

```

/PAL III
/PART II
/PAGE 11
/ROUTINE TO PACK SIX CHARACTERS IN SYM1, SYM2, SYM3
/LENGTH: 25 REGISTERS
*400

```

| | | |
|--------|-------------|---------------------------------------|
| GETS, | ISZ TEM1 | /SHOULD CHARACTER BE ROTATED? |
| | JMP ROTE | /YES |
| | TAD M2 | /RESET |
| PUT, | DCA TEM1 | /ROTATION COUNTER |
| | TAD PNTR | /WHICH WORD IS THIS? |
| | TAD TEM2 | /ADDRESS IS FORMED FROM CNTR2 AND A |
| | DCA TEM3 | /POINTER, TO GET RELEVANT SYMBOL WORD |
| | TAD CHAR | /GET CHARACTER |
| | TAD I TEM3 | /ADD IN PREVIOUS CHARACTER |
| | DCA I TEM3 | /STORE IT |
| | JMP I NXCHP | /GO GET NEXT CHARACTER |
| ROTE, | ISZ TEM2 | /THE FOURTH? |
| | JMP .+5 | /NO: ROTATE IT |
| | DCA TEM1 | /YES: ARRANGE COUNTERS |
| | CMA | /SO THAT ALL OTHERS ARE |
| | DCA TEM2 | /IGNORED |
| | JMP I NXCHP | /KEEP ACCEPTING CHARS. |
| | TAD CHAR | |
| | CLL RTL | /SIX BITS LEFT |
| | RTL | |
| | RTL | |
| | DCA CHAR | /PUT IT BACK IN CHAR |
| | JMP PUT | /GOT STORE CHARACTER |
| PNTR, | SYM3+1 | /WORD POINTER |
| NXCHP, | NXCH | /NEXT CHARACTER ROUTINE |

/PAGE 12
 /PAL SPACE HANDLER
 /WHEN CALLED, THE SYMBOL TYPE IS IN TYPE
 /AND THE SIGN ASSOCIATED WITH THE SYMBOL IS IN SIGN
 /VALUE OF SYMBOL IS IN SYMVAL.
 /MEMORY REFERENCE INSTRUCTIONS ARE IORED INTO BINW
 /IF CALLED WITH A + OR -, THE ADDRESS IS ADDED OR SUBTRACTED FROM ADDR
 /LENGTH: 28 REGISTERS

| | | |
|-------|-------------|--------------------------------------|
| SPAC, | TAD INFO | /ANYTHING TO PROCESS? |
| | SNA CLA | |
| | POPJ | /NO: EXIT |
| | TAD TYPE | /YES: GET SYMBOL TYPE |
| | SNA CLA | /IS IT ZERO? |
| | JMP INST | /YES, INSTRUCTION |
| CMBN, | TAD SIGN | /GET SIGN |
| | SNA | /IS IT ZERO? |
| | JMP IORC | /COMBINE SYMVAL AND ADDR WITH AN IOR |
| | SPA CLA | /WAS SIGN +? |
| | JMP SUBC | /NO, SUBTRACT |
| ADDC, | TAD SYMVAL | /ADD ROUTINE = ADD SYMVAL |
| | TAD ADDR | /AND ADDR |
| | DCA ADDR | /AND PUT RESULTS BACK IN ADDR |
| | POPJ | /RECURSIVE EXIT |
| SUBC, | TAD SYMVAL | /SUBTRACT ROUTINE = NEGATE |
| | CMA IAC | /SYMVAL |
| | JMP ADDC+1 | /JUMP INTO ADD ROUTINE |
| IORC, | TAD ADDR | /IOR ADDR AND SYMVAL |
| | JMS I IORIN | /GO TO IOR SUBROUTINE |
| | DCA ADDR | /LEAVE RESULTS IN ADDR |
| | POPJ | /RECURSIVE EXIT |
| INST, | CLA CMA | /SET INSTRUCTION FLAG |
| | DCA INSF | |
| | TAD BINW | /IOR BINW AND SYMVAL |
| | JMS I IORIN | |
| | DCA BINW | /LEAVE RESULTS IN BINW |
| | POPJ | /EXIT |

