

|      |        |        |        |       |                              |
|------|--------|--------|--------|-------|------------------------------|
| 1531 | 001370 | 000000 | 000117 |       |                              |
| 1532 | 001371 | 000000 | 000116 |       |                              |
| 1533 | 001372 | 000000 | 000124 |       |                              |
| 1534 | 001373 | 000000 | 000111 |       |                              |
| 1535 | 001374 | 000000 | 000116 |       |                              |
| 1536 | 001375 | 000000 | 000125 |       |                              |
| 1537 | 001376 | 000000 | 000105 |       |                              |
| 1538 | 001376 | 000000 | 000305 |       |                              |
| 1539 | 001377 | 000000 | 000000 |       |                              |
| 1540 |        |        | 000021 | 22640 | ERRCN==>                     |
| 1541 |        |        |        | 22660 | IFN                          |
| 1542 |        |        |        | 22680 | FUNCTS,<                     |
| 1543 | 001400 | 000000 | 000125 | 22700 | DCE*UF*                      |
| 1544 | 001401 | 000000 | 000116 |       | DCL*UNDEFINED USER FUNCTION" |
| 1545 | 001402 | 000000 | 000104 |       |                              |
| 1546 | 001403 | 000000 | 000105 |       |                              |
| 1547 | 001404 | 000000 | 000106 |       |                              |
| 1548 | 001405 | 000000 | 000111 |       |                              |
| 1549 | 001406 | 000000 | 000116 |       |                              |
| 1550 | 001407 | 000000 | 000105 |       |                              |
| 1551 | 001410 | 000000 | 000104 |       |                              |
| 1552 | 001411 | 000000 | 000048 |       |                              |
| 1553 | 001412 | 000000 | 000125 |       |                              |
| 1554 | 001413 | 000000 | 000123 |       |                              |
| 1555 | 001414 | 000000 | 000105 |       |                              |
| 1556 | 001415 | 000000 | 000122 |       |                              |
| 1557 | 001416 | 000000 | 000040 |       |                              |
| 1558 | 001417 | 000000 | 000106 |       |                              |
| 1559 | 001420 | 000000 | 000125 |       |                              |
| 1560 | 001421 | 000000 | 000116 |       |                              |
| 1561 | 001422 | 000000 | 000105 |       |                              |
| 1562 | 001423 | 000000 | 000124 |       |                              |
| 1563 | 001424 | 000000 | 000111 |       |                              |
| 1564 | 001425 | 000000 | 000117 |       |                              |
| 1565 | 001426 | 000000 | 000116 |       |                              |
| 1566 | 001426 | 000000 | 000316 |       |                              |
| 1567 | 001427 | 000000 | 000000 |       |                              |
| 1568 |        |        | 000022 | 22720 | ERRUF==>                     |
| 1569 |        |        |        |       |                              |
| 1570 |        |        |        | 22760 | PAGE                         |

|      |        |        |        |       |  |
|------|--------|--------|--------|-------|--|
| 1571 |        |        |        | 22780 | SUBTTL LOW SEGMENT -- RAM -- IE THIS STUFF IS NOT CONSTANT |
| 1572 |        |        |        | 22800 | ;  |
| 1573 |        |        |        | 22820 | ; THIS IS THE "VOLATILE" STORAGE AREA AND NONE OF IT       |
