```
Options in effect:
 NOADATA
 NOADV
   AFP(NOVOLATILE)
   APOST
   ARCH(11)
   ARITH(COMPAT)
   AWO
   BLOCK0
   BUFSIZE(16384)
 NOCICS
   CODEPAGE(1147)
 NOCOMPILE(S)
 NOCOPYLOC
 NOCOPYRIGHT
 NOCURRENCY
   DATA(31)
   DBCS
 NODECK
   DEFINE(ISBATCH=B'1')
   DIAGTRUNC
   DISPSIGN(SEP)
 NODLL
 NODUMP
   DYNAM
 NOEXIT
 NOEXPORTALL
   FASTSRT
   FLAG(I,I)
 NOFLAGSTD
   HGPR(PRESERVE)
   INITCHECK
 NOINITIAL
   INLINE
   INTDATE(ANSI)
   LANGUAGE(EN)
   LINECOUNT(60)
   LIST
   MAP(HEX)
   MAXPCF(100000)
 NOMDECK
   NAME(ALIAS)
   NSYMBOL(NATIONAL)
 NONUMBER
 NONUMCHECK
   NUMPROC(PFD)
   OBJECT
 NOOFFSET
   OPTIMIZE(1)
   OUTDD(SYSOUT)
 NOPARMCHECK
   PGMNAME(COMPAT)
   QUALIFY(COMPAT)
   RENT
   RMODE(AUTO)
   RULES(ENDPERIOD,NOEVENPACK,NOLAXPERF,OMITODOMIN,NOSLACKBYTES,UNREF)
 NOSERVICE
 NOSEQUENCE
```

```
  SOURCE
  SPACE(1)
NOSQL
NOSQLCCSID
NOSQLIMS
NOSSRANGE
  STGOPT
  SUPPRESS
NOTERM
NOTEST(NODWARF,NOSOURCE,NOSEPARATE)
NOTHREAD
  TRUNC(BIN)
NOVBREF
  VLR(STANDARD)
  VSAMOPENFS(SUCC)
NOWORD
  XMLPARSE(XMLSS)
  XREF(FULL)
  ZONEDATA(PFD)
  ZWB
```

```
  LineID  PL SL  ----+-*A-1-B--+----2----+----3----+----4----+----5----+----6----+----7-!--+----8 Map and Cross Reference

  000001               Identification division.
  000002               Program-id. TestLoop.
  000003               Environment division.
  000004               Data division.
  000005               Working-storage section.
  000006               01  STRING1-P pic s9(4) binary.                                          000000000 2C
  000007               01  STRING1-L pic s9(4) binary value 80.                                 000000000 2C
  000008               01  STRING2-P pic S9(4) binary.                                          000000000 2C
  000009               01  STRING2-L pic S9(4) binary value 80.                                 000000000 2C
  000010               01  STRING1.                                                             000000000 0CL80
  000011                   02  CAR1  pic X occurs 80.                                           000000000 1C
  000012               01  STRING2.                                                             000000000 0CL80
  000013                   02  CAR2  pic X occurs 80.                                           000000000 1C
  000014               01  ZONES.                                                               000000000 0CL400
  000015                   02  ZONE  pic X(80) occurs 5                                         000000000 80C
  000016                   indexed by XZONE.
  000017               Procedure division.
  000018               MAIN.
  000019              *
  000020                   accept STRING1                                                    10
  000021                   move 1 to STRING1-P                                               6
  000022                   perform varying XZONE from 1 by 1                                 16
  000023                   until XZONE > 5                                                   16
  000024     1               perform SPLIT thru SPLIT-END                                   31 55
  000025     1               move STRING2 to ZONE (XZONE)                                   12 15 16
  000026                   end-perform
  000027                   .
  000028               BYE.
  000029                   Goback
  000030                   .
  000031               SPLIT.
  000032                   move space to STRING2                                             IMP 12
  000033                   move 0 to STRING2-P                                               8
  000034                   .
  000035              * boucle do-while avec test multi-opérandes
  000036               LOOP.     if    STRING1-P <= STRING1-L                                6 7
  000037                         and   STRING2-P <= STRING2-L                                8 9
  000038                         and   CAR1 (STRING1-P) not = ';'                            11 6
  000039     1                       next sentence
  000040                         else
  000041     1                       go to LOOP-END.                                        48
  000042                   add  1 to STRING2-P                                               8
  000043                   move CAR1 (STRING1-P) to CAR2 (STRING2-P)                         11 6 13 8
  000044                   add  1 to STRING1-P                                               6
  000045                   .