/PAGE 13
 /ROUTINE TO HANDLE CARIAGE RETURNS
 /LENGTH: 40 REGISTERS

| | | |
|------------------|-----------|-------------------------------------|
| CARR, | PUSHJ | /RECURSIVE CALL TO SPACE HANDLER |
| | SPAC | /ADDRESS OF SPACE HANDLER FOR PUSHJ |
| | TAD ADDR | /GET 12 BIT ADDRESS |
| | ISZ INSF | /WAS THERE AN INSTRUCTION? |
| | JMP DPST | /NO |
| | AND PAGE | /LOOK AT PAGE BITS |
| | SZA | /PAGE ZERO REFERENCE? |
| | JMP TPAGE | /NO, CHECK ILLEGAL PAGE REFERENCE |
| IG02, | ISZ ZFLG | /PUT UP PAGE ZERO FLAG |
| | TAD ADDR | /GET ADDRESS |
| | AND SL7 | /SAVE SEVEN BITS |
| | DCA ADDR | /SAVE IT |
| | TAD ZFLG | /IS THIS A PAGE ZERO REFERENCE? |
| | SNA CLA | |
| | TAD C200 | /NO, PICK UP A 1 IN BIT 4 |
| | TAD ADDR | /GET REST OF ADDRESS |
| | TAD BINW | /GET INSTRUCTION PART |
| DPST, | DCA BINW | /THIS IS THE ASSEMBLED WORD |
| | POPJ | /RECURSIVE EXIT |
| TPAGE, | DCA TEM1 | /TEST EQUALITY OF PAGE BITS |
| | TAD PF | /BUT ONLY ON PASS2 OR PASS3 |
| | SNA CLA | /SINCE REFERENCE MAY STILL BE |
| | JMP IG02 | /UNDEFINED ON PASS1 |
| | TAD AACT | /LOOK AT PAGE BITS OF AACT |
| | AND PAGE | |
| | CMA IAC | /NEGATE IT |
| | TAD TEM1 | /COMPARE IT WITH CURRENT REFERENCE |
| | SNA CLA | /ARE THEY EQUAL? |
| | JMP IG02 | /YES, RETURN TO PROCESSOR |
| | TAD PF | /NO, CHANGE PF TO PASS1 |
| | PUSH | /TO FAKE OUT THE ERROR ROUTINE |
| | DCA PF | /SAVE PF ON PUSHDOWN, SET TO 0 |
| | TAD ADDR | /PUT ADDR ON STACK, ALSO |
| | PUSH | |
| | TAD C5 | /CODE FOR IR IS 5 |
| | PUSHJ | /GO TO ERROR ROUTINE |
| | ERROR2 | |
| | POP | /RESTORE PF |
| | DCA PF | |
| | JMP IG02 | /RETURN TO PROCESSOR |
| /SET PAGE 0 FLAG | | |
| ZPAGE, | ISZ ZFLG | /HANDLE "Z" AS A |
| | POPJ | /RETURN |

/PAGE 14
 /POP ROUTINE TO RECOVER ITEMS FROM THE
 /PUSHDOWN LIST
 /CHECKS FOR PUSHDOWN UNDERFLOW
 /LENGTH: 18 REGISTERS

```

POPM, 0
      TAD PSHPN1      /CHECK FOR PUSHDOWN UNDERFLOW
      CMA IAC
      TAD P1BGN      /PSHPN1 SHOULD BE GREATER THAN
                      /P1BGN
      SPA CLA
      JMP POPM2      /POINTER IS OK
      TAD PSHPN1      /UNDERFLOW
      PUSH           /PUT POINTER ON LIST
      TAD SL3        /PDL CODE IS 7
      PUSHJ          /CALL ERROR ROUTINE
      ERROR2
      HLT
      JMP ,=1        /DEAD END
POPM2, CMA           /RESET POINTER
      TAD PSHPN1      /SET IT BACK 1
      DCA PSHPN1
      TAD I PSHPN1     /GET ITEM FROM LIST
      JMP I POPM       /EXIT WITH ITEM IN AC
  
```

/ROUTINE TO FIX ENTIRE SYMBOL TABLE
 /LENGTH: 7 REGISTERS

```

FIXT, DCA INFO      /NO PUNCHING IN MAIN
      TAD PF         /ONLY ON PASS 1
      SZA CLA
      POPJ           /NOT PASS1: EXIT
      TAD SPNT       /ADJUST POINTER
      DCA IAM1
      POPJ           /EXIT
  
