

J E S 2 J O B L O G

15.44.17 JOB 1462 \$HASP373 MSZR01UA STARTED - INIT 1 - CLASS A - SYS TK4-
15.44.17 JOB 1462 IEF403I MSZR01UA - STARTED - TIME=15.44.17
15.44.19 JOB 1462 IEFACRTT - Stepname Procstep Program Retcode
15.44.19 JOB 1462 MSZR01UA ASMF IFOX00 RC= 0000
15.44.20 JOB 1462 IEC130I SYSLIN DD STATEMENT MISSING
15.44.20 JOB 1462 IEC141I 013-14,IGG0191B,MSZR01UA,LKED1B,SYSLMOD,290,PUB003,
15.44.20 JOB 1462 IEC141I MSZR01U.SPICE.LIB
15.44.20 JOB 1462 IEF450I MSZR01UA LKED1B - ABEND S013 U0000 - TIME=15.44.20
15.44.20 JOB 1462 MSZR01UA LKED1B IEWL AB S013
15.44.20 JOB 1462 IEF404I MSZR01UA - ENDED - TIME=15.44.20
15.44.20 JOB 1462 \$HASP395 MSZR01UA ENDED

----- JES2 JOB STATISTICS -----

02 JUN 20 JOB EXECUTION DATE

708 CARDS READ

1,162 SYSOUT PRINT RECORDS

0 SYSOUT PUNCH RECORDS

0.05 MINUTES EXECUTION TIME

```
1 //MSZR01UA JOB (AS),'SPICE2 ASM', JOB 1462
// CLASS=A,MSGCLASS=A,RESTART=ASMF,
// USER=MSZR01U,PASSWORD= GENERATED BY GDL
***-----
2 //ASMF EXEC PGM=IFOX00,REGION=2048K
3 //SYSLIB DD DSN=SYS1.MACLIB,DISP=SHR,DCB=BLKSIZE=32720
4 // DD DSN=SYS1.AMODGEN,DISP=SHR
5 // DD DSN=SYS2.MACLIB,DISP=SHR
6 // DD DSN=SYS2.PSU.ASSIST,DISP=SHR
7 //SYSUT1 DD DISP=(NEW,DELETE),SPACE=(1700,(900,100)),UNIT=SYSDA
8 //SYSUT2 DD DISP=(NEW,DELETE),SPACE=(1700,(600,100)),UNIT=SYSDA
9 //SYSUT3 DD DISP=(NEW,DELETE),SPACE=(1700,(600,100)),UNIT=SYSDA
10 //SYSPRINT DD SYSOUT=*
11 //SYSPUNCH DD DSN=MSZR01U.SPICE.OBJ,
// UNIT=SYSDA,SPACE=(CYL,1),DISP=(NEW,CATLG),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=400)
***SYSLMOD DD DSN=MSZR01U.SPICE.LIB(SPICEA),
*** UNIT=SYSDA,SPACE=(CYL,1),DISP=(NEW,CATLG),
*** DCB=(RECFM=FB,LRECL=80,BLKSIZE=19069)
*** AMODE(24)
12 //SYSIN DD *
13 //LKED1B EXEC PGM=IEWL,
// PARM='XREF,LIST,LET,TEST,AC=0',
// REGION=1024K
14 //SYSLMOD DD DSN=MSZR01U.SPICE.LIB,
// UNIT=SYSDA,SPACE=(CYL,1),DISP=(NEW,CATLG),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200)
15 //SYSUT1 DD UNIT=WORK,SPACE=(CYL,(8,1))
16 //SYSPRINT DD SYSOUT=*
***SYSLIN DD DSN=&&OBJLIB,DISP=(OLD,PASS,DELETE)
***SYSLIN DD *
*** INCLUDE SYSLMOD(SPICEA)
*** ENTRY ADDGEN
*** NAME SPICEA(R)
***
```

```

IEF236I ALLOC. FOR MSZR01UA ASMF
IEF237I 148 ALLOCATED TO SYSLIB
IEF237I 248 ALLOCATED TO
IEF237I 148 ALLOCATED TO
IEF237I 191 ALLOCATED TO
IEF237I 242 ALLOCATED TO SYSUT1
IEF237I 140 ALLOCATED TO SYSUT2
IEF237I 180 ALLOCATED TO SYSUT3
IEF237I JES2 ALLOCATED TO SYSPRINT
IEF237I 243 ALLOCATED TO SYSPUNCH
IEF237I 240 ALLOCATED TO SYS00068
IEF237I JES2 ALLOCATED TO SYSIN
IEF142I MSZR01UA ASMF - STEP WAS EXECUTED - COND CODE 0000
IEF285I   SYS1.MACLIB                      KEPT          *-----15
IEF285I   VOL SER NOS= MVSRES.
IEF285I   SYS1.AMODGEN                      KEPT          *-----1
IEF285I   VOL SER NOS= MVSDLB.
IEF285I   SYS2.MACLIB                      KEPT          *-----0
IEF285I   VOL SER NOS= MVSRES.
IEF285I   SYS2.PSU.ASSIST                  KEPT          *-----0
IEF285I   VOL SER NOS= MVSCAT.
IEF285I   SYS20154.T154417.RA000.MSZR01UA.R0000001 DELETED        *-----51
IEF285I   VOL SER NOS= MSZ01 .
IEF285I   SYS20154.T154417.RA000.MSZR01UA.R0000002 DELETED        *-----11
IEF285I   VOL SER NOS= WORK00.
IEF285I   SYS20154.T154417.RA000.MSZR01UA.R0000003 DELETED        *-----12
IEF285I   VOL SER NOS= WORK02.
IEF285I   JES2.JOB01462.SO0102             SYSOUT
IEF285I   MSZR01U.SPICE.OBJ                CATALOGED      *-----12
IEF285I   VOL SER NOS= PUB014.
IEF285I   SYS1.UCAT.TSO                    KEPT          *-----0
IEF285I   VOL SER NOS= PUB000.
IEF285I   JES2.JOB01462.SI0101             SYSIN
IEF373I STEP /ASMF      / START 20154.1544
IEF374I STEP /ASMF      / STOP 20154.1544 CPU      0MIN 01.75SEC SRB      0MIN 00.09SEC VIRT 2048K SYS 200K
*****
*      1. Jobstep of job: MSZR01UA      Stepname: ASMF      Program name: IFOX00      Executed on 02.06.20 from 15.44.17 to 15.44.19 *
*      elapsed time 00:00:02,58      CPU-Identifier: TK4-      Page-in: 0 *
*      CPU time 00:00:01,84      Virtual Storage used: 2048K      Page-out: 0 *
*      corr. CPU: 00:00:01,84      CPU time has been corrected by 1 / 1,0 multiplier *
* * * * *
*      I/O Operation *
*      Number of records read via DD * or DD DATA: 672 *
*      148.....15 248.....1 148.....0 191.....0 242.....51 140.....11 180.....12 DMY.....0 243.....12 240.....0 *
*      DMY.....0 *
* * * * *
*      Charge for step (w/o SYSOUT): 3,06 *
*****
IEF236I ALLOC. FOR MSZR01UA LKED1B
IEF237I 290 ALLOCATED TO SYSLMOD
IEF237I 240 ALLOCATED TO SYS00069
IEF237I 242 ALLOCATED TO SYSUT1
IEF237I JES2 ALLOCATED TO SYSPRINT
IEC130I SYSLIN DD STATEMENT MISSING
IEC141I 013-14,IGG0191B,MSZR01UA,LKED1B,SYSLMOD,290,PUB003,
IEC141I MSZR01U.SPICE.LIB
IEF472I MSZR01UA LKED1B - COMPLETION CODE - SYSTEM=013 USER=0000
IEF285I   MSZR01U.SPICE.LIB                CATALOGED      *-----0
IEF285I   VOL SER NOS= PUB003.
IEF285I   SYS1.UCAT.TSO                    KEPT          *-----0
IEF285I   VOL SER NOS= PUB000.
IEF285I   SYS20154.T154417.RA000.MSZR01UA.R0000004 DELETED        *-----0
IEF285I   VOL SER NOS= MSZ01 .

