

01  
02  
03  
04  
05  
06  
07  
08  
09  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47

;  
;  
;

RCSL: 43-GL10263  
AUTHOR: APR/DHA/SM/JHA  
EDITED: 80.05.13

;

MUC03

; KEYWORD:  
; ABSTRACT:  
;  
;

MUS, CHARACTER I/O, PROCEDURES, LISTING.  
MUS CHARACTER I/O PROCEDURES.  
ASCII PAPER TAPE RCSL 43-GL10262.  
REL. BINARY PAPER TAPE RCSL 43-GL10264.

10002 MUC03

01 ; \*\*\*\*\* CHARACTER I/O PROCEDURES \*\*\*\*\*

02

04 .TITL MUC03

05 .NREL

06 000001 .TXTM 1

07 000012 .RDX 10

08

09 00207 000207 .LOC INCHAR-GOS

10 000013\* A56 ;

11 00212 000212 .LOC OUTCHAR-GOS

12 000026\* A58 ;

13 00211 000211 .LOC OUTSPACE-GOS

14 000025\* A580

15 00213 000213 .LOC OUTNL-GOS

16 000050\* A590 ;

17 00214 000214 .LOC OUTEND-GOS

18 000051\* A59 ;

19 00215 000215 .LOC OUTTEXT-GOS

20 000057\* A60 ;

21 00216 000216 .LOC OUTOCTAL-GOS

22 000071\* A61 ;

23 00223 000223 .LOC INNAME-GOS

24 000107\* A82 ;

25

26 .XREL

10003 MUC03

```
01 ; LOCAL PROCEDURE TEST AND SET;
02 ; USED BY INCHAR AND OUTCHAR.
03
04 00000*021023 A545: LDA 0 ZREM,2 ; TEST AND SET:
05 00001*100405 NEG 0,0 SNR ; IF REM.ZONE=0 THEN
06 00002*001400 JMP 0,3 ; RETURN(0);
07 00003*100000 COM 0,0 ; REM.ZONE:=(-REM.ZONE)+1;
08 00004*041023 STA 0 ZREM,2 ;
09 00005*021020 LDA 0 ZTOP,2 ;
10 00006*041017 STA 0 ZFIRST,2 ; FIRST.ZONE:=TOP.ZONE;
11 00007*011020 ISZ ZTOP,2 ; INCR(TOP.ZONE);
12 00010*101220 MOVZR 0,0 ; VAL:=ADDR//2;
13 00011*041024 STA 0 Z0,2 ;
14 00012*001402 JMP 2,3 ; RETURN(2);
15
16 ; PROCEDURE INCHAR(ZONE,CHAR);
17 ; CALL: RETURN:
18 ; AC0 DESTROYED
19 ; AC1 CHAR
20 ; AC2 ZONE ZONE
21 ; AC3 LINK DESTROYED
22
23 00013*055026 A56: STA 3 Z2,2 ; INCHAR:
24 00014*004764 A560: JSR A545 ; WHILE TEST AND SET=0 DO
25 00015*006205 INBLOCK ; INBLOCK(ZONE);
26 00016*000776 JMP A560 ;
27 00017*027024 LDA 1 Z0,2 ; VALUE:= 0.(ADDR//2);
28 00020*125003 MOV 1,1 SNC ; IF ADDR(15)=0 THEN
29 00021*125300 MOVS 1,1 ; CHAR:= VALUE(0:7)
30 00022*034143 LDA 3 .255 ; ELSE
31 00023*167400 AND 3,1 ; CHAR:= VALUE(8:15);
32 00024*003026 JMP 2,2 ; RETURN;
33
```

```

01 ; PROCEDURE OUTSPACE(ZONE);
02 ; PERFORMS OUTCHAR(ZONE,32).
03 ; CALL: RETURN:
04 ; AC0 UNCHANGED
05 ; AC1 DESTROYED
06 ; AC2 ZONE ZONE
07 ; AC3 LINK DESTROYED
08
09 ; PROCEDURE OUTCHAR(ZONE,CHAR);
10 ; CHAR MUST BE IN THE RANGE 0 TO 255.
11 ; CALL: RETURN:
12 ; AC0 UNCHANGED
13 ; AC1 CHAR DESTROYED
14 ; AC2 ZONE ZONE
15 ; AC3 LINK DESTROYED
16
17 A580: ; OUTSPACE:
18 00025*024113 LDA 1 .32 ; CHAR:= 32;
19
20 00026*041026 A58: STA 0 Z2,2 ; OUTCHAR:
21 00027*055027 A581: STA 3 Z3,2 ; SAVE(AC0);
22 00030*045030 STA 1 Z4,2 ; SAVE(CHAR);
23 00031*004747 A583: JSR A545 ; WHILE TEST AND SET= 0 DO
24 00032*006206 OUTBLOCK ; OUTBLOCK;
25 00033*000776 JMP A583 ;
26 00034*025030 LDA 1 Z4,2 ;
27 00035*023024 LDA# 0 Z0,2 ; RESTORE(CHAR);
28 00036*034143 LDA 3 .255 ; VALUE:=0.ADDR//2
29 00037*167400 AND 3,1 ; IF ADDR(15) = 1 THEN
30 00040*101002 MOV 0,0 SZC ; VALUE:=VALUE-(VALUE EXTRACT 8)
31 00041*174001 COM 3,3 SKP ; + CHAR EXTRACT 8
32 00042*125300 MOVS 1,1 ; ELSE
33 00043*163400 AND 3,0 ; VALUE:=VALUE EXTRACT 8
34 00044*123000 ADD 1,0 ; + CHAR SHIFT 8;
35 00045*043024 STA# 0 Z0,2 ;
36 00046*021026 LDA 0 Z2,2 ; RESTORE(AC0);
37 00047*003027 JMP# Z3,2 ; RETURN;
38
39 ; PROCEDURE OUTNL(ZONE);
40 ; PERFORMS OUTEND(ZONE,NL).
41 ; CALL: RETURN:
42 ; AC0 DESTROYED
43 ; AC1 DESTROYED
44 ; AC2 ZONE ZONE
45 ; AC3 LINK DESTROYED
46
47 A590: ; OUTNL:
48 00050*024126 LDA 1 .NL ; CHAR:= NL;
49
50 ; PROCEDURE OUTEND(ZONE,CHAR);
51 ; CHAR MUST BE IN THE RANGE 0 TO 255.
52 ; CALL: RETURN:
53 ; AC0 DESTROYED
54 ; AC1 CHAR DESTROYED
55 ; AC2 ZONE ZONE
56 ; AC3 LINK DESTROYED
57
58 00051*055026 A59: STA 3 Z2,2 ; OUTEND:
59 00052*004755 JSR A581 ; OUTCHAR(ZONE,CHAR);
60 00053*021005 LDA 0 ZKIND,2 ;

