1     ERROR RECOVERY

                                        PRESS CONTINUE TO PROCEDE TO NEXT TEST.

                                6.2     ERROR LOOPING

                                        TO LOOP ON AN ERROR, PLACE A BRANCH TO THE PREVIOUS SCOPE
                                        INSTRUCTION IN PLACE OF THE HALT INSTRUCTION.  NOTE THAT IF THE
                                        ERROR IS INTERMITTANT THAT THE TEST WILL DROP THRU THE HALT AND
                                        PROCEED TO THE NEXT TEST.  THEREFORE, TO LOOP THE TEST CONTIN-
                                        UOUSLY REPLACE THE BEQ .+4 INSTRUCTION IMMEDIATELY PRECEEDING
                                        THE HALT WITH A BRANCH BACK TO THE PREVIOUS SCOPE.

                                        TO LOOP ON TRAP FAILURES, PATCH IN THE FOLLOWING ROUTINE AT THE
                                        ADDRESS OF THE TRAP VECTOR.

                                        TRAPVEC:        TRAPVEC+4
                                        TRAPVEC+2:      0
                                        TRAPVEC+4:      012716          ;MOVE SCOPE ADDRESS TO STACK
                                        TRAPVEC+6:      ADDRESS         ;ADDRESS OF PREVIOUS SCOPE
                                        TRAPVEC+10:     000006          ;RETURN TO TEST AT SCOPE

                                        RESTORE ALL LOCATIONS BEFORE PROCEEDING TO NEXT TEST.

                                7.0     RESTRICTIONS

                                        NONE

                                8.0     MISCELLANEOUS

                                        ON TRAP ERRORS THE STACK POINTER (R6) WILL CONTAIN ADDRESS WHERE
                                        THE TRAP OCCURRED.

                                8.1     EXECUTION TIME

                                        EACH PROGRAM TAKES ABOUT 1 MINUTE.

                                9.0     PROGRAM DESCRIPTIONS

                                        THIS IS A TEST OF THE STACK LIMIT REGISTER AND INSURES CORRECT
                                        OPERATION OF THE RED AND YELLOW ZONE BOUNDARIES.  OVERFLOW TRAPS
                                        ARE TESTED FOR ALL VALUES OF THE STACK LIMIT REGISTER.
```

```
                                        .TITLE  MAINDEC-11-DCKBF-B PDP11/40-45 STK LIM TEST
                                ;TEST DCKBFB- STACK LIMIT REGISTER TEST
                                ;THE STACK LIMIT REGISTER ALLOWS THE 'OVERFLOW' BOUNDARIES TO BE CHANGED.
                                ;FOR EXAMPLE IF THE STACK LIMIT REGISTER IS CLEAR THE BOUNDARY IS AT
                                ;400 (YELLOW ZONE) AND 340 (RED ZONE).  IN ALL CASES THE YELLOW ZONE
                                ;BOUNDARY IS AT 400(8) PLUS THE VALUE IN THE STACK LIMIT REGISTER, AND
                                ;THE RED ZONE BEGINS 20(8) WORDS BELOW THE YELLOW ZONE.  THIS TEST
                                ;CHECKS THAT THE STACK LIMIT IS 400 GREATER THAN THE CONTENTS OF THE
                                ;STACK LIMIT REGISTER (CORE PERMITTING), AND CHECKS THE LENGTH OF THE
                                ;YELLOW ZONE AND THE BEGINNING OF THE RED ZONE.

                                ;STARTING PROCEDURE
                                ;       LOAD ADDRESS=200
                                ;       PRESS START
                                ;       BELL WILL RING WHEN TEST IS COMPLETE

                                ;EQUATE STATEMENTS
000000                          R0=%0
000001                          R1=%1
000002                          R2=%2
000003                          R3=%3
000004                          R4=%4
000005                          R5=%5
000006                          SP=%6
000007                          PC=%7

                                ;REGISTER ADDRESSES
177776                                  PSW=177776              ;ADDRESS OF PROCESSER STATUS WORD
177770                                  UBREAK=177770           ;ADDRESS OF PDP11/45 MICRO BREAK REGISTER
177570                                  DSWR=177570             ;ADDRESS CONSOLE SWITCH REGISTER
177564                                  TPS=177564
177566                                  TPB=177566
177570                                  DDISP=177570            ;ADDRESS OF CONSOLE DISPLAY REGISTER
000000                          HLT=HALT
022626                          POP2=22626
010701                          SCOPE=010701                    ;MOVE PC TO R1
000340                          PRTY7=340                       ;PRIORITY LEVEL 7
000004                          ERRVEC=4                        ;ADDRESS OF ERROR VECTOR