```

/ROUTINE TO HANDLE "."
 /LENGTH: 6 REGISTERS

```

POINT, PUSHJ        /REVISION TO TREAT . AS A TERMINATOR
      SPAC          / EXAMPLE = JMP. = JMP .
      TAD AACT      /VALUE OF . IS AACT
      DCA SYMVAL
      ISZ INFO
      CLA IAC        /REPRESENTS VALID INFO
                      /TREAT AS SYMBOL
      DCA TYPE
      JMP I SPACI
  
```

L04,

/PAGE 15
/DOLLAR ROUTINE
/LENGTH: 65 REGISTERS

*600

| | | |
|---------|-------------|---------------------------------|
| DOLLAR, | TAD PF | /WHICH PASS IS THIS? |
| | SZA SMA CLA | /PASS 1 OR PASS3? |
| | JMP EPASS2 | /NOPE, IT'S PASS2 |
| | CLA DSR | |
| | RAR | |
| | SNL CLA | /LOOK AT BIT 11 |
| | JMP .+3 | /IT IS 0; FORGET IT |
| | TAD BBB | /IT IS 1; SET TO PUNCH |
| | DCA DDD | |
| | TAD CRLF | /CR,LF |
| | JMS I EEE | |
| | JMS I LDTI | /OUTPUT LEADER-TRAILER |
| | TAD C377 | /OUTPUT A RUBOUT FOR DDT |
| | JMS I DDD | |
| | DCA LOW | /INITIALIZE SORT ROUTINE |
| | DCA LOW+1 | |
| | DCA LOW+2 | |
| ELOOP, | TAD CRLF | /CARRIAGE RETURN, LINE FEED |
| | JMS I EEE | |
| | JMP SORT | /GET A SYMBOL FROM SORT |
| ELUP2, | TAD I TEM3 | /GET FIRST WORD OF SYMBOL |
| | SPA | /WAS IT DEFINED? |
| | JMP UASYM | /NO, BIT0 IS 1, ERROR |
| | JMS I EEE | /YES, OUTPUT FIRST WORD |
| | ISZ TEM3 | /INCREMENT POINTER TO NEXT WORD |
| | TAD I TEM3 | /PICK IT UP |
| | JMS I EEE | /TYPE IT OUT |
| | ISZ TEM3 | /POINTER TO THIRD WORD |
| | TAD I TEM3 | /PICK IT UP |
| | JMS I EEE | /TYPE IT |
| | JMS I EEE | /TYPE TWO SPACES |
| | ISZ TEM3 | /INCREMENT POINTER TO VALUE |
| | TAD I TEM3 | /GET VALUE OF SYMBOL |
| | JMS I OCTI | /GO TO OCTAL OUTPUT ROUTINE |
| | JMP ELOOP | /RETURN FOR MORE SYMBOLS |
| OUT1, | TAD C204 | /OUTPUT AN EOT FOR DDT |
| | JMS I DDD | |
| | TAD CRLF | /TYPE A CRLF FOR DDT |
| | JMS I EEE | |
| | TAD C377 | /TYPE A RUBOUT FOR DDT |
| | JMS I DDD | |
| | JMS I LDTI | /MORE LEADER-TRAILER |
| | JMP I NEXI | /GO TO NEXT PASS |

/PAGE 16

| | | |
|---------|------------|---------------------------|
| EPASS2, | CLL | /CLEAR LINK FOR BIN PUNCH |
| | TAD CKSM | /PUNCH CHECKSUM |
| | JMS I PUNI | |
| | JMS I LDTI | /PUNCH LEADER-TRAILER |
| | JMP I NEXI | /GO TO NEXT PASS |
| UASYM, | CLA | |
| | TAD TEM3 | /GET ADDRESS OF SYMBOL |
| | PUSH | /ON PUSHDOWN LIST |
| | TAD TEM3 | /GET CURRENT LOCATION |
| | TAD C3 | /COUNTER WHEN THE SYMBOL |
| | DCA TEM3 | /WAS LAST SEEN |
| | TAD I TEM3 | /PUT INTO AACT |
| | DCA AACT | |
| | IAC | |
| | PUSHJ | /GO TO ERROR ROUTINE |
| | ERROR1 | /CODE FOR UA IS 1 |
| | ISZ TEM2 | |
| | TAD I TEM2 | /RESET VALUE TO HIGHEST |
| | AND PAGE | /ADDRESS ON PAGE |
| | TAD SL7 | /SET LOW 7 BITS TO 1 |
| | DCA I TEM2 | |
| | JMP ELOOP | /RETURN FOR MORE SYMBOLS |