```

0005 MUC03

01 00054\*101202

MOVR 0,0 SZC ;

IF KIND.ZONE(15) THEN

02 00055\*006206

OUTBLOCK ;

OUTBLOCK(ZONE);

03 00056\*003026

JMP# Z2,2 ;

RETURN;

04

```

01 ; PROCEDURE OUTTEXT(ZONE,ADDR);
02 ; CALL: RETURN:
03 ; AC0 ADDR DESTROYED
04 ; AC1 DESTROYED
05 ; AC2 ZONE ZONE
06 ; AC3 LINK DESTROYED
07
08 00057*055031 A60: STA 3 Z5,2 ; OUTTEXT:
09 00060*115220 A601: MOVZR 0,3 ; VALUE:= 0.(ADDR//2);
10 00061*025400 LDA 1 +0,3 ; IF ADDR(15:15)=0 THEN
11 00062*101403 INC 0,0 SNC ; VALUE:= VALUE SHIFT -8;
12 00063*125300 MOVS 1,1 ; INCR(ADDR);
13 00064*034143 LDA 3 .255 ; CHAR:= VALUE(8:15);
14 00065*167405 AND 3,1 SNR ; IF CHAR=0 THEN
15 00066*003031 JMP# Z5,2 ; RETURN;
16 00067*004737 JSR A58 ; OUTCHAR(ZONE,CHAR);
17 00070*000770 JMP A601 ; GOTO OUTTEXT;
18
19
20 ; PROCEDURE OUTOCTAL(ZONE,VALUE);
21 ; CALL: RETURN:
22 ; AC0 VALUE DESTROYED
23 ; AC1 DESTROYED
24 ; AC2 ZONE ZONE
25 ; AC3 LINK DESTROYED
26
27 00071*055031 A61: STA 3 Z5,2 ; OUTOCTAL:
28 00072*024132 LDA 1 .24 ; CHAR:=24;
29 00073*101140 MOVOL 0,0 ; VALUE:=VALUE SHIFT 1+1;
30 00074*000406 JMP A612 ; GOTO LAST BIT;
31 A611: ; NEXT CHAR:
32 00075*024123 LDA 1 .6 ; CHAR:= 6;
33 00076*125100 MOVL 1,1 ; CHAR:=CHAR SHIFT 1+CARRY;
34 00077*101120 MOVZL 0,0 ; VALUE:=VALUE SHIFT 1;
35 00100*125100 MOVL 1,1 ; CHAR*= CHAR SHIFT 1+CARRY;
36 00101*101120 MOVZL 0,0 ; VALUE:=VALUE SHIFT 1;
37 A612: ; LAST BIT:
38 00102*125100 MOVL 1,1 ; CHAR:=CHAR SHIFT 1+CARRY;
39 00103*004723 JSR A58 ; OUTCHAR(ZONE,CHAR);
40 00104*101124 MOVZL 0,0 SZR ; IF VALUE:=VALUE SHIFT 1<>0 THEN
41 00105*000770 JMP A611 ; GOTO NEXT CHAR;
42 00106*003031 JMP# Z5,2 ; RETURN;
43