000000                                  .=0


000046                                  .=46
000046  002212                          $ENDAD

000174                                  .=174
000174  000000                  DISPREG:0
000176  000000                  SWREG:  0

000200                                  .=200
000200  000167  000604                  JMP     START

000500                                  .=500
                                ;TAGS
000500  000000                  ICNT:   0                       ;CONTAINS PASS COUNT
```

```
000502  000000                  ICNTA:  0
000504  177570                  SWR:    DSWR
000506  177570                  DISPLAY:DDISP
000510  000760                  SPBOT:  760
000512  177774                  SLR:    177774                  ;ADDRESS OF STACK LIMIT REGISTER
000514  177775                  SLH:    177775                  ;HIGH (ODD BYTE)
000516  000000                  TEMP:   0

.       001010                          .=1010
001010  016706  177474          START:  MOV     SPBOT,%6        ;INITIALIZE STACK POINTER
001014  012737  000340  177776          MOV     #PRTY7,@#PSW    ;LOCK OUT INTERRUPTS
                        ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
                        ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
001022  013746  000004          MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
001026  012737  001062  000004  MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
001034  012767  177570  177442  MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
001042  012767  177570  177436  MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
001050  022777  177777  177426  CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
001056  001012                  BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
                                                        ;;AND  THE HARDWARE SWR IS NOT = -1
001060  000403                  BR      65$             ;;BRANCH IF NO TIMEOUT
001062  012716  001070          64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
001066  000002                  RTI
001070  012767  000176  177406  65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
001076  012767  000174  177402          MOV     #DISPREG,DISPLAY
001104  012637  000004          66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR
001110  005067  177364                  CLR     ICNT            ;CLEAR PASS COUNT
001114  005767  177362                  TST     ICNTA
001120  001412                  BEQ     BEGIN
001122  022767  177777  001066  CMP     #-1,$ENDAD+4
001130  001006                  BNE     BEGIN
001132  022767  001477  177340  CMP     #1477,ICNT
001140  001002                  BNE     BEGIN
001142  005067  177334          CLR     ICNTA

001146  016706  177336  BEGIN:  MOV     SPBOT,%6        ;INITIALIZE STACK POINTER
001152  016777  177322  177326  MOV     ICNT,@DISPLAY   ;DISPLAY PASS COUNT
001160  032777  000400  177316  BIT     #400,@SWR       ;LOAD PDP11/45 MICRO BREAK REGISTER
001166  001403                  BEQ     .+10
001170  117737  177310  177770  MOVB    @SWR,@#UBREAK   ;LOAD MICRO BREAK REG WITH SPD-7

                        ;CHECK THAT CP CAN TIME OUT TRAP
001176  012737  001214  000004          MOV     #TORET,@#ERRVEC ;LOAD TIMEOUT TRAP VECTOR
001204  005037  173000                  CLR     @#173000        ;ADDRESS 173000 ALWAYS TIMES OUT ON
                                                                ;DATIP/DATO BUS CYCLE
001210  000000                  HLT                             ;ERROR! FAILED TO TIME OUT TRAP
001212  000755                  BR      BEGIN                   ;LOOP TEST
001214  022626                  TORET:  CMP     (6)+,(6)+       ;RESTORE THE STACK

                        ;TEST THAT THE STACK LIMIT REGISTER CAN BE REFERENCED USING DATI.
                        ;DATIP/DATO
001216  010701                  TO:     SCOPE
001220  012767  001242  176556          MOV     #TOA,ERRVEC     ;LOAD ERROR VECTOR
001226  005067  176554                  CLR     ERRVEC+2
001232  017737  177254  177774          MOV     @SLR,@#177774   ;REFERENCE STACK LIMIT REGISTER
001240  000403                  BR      TOB             ;GO TO NEXT TEST
```

```
001242  022626                  TOA:    POP2
001244  000000                          HLT                     ;ERROR:  CANNOT REFERENCE STACK LIMIT REG.
001246  000763                          BR      TO              ;LOOP TEST IF ERROR

                                ;USING DATI, DATIP/DATOB

001250  012767  001266  176526  TOB:    MOV     #TOC,ERRVEC     ;LOAD ERROR VECTOR
001256  117777  177230  177230          MOVB    @SLR,@SLH       ;REFERENCE ODD BYTE
001264  000403                          BR      TOD
001266  022626                  TOC:    POP2
001270  000000                          HLT                     ;ERROR! CANNOT REFERENCE STACK LIMIT
001272  000766                          BR      TOB             ;USING BYTE INSTRUCTION.

