

1

TITLE 'GAW8051 (DE-)COMPRESSIE VAN RECORDS'  
\*\*\*\*\*  
\* GAW8051 (DE-)COMPRESS A GIVEN STRING, COMPRESSING STRINGS \*  
\* OF THREE OR MORE CONSECUTIVE EQUAL CHARACTERS. \*  
\*  
\* CALL FROM PL/1: CALL GAW8051 (OPER, RC, STRING) \*  
\* WHERE: OPER : CHAR (4) \*  
\* RC : CHAR (2) WHERE 00 = OK \*  
\* \* 02 = NOT OK \*  
\* \* 99 = OPER NOT OK \*  
\* STRING: FIXED BIN (15) CONTAINS LENGTH \*  
\* CHAR (8192) \*  
\*  
\* PGMR: GERARD A. WASSINK RI, DATA PROCESS INFORMATICA \*  
\*  
\*\*\*\*\*  
\* \*\*\*\*\* SYMBOLIC NAMES FOR USED REGISTERS \*\*\*\*\*  
\*  
R1 EQU 1 PARAMETER LIST ADDRESS ON ENTRY.  
R2 EQU 2 )  
R3 EQU 3 )  
R4 EQU 4 )  
R5 EQU 5 ) FREE FOR ROUTINE USAGE  
R6 EQU 6 )  
R7 EQU 7 )  
R8 EQU 8 )  
R9 EQU 9 BASE-ADDR OPERATIE / BUFFER.  
R10 EQU 10 BASE-ADDR RET-COD.  
R11 EQU 11 BASE-ADDR RECORD.  
R12 EQU 12 BASE-REGISTER FOR THIS PROGRAM.  
R13 EQU 13 SAVE-AREA ADDRESS.  
R14 EQU 14 RETURN ADDRESS.  
R15 EQU 15 INITIAL VALUE FOR BASE REGISTER.

35

2

```

*                                         MAIN LINE CODING
*-----+
GAW8051 CSECT
        STM    R14,R12,12(R13)
*
        LR     R12,R15          ESTABLISH
        USING GAW8051,R12      ADDRESSABILITY.

*
        LA     R2,SAVEAREA      LINK
        ST     R13,SAVEAREA+4   SAVEAREA
        ST     R2,8(R13)        TO PREVIOUS.
        LA     R13,SAVEAREA     ADDRESS OUR OWN.

*
        L      R9,0(R1)         R9: BASE-ADDR OPERATION.
        USING PRMOPER,R9
        L      R10,4(R1)        R10: BASE-ADDR RET-COD.
        USING PRMRCOD,R10
        L      R11,8(R1)        R11: BASE-ADDR RECORD.
        USING PRMRECD,R11

*
        MVC   RETCODE,=C'00'    INITIALIZE
        B     SELECT            RETURN-CODE.

*
BACK    DS    OH
        L     R13,4(R13)       RESTORE R13.
        LM   R14,R12,12(R13)   RESTORE CALLERS REGISTERS.
        OI   15(R13),X'01'
        LA   R15,0(0,0)        RETURN CODE ZERO.
        BR   R14               GO BACK TO CALLER.

```

```

* SELECT OPERATION                               SELECT
*
SELECT    DS      OH

*
* --- CHECK THE SPECIFIED LENGTH ---
*
          XR      R7,R7           ZERO OUT R7
          LH      R7,RECLEN        LOAD RECORD LENGTH.
          C       R7,BUFMAX        GREATER THAN MAXIMUM ?
          BNH    SEL005          NO, CONTINUE.

*
          MVC    RETCODE,=C'02'     NOT_OK
          B      BACK             AND GO BACK.

*
SEL005    DS      OH

*
* --- SAVE THE CONTENTS OF THE SPECIFIED RECORD IN THE BUFFER ---
*
          STM    R6,R9,SELSAV      STORE REGISTERS.
          LA     R6,BUFFER         PREPARE
*
          LA     R8,2(R11)         R7 CONTAINS LENGTH
          LR     R9,R7             AND
          MVCL   R6,R8             EXECUTE
          LM     R6,R9,SELSAV      MOVE CHARACTERS LONG
                                RESTORE REGISTERS.

*
* --- SELECT THE OPERATION TO PERFORM ---
*
          CLC    OPERAT,=C'COMP'    COMPRESS REQUESTED ?
          BE     COMPRESS
          CLC    OPERAT,=C'DECO'    DE-COMPRESS REQUESTED ?
          BE     DECOMP

*
          DROP   R9                NO USE FOR THIS BASE NOMORE.

*
          MVC    RETCODE,=C'99'    NEITHER, RETURN ERROR.
          B      BACK

*
SELSAV    DS      18F            LOCAL SAVE AREA.