  000046               LOOP-NEXT.
  000047                   go to LOOP.                                                       36
  000048               LOOP-END.
  000049                   exit.
  000050               SKIP-SEMI.
  000051                   add 1 to STRING1-P                                                6
  000052                   .
  000053               SKIP-SEMI-END.
  000054                   exit.
  000055               SPLIT-END.
  000056                   exit.
  000057               End program TestLoop.                                                 2
```

```
PP 5655-EC6 IBM Enterprise COBOL for z/OS  6.2.0 P190312       TESTLOOP  Date 04/19/2019  Time 17:05:36   Page      9

 000002:         Program-id. TestLoop.
…
    000180                    000020  USER-ENTRY: EQU     *
 000020:              accept STRING1
    000180  D20B D0E8 30F0      000020            MVC     232(12,R13),240(R3)   £                          +240
    000186  4120 9038           000020            LA      R2,56(,R9)            £  STRING1
    00018A  5020 D0EC           000020            ST      R2,236(,R13)          £
    00018E  4110 D0E8           000020            LA      R1,232(,R13)          £  _ArgumentList
    000192  58F0 3020           000020            L       R15,32(,R3)           £
    000196  58C0 D080           000020            L       R12,128(,R13)         £
    00019A  0DEF                000020            BASR    R14,R15               £  Call "IGZXACP"
 000021:              move 1 to STRING1-P
    00019C  E544 9018 0001      000021            MVHHI   24(,R9),X'0001'       £
 000022:              perform varying XZONE from 1 by 1
    0001A2  E54C 8000 0000      000022            MVHI    0(,R8),X'0000'        £
 000026:              end-perform
    0001A8  E55C 8000 0140      000026            CHSI    0(,R8),X'0140'        £
    0001AE  A724 0049           000026            JH      L0007
    0001B2                      000024  L0010:    EQU     *
    0001B2                      000031  SPLIT:    EQU     *
 000032:              move space to STRING2
    0001B2  9240 9088           000032            MVI     136(,R9),X'40'        £  STRING2
    0001B6  D24E 9089 9088      000032            MVC     137(79,R9),136(R9)    £  STRING2                 STRING2
 000033:              move 0 to STRING2-P
    0001BC  E544 9028 0000      000033            MVHHI   40(,R9),X'0000'       £
    0001C2                      000033  LOOP:     EQU     *
 000036:         LOOP.     if    STRING1-P <= STRING1-L
 000037                          and   STRING2-P <= STRING2-L
 000038                          and   CAR1 (STRING1-P) not = ';'
    0001C2  E554 9018 0050      000036            CHHSI   24(,R9),X'0050'       £     ← duplicate code for first test
    0001C8  A7C4 0009           000036            JNH     L0008
    0001CC  A7F4 0022           000033            J       LOOP-END
    0001D0                      000036  L0009:    EQU     *
    0001D0  E554 9018 0050      000036            CHHSI   24(,R9),X'0050'       £
    0001D6  A724 001D           000036            JH      LOOP-END
    0001DA                      000036  L0008:    EQU     *
    0001DA  4850 9028           000036            LH      R5,40(,R9)            £
    0001DE  EC52 0019 507E      000036            CIJ     R5,LOOP-END,80,(mask=0x2),
    0001E4  4820 9018           000036            LH      R2,24(,R9)            £
    0001E8  E342 9037 0094      000036            LLC     R4,55(R2,R9)          £  CAR1
    0001EE  EC48 0011 5E7F      000036            CLIJ    R4,LOOP-END,94,(mask=0x8),
 000042:              add  1 to STRING2-P
    0001F4  A75A 0001           000042            AHI     R5,0x1
    0001F8  4050 9028           000042            STH     R5,40(,R9)            £
    0001FC  B927 0055           000042            LHR     R5,R5
 000043:              move CAR1 (STRING1-P) to CAR2 (STRING2-P)
    000200  4245 9087           000043            STC     R4,135(R5,R9)         £  CAR2
 000044:              add  1 to STRING1-P
    000204  A72A 0001           000044            AHI     R2,0x1
    000208  4020 9018           000044            STH     R2,24(,R9)            £
000047:               go to LOOP.