```

IEF285I JES2.JOB01462.S00103
IEF373I STEP /LKED1B / START 20154.1544
IEF374I STEP /LKED1B / STOP 20154.1544 CPU OMIN 00.14SEC SRB OMIN 00.01SEC VIRT 36K SYS 160K

* 2. Jobstep of job: MSZR01UA Stepname: LKED1B Program name: IEWL Executed on 02.06.20 from 15.44.19 to 15.44.20 *
* elapsed time 00:00:00,42 CPU-Identifier: TK4- Page-in: 0 *
* CPU time 00:00:00,15 Virtual Storage used: 36K Page-out: 0 *
* corr. CPU: 00:00:00,15 CPU time has been corrected by 1 / 1,0 multiplier *
* *
* I/O Operation *
* Number of records read via DD * or DD DATA: 0 *
* 290.....0 240.....0 242.....0 DMY.....0 *
* *
* Charge for step (w/o SYSOUT): 0,25 *

IEF375I JOB /MSZR01UA/ START 20154.1544
IEF376I JOB /MSZR01UA/ STOP 20154.1544 CPU OMIN 01.89SEC SRB OMIN 00.10SEC

SYMBOL	TYPE	ID	ADDR	LENGTH	LDID	ASM 0201 15.44 06/02/20
ADDGEN	SD	0001	000000	000138		
TRPNUM	LD		000110		0001	
REGVAL	LD		0000C0		0001	
NXTCOD	ER	0002				
SECOND	SD	0003	000138	000078		
CODEXC	SD	0004	0001B0	0000AC		
SCRTCH	LD		000240		0004	
NXTCOD	LD		000238		0004	
DATIME	SD	0005	000260	000058		
MINS3A	SD	0006	0002B8	000108		
ADDGEN	ER	0007				
LOC	SD	0008	0003C0	000018		
MINS0	SD	0009	0003D8	0000CC		
SCRTCH	ER	000A				
TRPNUM	ER	000B				
REGVAL	ER	000C				
MINS1	SD	000D	0004A8	0000F8		
MINS2	SD	000E	0005A0	0000F0		
MINS4	SD	000F	000690	0000B4		
MINS3	SD	0010	000748	000120		
GTMAIN	SD	0011	000868	000114		

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 15.44 06/02/20
000000				1	ADDGEN CSECT	
				2 *	CREATE THE SECOND HALFWORD OF AN RX TYPE INSTRUCTION.	
				3 *	NO ARGUMENTS - THE VALUE ARRAY SUBSCRIPT MUST BE IN R1 ON	
				4 *	CALL. LEAVES RESULT IN LOWER HALF OF REGISTER 1, THE NUMBER O	
				5 *	HALFWORDS OF CODE PRODUCED IS ALSO PLACED IN REGISTER 0.	
				6 *	PROGRAMMED BY T. QUARLES 4/2/81 AS PART OF THE SPICE 2G.2 CODE	
				7 *	GENERATION PACKAGE FOR IBM SYSTEM/360, /370 ARCHITECTURE MACHI	
				8 *	REGEQU	
				9	\$REGS	
		00000		11+R0	EQU 0	00000070
		00001		12+R1	EQU 1	00000080
		00002		13+R2	EQU 2	00000090
		00003		14+R3	EQU 3	00000100
		00004		15+R4	EQU 4	00000110
		00005		16+R5	EQU 5	00000120
		00006		17+R6	EQU 6	00000130
		00007		18+R7	EQU 7	00000140
		00008		19+R8	EQU 8	00000150
		00009		20+R9	EQU 9	00000160
		0000A		21+R10	EQU 10	00000170
		0000B		22+R11	EQU 11	00000180
		0000C		23+R12	EQU 12	00000190
		0000D		24+R13	EQU 13	00000200
		0000E		25+R14	EQU 14	00000210
		0000F		26+R15	EQU 15	00000220
		00000		28+REG0	EQU 0	00000240
		00001		29+REG1	EQU 1	00000250
		00002		30+REG2	EQU 2	00000260
		00003		31+REG3	EQU 3	00000270
		00004		32+REG4	EQU 4	00000280
		00005		33+REG5	EQU 5	00000290
		00006		34+REG6	EQU 6	00000300
		00007		35+REG7	EQU 7	00000310
		00008		36+REG8	EQU 8	00000320
		00009		37+REG9	EQU 9	00000330
		0000A		38+REG10	EQU 10	00000340
		0000B		39+REG11	EQU 11	00000350
		0000C		40+REG12	EQU 12	00000360
		0000D		41+REG13	EQU 13	00000370
		0000E		42+REG14	EQU 14	00000380
		0000F		43+REG15	EQU 15	00000390
		00000		45	USING *,R15	
000000	47F0 F00C	0000C		46	B START	
000004	07			47	DC AL1(7)	
000005	C1C4C4C7C5D540			48	DC CL7'ADDGEN '	
				49	ENTRY TRPNUM	
				50	ENTRY REGVAL	
00000C	90EC D00C	0000C		51	START STM R14,R12,12(R13) COMPLETE THE STANDARD ENTRY SEQUENCE	
000010	8910 0003	00003		52	SLL R1,3 MULTIPLY SUBSCRIPT BY 8 TO GET BYETE NUMB	
000014	5410 F118	00118		53	N R1,=X'00FFFFFF' CLEAR HIGH ORDER GARBAGE	
000018	1B00			54	SR R0,R0 CLEAR THE CODE PRODUCED REGISTER	
00001A	1821			55	LR R2,R1 COPY ADDRESS	

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 15.44 06/02/20
00001C	5410 F11C	0011C		56	N	R1,=X'00000FFF'	GET DISPLACEMENT PART
000020	8A20 000C	0000C		57	SRA	R2,12	SHIFT PAGE NUMBER TO WHERE WE WANT IT
000024	4780 F0B2	000B2		58	BZ	DONE	PAGE ZERO, SO DON'T HAVE ANY WORK TO DO
000028	5860 F120	00120		59	L	R6,=X'7FFFFFFF'	GET AN IMPOSSIBLE TRIP NUMBER
00002C	5830 F124	00124		60	L	R3,=F'10'	GET THE NUMBER OF REGISTERS
000030	5840 F128	00128		61	L	R4,=A(REGVAL)	GET POINTER TO THE TABLE OF VALUES
000034	5924 0000	00000		62	LOOP	C R2,0(R4)	CHECK FOR A MATCH BETWEEN PAGE NUMBER & T
000038	4780 F09C	0009C		63	BE	FOUND	GOOD - TABLE ENTRY MATCHED, DON'T NEED TO
00003C	5964 0004	00004		64	C	R6,4(R4)	CHECK IF THIS IS LEAST RECENTLY USED ENTR
000040	47D0 F04C	0004C		65	BNH	NOSWP	NO, SO DON'T SWAP POINTERS
000044	5864 0004	00004		66	L	R6,4(R4)	GET THE TRIP NUMBER IN A CONVENIENT REG.
000048	1874			67	LR	R7,R4	COPY THE TABLE ENTRY POINTER
00004A	1883			68	LR	R8,R3	COPY THE REGISTER NUMBER
00004C	5A40 F12C	0012C		69	NOSWP	A R4,=F'8'	ADD TO POINTER
000050	4630 F034	00034		70	BCT	R3,LOOP	SUBTRACT FROM REG
				71	*		AND LOOP IF NOT 0.
				72	*		FELL OUT OF THE BOTTOM OF THE LOOP - DIDN'T FIND A VALUE, SO
				73	*		WE MUST PUT THE CORRECT VALUE IN THE DESIRED REGISTER (LRU).
				74	*		CODE PRODUCED TO LOAD THE REGISTER IS
				75	*	LA R?,PAGENUMBER	
				76	*	SLL R?,12	(MULTIPLY BY 4096 IN A HURRY)
				77	*		
000054	1838			78	LR	R3,R8	COPY TO THE DESIRED REGISTER SET
000056	1847			79	LR	R4,R7	
000058	5024 0000	00000		80	ST	R2,0(R4)	UPDATA TABLE\
00005C	5800 F130	00130		81	L	R0,=F'4'	SAY PRODUCED 4 HALFWORDS OF CODE
000060	5890 F134	00134		82	L	R9,=V(NXTCOD)	GET PLACE TO PUT CODE
000064	58A9 0000	00000		83	L	R10,0(R9)	AND THE ACTUAL CODE ADDRESS
000068	18BA			84	LR	R11,R10	COPY TO ANOTHER REGISTER ALSO
00006A	5AA0 F12C	0012C		85	A	R10,=F'8'	ADD LENGTH OF THE CODE TO IT
00006E	50A9 0000	00000		86	ST	R10,0(R9)	AND PUT BACK FOR THE REST OF THE WORLD
000072	58A0 F0B8	000B8		87	L	R10,LAINST	GET THE LOAD ADDRESS INSTRUCTION
000076	8980 0014	00014		88	SLL	R8,20	SHIFT THE REGISTER NUMBER TO THE DESIRED
00007A	16A8			89	OR	R10,R8	AND PUT THE REGISTER NUMBER IN LA INST
00007C	16A2			90	OR	R10,R2	AND NOW PUT IN THE PAGE NUMBER
00007E	40AB 0002	00002		91	STH	R10,2(R11)	AND PUT OUT AS CODE
000082	88A0 0010	00010		92	SRL	R10,16	
000086	40AB 0000	00000		93	STH	R10,0(R11)	
00008A	58A0 F0BC	000BC		94	L	R10,SLLINST	NOW GET THE SLL INSTRUCTION
00008E	16A8			95	OR	R10,R8	AND PUT THE REGISTER NUMBER IN IT TOO.
000090	40AB 0006	00006		96	STH	R10,6(R11)	AND PUT OUT AS CODE ALSO
000094	88A0 0010	00010		97	SRL	R10,16	
000098	40AB 0004	00004		98	STH	R10,4(R11)	
00009C	8930 000C	0000C		99	FOUND	SLL R3,12	SHIFT REGISTER TO DESIRED LOCATION
0000A0	1613			100	OR	R1,R3	PUT REGISTER TOGETHER WITH THE DISPLACEME
0000A2	5870 F110	00110		101	L	R7,TRPNUM	GET THE TRIP COUNT
0000A6	4177 0001	00001		102	LA	R7,1(R7)	INCREMENT IT
0000AA	5070 F110	00110		103	ST	R7,TRPNUM	SAVE
0000AE	5074 0004	00004		104	ST	R7,4(R4)	AND PUT IN TABLE FOR LRU ALG. TO FIND
0000B2	982C D01C	0001C		105	DONE	LM R2,R12,28(R13)	RESTORE ALL USED REGISTERS EXCEPT RETURNS
0000B6	07FE			106	BR	R14	AND RETURN TO CALLER
0000B8	4100 0000	00000		107	LAINST	LA R0,0(R0)	THE SAMPLE LOAD ADDRESS INSTRUCTIN
0000BC	8900 000C	0000C		108	SLLINST	SLL R0,12	THE SAMPLE SHIFT LEFT LOGICAL INSTRUCTION
0000C0	FFFFFFFFFFFFFFFF			109	REGVAL	DC 20F'-1'	THE INITIAL REGISTER USE TABLE
000110	00000000			110	TRPNUM	DC F'0'	THE COUNT OF NUMBER OF TIMES THIS ROUTINE