```

\* -----  
\* COMPRESSION ROUTINE  
\* -----  
\* THIS ROUTINE WILL PERFORM COMPRESSION OF THREE TYPES OF FIELDS:  
\*  
\* 1) SPACE SUPPRESSION  
\* 2) REPLICATING NON-SPACE & NON-DIGIT CHARACTERS  
\* 3) CONVERTING ZONED DATA TO BCD  
\*  
\* TO BE ABLE TO PROCESS ANY RECORD, EVERY FIELD IN THE COMPRESSED  
\* RECORD IS PRECEDED BY A CONTROL CODE. THIS CONTROL CODE IS ONE BYTE  
\* LONG AND IS USED AS FOLLOWS:  
\*  
\* BIT: 76543210  
\* CE.....  
\*  
\* BIT 7 INDICATES COMPRESSION '1': NEXT AREA CONTAINS COMPRESSED DATA  
\* '0': NEXT AREA CONTAINS UNCOMPRESSED DATA  
\*  
\* BIT 6, WHEN '1', INDICATES AN EXTRA LENGTH BYTE IS PROVIDED.  
\*  
\* FURTHERMORE, WHEN BIT 7 IS '1', THE LAYOUT OF THE REST OF THE  
\* CONTROL CODE IS AS FOLLOWS:  
\*  
\* BIT 5 & 4 INDICATE THE TYPE OF COMPRESSION,  
\* '10' - REPLICATE CHARACTER COMPRESSION  
\* '01' - SPACE SUPPRESSION  
\* '00' - ZONED TO BCD COMPRESSION  
\*  
\* BIT 3 TO 0 SPECIFY THE LENGTH OF THE COMPRESSED AREA  
\*  
\* WHEN BIT 7 IS '0', THE LAYOUT OF THE REST OF THE CONTROL CODE IS  
\* AS FOLLOWS:  
\*  
\* BIT 5 TO 0 SPECIFY THE LENGTH OF THE UN-COMPRESSED AREA  
\*  
\* WHEN BIT 6 WAS '1', THE NEXT BYTE ALSO CONTAINS CONTROL INFORMATION,  
\* IT CONTAINS AN EXTRA 8 BITS OF LENGTH INFORMATION. THE TOTAL LENGTH  
\* OF THE FIELD IS CALCULATED BY MULTIPLYING THE LENGTH FROM THE FIRST  
\* CONTROL BYTE BY 256 AND ADDING THE CONTENTS OF THE SECOND BYTE.  
\*  
\*  
\* IN CASE OF COMPRESSION, WHEN THE COMPRESSION TYPE IS '10',  
\* REPLICATING CHARACTERS, THE NEXT (SECOND OR THIRD) BYTE CONTAINS  
\* THE REPLICATING CHARACTER.  
\*  
\* -----

```

* -----
* COMPRESSION PROCESS:
*
* 1) INITIALIZE POINTERS
*
* 2) FIND POINTERS TO FIRST FIELD OF EACH COMPRESSION TYPE
*
* 3) DETERMINE WHICH TYPE-POINTER IS THE SMALLEST
*
* 4) WHEN APPROPRIATE, TAKE OVER THE AREA PRECEDING THIS SMALLEST
*    POINTER, INDICATING IT'S BEING UN-COMPRESSED
*
* 5) PERFORM THE COMPRESSION OF THE SPECIFIED TYPE
*
* 6) FIND THE NEXT AREA OF THE SAME TYPE
*
* 7) IF NOT END OF RECORD, LOOP AROUND TO 3)
*
* 8) GIVE CONTROL BACK TO CALLER
*
* -----