001274  012767  000006  176502  TOD:    MOV     #6,ERRVEC       ;RESTORE ERROR TRAP VECTOR
001302  000400                          BR      T1              ;GO TO NEXT TEST

                                ;TEST THAT EACH BIT OF THE STACK LIMIT REGISTER BITS CAN BE SET
                                ;AND CLEARED. THIS TEST ROTATES A BIT THROUGH THE STACK LIMIT REGISTER.

001304  010701                  T1:     SCOPE
001306  012702  000400                  MOV     #400,R2         ;LOAD TEST VALUE
001312  010277  177174          T1A:    MOV     R2,@SLR         ;LOAD TEST VALUE INTO STACK LIM. REG.
001316  017700  177170                  MOV     @SLR,R0         ;GET RESULT
001322  020002                          CMP     R0,R2           ;CHECK RESULT
001324  001401                          BEQ     .+4             ;BRANCH IF RESULT IS CORRECT
001326  000000                          HLT                     ;ERROR! INCORRECT RESULT. R2 HAS CORRECT
                                                                ;RESULT AND R0 HAS INCORRECT RESULT.
001330  005077  177156                  CLR     @SLR            ;CLEAR STACK LIM. REG.
001334  017700  177152                  MOV     @SLR,R0         ;GET AND CHECK RESULT
001340  001401                          BEQ     .+4
001342  000000                          HLT                     ;ERROR! INCORRECT RESULT
001344  006302                          ASL     R2              ;SHIFT TEST VALUE
001346  103361                          BCC     T1A             ;BRANCH IF NOT DONE
001350  000417                          BR      T2
        001410                          .=1410

                                ;THIS TEST INCREMENTS THE STACK LIMIT REGISTER

001410  010701                  T2:     SCOPE
001412  005067  177[illegible]          CLR     TEMP
001416  005077  177070                  CLR     @SLR            ;CLEAR STACK LIMIT REGISTER
001422  017700  177064          T2A:    MOV     @SLR,R0         ;GET RESULT
001426  020067  177064                  CMP     R0,TEMP         ;CHECK RESULT
001432  001401                          BEQ     .+4
001434  000000                          HLT                     ;ERROR! STACK LIM. REG. WAS INCORRECT DATA
                                                                ;TEMP # CORRECT RESULT
001436  105277  177052                  INCB    @SLH            ;INCREMENT VALUE IN STACK LIM. REG.
001442  105267  177051                  INCB    TEMP+1          ;INCREMENT TEST VALUE
001446  001365                          BNE     T2A             ;BRANCH IF ALL VALUES NOT TESTED

                                ;TEST THAT RESET CLEARS THE STACK LIMIT REGISTER

001450  010701                  T3:     SCOPE
001452  012777  177777  177032        MOV     #-1,@SLR        ;PRESET SLR
001460  017700  177026                  MOV     @SLR,R0         ;SAVE SLR
001464  022700  177400                  CMP     #177400,R0      ;CHECK THAT SLR WAS LOADED
```

```
001470  001431                          BEQ     .+4
001472  000000                          HLT                     ;! SLR FAILED TO LOAD
001474  000005                          RESET                   ;RESET CLEARS SLR
001476  017700  177010                  MOV     @SLR,R0         ;GET RESULT OF RESET AND CHECK RESULT
001502  001402                          BEQ     T4              ;GO TO NEXT TEST IF RESET CLEARED SLR
001504  000000                          HLT                     ;ERROR! RESET FAILED TO CLEAR SLR
001506  000760                          BR      T3              ;LOOP TEST IF ERROR

                                ;TEST THAT THE CLEAR INSTRUCTION CLEARS THE STACK LIMIT REGISTER

001510  010701                  T4:     SCOPE
001512  112777  177777  176774        MOVB    #-1,@SLH        ;PRESET ODD BYTE
001520  017700  176766                  MOV     @SLR,R0         ;GET RESULT
001524  022700  177400                  CMP     #177400,R0      ;CHECK RESULT
001530  001401                          BEQ     .+4
001532  000000                          HLT                     ;ERROR! SLR DID NOT PRESET
001534  005077  176752                  CLR     @SLR
001540  017700  176746                  MOV     @SLR,R0         ;GET RESULT OF CLEAR & BRANCH IF CLEAR
001544  001402                          BEQ     T5              ;GO TO NEXT TEST
001546  000000                          HLT                     ;ERROR! CLR INST FAILED TO CLEAR SLR
001550  000757                          BR      T4              ;LOOP TEST IF ERROR


                                ;TEST THAT AN OVERFLOW ERROR OCCURS FOR ALL STACK LIMIT REGISTER VALUES.
                                ;(PROVIDED CORE IS AVAILABLE).