/PAGE 17
/SYMBOL SORTING ROUTINE
/LENGTH: 63 REGISTERS

| | | |
|-------|--------------|---|
| SORT, | DCA INFO | /INDICATOR TELLS IF ANY SYMBOLS WERE SORTED |
| | TAD IAM1 | /POINTER TO BEGINNING OF SYMBOL TABLE |
| | DCA PNT1 | /PUT IT IN ADDRESS POINTER |
| | CMA | /SET HIGH SYMBOL TO 7777 7777 7777 |
| | DCA HIGH | |
| | CMA | |
| | DCA HIGH+1 | |
| | CMA | |
| | DCA HIGH+2 | |
| LCHK, | TAD PNT1 | /HAVE WE CHECKED ALL OF SYMBOL TABLE? |
| | CLL CMA IAC | /COMPARE POINTER WITH ANOTHER POINTER |
| | TAD SPNT | /TO THE END OF THE SYMBOL TABLE |
| | SNA CLA | |
| | JMP ADUT | /ALL THROUGH |
| | TAD LOWPNT | /SET UP POINTERS FOR SYMBOL COMPARISON |
| | DCA SORT1 | /TO SEE IF SYM IS GREATER THAN LOW |
| | TAD PNT1 | /GET ADDRESS OF CURRENT SYMBOL |
| | DCA SORT2 | |
| | JMS I SYMCM1 | /GO TO SYMBOL COMPARISON ROUTINE |
| | JMP INC | /SYM IS EQUAL TO LOW |
| | JMP INC | /SYM IS LESS THAN LOW |
| | TAD HIPNT | /SYM IS GREATER THAN LOW |
| | DCA SORT2 | /NOW SET UP POINTERS FOR CHECK3 ROUTINE |
| | TAD PNT1 | /TO SEE IF SYM IS LESS THAN HIGH |
| | DCA SORT1 | |
| | JMS I SYMCM1 | /GO TO SYMBOL COMPARISON ROUTINE |
| | JMP INC | /SYM IS EQUAL TO HIGH |
| | JMP INC | /SYM IS GREATER THAN HIGH |
| | ISZ INFO | /SYM IS LESS THAN HIGH |
| | TAD PNT1 | /GET ADDRESS OF SYMBOL |
| | DCA SORT1 | /IN AN INDEX REGISTER |
| | TAD I SORT1 | /PUT THE SYMBOL IN HIGH |
| | AND MASK | /GET RID OF THE UNDEFINED BIT |
| | DCA HIGH | |
| | TAD SORT1 | /PUT ADDRESS OF SYMBOL IN ADDR FOR DOLL |
| | DCA TEM3 | |
| | TAD I SORT1 | /FINISH PUTTING SYMBOL IN HIGH |
| | DCA HIGH+1 | |
| | TAD I SORT1 | |
| | DCA HIGH+2 | |

/PAGE 18

| | | |
|---------|------------|--|
| INC, | TAD C4 | /INCREMENT POINTER |
| | TAD PNT1 | /FOR NEXT COMPARE |
| | DCA PNT1 | /PUT IT BACK IN PNT1 |
| | JMP LCHK | /GO LOOK AT NEXT SYMBOL |
| AOUT, | TAD HIGH | /ALL THROUGH, NOW PUT HIGH INTO LOW |
| | DCA LOW | |
| | TAD HIGH+1 | /HIGH+1 GOES INTO LOW+1 |
| | DCA LOW+1 | |
| | TAD HIGH+2 | /HIGH+2 GOES INTO LOW+2 |
| | DCA LOW+2 | |
| | TAD INFO | /WERE ANY SWITCHES MADE IN THIS PASS? |
| | SNA CLA | |
| | JMP OUT1 | /NO: ALL DONE |
| | JMP ELUP2 | |
| HIGH, | 0 | /STORAGE FOR HIGH SYMBOL |
| | 0 | |
| | 0 | |
| LOW, | 0 | /STORAGE FOR LOW SYMBOL |
| | 0 | |
| | 0 | |
| PNT1, | 0 | /STORAGE FOR ADDRESS OF CURRENT SYMBOL |
| LOWPNT, | LOW=1 | /POINTER USED WITH AUTO INDEX REGISTER |
| HIPNT, | HIGH=1 | /POINTER USED WITH AUTO INDEX REGISTER |

L06, /END OF PAGE

PAUSE