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM	0201	15.44	06/02/20
000118				111	LTORG				
000118	00FFFFFF			112	=X'00FFFFFF'				
00011C	00000FFF			113	=X'00000FFF'				
000120	7FFFFFFF			114	=X'7FFFFFFF'				
000124	0000000A			115	=F'10'				
000128	000000C0			116	=A(REGVAL)				
00012C	00000008			117	=F'8'				
000130	00000004			118	=F'4'				
000134	00000000			119	=V(NXTCOD)				
				120	DROP				

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 15.44 06/02/20
000138					122	SECOND	CSECT	
000138	47FF	000C	0000C		123		BC 15,12(15)	
00013C	07				124		DC X'7'	
00013D	E2C5C3D6D5C440				125		DC CL7'SECOND '	
000144	90EC	D00C	0000C		126		STM 14,12,12(13)	
000148	05C0				127		BALR 12,0	
				0014A	128		USING *,12	
00014A	5820	1000	00000		129		L 2,0(,1)	
00014E	4130	C036	00180		130		LA 3,AREA	
000152	8330000C				131		DC X'8330000C'	DIAGNOSE INSTRUCTION FOR TIME INFO
000156	5843	0010	00010		132		L 4,16(3)	
00015A	5440	C05E	001A8		133		N 4,=X'00FFFFFF'	
00015E	5640	C062	001AC		134		O 4,=X'4E000000'	
000162	5043	0010	00010		135		ST 4,16(3)	
000166	6803	0010	00010		136		LD 0,16(3)	
00016A	6D00	C056	001A0		137		DD 0,=D'1000000.0'	
00016E	6002	0000	00000		138		STD 0,0(2)	
000172	98EC	D00C	0000C		139		LM 14,12,12(13)	
000176	92FF	D00C	0000C		140		MVI 12(13),X'FF'	
00017A	07FE				141		BCR 15,14	
000180					142	AREA	DS 4D	
0001A0					143		LTORG	
0001A0	45F4240000000000				144		=D'1000000.0'	
0001A8	00FFFFFF				145		=X'00FFFFFF'	
0001AC	4E000000				146		=X'4E000000'	
					147		DROP	

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 15.44 06/02/20
0001B0				149	CODEXC CSECT	
				150	*****	
				151	*	
				152	*	
				153	* CODEXC(NODPLC(LV1),VALUE(LVN)):	
				154	* ASSEMBLER SUBROUTINE TO CAUSE EXECUTION OF THE CODE GENERATED	
				155	* BY CALLS TO THE MINS? ROUTINES. FIRST ARGUMENT IS A POINTER	
				156	* TO THE CODE TO EXECUTE, AND THE SECOND IS A POINTER TO THE	
				157	* FIRST ELEMENT OF THE VALUE SUBARRAY.	
				158	*	
				159	*	
				160	* WRITTEN BY TOM QUARLES 4/1/81 AS PART OF THE MACHINE CODE	
				161	* GENERATION PACKAGE FOR IBM S/360 AND S/370 MACHINES FOR SPICE	
				162	* 2G.2	
				163	*	
				164	*****	
0001B0	47F0 F00C	001BC	001B0	165	USING *,R15 A FAIRLY STANDARD ENTRY SEQUENCE	
0001B4	07			166	B NAME	
0001B5	C3D6C4C5E7C340			167	DC AL1(7)	
				168	DC CL7'CODEXC'	
				169	ENTRY SCRTCH	
				170	ENTRY NXTCOD	
0001BC	90EC D00C	0000C		171	NAME STM R14,R12,12(R13) SAVE REGISTERS	
0001C0	4130 F040	001F0		172	LA R3,AREAEX	
0001C4	503D 0008	00008		173	ST R3,8(R13)	
0001C8	50D3 0004	00004		174	ST R13,4(R3)	
0001CC	180D			175	LR R0,R13 ACTIVATE NEW SAVE AREA	
0001CE	189F			176	LR R9,R15 SWITCH TO UNUSUAL BASE REGISTER	
				177	DROP R15	
			001B0	178	USING CODEXC,R9	
0001D0	58F0 90A8	00258		179	L R15,=A(SCRTCH)	
0001D4	58B1 0004	00004		180	L R11,4(R1) GET DATA ADDRESS	
0001D8	58CF 0010	00010		181	L R12,16(R15) PICK UP PIVTOL ADDR	
0001DC	684C 0000	00000		182	LD R4,0(R12) AND PIVTOL ITSELF	
0001E0	58CF 000C	0000C		183	L R12,12(R15) IGOOF ADDR	
0001E4	58CC 0000	00000		184	L R12,0(R12) AND IGOOF ITSELF	
0001E8	5851 0000	00000		185	L R5,0(R1) R5=ADDR OF CODE TO EXEC.	
0001EC	07F5			186	BR R5 SO GO DO IT	
0001F0				187	AREAEX DS 18F	
000238				188	NXTCOD DS F	
000240				189	SCRTCH DS 1D SCRATCH SPACE	
000248	600D 0000	00000		190	STD F0,0(R13) OBJECT OF EXECUTE INSTRUCTION	
00024C				191	DS 2F MORE SCRATCH SPACE OF A DIFF. TYPE	
000258				192	LTORG	
000258	00000240			193	=A(SCRTCH)	
				194	DROP	

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM	0201	15.44	06/02/20
000260					196	DATIME	CSECT				
000260	47FF	000C	0000C		197		BC 15,12(15)				
000264	07				198		DC X'7'				
000265	C4C1E3C9D4C540				199		DC CL7'DATIME '				
00026C	90EC	D00C	0000C		200		STM 14,12,12(13)				
000270	05C0				201		BALR 12,0				
				00272	202		USING *,12				
000272	4160	C026	00298		203		LA 6,SOURCE				
000276	5871	0000	00000		204		L 7,0(1)				
00027A	5881	0004	00004		205		L 8,4(1)				
00027E	8360000C				206		DC X'83',XL3'60000C'				
000282	D207	7000	C026	00000	00298	207	MVC 0(8,7),SOURCE				
000288	D207	8000	C02E	00000	002A0	208	MVC 0(8,8),SOURCE+8				
00028E	98EC	D00C	0000C		209		LM 14,12,12(13)				
000292	92FF	D00C	0000C		210		MVI 12(13),X'FF'				
000296	07FE				211		BCR 15,14				
000298					212	SOURCE	DS 4D				
0002B8					213		LTORG				
					214		DROP				