```

COMPRESS DS OH

```

* --- INITIALIZE VARIABLES & REGISTERS ---
*
```

LA	R9,BUFFER	BUFFER BASE ADDRESS.
LR	R7,R9	BUFFER START
LH	R6,RECLEN	PLUS RECORD LENGTH
AR	R7,R6	GIVES BUFFER-END + 1.
MVI	0(R7),C' '	MOVE IN ONE SPACE PAST END
BCTR	R7,0	MINUS ONE EQUALS BUFFER END.
LA	R3,BUFFER	POINTER TO BUFFER.
LR	R8,R3	POINTER IN BUFFER.
LA	R5,RECORD	POINTER TO RECORD.
*		
ST	R3,T1	STORE T1
ST	R8,T2	STORE T2
ST	R5,T3	STORE T3
ST	R7,EOR	STORE EOR

```

* --- FIND THE FIRST OCCURRENCE OF EACH POSSIBLE COMPRESSION TYPE ---
*
```

BAL	R14,NEXTZON	FIND NEXT ZONED AREA.
BAL	R14,NXTSPA	FIND NEXT SPACE AREA.
BAL	R14,NEXTREP	FIND NEXT REPLICATE AREA.

CMP010	DS OH	
	CLC NXTSPA,NXTZON	SPACES > ZONED ?
	BH CMP012	YES
	CLC NXTSPA,NEXTREP	SPACES > REPL ?
	BH CMP012	YES
	B CMP100	DO SPACES.

CMP012	DS OH	
	CLC NXTREP,NXTZON	REPL <= ZONED ?
	BNH CMP200	YES, DO REPL'S
	B CMP300	NO, DO ZONED.

6

---

\* -----  
 \* SPACE COMPRESSION SPACES  
 \* -----

CMP100 DS OH  
 L R8,NXTSPA ADDRESS OF NEXT SPACE AREA  
 BAL R14,TAKEOVER TAKE OVER TO HERE UNCOMPRESSED.

\*  
 L R3,T1 LOAD CHANGED T1  
 L R5,T3 AND T3

\*  
 C R8,EOR PAST EOR ?  
 BH CMP900 YEP, END.

\*  
 \* --- DETERMINE LENGTH OF SPACE AREA WITH T2 ---

\*  
 CMP102 DS OH  
 CLI O(R8),C' ' SPACE ?  
 BNE CMP105 NO, END OF SPACE AREA.  
 LA R8,1(R8) INCREMENT T2.  
 C R8,EOR PAST EOR ?  
 BNH CMP102 NO, NEXT.

\*  
 \* --- R8 POINTS TO EOR+1 OR TO FIRST NON-SPACE CHARACTER ---

\*  
 CMP105 DS OH  
 MVI CONTROL,X'00' ZERO OUT CONTROL CODE.  
 LR R4,R8 LENGTH EQUALS  
 SR R4,R3 AREA END -/- AREA START.  
 STH R4,ARLEN REMEMBER AREA LENGTH.  
 C R4,MAXL4 EXTRA LENGTH BYTE NEEDED ?  
 BNH CMP107 NO, FITS IN 4 BITS.  
 STC R4,CONTROL2 STORE LENGTH IN 2ND CTRL BYTE.  
 SRL R4,8(0) SHIFT LENGTH TO RIGHT  
 STC R4,CONTROL STORE LENGTH IN 1ST CTRL BYTE.  
 NI CONTROL,X'OF' ZERO OUT HIGH HALF  
 OI CONTROL,PACKED INDICATE COMPRESSION ACTIVE  
 OI CONTROL,EXTLEN INDICATE EXTRA LENGTH BYTE  
 OI CONTROL,SPACES INDICATE SPACE COMPRESSION  
 MVC 0(1,R5),CONTROL STORE CTRL IN RECORD  
 MVC 1(1,R5),CONTROL2 STORE CTRL2 IN RECORD  
 LA R5,2(,R5) BUMP UP R5.  
 B CMP190 END OF COMPRESSION.

\*  
 CMP107 DS OH  
 STC R4,CONTROL STORE LENGTH IN 1ST CTRL BYTE.  
 NI CONTROL,X'OF' ZERO OUT HIGH HALF  
 OI CONTROL,PACKED INDICATE COMPRESSION ACTIVE  
 OI CONTROL,SPACES INDICATE SPACE COMPRESSION  
 MVC 0(1,R5),CONTROL STORE CTRL IN RECORD  
 LA R5,1(,R5) BUMP UP R5.

\*  
 CMP190 DS OH  
 LR R3,R8 T1 = T2.

\*  
 ST R3,T1 STORE T1  
 ST R8,T2 STORE T2  
 ST R5,T3 STORE T3

\*  
 BAL R14,NEXTSPA FIND NEXT SPACE AREA.

\*  
 B CMP900 BACK TO CALLER.