| 1574 |        |        |        | 22840 | ; CAN BE KEPT IN RAM, ANY CONSTANTS IN THIS AREA CANNOT    |
| 1575 |        |        |        | 22860 | ; BE KEPT IN A RAM, BUT MUST BE LOADED IN BY THE           |
| 1576 |        |        |        | 22900 | ; PROGRAM INSTRUCTIONS IN RAM.                             |
| 1577 |        |        |        | 22900 | ;  |
| 1578 |        |        |        |       |  |
| 1579 | 001430 | 000000 | 000054 | 22940 | BUFMIN: 44   |
| 1580 |        |        |        | 22960 |  |
| 1581 |        |        |        | 22980 |  |
| 1582 |        |        |        | 23000 |  |
| 1583 | 001431 |        |        | 23020 | BUF: BLOCK BUFLen  |
| 1584 |        |        |        | 23040 |  |
| 1585 |        |        |        | 23060 |  |
| 1586 |        |        |        | 23080 |  |
| 1587 |        |        |        | 23100 |  |
| 1588 |        |        |        | 23120 |  |
| 1589 |        |        |        | 23140 |  |
| 1590 |        |        |        | 23160 | IFN LPTSW,<  |
| 1591 |        |        |        | 23180 | LPTPOS: BLOCK 1  |
| 1592 |        |        |        | 23200 | PRTFLG: BLOCK 1  |
| 1593 |        |        |        | 23220 |  |
| 1594 |        |        |        | 23240 | IFN CONTRH,<   |
| 1595 | 001541 |        |        | 23260 | CNTWFL: BLOCK 1  |
| 1596 | 001542 |        |        | 23280 | DIMFLG: BLOCK 1  |
| 1597 |        |        |        | 23300 |  |
| 1598 |        |        |        | 23320 |  |
| 1599 |        |        |        | 23340 |  |
| 1600 |        |        |        | 23360 |  |
| 1601 |        |        |        | 23380 | IFN STRING,<   |
| 1602 | 001543 |        |        | 23400 | VALTYP: BLOCK 1  |
| 1603 |        |        |        | 23420 |  |
| 1604 | 001544 |        |        | 23435 | UPRTYP:  |
| 1605 |        |        |        | 23435 |  |
| 1606 |        |        |        | 23437 |  |
| 1607 | 001544 |        |        | 23440 | DORES: BLOCK 1   |
| 1608 |        |        |        | 23460 |  |
| 1609 |        |        |        | 23480 |  |
| 1610 |        |        |        | 23500 |  |
| 1611 | 001545 |        |        | 23520 | MEMSIZ: BLOCK 2  |
| 1612 | 001547 |        |        | 23540 | TEMPPT: BLOCK 2  |
| 1613 |        |        |        | 23560 |  |
| 1614 | 001551 |        |        | 23580 | TEMPST: BLOCK STRSIZ*NUMTMP                                |
| 1615 | 001576 |        |        | 23600 | DSCTMP: BLOCK STRSIZ                                       |
| 1616 | 001573 |        |        | 23620 | PHCTOP: BLOCK 2  |
| 1617 |        |        |        | 23640 | IFN LENGTH*STRING,<  |
| 1618 | 001575 |        |        | 23660 | TEMP3: BLOCK 2   |
| 1619 |        |        |        | 23680 |  |
| 1620 |        |        |        | 23700 |  |
| 1621 |        |        |        | 23720 |  |
| 1622 |        |        |        | 23740 |  |
| 1623 |        |        |        | 23760 | IFN LENGTH,<   |

|      |         |       |                  |   |
|------|---------|-------|------------------|---|
| 1624 | 001577' | 23780 | DATLIN: BLOCK 2  | DATA LINE # -- REMEMBER FOR ERRORS            |
| 1625 | 001601' | 23800 | SUBFLG: BLOCK 1> | IFLAG WHETHER SUBSCRIPTED VARIABLE ALLOWED    |
| 1626 |         | 23820 |                  | IF "FOR" AND USER-DEFINED FUNCTION            |
| 1627 |         | 23840 |                  | IF POINTER FETCHING TURN                      |
| 1628 |         | 23860 |                  | IF THIS ON BEFORE CALLING PTRGET              |
| 1629 |         | 23880 |                  | IFSO ARRAYS WON'T BE DETECTED.                |