    00020C  A7F4 FFE2           000047            J       L0009
    000210                      000048  LOOP-END: EQU     *
000051:               add 1 to STRING1-P
    000210  4820 9018           000051            LH      R2,24(,R9)            £
    000214  A72A 0001           000051            AHI     R2,0x1
    000218  4020 9018           000051            STH     R2,24(,R9)            £
    00021C                      000056  SPLIT-END: EQU     *
```

on iteration, don't go to
offset assigned to « IF » statement
→ not tracable under debuger
if compiled with « OFFSET »

```
000024:                perform SPLIT thru SPLIT-END
   00021C  D203 D0D8 D178     000024            MVC     216(4,R13),376(R13)   £
000025:                move STRING2 to ZONE (XZONE)
   000222  5820 8000          000025            L       R2,0(,R8)             £
   000226  4142 9000          000025            LA      R4,0(R2,R9)           £
   00022A  D24F 40D8 9088     000025            MVC     216(80,R4),136(R9)    £  ZONE                    STRING2
   000230  A72A 0050          000026            AHI     R2,0x50
   000234  5020 8000          000026            ST      R2,0(,R8)             £
   000238  A72E 0140          000026            CHI     R2,0x140
   00023C  A7C4 FFBB          000026            JNH     L0010
   000240                     000029  L0007:    EQU     *
000029:              Goback
   000240  9181 D11C          000029            TM      284(,R13),X'81'       £  DSAFlags
   000244  A784 0020          000029            JE      L0011
   000248  A718 0000          000029            LHI     R1,0x0
   00024C  58F0 3024          000029            L       R15,36(,R3)           £
   000250  58C0 D080          000029            L       R12,128(,R13)         £
   000254  0DEF               000029            BASR    R14,R15               £  Call "IGZXRTN"
   000256  9180 D11C          000029            TM      284(,R13),X'80'       £  DSAFlags
   00025A  A784 0015          000029            JE      L0011
   00025E  E54C D0E4 0000     000029            MVHI    228(,R13),X'0000'     £
   000264  E54C D0E0 0000     000029            MVHI    224(,R13),X'0000'     £
   00026A  4120 D0E4          000029            LA      R2,228(,R13)          £  TS2=10
   00026E  5020 D0E8          000029            ST      R2,232(,R13)          £
   000272  4120 D0E0          000029            LA      R2,224(,R13)          £  TS2=11
   000276  5020 D0EC          000029            ST      R2,236(,R13)          £
   00027A  4110 D0E8          000029            LA      R1,232(,R13)          £  _ArgumentList
   00027E  58F0 3028          000029            L       R15,40(,R3)           £
   000282  0DEF               000029            BASR    R14,R15               £  Call "IGZXTREN"
   000284                     000056  L0011:    EQU     *
000056:              exit.
   000284  EBFF 8054 006A     000056            ASI     84(,R8),X'FF'         £  IPCB_NumEnts
   00028A  A7F8 0000          000056            LHI     R15,0x0
   00028E  EB1F D08C 0096     000056            LMH     R1,R15,140(,R13)      £
   000294  58D0 D004          000056            L       R13,4(,R13)           £  Caller's DSA
   000298  58E0 D00C          000056            L       R14,12(,R13)          £  Return Address
   00029C  980C D014          000056            LM      R0,R12,20(,R13)       £  Restore Caller's registers
   0002A0  07FE               000056            BR      R14                   £  Return


PROCESS(CBL) statements:
 CBL OPTIMIZE(1)
 CBL OFFSET

PP 5655-EC6 IBM Enterprise COBOL for z/OS  6.2.0 P190312       TESTLOOP  Date 04/19/2019  Time 17:21:56   Page     9
LINEID  HEXLOC  VERBCODE                  LINEID  HEXLOC  VERBCODE                  LINEID  HEXLOC  VERBCODE
000020  000180  ACCEPT                    000021  00019C  MOVE                      000022  0001A2  PERFORM
000026  0001A8  END-PERFORM               000032  0001B2  MOVE                      000033  0001BC  MOVE
000036  0001C2  IF                        000041  0001F4  GO                        000042  0001F4  ADD
000043  000200  MOVE                      000044  000204  ADD                       000047  00020C  GO
000049  000210  EXIT                      000051  000210  ADD                       000024  00021C  PERFORM
000025  000222  MOVE                      000029  000240  GOBACK                    000054  000284  EXIT
000056  000284  EXIT
```