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 15.44 06/02/20
0002B8				216	MINS3A CSECT	
				217	*****	
				218	*	
				219	* MINS3A(X,Z):	
				220	*	
				221	*FUNCTIN TO PROVIDE MACHINE CODE FOR A MINS3A OPERATION. THE CODE	
				222	*PRODUCED IS EQUIVALENT TO VALUE(X)=VALUE(X)-VALUE(Y)*VALUE(Z) WERE	
				223	* Y IS ASSUMED TO BE LEFT IF FPR2 FROM A PREVIOUS OPERATION(MINS2),	
				224	* AND X AND Z ARE THE TWO PARAMETERS. THE RETURN VALUE IS	
				225	*THE NUMBER OF HALFWORDS OF CODE PRODUCED.	
				226	*	
				227	* PROGRAMMED BY T. QUARLES 4/9/81 AS PART OF THE SPICE 2G.2	
				228	* CODE GENERATION PACKAGE FOR IBM SYSTEM/360 /370 ARCHITECTURE MACHINES	
				229	*	
				230	*	
				231	*	
				232	*****	
0002B8	47F0 F00C	002C4	002B8	233	USING *,R15	
0002BC	07			234	B S3A STANDARD HEADER AND ENTRY SEQUENCE	
0002BD	D4C9D5E2F3C140			235	DC AL1(7)	
0002C4	90EC D00C	0000C		236	DC CL7'MINS3A'	
				237	S3A STM R14,R12,12(R13)	
				238	DROP R15	
			002B8	239	USING MINS3A,R12	
0002C8	18CF			240	LR R12,R15	
0002CA	4130 C09C	00354		241	LA R3,AREA3A	
0002CE	503D 0008	00008		242	ST R3,8(R13)	
0002D2	50D3 0004	00004		243	ST R13,4(R3)	
0002D6	18D3			244	LR R13,R3	
0002D8	58B0 C0F8	003B0		245	L R11,=V(NXTCOD) GET ADDRESS OF CODE POINTER	
0002DC	5831 0000	00000		246	L R3,0(R1) GET FIRST PRAMETER ADDRESS	
0002E0	5851 0004	00004		247	L R5,4(R1) GET SECOND PARAMETER ADDRESS	
0002E4	5813 0000	00000		248	L R1,0(R3) GET FIRST ADDRESS	
0002E8	58F0 C0FC	003B4		249	L R15,=V(ADDGEN) TRANSLATE TO RX FORM OF ADDRESS	
0002EC	05EF			250	BALR R14,R15	
0002EE	1831			251	LR R3,R1 SAVE RESULT	
0002F0	1860			252	LR R6,R0 SAVE AMT. OF CODE PRODUCED	
0002F2	5815 0000	00000		253	L R1,0(R5) REPEAT FOR SECOND PARAMETER	
0002F6	58F0 C0FC	003B4		254	L R15,=V(ADDGEN)	
0002FA	05EF			255	BALR R14,R15	
0002FC	1851			256	LR R5,R1	
0002FE	1A60			257	AR R6,R0	
000300	5A60 C100	003B8		258	A R6,=F'8' ADD NUMBER OF 1/2 WORDS WE WILL PRODUCE	
000304	58AB 0000	00000		259	L R10,0(R11) GET CODE DEST. ADDRESS	
000308	4890 C0E4	0039C		260	LH R9,LD6INST GET THE LD R6, INSTRUCTIN	
00030C	409A 0000	00000		261	STH R9,0(R10) AND OUTPUT	
000310	405A 0002	00002		262	STH R5,2(R10) WITH AN ADDRESS INSERTED	
000314	4880 C0E8	003A0		263	LH R8,MDRINST GET A MDR INSTRUCTION	
000318	408A 0004	00004		264	STH R8,4(R10) OUTPUT	
00031C	4890 C0EA	003A2		265	LH R9,LD0INST GET A LD R0, INSTRUCTION	
000320	409A 0006	00006		266	STH R9,6(R10) OUTPT	
000324	403A 0008	00008		267	STH R3,8(R10) WITH AN ADDRESS FIELD	
000328	4890 C0EE	003A6		268	LH R9,SDRINST GET AN SDR INSTRUCTION	
00032C	409A 000A	0000A		269	STH R9,10(R10) AND OUTPUT	
000330	4890 C0F0	003A8		270	LH R9,STDINST GET A STD INSTRUCTION	

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 15.44 06/02/20
000334	409A	000C	0000C		271	STH	R9,12(R10)	AND WRITE OUT
000338	403A	000E	0000E		272	STH	R3,14(R10)	WITH THE FINAL ADDRESS
00033C	5AA0	C104	003BC		273	A	R10,=F'16'	ADD THE CODE SPACE WE USED TO ADDRESS
000340	50AB	0000	00000		274	ST	R10,0(R11)	AND PUT ADDRESS BACK WHERE WE FOUND IT
000344	1806				275	LR	R0,R6	COPY 1/2 WORDS USED COUNT TO RETURN REG.
000346	58DD	0004	00004		276	L	R13,4(R13)	GET OLD SAVE AREA POINTER
00034A	98EF	D00C	0000C		277	LM	R14,R15,12(R13)	RECOVER SAVED REGISTERS
00034E	981C	D018	00018		278	LM	R1,R12,24(R13)	... CONTINUED ...
000352	07FE				279	BR	R14	AND RETURN
					280	*****		
					281	*		
					282	*		
					283	*		
					284	CONSTANTS AND DATA AREAS		
					285	*		
000354					286	AREA3A	DS	18F
					287	*****		
					288	*		
					289	CODE PRODUCED		
					290	*		
00039C	686B	B00B	0000B		291	LD6INST	LD	F6,11(R11,R11) F6=Z
0003A0	2C62				292	MDRINST	MDR	F6,F2 F6=Y*Z
0003A2	680B	B00B	0000B		293	LD0INST	LD	F0,11(R11,R11) F0=X
0003A6	2B06				294	SDRINST	SDR	F0,F6 F0=X-Y*Z
0003A8	600B	B00B	0000B		295	STDINST	STD	F0,11(R11,R11) X=X-Y*Z
0003B0					296	LTORG		
0003B0	00000000				297	=V(NXTCOD)		
0003B4	00000000				298	=V(ADDGEN)		
0003B8	00000008				299	=F'8'		
0003BC	00000010				300	=F'16'		
					301	DROP		

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM	0201	15.44	06/02/20
0003C0					303	LOC	CSECT				
					304	*	I = LOC (X)				
				003C0	306		USING *,15				
0003C0	5800	1000	00000		307		L 0,0(0,1)				
0003C4	5400	F010	003D0		308		N 0,MASK				
0003C8	1BFF				309		SR 15,15				
0003CA	07FE				310		BR 14				
0003CC					312		DS F				
0003D0	00FFFFFF				313	MASK	DC X'00FFFFFF'				
0003D8					314		LTORG				
					315		DROP				

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 15.44 06/02/20
0003D8				317	MINS0 CSECT	
				318	*****	
				319	*	
				320	* MINS0(PIVTOL,IGOOF,NODPLC(LV1))	
				321	*	
				322	* FORTRAN CALLABLE FUNCTION TO PERFORM INITIALIZATION FOR THE	
				323	* MACHINE CODE GENERATION PACKAGE FOR SPICE 2G.2 ON THE IBM	
				324	* SYSTEM/360 /370 ARCHITECTURE MACHINES.	
				325	*	
				326	* WRITTEN BY T. QUARLES 5/1/81	
				327	*****	
			003D8	328	USING *,R15	
0003D8	47F0 F00A	003E2		329	B S0 USE A STANDARD HEADER AND ENTRY SEQUENCE	
0003DC	05			330	DC AL1(5)	
0003DD	D4C9D5E2F0			331	DC CL5'MINS0'	
0003E2	90EC D00C	0000C		332	S0 STM R14,R12,12(R13)	
0003E6	4130 F06C	00444		333	LA R3,AREA0	
0003EA	50D3 0004	00004		334	ST R13,4(R3)	
0003EE	503D 0008	00008		335	ST R3,8(R13)	
0003F2	18D3			336	LR R13,R3	
0003F4	18CF			337	LR R12,R15	
				338	DROP R15	
			003D8	339	USING MINS0,R12	
0003F6	5831 0000	00000		340	L R3,0(R1) ADDR OF PIVTOL	
0003FA	5841 0004	00004		341	L R4,4(R1) ADDR OF IGOOF	
0003FE	5851 0008	00008		342	L R5,8(R1) ADDR OF WHERE TO PUT CODE	
000402	5860 C0B8	00490		343	L R6,=V(SCRATCH) ADDR OF SCRATCH AREA	
000406	5046 000C	0000C		344	ST R4,12(R6) SAVE IGOOF ADDR FOR LATER REF. (CODEXC)	
00040A	5036 0010	00010		345	ST R3,16(R6) SAVE PIVTOL ADDR FOR LATER REF. (CODEXC)	
00040E	5860 C0BC	00494		346	L R6,=V(NXTCOD)	
000412	5056 0000	00000		347	ST R5,0(R6) REMEMBER WHERE TO PUT CODE	
000416	5860 C0C0	00498		348	L R6,=V(TRPNUM) AND SET NUMBER OF TRIPS THROUGH ADDR. FCT	
00041A	1755			349	XR R5,R5 TO 0 XR CLEARS REGISTER 5	
00041C	5056 0000	00000		350	ST R5,0(R6)	
000420	5860 C0C4	0049C		351	L R6,=V(REGVAL) CLEAR REGVAL ARRAY TO -1	
000424	5850 C0C8	004A0		352	L R5,=F'-1'	
000428	5056 0000	00000		353	ST R5,0(R6) PUT IN TABLE	
00042C	D24B 6004 6000	00004 00000		354	MVC 4(76,R6),0(R6) AND PROPAGATE THROUGH NEXT 76 BYTES(19 WD	
000432	58DD 0004	00004		355	L R13,4(R13) STANDARD EXIT SEQUENCE	
000436	98EC D00C	0000C		356	LM R14,R12,12(R13)	
00043A	1700			357	XR R0,R0	
00043C	17FF			358	XR R15,R15	
00043E	92FF D00C	0000C		359	MVI 12(R13),X'FF'	
000442	07FE			360	BR R14	
000444				361	AREA0 DS 18F	
000490				362	LTORG	
000490	00000000			363	=V(SCRATCH)	
000494	00000000			364	=V(NXTCOD)	
000498	00000000			365	=V(TRPNUM)	
00049C	00000000			366	=V(REGVAL)	
0004A0	FFFFFFFF			367	=F'-1'	
				368	DROP	