| 1630 |         | 23900 |                  | IFSTKINI AND PTRGET CLEAR IT.                 |
| 1631 | 001602' | 23920 | FLG1NP: BLOCK 1  | IFLAGS WHETHER WE ARE DOING INPUT             |
| 1632 |         | 23940 |                  | FOR A READ                                    |
| 1633 | 001603' | 23960 | TEMP: BLOCK 2    | IFTEMPORARY FOR STATEMENT CODE                |
| 1634 |         | 23980 |                  | IFNEWSTT SAVES (H,L) HERE FOR INPUT AND "C    |
| 1635 |         | 24000 |                  | IF"LET" SAVES NUMERIC VARIABLE                |
| 1636 |         | 24020 |                  | IFPOINTERS HERE FOR "FOR"                     |
| 1637 |         | 24040 |                  | IF"NEXT" SAVES ITS TEXT POINTER HERE          |
| 1638 |         | 24060 |                  | IFCLEARC SAVE (H,L) HERE                      |
| 1639 | 001605' | 24080 | TEMP2: BLOCK 2   | IFFORMULA EVALUATOR TEMP                      |
| 1640 |         | 24100 |                  | IFMUST BE PRESERVED BY OPERATORS              |
| 1641 |         | 24120 |                  | IFUSED IN EXTENDED BY FOUT                    |
| 1642 |         | 24140 |                  | IFARRAY VARIABLE HANDLER TEMPORARY            |
| 1643 | 001607' | 24160 | CURLIN: BLOCK 2  | IFCURRENT LINE #                              |
| 1644 |         | 24180 |                  | IFSET TO 65535 WHEN DIRECT STATEMENTS EXECUTE |
| 1645 |         | 24200 | IFN LENGTH,<     |   |
| 1646 | 001611' | 24220 | OLDLIN: BLOCK 2  | IFOLD LINE NUMBER                             |
| 1647 | 001613' | 24240 | OLDTXT: BLOCK 2> | IFOLD TEXT POINTER                            |
| 1648 |         | 24260 |                  | IFPOINTS AT STATEMENT TO BE EXECUTED NEXT     |
| 1649 | 001615' | 24280 | STKTOP: BLOCK 2  | IFTOP LOCATION TO USE FOR THE STACK           |
| 1650 |         | 24300 |                  | IFINITIALLY SET UP BY INIT                    |
| 1651 |         | 24320 |                  | IFACCORDING TO MEMORY SIZE                    |
| 1652 |         | 24340 |                  | IFTO ALLOW FOR 50 BYTES OF STRING SPACE,      |
| 1653 |         | 24360 |                  | IFCHANGED BY A CLEAR COMMAND WITH             |
| 1654 |         | 24380 |                  | IFAN ARGUMENT.                                |
| 1655 | 001617' | 24400 | TXTTAB: BLOCK 2  | IFPOINTER TO BEGINNING OF TEXT                |
| 1656 |         | 24420 |                  | IFDOESN'T CHANGE AFTER BEING                  |
| 1657 |         | 24440 |                  | IFSETUP BY INIT.                              |
| 1658 | 001621' | 24460 | VARTAB: BLOCK 2  | IFPOINTER TO START OF SIMPLE                  |
| 1659 |         | 24480 |                  | IFVARIABLE SPACE                              |
| 1660 |         | 24500 |                  | IFUPDATED WHENEVER THE SIZE OF THE            |
| 1661 |         | 24520 |                  | IFPROGRAM CHANGES, SET TO (TXTTAB)            |
| 1662 |         | 24540 |                  | IFBY SCRATCH ("NEW").                         |
| 1663 | 001623' | 24560 | ANYTAB: BLOCK 2  | IFPOINTER TO BEGINNING OF ARRAY               |
| 1664 |         | 24580 |                  | IFTABLE                                       |
| 1665 |         | 24600 |                  | IFINCREMENTED BY 6 WHENEVER                   |
| 1666 |         | 24620 |                  | IFA NEW SIMPLE VARIABLE IS FOUND, AND         |
| 1667 |         | 24640 |                  | IFSET TO (VARTAB) BY CLEARC.                  |
| 1668 | 001625' | 24660 | STREND: BLOCK 2  | IFEND OF STORAGE IN USE                       |