```

```

01 ; PROCEDURE INNAME(ZONE,NAMEADDR);
02 ; CALL: RETURN:
03 ; AC0 DESTROYED
04 ; AC1 NAMEADDR DESTROYED
05 ; AC2 ZONE ZONE
06 ; AC3 LINK DESTROYED
07 00107*055031 A82: STA 3 Z5,2 ; INNAME:
08 00110*135000 MOV 1,3 ;
09 00111*125120 MOVZL 1,1 ; NAMEADDR:=NAMEADDR*2;
10 00112*102400 SUB 0,0 ;
11 00113*041400 STA 0 0,3 ;
12 00114*041401 STA 0 1,3 ; NAME(0:3):=0;
13 00115*041402 STA 0 2,3 ;
14 00116*045030 STA 1 Z4,2 ;
15 00117*024123 LDA 1 .6 ;
16 00120*045027 STA 1 Z3,2 ; COUNT:=6;
17 00121*006207 A820: INCHAR ; REPEAT
18 00122*020113 LDA 0 .32 ; INCHAR(ZONE,CHAR);
19 00123*034142 LDA 3 .127 ;
20 00124*167400 AND 3,1 ; CHAR:=CHAR EXTRACT 7;
21 00125*122415 SUB# 1,0 SNR ; UNTIL CHAR<>32;
22 00126*000773 JMP A820 ;
23 00127*000414 JMP A822 ;
24 00130*035030 A821: LDA 3 Z4,2 ; PACKNEXT:
25 00131*175223 MOVZR 3,3 SNC ; IF NAMEADDR(15)=0 THEN
26 00132*125300 MOVS 1,1 ; SWOP(CHAR);
27 00133*021400 LDA 0 +0,3 ; WORD(INDEX):=
28 00134*107000 ADD 0,1 ; WORD(INDEX)+CHAR;
29 00135*045400 STA 1 +0,3 ;
30 00136*011030 ISZ Z4,2 ; INCR(NAMEADDR);
31 00137*006207 A823: INCHAR ; INNEXT:
32 00140*020113 LDA 0 .32 ; INCHAR(ZONE,CHAR);
33 00141*034142 LDA 3 .127 ;
34 00142*167400 AND 3,1 ; CHAR:=CHAR EXTRACT 7;
35 00143*106112 A822: ADCL# 0,1 SZC ; IF CHAR<=32 THEN
36 00144*003031 JMP# Z5,2 ; RETURN;
37 00145*015027 DSZ Z3,2 ; IF DECR(COUNT)<>0 THEN
38 00146*000762 JMP A821 ; GOTO PACKNEXT;
39 00147*011027 ISZ Z3,2 ; INCR(COUNT);
40 00150*000767 JMP A823 ; GOTO INNEXT;
41
42 ; ***** END OF CHARACTER I/O PROCEDURES *****
43
44 .END
0000 SOURCE LINES IN ERROR

```

## 0008 MUC03

A545	000000*	3/04	3/24	4/23	
A56	000013*	2/09	3/23		
A560	000014*	3/24	3/26		
A58	000026*	2/11	4/20	6/16	6/39
A580	000025*	2/13	4/17		
A581	000027*	4/21	4/59		
A583	000031*	4/23	4/25		
A59	000051*	2/17	4/58		
A590	000050*	2/15	4/47		
A60	000057*	2/19	6/08		
A601	000060*	6/09	6/17		
A61	000071*	2/21	6/27		
A611	000075*	6/31	6/41		
A612	000102*	6/30	6/37		
A82	000107*	2/23	7/07		
A820	000121*	7/17	7/22		
A821	000130*	7/24	7/38		
A822	000143*	7/23	7/35		
A823	000137*	7/31	7/40		



4/25

A59	000051*	2/17	4/58
A590	000050*	2/15	4/47
A60	000057*	2/19	6/08
A601	000060*	6/09	6/17
A61	000071*	2/21	6/27
A611	000075*	6/31	6/41
A612	000102*	6/30	6/37
A82	000107*	2/23	7/07
A820	000121*	7/1	