



.LSTON  
.Page  
.FIN

Init\_SprTbl:

```
Ld    !r2,*.HIBYTE. SpareArray
Ld    !r3,*.LOWBYTE. SpareArray
Call  ZeroBlock

Ld    !rE,*.HIBYTE. SparePw1
Ld    !rF,*.LOWBYTE. SparePw1
Call  Load_Password ;load SparePw1

Ld    !r2,*.HIBYTE. FmtOffset
Ld    !r3,*.LOWBYTE. FmtOffset

Lde   @!!r2,!r4 ;store offset value
Incw  !!r2
Lde   @!!r2,!r5 ;store interleave value
Incw  !!r2
```

I\_S\_Tbl\_Lp2:

```
Ld    !r0,*Nil
Ld    !r1,*( SprCount - SegPtrArray )
Lde   @!!r2,!r0
Incw  !!r2
Djnz  !r1,I_S_Tbl_Lp2
```

I\_S\_Tbl\_Lp4:

```
Ld    !r0,$00
Ld    !r4,*.HIBYTE. ( SpareCheck - SprCount )
Ld    !r5,*.LOWBYTE. ( SpareCheck - SprCount )
Lde   @!!r2,!r0
Incw  !!r2
Decw  !!r4
Jr    Nz,I_S_Tbl_Lp4
```

Create\_Tbl:

```
Ld    Data_Type,*SprTbl_Type
Ld    !r5,#2 ;create two tables
Clr   !r8
Call  Spr
Ld    !r2,*.HIBYTE. GetNewSpare
Ld    !r3,*.LOWBYTE. GetNewSpare
Call  Bank_Call
Push  !r8 ;save counter
Ld    !rF,!r0
Ld    !r8,*SprTbl_Type
Or    !r8,*Spare
Ld    !r2,*.HIBYTE. AddSpare
Ld    !r3,*.LOWBYTE. AddSpare
Call  Bank_Call
Pop   !r8
Inc   !r8 ;go to next table
Djnz  !r5,Create_Tbl
```

I\_Map\_Lp:

```
Ld    !r2,*.HIBYTE. Map_Table
Ld    !r3,*.LOWBYTE. Map_Table
Ld    !r1,*NbrSctrs
Ld    !r0,*0
Lde   @!!r2,!r0
Incw  !!r2
Add   !r0,*Map_DfIt
Cp    !r0,*NbrSctrs
Jr    Lt,I_Map_Lp2
Sub   !r0,*NbrSctrs
```





```

L_SprTbl_More:  Inc    Sector
                Djnz   !r6,L_Rd_Lp
                Djnz   !r5,L_SprTbl_Lp

                Or     !r4,!r4 ;check if any spare table found
                Jr     Nz,L_Spr_End

                Call   Abort

Chk_SprTbl:    Ld     ScrReg2,#.HIBYTE. ( RBuffer1+BlockID )
                Ld     ScrReg3,#.LOWBYTE. ( RBuffer1+BlockID )
                Call   Chk_PassWord
                Jr     Z,L_SprTbl_More

                Ld     ScrReg2,#.HIBYTE. RBuffer1
                Ld     ScrReg3,#.LOWBYTE. RBuffer1
                Call   Chk_PassWord
                Jr     Z,L_SprTbl_More

                Ld     !r2,#.HIBYTE. ( RBuffer1+SpareCheck-SpareArray )
                Ld     !r3,#.LOWBYTE. ( RBuffer1+SpareCheck-SpareArray )
                Lde    !r0,@!!r2 ;check possible check byte
                Incw   !!r2
                Lde    !r1,@!!r2

                Srp    #Wrk_Scr
                Ld     !rC,#.HIBYTE. RBuffer1
                Ld     !rD,#.LOWBYTE. RBuffer1
                Call   SprChk2

                Srp    #Wrk_Sys
                Call   Chk_Spr2
                Jr     Z,L_SprTbl_More

                Or     !r4,!r4 ;check for a SpareTable already found
                Jr     Z,L_Spr_Move

                Srp    #Wrk_Scr
                Ld     !r6,#.HIBYTE. SpareTmStmp
                Ld     !r7,#.LOWBYTE. SpareTmStmp
                Ld     !r4,#ScrReg0
                Call   Ld_TmStmp
                Ld     !r6,#.HIBYTE. ( RBuffer1+SpareTmStmp-SpareArray )
                Ld     !r7,#.LOWBYTE. ( RBuffer1+SpareTmStmp-SpareArray )
                Ld     !r4,#ScrRegC
                Call   Ld_TmStmp
                Sub    !r3,!rF
                Sbc   !r2,!rE
                Sbc   !r1,!rD
                Sbc   !r0,!rC
                Srp    #Wrk_Sys
                Jr     Ge,L_Spr_Inc
                Dec    !r4 ;account for old, bogus spare table

L_Spr_Move:    Ld     !r2,#.HIBYTE. RBuf_To_Spr
                Ld     !r3,#.LOWBYTE. RBuf_To_Spr
                Call   Bank_Call

L_Spr_Inc:     Inc     !r4 ;note the arrival of a SpareTable
                Cp     !r4,#2 ;see if that's all there is
                Jr     Nz,L_SprTbl_More

L_Spr_End:     Call   UpDate_SprTbl

```

















```

Ld    !r1,!r0 ;get command
And   !r1,#$FC ;check for illegal command
Jr    Z,SprCnt_1

Ld    !r9,!r0
Call  Abort

SprCnt_1:
Ld    !r2,!.HIBYTE. SprCount
Ld    !r3,!.LOWBYTE. SprCount
Lde   !r1,@!!r2 ;assume spare count increment
Inc   !r1!!r2 ;get bad block count

Cp    !r0,#01 ;check for Inc_BadBlock
Jr    Z,S_C_BadInc

S_C_BadInc:
Inc   !r1
S_C_BadEnd:
Lde   @!!r2,!r1 ;store new count
Jr    SprCnt_End

S_C_Spare:
!r1 ;store new count
SprCnt_End:
Call  Chk_SprCnt
SprCnt_Ret:
Jp    Bank_Ret

Chk_SprCnt:
Ld    !r2,!.HIBYTE. SprCount
Ld    !r3,!.LOWBYTE. SprCount
Lde   !r0,@!!r2 ;get spare count
Incw  !!r2
Lde   !r1,@!!r2 ;get bad co    Lt,Chk_Spr_Ret

Or    Excpt_Stat,#SprTbl_Warn
Call  SS_SprWarn

Chk_Spr_Ret:
Jp

```