| 1669 |         | 24680 |                  | IFINCREASED WHENEVER A NEW ARRAY              |
| 1670 |         | 24700 |                  | IFOR SIMPLE VARIABLE IS ENCOUNTERED           |
| 1671 |         | 24720 |                  | IFSET TO (VARTAB) BY CLEARC.                  |
| 1672 | 001627' | 24740 | DATPTR: BLOCK 2  | IFPOINTER TO DATA, INITIALIZED TO POINT       |
| 1673 |         | 24760 |                  | IFAT THE ZERO IN FRONT OF (TXTTAB)            |
| 1674 |         | 24780 | IFE LENGTH=2,<   |   |
| 1675 | 001631' | 24800 | TRCFLG: BLOCK 1> | IFB MEANS NO TRACE IN PROGRESS                |
| 1676 |         |       |                  |   |

|      |         |       |                            |  |
|------|---------|-------|----------------------------|--|
| 1677 |         | 24840 | IFTE FLOATING ACCUMULATOR  |  |
| 1678 |         | 24860 | IFE LENGTH=2,<             |  |
| 1679 | 001632' | 24880 | BLOCK 1                    | IF [TEMPORARY LEAST SIGNIFICANT BYTE]        |
| 1680 | 001633' | 24900 | OFACLO: BLOCK 4>           | IF [FOUR LOWEST ORDERS FOR DOUBLE PRECISION] |
| 1681 | 001637' | 24920 | FACLO: BLOCK 3             | IF [LOW ORDER OF MANTISSA]                   |
| 1682 |         | 24940 |                            | IF [MIDDLE ORDER OF MANTISSA]                |
| 1683 |         | 24960 |                            | IF [HIGH ORDER OF MANTISSA]                  |
| 1684 | 001642' | 24980 | FAC: BLOCK 2               | IF [EXPONENT]                                |
| 1685 |         | 25000 |                            | IF [TEMPORARY COMPLEMENT OF SIGN IN MSB]     |
| 1686 |         | 25020 | IFE LENGTH=2,<             |  |
| 1687 | 001644' | 25030 | BLOCK 1                    | IF [TEMPORARY LEAST SIGNIFICANT BYTE]        |
| 1688 | 001645' | 25040 | ARGLO: BLOCK 7             | IF [LOCATION OF SECOND ARGUMENT FOR DOUBLE   |
| 1689 | 001654' | 25060 | ARG: BLOCK 1>              | IF PRECISION]                                |
| 1690 | 001655' | 25080 | FBUFFR: BLOCK 13           | IF BUFFER FOR FOUT                           |
| 1691 | 001672' | 25100 | IFE LENGTH=2,<BLOCK 35=13> |  |
| 1692 |         | 25120 | PAGE                       |  |

```

1693 SUBTTL TEXT CONSTANTS FOR PRINT OUT
1694 ;
1695 ; NEEDED FOR MESSAGES IN ALL VERSIONS
1696 ; MUST BE STORED ABOVE DSGTMP UN ELSE STRLIT
1697 ; WILL COPY THEM BEFORE STRPAT PRINTS THEM, THIS IS BAD, SINCE IF THE
1698 ; USER IS OUT OF STRING SPACE BASIC WILL LOOP GETTING "OUT OF STRING SPACE"
1699 ; ERRORS,
1700 ;
1701 ; IFN LENGTH=2,<
1702 ERR: DC" ERROR"
1703 ;
1704 001720' 000000 000040
1705 001721' 000000 000111
1706 001722' 000000 000116
1707 001723' 000000 000040
1708 001723' 000000 000240
1709 001724' 000000 000000
1710 001725' 000000 000015
1711 001726' 000000 000012
1712 001727' 000000 000117
1713 001730' 000000 000113
1714 001730' 000000 000313
1715 001731' 000000 000015
1716 001732' 000000 000012
1717 001733' 000000 000000
1718 ;
1719 001734' 000000 000015
1720 001735' 000000 000012
1721 001736' 000000 000102
1722 001737' 000000 000122
1723 001740' 000000 000105
1724 001741' 000000 000101
1725 001742' 000000 000113
1726 001742' 000000 000313
1727 001743' 000000 000000
1728 ;
1729 ;
25140 SUBTTL TEXT CONSTANTS FOR PRINT OUT
25160 ;
25180 ;
25200 ; MUST BE STORED ABOVE DSGTMP UN ELSE STRLIT
25220 ; WILL COPY THEM BEFORE STRPAT PRINTS THEM, THIS IS BAD, SINCE IF THE
25240 ; USER IS OUT OF STRING SPACE BASIC WILL LOOP GETTING "OUT OF STRING SPACE"
25260 ; ERRORS,
25280 ;