001552  010701                  T5:     SCOPE
001554  012702  000010                  MOV     #10,R2          ;INITIALIZE STACK VALUE AND
001560  012703  177400                  MOV     #-400,R3        ;STACK LIMIT REGISTER VALUE
001564  062702  000400          T5A:    ADD     #400,R2         ;LOAD NEW STACK VALUE
001570  062703  000400                  ADD     #400,R3         ;AND NEW STACK LIM. REG. VALUE
001574  005037  000000                  CLR     @#0             ;CLEAR ADDRESS 0
001600  016706  176704                  MOV     SPBOT,SP        ;INITIALIZE THE STACK POINTER
001604  012767  002072  176172  T5B:    MOV     #LIMX,ERRVEC    ;LOAD TIME OUT TRAP
001612  016204  177776                  MOV     -2(2),R4        ;SAVE STACK LOCATIONS
001616  016205  177774                  MOV     -4(2),R5        ;EXIT TEST IF EITHER INST TIMES OUT
001622  012767  001646  176154        MOV     #LIMA,ERRVEC    ;LOAD OVERFLOW VECTOR
001630  010206                          MOV     R2,SP           ;LOAD STACK POINTER
001632  010377  176654                  MOV     R3,@SLR         ;LOAD STACK LIM. REG.
001636  016666  177770  177770        MOV     -10(6),-10(6)   ;REFERENCE LIMIT ADDRESS
001644  000401                          BR      .+4             ;SHOULD NOT HAVE TRAPPED
001646  000000                  LIMA:   HLT                     ;ERROR! REFERENCE TO LIMIT ADDRESS
                                                                ;CAUSED AN OVERFLOW
001650  010206                          MOV     R2,SP           ;REPOINT STACK POINTER
001652  012767  001670  176124        MOV     #LIMB,ERRVEC    ;REPOINT OVERFLOW VECTOR
001660  016666  177766  177766        MOV     -12(6),-12(6)   ;REFERENCE FIRST 'YELLOW' ADDRESS
001666  000000                          HLT                     ;ERROR! SHOULD HAVE TRAPPED
001670  005737  000000          LIMB:   TST     @#0             ;WAS IT 'RED' OVERFLOW?
001674  001401                          BEQ     .+4
001676  000000                          HLT                     ;ERROR! A 'RED' OVERFLOW OCCURRED WHEN
                                                                ;A 'YELLOW' ADDRESS WAS REFERENCED.
001700  010206                          MOV     R2,SP           ;REPOINT STACK POINTER
001702  005037  000000                  CLR     @#0             ;CLEAR ADDRESS 0
001706  012767  001724  176070        MOV     #LIMC,ERRVEC    ;REPOINT OVERFLOW VECTOR
001714  016666  177730  177730        MOV     -50(6),-50(6)   ;REFERENCE LAST 'YELLOW' ADDRESS
001722  000000                          HLT                     ;ERROR! SHOULD HAVE TRAPPED
```