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 15.44 06/02/20
0004A8				370	MINS1 CSECT	
				371	*****	
				372	* MINS1(X): IBM ASSEMBLER ROUTINE (FORTRAN CALLABLE) TO GENERATE	
				373	* THE MACHINE CODE TO PERFORM A MINS1 OPERATION:	
				374	* CODE GENERATED IS EQUIVALENT TO	
				375	* IF(ABS(X).GE.PIVTOL) GO TO 10	
				376	* X=PIVTOL	
				377	* IGOOF=IGOOF+1	
				378	* 10 CONTINUE	
				379	*	
				380	* WRITTEN BY T. QUARLES 4/30/81	
				381	*****	
		004A8		382	USING *,R15 STANDARD HEADER AND ENTRY SEQUENCE CODE	
0004A8	47F0 F00A	004B2		383	B S1	
0004AC	05			384	DC AL1(5)	
0004AD	D4C9D5E2F1			385	DC CL5'MINS1'	
0004B2	90EC D00C	0000C		386	S1 STM R14,R12,12(R13)	
0004B6	18CF			387	LR R12,R15	
				388	DROP R15	
		004A8		389	USING MINS1,R12	
0004B8	4130 C07C	00524		390	LA R3,AREA1	
0004BC	503D 0008	00008		391	ST R3,8(R13)	
0004C0	50D3 0004	00004		392	ST R13,4(R3)	
0004C4	18D3			393	LR R13,R3	
0004C6	5811 0000	00000		394	L R1,0(R1) GET PARAMETER	
0004CA	5811 0000	00000		395	L R1,0(R1)	
0004CE	58F0 C0E8	00590		396	L R15,=V(ADDGEN) GET CONVERTED TO RX STYLE ADDRESS	
0004D2	05EF			397	BALR R14,R15	
0004D4	1831			398	LR R3,R1 SAVE RESULT	
0004D6	5890 C0EC	00594		399	L R9,=V(NXTCOD) FIND WHERE TO PUT CODE	
0004DA	58A9 0000	00000		400	L R10,0(R9)	
0004DE	4850 C0C4	0056C		401	LH R5,LDINST MOVE 1ST INSTRUCTION & PUT IN ADDR.	
0004E2	405A 0000	00000		402	STH R5,0(R10)	
0004E6	403A 0002	00002		403	STH R3,2(R10)	
0004EA	D215 A004 C0C8	00004 00570		404	MVC 4(22,R10),LDRINST MOVE BLOCK OF FIXED INSTRUCTIONS	
0004F0	4850 C0DE	00586		405	LH R5,LAINST1 PUT IN LAST TWO INSTRUCTIONS	
0004F4	405A 001A	0001A		406	STH R5,26(R10)	
0004F8	403A 001C	0001C		407	STH R3,28(R10)	
0004FC	D203 A01E C0E2	0001E 0058A		408	MVC 30(4,R10),EXINST	
000502	5AA0 C0F0	00598		409	A R10,=F'34' ADD NUMBER OF BYTES USED	
000506	5A00 C0F4	0059C		410	A R0,=F'17' RETURN NUMBER OF HALF WORDS USED	
00050A	50A9 0000	00000		411	ST R10,0(R9)	
00050E	58DD 0004	00004		412	L R13,4(R13) NORMAL RETURN SEQUENCE	
000512	98EF D00C	0000C		413	LM R14,R15,12(R13)	
000516	981C D018	00018		414	LM R1,12,24(R13)	
00051A	17FF			415	XR R15,R15	
00051C	92FF D00C	0000C		416	MVI 12(R13),X'FF'	
000520	07FE			417	BR R14	
000524				418	AREA1 DS 18F SAVE AREA	
				419	*****	
				420	*	
				421	* CODE PRODUCED	
				422	*	
				423	*****	
00056C	680B B00B	0000B		424	LDINST LD F0,11(R11,R11) 2ND OP IS X	

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 15.44 06/02/20
000570	2060				425	LDRINST	LPDR F6,F0	
000572	2B64				426		SDR F6,F4	
000574	606F	0000	00000		427		STD F6,0(R15)	
000578	58EF	0000	00000		428		L R14,0(R15)	
00057C	88E0	001F	0001F		429		SRL R14,31	
000580	1ACE				430		AR R12,R14	
000582	89E0	0006	00006		431		SLL R14,6	
000586	41DB	B00B	0000B		432	LAINST1	LA R13,11(R11,R11)	2ND OP IS X
00058A	44EF	0008	00008		433	EXINST	EX R14,8(R15)	
000590					434		LTORG	
000590	00000000				435		=V(ADDGEN)	
000594	00000000				436		=V(NXTCOD)	
000598	00000022				437		=F'34'	
00059C	00000011				438		=F'17'	
					439		DROP	

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 15.44 06/02/20
0005A0				441	MINS2 CSECT	
				442	*****	
				443	*	
				444	* MINS2(X,Y):	
				445	* FUNCTION TO PRODUCE MACHINE CODE FOR A MINS2 OPERATION.	
				446	* THE CODE PRODUCED IS EQUIVALENT TO VALUE(X)=VALUE(X)/VALUE(Y),	
				447	* THE RETURN VALUE OF THE CALL IS THE NUMBER OF HALF WORDS USED	
				448	* FOR THE CODE PRODUCED.	
				449	* LEAVES RESULT IN FPR 2 IN ADDITION TO STORING IN PROPER PLACE	
				450	* IN VALUE.	
				451	*	
				452	*	
				453	* WRITTEN BY TOM QUARLES 5/1/81 AS PART OF THE IBM S/360 AND	
				454	* S/370 CODE GENERATION PACKAGE FOR SPICE 2G.2	
				455	*****	
		005A0		456	USING *,R15	
0005A0	47F0 F00A	005AA		457	B S2 USE A STANDARD ENTRY SEQUENCE AND HEADER	
0005A4	05			458	DC AL1(5)	
0005A5	D4C9D5E2F2			459	DC CL5'MINS2'	
0005AA	90EC D00C	0000C		460	S2 STM 14,12,12(R13)	
				461	DROP R15	
		005A0		462	USING MINS2,R12	
0005AE	18CF			463	LR R12,R15 SWITCH TO ANOTHER BASE REGISTER	
0005B0	4130 C08C	0062C		464	LA R3,AREA2	
0005B4	503D 0008	00008		465	ST R3,8(R13)	
0005B8	50D3 0004	00004		466	ST R13,4(R3)	
0005BC	18D3			467	LR R13,R3	
0005BE	58B0 C0E0	00680		468	L R11,=V(NXTCOD) GET ADDRESS OF ADDRESS WHERE CODE GOES	
0005C2	5831 0000	00000		469	L R3,0(R1) GET ADDRESS OF FIRST PARAMETER	
0005C6	5841 0004	00004		470	L R4,4(R1) GET ADDRESS OF SECOND PARAMETER	
0005CA	5813 0000	00000		471	L R1,0(R3)	
0005CE	58F0 C0E4	00684		472	L R15,=V(ADDGEN) AND GET IT TRANSLATED TO RX FORM	
0005D2	05EF			473	BALR R14,R15	
0005D4	1831			474	LR R3,R1 SAVE THE RESULTING HALFWORD	
0005D6	1860			475	LR R6,R0 SAVE SPACE USED TO SET UP BASE REGISTERS	
0005D8	5814 0000	00000		476	L R1,0(R4) REPEAT ABOVE FOR SECOND PARAMETER	
0005DC	58F0 C0E4	00684		477	L R15,=V(ADDGEN)	
0005E0	05EF			478	BALR R14,R15	
0005E2	1841			479	LR R4,R1	
0005E4	1A60			480	AR R6,R0	
0005E6	58AB 0000	00000		481	L R10,0(R11) GET ADDRESS FOR NEXT CODE	
0005EA	4890 C0D4	00674		482	LH R9,LDINST2 GET THE LOAD DOUBLE INSTRUCTION	
0005EE	409A 0000	00000		483	STH R9,0(R10) AND PUT OUT AS PART OF THE CODE	
0005F2	403A 0002	00002		484	STH R3,2(R10) AND PUT AN ADDRESS IN IT	
0005F6	4890 C0D8	00678		485	LH R9,DDINST GET A DIVIDE DOUBLE INSTRUCTION	
0005FA	409A 0004	00004		486	STH R9,4(R10) AND GENERATE IT AS CODE	
0005FE	404A 0006	00006		487	STH R4,6(R10) WITH AN APPROPRIATE ADDRESS	
000602	4890 C0DC	0067C		488	LH R9,STDINST2 GET A STORE DOUBLE INSTRUCTION	
000606	409A 0008	00008		489	STH R9,8(R10) GENERATE IT AS CODE	
00060A	403A 000A	0000A		490	STH R3,10(R10) WITH AN ADDRESS	
00060E	5AA0 C0E8	00688		491	A R10,=F'12' NEED TO UPDATE THE NEXT INSTRUCTION ADDRESS	
000612	50AB 0000	00000		492	ST R10,0(R11) AND PUT IT BACK WHERE WE FOUND IT	
000616	5A60 C0EC	0068C		493	A R6,=F'6' AND GENERATE THE HALFWORDS PRODUCED COUNT	
00061A	1806			494	LR R0,R6 AND PUT IT IN THE RIGHT REGISTER	
00061C	58DD 0004	00004		495	L R13,4(R13) GET BACK SAVE AREA POINTER	