25300 ; IFN LENGTH=2,<
25320 ERR: DC" ERROR"
25340 ;
25360 INTXT: DC" IN "
25380 ;
25400 REDDY: ACRLF
25420 DC"OK"
25440 ACRLF
25460 ;
25480 IFN LENGTH,<
25500 BRKXT: ACRLF
25520 DC"BREAK"
25540 ;
25560 PAGE
  
```

```

1730 SUBTTL GENERAL STORAGE MANAGEMENT ROUTINES
1731 ;
1732 ; FIND A FOR ENTRY ON THE STACK WITH THE VARIABLE POINTER
1733 ; PASSED IN [D,E],
1734 ;
1735 001744' 001000 000041
1736 001745' 000000 000004
1737 001746' 000000 000720
1738 ;
1739 001747' 001000 000071
1740 001750' 001000 000176
1741 001751' 001000 000043
1742 001752' 001000 000376
1743 001753' 000000 000001
1744 001754' 001000 000300
1745 ;
1746 ;
1747 ;
1748 ;
1749 001755' 001000 000116
1750 001756' 001000 000043
1751 001757' 001000 000106
1752 001760' 001000 000043
1753 001761' 001000 000345
1754 001762' 001000 000151
1755 001763' 001000 000140
1756 001764' 001000 000172
1757 001765' 001000 000263
1758 001766' 001000 000353
1759 001767' 001000 000312
1760 001770' 000000 001774
1761 001771' 000000 001745
1762 001772' 001000 000353
1763 001773' 001000 000347
1764 001774' 001000 000001
1765 001775' 000000 000015
1766 001776' 000000 001770
1767 001777' 001000 000341
1768 002000' 001000 000310
1769 002001' 001000 000011
1770 002002' 001000 000303
1771 002003' 000000 001750
1772 002004' 000000 001775
1773 ;
1774 ;
1775 ;
1776 ;
1777 ;
1778 ;
1779 ;
1780 ;
1781 ;
1782 ;
25600 SUBTTL GENERAL STORAGE MANAGEMENT ROUTINES
25620 ;
25640 ; FIND A FOR ENTRY ON THE STACK WITH THE VARIABLE POINTER
25660 ; PASSED IN [D,E],
25680 ;
25700 PNOFOR: LXI H,4*SCODE ;IGNORING EVERYONES "NEWSTT"
25720 ;
25740 LOOPER: MOV SP ;AND THE RETURN ADDRESS OF THIS
25760 INX A,M ;SEE WHAT TYPE OF THING IS ON THE STACK
25780 CPI H ;IS THIS STACK ENTRY A FOR?
25800 ;
25820 ;
25840 IFN LENGTH,< ;NO SO OK
25860 PUSHM ;GET VARIABLE NAME
25880 XTHL>
25900 IFN LENGTH,<
25920 MOV C,H
25940 INX H ;DO EQUIVALENT OF PUSHM / XTHL
25960 MOV B,M
25980 INX H
26000 ;
26020 MOV L,C ;PUT H ON
26040 MOV M,B ;PUSH B / XTHL IS SLOWER
26060 MOV A,D ;FOR THE "NEXT" STATEMENT WITHOUT AN ARGUMENT
26080 ORA E ;WE MATCH ON ANYTHING
26100 XCHG ;MAKE SURE WE RETURN [D,E]
26120 JZ POPGOF ;POINTING TO THE VARIABLE
26140 ;
26160 XCHG>
26180 COMPAR
26200 POPGOF: LXI B,SCODE+13 ;TO WIPE OUT A FOR ENTRY
26220 ;
26240 POP H ;IF FOR MATCHES GOOD
26260 RZ ;TRY THE NEXT ONE
26280 ;
26300 ;
26320 ;
26340 ;
26360 ; [H,L] = DESTINATION OF HIGH ADDRESS
26380 ; [D,E] = LOW ADDRESS TO BE TRANSFERRED
26400 ; [B,C] = HIGH ADDRESS TO BE TRANSFERRED
26420 ;
26440 ; A CHECK IS MADE TO MAKE SURE A REASONABLE AMOUNT
26460 ; OF SPACE REMAINS BETWEEN THE TOP OF THE STACK AND THE HIGHEST LOCATION
  
```

```

1783      26480      ; TRANSFERRED INTO
1784      26500      ;
1785      26520      ; ON EXIT [M,L]=[D,E]=LOW [B,C]=LOCATION LOW HAS MOVED INTO
1786      26540      ;
1787
1788      002007' 001000 000315      26580      BLTU:   CALL      REASON      ;CHECK DESTINATION TO MAKE
1789      002008' 000000 002045'
1790      002007' 000000 002003'
1791
1792      002010' 001000 000305      26600      BLTUC:   PUSH      B      ;SURE THE STACK WON'T BE OVERRUN
1793      002011' 001000 000343      26620      XTHL      ;EXCHANGE [B,C] AND [H,L]
1794      002012' 001000 000301      26640      POP       B
1795      002013' 001000 000307      26660      BLTLOP:  COMPAR     ;SEE IF WE ARE DONE
1796      002014' 001000 000176      26700      MOV       A,M      ;GET THE WORD TO TRANSFER
1797      002015' 001000 000002      26720      STAX      B      ;TRANSFER IT
1798      002016' 001000 000310      26740      RZ