```
001724  005737  000000          LIMC:   TST     @#0             ;WAS IT 'RED' OVERFLOW
001730  001401                          BEQ     .+4
001732  000000                          HLT                     ;ERROR! A 'RED' OVERFLOW OCCURRED WHEN
                                                                ;THE LAST 'YELLOW' ADDRESS WAS REFERENCED
001734  005037  000000                  CLR     @#0             ;CLEAR ADDRESS 0
001740  010206                          MOV     R2,SP           ;REPOINT THE STACK POINTER
001742  012767  002010  176034          MOV     #LIMD,ERRVEC    ;REPOINT THE OVERFLOW TRAP
001750  016267  177726  176540          MOV     -52(2),TEMP     ;GET 'RED' LOCATION
001756  005166  177726                  COM     -52(6)          ;REFERENCE 'RED' ADDRESS
001762  000000          LIMCC:  HLT                             ;ERROR! NO OVERFLOW TRAP WHEN 'RED' ADDRESS
001764  000411                          BR      LIMD                    ;WAS REFERENCED
        002010                          .=2010
002010  026267  177726  176500  LIMD:   CMP     -52(2),TEMP     ;WAS INSTRUCTION ABORTED?
002016  001401                          BEQ     .+4
002020  000000                          HLT                     ;ERROR! COM -52(6) WAS ALLOWED TO CHANGE
                                                                ;A 'RED' ADDRESS.
002022  016762  176470  177726          MOV     TEMP,-52(2)     ;RESTORE 'RED' LOCATION IN ANY EVENT
002030  005706                          TST     %6              ;WAS STACK POINTER CHANGED TO 0?
002032  001401                          BEQ     .+4
002034  000000                          HLT                     ;'RED' OVERFLOW DID NOT ASSUME NEW STACK
002036  022737  001762  000000          CMP     #LIMCC,@#0      ;IS RETURN ADDRESS ON NEW STACK?
002044  001401                          BEQ     .+4
002046  000000                          HLT                     ;ERROR! RETURN ADDRESS NOT SAVED ON NEW
                                                                ;STACK
002050  005077  176436                  CLR     @SLR            ;GET READY TO GET NEW VALUES FOR TEST
002054  016706  176430                  MOV     SFBOT,SP        ;INITIALIZE THE STACK POINTER
002060  010462  177776                  MOV     R4,-2(2)        ;RESTORE STACK ADDRESS DATA
002064  010562  177774                  MOV     R5,-4(2)
002070  000635                          BR      T5A             ;GET NEW VALUES
002072  012767  000006  175704  LIMX:   MOV     #6,ERRVEC       ;RESTORE TIME OUT TRAP VECTOR
```

```
002100  005767  176376                  TST     ICNTA                   ;TEST FOR
002104  001013                          BNE     END
002106  022767  177777  000102          CMP     #-1,$ENDAD+4            ;SCRIPT
002114  001417                          BEQ     DONE                    ;CONDITION
002116  026767  175724  175716          CMP     46,42                   ;AND
002124  001003                          BNE     END                     ;SET
002126  005267  176350                  INC     ICNTA                   ;FLAG
002132  000410                          BR      DONE
002134  005267  176340          END:    INC     ICNT
002140  026727  176334  001500          CMP     ICNT,#1500      ;HAVE 1500 PASSES BEEN COMPLETED
002146  001402                          BEQ     DONE
002150  000167  176772                  JMP     BEGIN
002154  012767  000007  175404  DONE:   MOV     #7,TPB          ;RING BELL
002162  105767  175376                  TSTB    TPS
002166  100375                          BPL     .-4
002170  012767  000000  175370          MOV     #0,TPB          ;INSERT
002176  105767  175362                  TSTB    TPS             ;NULL
002202  100375                          BPL     .-4             ;CHARACTER
002204  013702  000042                  MOV     @#42,%2         ;GET DECTAP MONITOR RETURN ADDRESS
002210  001404                          BEQ     DONE1           ;DO NOT RETURN IF (42)=0
002212  004712          $ENDAD: JSR     7,(2)           ;RETURN TO DECTAPE MONITOR
002214  000240                          NOP                     ;ACT11
002216  000240                          NOP                     ;OVERLAY
002220  000240                          NOP                     ;AREA
```

```
        002222  000167  176562          DONE1:  JMP     START           ;RESTART

                000000:                         .END
```

```
BEGIN   001146          ICNT    000500          PSW   = 177776          TPB   = 177566          T2A     001422
DOISP = 17757C          ICNTA   000502          SCOPE = 010701          TPS   = 177564          T3      001450
DISPLA  000506          LIMA    001646          SLH     000514          T0      001216          T4      001510
DISPRE  000174          LIMB    001670          SLR     000512          T0A     001242          T5      001552
DONE    002154          LIMC    001724          SPBOT   00051C          T0B     001250          T5A     001564
DONE1   002222          LIMCC   001762          START   001010          T0C     001266          T5B     001604
DSWR  = 177570          LIMD    002010          SWR     000504          T0D     001274          UBREAK= 17777C
END     002134          LIMX    002072          SWREG   000176          T1      001304          $ENDAD  002212
ERRVEC= 000004          POP2  = 022626          TEMP    000516          T1A     001312          .       = 002226
HLT   = 000000          PRTY7 = 000340          TORET   001214          T2      001410

. ABS.  002226      000
```

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

DCKBFB,DCKBFB/SOL=DCKBFB
RUN-TIME: 3 2 .1 SECONDS
RUN-TIME RATIO: 52/6=7.7
CORE USED: 5K (9 PAGES)