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 15.44 06/02/20
000620	98EF	D00C	0000C		496	LM	R14,R15,12(R13) RECOVER CALLER'S REG. SET	
000624	981C	D018	00018		497	LM	R1,R12,24(R13) CONTINUED...	
000628	07FE				498	BR	R14 AD RETURN TO CALLER	
					499	*****		
					500	*		
					501	*		
					502	*		
					503	*		
					504	*		
					505	*		
					506	*		
00062C					507	AREA2 DS 18F	SAVE AREA	
					508	*****		
					509	*		
					510	*		
					511	*		
000674	682B	B00B	0000B		512	LDINST2 LD	F2,11(R11,R11) FAKE LOAD DOUBLE INSTRUCTION	
000678	6D2B	B00B	0000B		513	DDINST DD	F2,11(R11,R11) FAKE DIVIDED DOUBLE INSTRUCTION	
00067C	602B	B00B	0000B		514	STDINST2 STD	F2,11(R11,R11) FAKE STORE DOUBLE INSTRUCTION	
					515	*		
					516	*		
					517	*		
000680					518	LTORG		
000680	00000000				519	=V(NXTCOD)		
000684	00000000				520	=V(ADDGEN)		
000688	0000000C				521	=F'12'		
00068C	00000006				522	=F'6'		
					523	DROP		

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 15.44 06/02/20
000690				525	MINS4 CSECT	
				526	*****	
				527	* MINS4: ASSEMBLER SUBROUTINE TO GENERATE MACHINE CODE FOR	
				528	* A MINS4 OPERATION.	
				529	* FORTRAN CALLABLE!	
				530	* PRODUCES CODE TO TERMINATE MINS SEQUENCE BY RESTORING IGOOF TO	
				531	* MEMORY, PERFORMING CLEAN UP AND RETURNING TO THE CALLER OF	
				532	* CODEXC	
				533	*	
				534	* WRITTEN BY T. QUARLES 4/30/81	
				535	*****	
		00690		536	USING *,R15	
000690	47F0 F00A	0069A		537	B S4 STANDARD HEADER AND ENTRY SEQUENCE	
000694	05			538	DC AL1(5)	
000695	D4C9D5E2F4			539	DC CL5'MINS4'	
00069A	90EC D00C	0000C		540	S4 STM R14,R12,12(R13)	
00069E	18CF			541	LR R12,R15	
				542	DROP R15	
		00690		543	USING MINS4,R12	
0006A0	4130 C048	006D8		544	LA R3,AREA4	
0006A4	503D 0008	00008		545	ST R3,8(R13)	
0006A8	50D3 0004	00004		546	ST R13,4(R3)	
0006AC	18D3			547	LR R13,R3	
0006AE	5890 C0A8	00738		548	L R9,=V(NXTCOD) FIND OUT WHERE TO PUT CODE	
0006B2	58A9 0000	00000		549	L R10,0(R9)	
0006B6	D215 A000 C090	00000	00720	550	MVC 0(22,R10),CODE MOVE CODE IN	
0006BC	5AA0 C0AC	0073C		551	A R10,=F'22' UPDATE WHERE TO PUT CODE POINTER	
0006C0	50A9 0000	00000		552	ST R10,0(R9)	
0006C4	5800 C0B0	00740		553	L R0,=F'11'	
0006C8	58DD 0004	00004		554	L R13,4(R13) RETURN SEQUENCE	
0006CC	981C D018	00018		555	LM R1,R12,24(R13)	
0006D0	92FF D00C	0000C		556	MVI 12(R13),X'FF'	
0006D4	17FF			557	XR R15,R15	
0006D6	07FE			558	BR R14	
				559	*****	
				560	*	
				561	*	
				562	*	
				563	* CONSTANTS AND DATA AREAS	
				564	*	
				565	*	
0006D8				566	AREA4 DS 18F	
				567	*****	
				568	*	
				569	*	
				570	* CODE PRODUCED	
				571	*	
000720	58AF 000C	0000C		572	CODE L R10,12(R15) RECOVER IGOOF ADDRESS	
000724	50CA 0000	00000		573	ST R12,0(R10) SAVE IGOOF	
000728	18D0			574	LR R13,R0 RESTORE SAVE AREA POINTER	
00072A	98EC D00C	0000C		575	LM R14,R12,12(R13) RESTORE REGS.	
00072E	17FF			576	XR R15,R15 SAY DONE	
000730	92FF D00C	0000C		577	MVI 12(R13),X'FF' SAY DONE	
000734	07FE			578	BR R14 AND LEAVE	
000738				579	LTORG	

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 15.44 06/02/20
000738	00000000			580	=V(NXTCOD)	
00073C	00000016			581	=F ' 22 '	
000740	0000000B			582	=F ' 11 '	
				583	DROP	

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 15.44 06/02/20
000748				585	MINS3 CSECT	
				586	*****	
				587	*	
				588	* MINS3(X,Y,Z):	
				589	*	
				590	*FUNCTIN TO PROVIDE MACHINE CODE FOR A MINS3 OPERATION. THE CODE	
				591	*PRODUCED IS EQUIVALENT TO VALUE(X)+VALUE(X)-VALUE(Y)*VALUE(Z) WERE	
				592	*THE SUBSCRIPTS X,Y,Z ARE THE THREE PARAMETERS. THE RETURN VALUE IS	
				593	*THE NUMBER OF HALFWORDS OF CODE PRODUCED.	
				594	*	
				595	* PROGRAMMED BY T. QUARLES 4/9/81 AS PART OF THE SPICE 2G.2	
				596	* CODE GENERATION PACKAGE FOR IBM SYSTEM/360 /370 ARCHITECTURE MACHINES	
				597	*	
				598	*	
				599	*	
				600	*****	
000748	47F0 F00A	00752	00748	601	USING *,R15	
00074C	05			602	B S3 STANDARD HEADER AND ENTRY SEQUENCE	
00074D	D4C9D5E2F3			603	DC AL1(5)	
000752	90EC D00C	0000C		604	DC CL5'MINS3'	
				605	S3 STM R14,R12,12(R13)	
				606	DROP R15	
000756	18CF		00748	607	USING MINS3,R12	
000758	4130 C0B0	007F8		608	LR R12,R15	
00075C	503D 0008	00008		609	LA R3,AREA3	
000760	50D3 0004	00004		610	ST R3,8(R13)	
000764	18D3			611	ST R13,4(R3)	
000766	58B0 C110	00858		612	LR R13,R3	
00076A	5831 0000	00000		613	L R11,=V(NXTCOD) GET ADDRESS OF CODE POINTER	
00076E	5841 0004	00004		614	L R3,0(R1) GET FIRST PRAMETER ADDRESS	
000772	5851 0008	00008		615	L R4,4(R1) GET SECOND PARAMETER ADDRESS	
000776	5813 0000	00000		616	L R5,8(R1) GET THIRD PARAMETER ADDRESS	
00077A	58F0 C114	0085C		617	L R1,0(R3) GET FIRST ADDRESS	
00077E	05EF			618	L R15,=V(ADDGEN) TRANSLATE TO RX FORM OF ADDRESS	
000780	1831			619	BALR R14,R15	
000782	1860			620	LR R3,R1 SAVE RESULT	
000784	5814 0000	00000		621	LR R6,R0 SAVE AMT. OF CODE PRODUCED	
000788	58F0 C114	0085C		622	L R1,0(R4) REPEAT FOR SECOND PARAMETER	
00078C	05EF			623	L R15,=V(ADDGEN)	
00078E	1841			624	BALR R14,R15	
000790	1A60			625	LR R4,R1	
000792	5815 0000	00000		626	AR R6,R0	
000796	58F0 C114	0085C		627	L R1,0(R5) AND AGAIN FOR THE THIRD PARAMETER	
00079A	05EF			628	L R15,=V(ADDGEN)	
00079C	1851			629	BALR R14,R15	
00079E	1A60			630	LR R5,R1	
0007A0	5A60 C118	00860		631	AR R6,R0	
0007A4	58AB 0000	00000		632	A R6,=F'9' ADD NUMBER OF 1/2 WORDS WE WILL PRODUCE	
0007A8	4890 C0F8	00840		633	L R10,0(R11) GET CODE DEST. ADDRESS	
0007AC	409A 0000	00000		634	LH R9,LD6INST3 GET THE LD R6, INSTRUCTIN	
0007B0	405A 0002	00002		635	STH R9,0(R10) AND OUTPUT	
0007B4	4880 C0FC	00844		636	STH R5,2(R10) WITH AN ADDRESS INSERTED	
0007B8	408A 0004	00004		637	LH R8,MDINST GET A MD INSTRUCTION	
0007BC	404A 0006	00006		638	STH R8,4(R10) OUTPUT	
				639	STH R4,6(R10) WITH AN ADDRESS	