1799      002017' 001000 000613      26760      DCX       B
1800      002020' 001000 000053      26780      DCX       H      ;BACKUP FOR NEXT GUY
1801      002021' 001000 000303      26800      JMP       BLTLOP
1802      002022' 000000 002013'
1803      002023' 000000 002006'
1804
1805      26820      ;
1806      26840      ; THIS ROUTINE IS USED TO MAKE SURE A CERTAIN NUMBER
1807      26860      ; OF LOCATIONS REMAIN AVAILABLE FOR THE
1808      26880      ; STACK, THE CALL IS :
1809      26900      ; CALL GETSTK
1810      26920      ; NUMBER OF 2 BYTE ENTRIES NECESSARY
1811      26940      ;
1812      26960      ; THIS ROUTINE MUST BE CALLED BY ANY ROUTINE WHICH PUTS
1813      26980      ; AN ARBITRARY AMOUNT OF STUFF ON THE STACK
1814      27000      ; (I.E. ANY RECURSIVE ROUTINE LIKE PREVL)
1815      27020      ; IT IS ALSO CALLED BY ROUTINES SUCH AS GOSUB AND FOR
1816      27040      ; WHICH MAKE PERMANENT ENTRIES ON THE STACK
1817      27060      ;
1818      27100      ; ROUTINES WHICH MERELY USE AND FREE UP THE GUARANTEED
1819      27120      ; NUMLEV STACK LOCATIONS NEED NOT CALL THIS
1820      27140      ;
1821      002024' 001000 000343      27160      GETSTK:  XTHL
1822      002025' 001000 000116      27180      MOV       C,M      ;GET ARGUMENT INTO [C]
1823
1824      002026' 001000 000043      27220      INX       H
1825      002027' 001000 000343      27240      XTHL
1826      002030' 001000 000345      27260      PUSH      H      ;PUT BACK RETURN ADDRESS
1827      002031' 001000 000052      27280      PSHD      STREND   ;SAVE [H,L]
1828      002032' 000000 001025'
1829      002033' 000000 002022'
1830      002034' 001000 000006      27300      MVI       B,0
1831      002035' 000000 000000
1832      002036' 001000 000011      27320      DAD       B
1833      002037' 001000 000011      27340      DAD       B      ;SEE IF WE CAN HAVE THIS MANY
1834      002040' 001000 000315      27360      CALL      REASON
1835      002041' 000000 002045'

```

```

1836      002042' 000000 002032'
1837      002043' 001000 000341      27380      POP       H      ;RESTORE [H,L]
1838      002044' 001000 000311      27400      RET
1839
1840      27440      ;
1841      27460      ; [H,L]= SOME ADDRESS
1842      27480      ; [H,L] IS EXAMINED TO MAKE SURE AT LEAST NUMLEV
1843      27500      ; LOCATIONS REMAIN BETWEEN IT AND THE TOP OF THE STACK
1844      27520      ;
1845
1846      002045' 001000 000325      27560      REASON:  PUSH      D      ;SAVE [D,E]
1847      002046' 001000 000353      27580      XCHG
1848      002047' 001000 000041      27600      LXI       H,SCODE+65536-2+NUMLEV ;SETUP OFFSET OF GUARANTEED
1849      002050' 000000 177726'
1850      002051' 000000 002041'
1851
1852      002052' 001000 000071      27620      DAD       SP      ;LOCATIONS
1853      002053' 001000 000347      27640      COMPAR     ;[M,L]=STACK POINTER + OFFSET
1854      002054' 001000 000353      27660      XCHG
1855      002055' 001000 000321      27700      POP       D      ;SEE IF THIS IS <GT, ENTERING [H,L]
1856      002056' 001000 000320      27720      RNC
1857      002057' 001000 000056      27740      OMERR:  MVI       E,EHROM ;RESTORE [H,L] FROM [D,E]
1858      002060' 000000 000007      27760      IFN      LENGTH,< ;GET [D,E] BACK
1859
1860      27780      XWD      <01000,1> ;WAS OK?
1861      27800      IFN      LENGTH,< ;"OUT OF MEMORY"
1862      002061' 001000 000303      27820      JMP       ERROR>
1863      002062' 000000 002102'
1864      002063' 000000 002050'
1865
1866      27840      PAGE

```