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 15.44 06/02/20
0007C0	4890	C100	00848		640	LH	R9,LD0INST3	GET A LD R0, INSTRUCTION
0007C4	409A	0008	00008		641	STH	R9,8(R10)	OUTPT
0007C8	403A	000A	0000A		642	STH	R3,10(R10)	WITH AN ADDRESS FIELD
0007CC	4890	C104	0084C		643	LH	R9,SDRINST3	GET AN SDR INSTRUCTION
0007D0	409A	000C	0000C		644	STH	R9,12(R10)	AND OUTPUT
0007D4	4890	C106	0084E		645	LH	R9,STDINST3	GET A STD INSTRUCTION
0007D8	409A	000E	0000E		646	STH	R9,14(R10)	AND WRITE OUT
0007DC	403A	0010	00010		647	STH	R3,16(R10)	WITH THE FINAL ADDRESS
0007E0	5AA0	C11C	00864		648	A	R10,=F'18'	ADD THE CODE SPACE WE USED TO ADDRESS
0007E4	50AB	0000	00000		649	ST	R10,0(R11)	AND PUT ADDRESS BACK WHERE WE FOUND IT
0007E8	1806				650	LR	R0,R6	COPY 1/2 WORDS USED COUNT TO RETURN REG.
0007EA	58DD	0004	00004		651	L	R13,4(R13)	GET OLD SAVE AREA POINTER
0007EE	98EF	D00C	0000C		652	LM	R14,R15,12(R13)	RECOVER SAVED REGISTERS
0007F2	981C	D018	00018		653	LM	R1,R12,24(R13)	... CONTINUED ...
0007F6	07FE				654	BR	R14	AND RETURN
					655	*****		
					656	*		
					657	*		
					658	*		
					659	CONSTANTS AND DATA AREAS		
					660	*		
0007F8					661	AREA3	DS	18F
					662	*****		
					663	*		
					664	CODE PRODUCED		
					665	*		
000840	686B	B00B	0000B		666	LD6INST3	LD	F6,11(R11,R11) F6=Z
000844	6C6B	B00B	0000B		667	MDINST	MD	F6,11(R11,R11) F6=Y*Z
000848	680B	B00B	0000B		668	LD0INST3	LD	F0,11(R11,R11) F0=X
00084C	2B06				669	SDRINST3	SDR	F0,F6 F0=X-Y*Z
00084E	600B	B00B	0000B		670	STDINST3	STD	F0,11(R11,R11) X=X-Y*Z
000858					671	LTORG		
000858	00000000				672	=V(NXTCOD)		
00085C	00000000				673	=V(ADDGEN)		
000860	00000009				674	=F'9'		
000864	00000012				675	=F'18'		
					676	DROP		

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	ASM 0201 15.44 06/02/20
000868				678	GTMAIN CSECT	
				679	*****	
				680	*	
				681	* GTMAIN(KFAMWA,MAXMEM)	
				682	*	
				683	* RETURNS KFAMWA=ADDRESS OF FIRST AVAILABLE WORD OF GETMAIN	
				684	* FREE STORAGE, MAXMEM=NUMBER OF WORDS AVAILABLE FOR USE	
				685	*	
				686	*****	
			00868	687	USING *,R15	
000868	47F0 F00C	00874		688	B GMN	
00086C	07			689	DC AL1(7)	
00086D	C7E3D4C1C9D540			690	DC CL7'GTMAIN'	
000874	90EC D00C	0000C		691	GMN STM R14,R12,12(R13)	
000878	4130 F0B4	0091C		692	LA R3,MAREA STANDARD ENTRY SEQUENCE FOR ROUTINE	
00087C	50D3 0004	00004		693	ST R13,4(R3)	
000880	503D 0008	00008		694	ST R3,8(R13)	
000884	18D3			695	LR R13,R3	
000886	18CF			696	LR R12,R15	
000888	1891			697	LR R9,R1 SAVE PARM POINTER	
				698	DROP R15	
			00868	699	USING GTMAIN,R12	
				700	GTMAIN VC,LA=SIZE,A=WHERE1	
				701+*	OS/V52 RELEASE 4 VERSION -- 10/21/75	00004804
00088A	0700			702+	CNOP 0,4	00366000
00088C	4510 C032	0089A		703+	BAL 1,*,+14 BRANCH AROUND LIST	00368000
000890	00000964			704+	DC A(SIZE) ADDR. OF LENGTH LIST	00386000
000894	0000096C			705+	DC A(WHERE1) ADDR. OF ADDR. LIST	00406002
000898	E0			706+	DC BL1'11100000' MODE AND OPTION FLAGS	00462002
000899	00			707+	DC AL1(0) SUBPOOL VALUE	00510000
00089A	0A04			708+	SVC 4 ISSUE GETMAIN SVC	00820002
00089C	58A0 C108	00970		709	L R10,WHERE1+4 GET SIZE OF SPACE AVAILABLE	
				710	FREEMAIN V,A=WHERE1 RETURN THE SPACE TO CMS	
				711+*	OS/V52 RELEASE 3 VERSION -- 10/25/74	00001603
0008A0				712+	CNOP 0,4	00092000
0008A0	4510 C046	008AE		713+	BAL 1,*,+14 BRANCH AROUND LIST	00093000
0008A4	00000000			714+	DC A(0)	00107000
0008A8	0000096C			715+	DC A(WHERE1) AREA LIST ADDRESS	00109000
0008AC	C0			716+	DC AL1(192) MODE BYTE	00119000
0008AD	00			717+	DC AL1(0) SUBPOOL VALUE	00127000
0008AE	0A05			718+	SVC 5 ISSUE FREEMAIN SVC	00311202
0008B0	5BA0 C110	00978		719	S R10,=F'32768' OFF A CHUNK FOR THE O/S =F'32768'	
0008B4	47D0 C0A2	0090A		720	BNP NOSPACE SINCE NOT ENOUGH ROOM LEFT FOR SYSTEM	
0008B8	50A0 C100	00968		721	ST R10,SIZE+4 PUT IN AS NEW MAX. NUM. BYTES	
				722	GTMAIN VC,LA=SIZE,A=WHERE1 AND ASK AGAIN.	
				723+*	OS/V52 RELEASE 4 VERSION -- 10/21/75	00004804
0008BC				724+	CNOP 0,4	00366000
0008BC	4510 C062	008CA		725+	BAL 1,*,+14 BRANCH AROUND LIST	00368000
0008C0	00000964			726+	DC A(SIZE) ADDR. OF LENGTH LIST	00386000
0008C4	0000096C			727+	DC A(WHERE1) ADDR. OF ADDR. LIST	00406002
0008C8	E0			728+	DC BL1'11100000' MODE AND OPTION FLAGS	00462002
0008C9	00			729+	DC AL1(0) SUBPOOL VALUE	00510000
0008CA	0A04			730+	SVC 4 ISSUE GETMAIN SVC	00820002
0008CC	1299			731	LTR R9,R9 CHECK IF PARMS SUPPLIED	
0008CE	4780 C0A4	0090C		732	BZ NOPARM	

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	SOURCE	STATEMENT	ASM 0201 15.44 06/02/20
0008D2	5869	0000	00000		733	L	R6,0(R9)	GET ADDR OF KFAMWA
0008D6	95FF	9000	00000		734	CLI	0(R9),X'FF'	CHECK FOR SECOND PARM
0008DA	4780	C0A4	0090C		735	BZ	NOPARM	
0008DE	5879	0004	00004		736	L	R7,4(R9)	GET ADDR OF MAXMEM
0008E2	5840	C104	0096C		737	L	R4,WHERE1	GET LOCATION OF AREA
0008E6	8840	0002	00002		738	SRL	R4,2	DIVIDE BY 2 TO GET ADDRESS IN WORDS
0008EA	5046	0000	00000		739	ST	R4,0(R6)	PUT IN KFAMWA
0008EE	5840	C108	00970		740	L	R4,WHERE1+4	GET SIZE OF AREA
0008F2	8840	0002	00002		741	SRL	R4,2	DIVIDE BY 4 TO GET SIZE IN WORDS
0008F6	5047	0000	00000		742	ST	R4,0(R7)	PUT IN MAXMEM
0008FA	58DD	0004	00004		743	NRET	L R13,4(R13)	
0008FE	98EC	D00C	0000C		744	LM	R14,R12,12(R13)	STANDARD RETURN SEQUENCE
000902	92FF	D00C	0000C		745	MVI	12(R13),X'FF'	
000906	17FF				746	XR	R15,R15	
000908	07FE				747	BR	R14	
00090A	0700				748	NOSPACE	NOPR R0	
00090C	58DD	0004	00004		749	NOPARM	L R13,4(R13)	
000910	98EC	D00C	0000C		750	LM	R14,R12,12(R13)	ERROR RETURN SEQUENCE
000914	41F0	0FFF	00FFF		751	LA	R15,4095	
000918	07FE				752	BR	R14	
00091C					753	MAREA	DS 18F	
000964	00000008				754	SIZE	DC F'8'	
000968	00FFFFFF8				755	DC	X'00FFFFFF8'	
00096C					756	WHERE1	DS 2F	
					757	DROP		
000978					758	LTORG		
000978	00008000				759		=F'32768'	
			00000		760	F0	EQU 0	
			00002		761	F2	EQU 2	
			00004		762	F4	EQU 4	
			00006		763	F6	EQU 6	
			00002		764	RSTAR4	EQU 2	
			00003		765	RSTAR8	EQU 3	
			00003		766	CSTAR8	EQU 3	
			00004		767	CSTAR16	EQU 4	
					768	END		

ASM 0201 15.44 06/02/20

POS.ID	REL.ID	FLAGS	ADDRESS
0001	0001	0C	000128
0001	0002	1C	000134
0004	0004	0C	000258
0006	0002	1C	0003B0
0006	0007	1C	0003B4
0009	0002	1C	000494
0009	000A	1C	000490
0009	000B	1C	000498
0009	000C	1C	00049C
000D	0002	1C	000594
000D	0007	1C	000590
000E	0002	1C	000680
000E	0007	1C	000684
000F	0002	1C	000738
0010	0002	1C	000858
0010	0007	1C	00085C
0011	0011	0C	000890
0011	0011	0C	000894
0011	0011	0C	0008A8
0011	0011	0C	0008C0
0011	0011	0C	0008C4

SYMBOL	LEN	VALUE	DEFN	REFERENCES	ASM 0201 15.44 06/02/20														
AREA	00008	00000180	00142	00130															
AREAEX	00004	000001F0	00187	00172															
AREA0	00004	00000444	00361	00333															
AREA1	00004	00000524	00418	00390															
AREA2	00004	0000062C	00507	00464															
AREA3	00004	000007F8	00661	00609															
AREA3A	00004	00000354	00286	00241															
AREA4	00004	000006D8	00566	00544															
CODE	00004	00000720	00572	00550															
CODEXC	00001	000001B0	00149	00178															
DDINST	00004	00000678	00513	00485															
DONE	00004	000000B2	00105	00058															
EXINST	00004	0000058A	00433	00408															
FOUND	00004	0000009C	00099	00063															
F0	00001	00000000	00760	00190	00293	00294	00295	00424	00425	00668	00669	00670							
F2	00001	00000002	00761	00292	00512	00513	00514												
F4	00001	00000004	00762	00426															
F6	00001	00000006	00763	00291	00292	00294	00425	00426	00427	00666	00667	00669							
GMN	00004	00000874	00691	00688															
GTMAIN	00001	00000868	00678	00699															
LAINST	00004	000000B8	00107	00087															
LAINST1	00004	00000586	00432	00405															
LDINST	00004	0000056C	00424	00401															
LDINST2	00004	00000674	00512	00482															
LDRINST	00002	00000570	00425	00404															
LD0INST	00004	000003A2	00293	00265															
LD0INST3	00004	00000848	00668	00640															
LD6INST	00004	0000039C	00291	00260															
LD6INST3	00004	00000840	00666	00634															
LOOP	00004	00000034	00062	00070															
MAREA	00004	0000091C	00753	00692															
MASK	00004	000003D0	00313	00308															
MDINST	00004	00000844	00667	00637															
MDRINST	00002	000003A0	00292	00263															
MINS0	00001	000003D8	00317	00339															
MINS1	00001	000004A8	00370	00389															
MINS2	00001	000005A0	00441	00462															
MINS3	00001	00000748	00585	00607															
MINS3A	00001	000002B8	00216	00239															
MINS4	00001	00000690	00525	00543															
NAME	00004	000001BC	00171	00166															
NOPARM	00004	0000090C	00749	00732	00735														
NOSPACE	00002	0000090A	00748	00720															
NOSWP	00004	0000004C	00069	00065															
NXTCOD	00004	00000238	00188	00170															
REGVAL	00004	000000C0	00109	00050	00116														
R0	00001	00000000	00011	00054	00054	00081	00107	00107	00108	00175	00252	00257	00275	00357	00357	00410	00475	00480	
				00494	00553	00574	00621	00626	00631	00650	00748								
R1	00001	00000001	00012	00052	00053	00055	00056	00100	00180	00185	00246	00247	00248	00251	00253	00256	00278	00340	
				00341	00342	00394	00394	00395	00395	00398	00414	00469	00470	00471	00474	00476	00479	00497	
				00555	00614	00615	00616	00617	00620	00622	00625	00627	00630	00653	00697				
R10	00001	0000000A	00021	00083	00084	00085	00086	00087	00089	00090	00091	00092	00093	00094	00095	00096	00097	00098	
				00259	00261	00262	00264	00266	00267	00269	00271	00272	00273	00274	00400	00402	00403	00404	
				00406	00407	00408	00409	00411	00481	00483	00484	00486	00487	00489	00490	00491	00492	00549	
				00550	00551	00552	00572	00573	00633	00635	00636	00638	00639	00641	00642	00644	00646	00647	

SYMBOL	LEN	VALUE	DEFN	REFERENCES	ASM 0201 15.44 06/02/20
=X'00FFFFFF'					
	00004	00000118	00112	00053	
=X'00000FFF'					
	00004	0000011C	00113	00056	
=X'7FFFFFFF'					
	00004	00000120	00114	00059	
=F'10'	00004	00000124	00115	00060	
=A(REGVAL)					
	00004	00000128	00116	00061	
=F'8'	00004	0000012C	00117	00069	00085
=F'4'	00004	00000130	00118	00081	
=V(NXTCOD)					
	00004	00000134	00119	00082	
=D'1000000.0'					
	00008	000001A0	00144	00137	
=X'00FFFFFF'					
	00004	000001A8	00145	00133	
=X'4E000000'					
	00004	000001AC	00146	00134	
=A(SCRATCH)					
	00004	00000258	00193	00179	
=V(NXTCOD)					
	00004	000003B0	00297	00245	
=V(ADDGEN)					
	00004	000003B4	00298	00249	00254
=F'8'	00004	000003B8	00299	00258	
=F'16'	00004	000003BC	00300	00273	
=V(SCRATCH)					
	00004	00000490	00363	00343	
=V(NXTCOD)					
	00004	00000494	00364	00346	
=V(TRPNUM)					
	00004	00000498	00365	00348	
=V(REGVAL)					
	00004	0000049C	00366	00351	
=F'-1'	00004	000004A0	00367	00352	
=V(ADDGEN)					
	00004	00000590	00435	00396	
=V(NXTCOD)					
	00004	00000594	00436	00399	
=F'34'	00004	00000598	00437	00409	
=F'17'	00004	0000059C	00438	00410	
=V(NXTCOD)					
	00004	00000680	00519	00468	
=V(ADDGEN)					
	00004	00000684	00520	00472	00477
=F'12'	00004	00000688	00521	00491	
=F'6'	00004	0000068C	00522	00493	
=V(NXTCOD)					
	00004	00000738	00580	00548	
=F'22'	00004	0000073C	00581	00551	
=F'11'	00004	00000740	00582	00553	
=V(NXTCOD)					
	00004	00000858	00672	00613	
=V(ADDGEN)					

SYMBOL					LEN					VALUE					DEFN					REFERENCES					ASM 0201 15.44 06/02/20				
										00004	0000085C	00673	00618	00623	00628														
=F'9'										00004	00000860	00674	00632																
=F'18'										00004	00000864	00675	00648																
=F'32768'										00004	00000978	00759	00719																

ASM 0201 15.44 06/02/20

NO STATEMENTS FLAGGED IN THIS ASSEMBLY

HIGHEST SEVERITY WAS 0

OPTIONS FOR THIS ASSEMBLY

ALIGN, ALOGIC, BUFSIZE(STD), DECK, ESD, FLAG(0), LINECOUNT(55), LIST, NOMCALL, YFLAG, WORKSIZE(2097152)

NOMLOGIC, NONUMBER, NOOBJECT, NORENT, RLD, NOSTMT, NOLIBMAC, NOTERMINAL, NOTEST, XREF(SHORT)

SYSPARM()

WORK FILE BUFFER SIZE/NUMBER =19066/ 1

TOTAL RECORDS READ FROM SYSTEM INPUT 672

TOTAL RECORDS READ FROM SYSTEM LIBRARY 1497

TOTAL RECORDS PUNCHED 60

TOTAL RECORDS PRINTED 1034

