


```

GGGGGGGG  SSSSSSSS  LL      UU      UU
GGGGGGGG  SSSSSSSS  LL      UU      UU
GG        SS        LL      UU      UU
GG        SS        LL      UU      UU
GG        SS        LL      UU      UU
GG        SS        LL      UU      UU
GG        SSSSSS    LL      UU      UU
GG        SSSSSS    LL      UU      UU
GG        SS        LL      UU      UU
GG        SS        LL      UU      UU
GG        SS        LL      UU      UU
GGGGGG    SSSSSSSS  LLLLLLLLLL  UUUUUUUUUU  ....
GGGGGG    SSSSSSSS  LLLLLLLLLL  UUUUUUUUUU  ....

```

```

LL        IIIIII  SSSSSSSS
LL        IIIIII  SSSSSSSS
LL        II      SS
LL        II      SS
LL        II      SS
LL        II      SS
LL        II      SSSSSS
LL        II      SSSSSS
LL        II      SS
LL        II      SS
LL        II      SS
LLLLLLLLLL  IIIIII  SSSSSSSS
LLLLLLLLLL  IIIIII  SSSSSSSS

```

.....

1
2
3
4
5
6
7
8
9
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

```

P 0001 0 MODULE gslu ( IDENT = 'V04-000'
0002 0 %BLISS32C, ADDRESSING_MODE (EXTERNAL = LONG_RELATIVE,
0003 0 NONEXTERNAL = LONG_RELATIVE)
0004 0 ) =
0005 1 BEGIN
0006 1
0007 1 *****
0008 1 *
0009 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *
0010 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *
0011 1 * ALL RIGHTS RESERVED. *
0012 1 *
0013 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
0014 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *
0015 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *
0016 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
0017 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *
0018 1 * TRANSFERRED. *
0019 1 *
0020 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *
0021 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *
0022 1 * CORPORATION. *
0023 1 *
0024 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *
0025 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *
0026 1 *
0027 1 *
0028 1 *****
0029 1
0030 1 ++
0031 1 FACILITY: DSR (Digital Standard RUNOFF) / DSRPLUS
0032 1
0033 1 ABSTRACT:
0034 1
0035 1 Collect an unbroken sequence of letters, converting
0036 1 all letters to uppercase.
0037 1
0038 1 ENVIRONMENT: Transportable
0039 1
0040 1 AUTHOR: R.W.Friday CREATION DATE: April, 1978
0041 1
0042 1

```


Module Level Declarations

```

: 55      0053 1 %SBTTL 'Module Level Declarations'
: 56      0054 1
: 57      0055 1
: 58      0056 1 : TABLE OF CONTENTS:
: 59      0057 1
: 60      0058 1 : INCLUDE FILES:
: 61      0059 1
: 62      0060 1 LIBRARY 'NXPORT:XPORT';      ! XPORT Library
: 63      0061 1 REQUIRE 'REQ:RNODEF';      ! RUNOFF variant definitions
: 64      0192 1
: 65      U 0193 1 %IF DSRPLUS %THEN
: 66      U 0194 1 LIBRARY 'REQ:DPLLIB';      ! DSRPLUS BLISS Library
: 67      0195 1 %ELSE
: 68      0196 1 LIBRARY 'REQ:DSRLIB';      ! DSR BLISS Library
: 69      0197 1 %FI
: 70      0198 1
: 71      0199 1
: 72      0200 1 : EXTERNAL REFERENCES:
: 73      0201 1
: 74      0202 1 EXTERNAL
: 75      0203 1     KHAR;
: 76      0204 1

```

Module Level Declarations

```

78 0205 1 GLOBAL ROUTINE GSLU (INPUT_STRING, OUTPUT_STRING) =
79 0206 1
80 0207 1
81 0208 1 ++
82 0209 1 FUNCTIONAL DESCRIPTION:
83 0210 1     GSLU processes an unbroken sequence of letters,
84 0211 1     stopping when 1) a non-letter is found, or
85 0212 1     2) the end of IRA is reached, or
86 0213 1     3) OUTPUT_STRING is about to overflow.
87 0214 1     All letters encountered are transferred to OUTPUT_STRING
88 0215 1     after being first converted to upper case.
89 0216 1
90 0217 1 FORMAL PARAMETERS:
91 0218 1
92 0219 1     INPUT_STRING is the string to be scanned.
93 0220 1     The collected string of letters is left in OUTPUT_STRING.
94 0221 1
95 0222 1 IMPLICIT INPUTS:      None
96 0223 1
97 0224 1 IMPLICIT OUTPUTS:    None
98 0225 1
99 0226 1 ROUTINE VALUE:
100 0227 1 COMPLETION CODES:
101 0228 1
102 0229 1     See GSLUCC.REQ for completion codes returned.
103 0230 1
104 0231 1 SIDE EFFECTS:        None
105 0232 1 --
106 0233 1
107 0234 2 BEGIN
108 0235 2 MAP
109 0236 2     OUTPUT_STRING : REF FIXED_STRING;
110 0237 2 BIND
111 0238 2     IRA = INPUT_STRING : REF FIXED_STRING;
112 0239 2
113 0240 2 WHILE 1 DO
114 0241 3 BEGIN
115 0242 3
116 0243 3     !Stop on a nonletter.
117 0244 4 IF NOT LETTER (.KHAR)
118 0245 3 THEN
119 0246 4     RETURN (IF .FS_LENGTH (OUTPUT_STRING) EQL 0
120 0247 3     THEN GSLU_NONE ELSE GSLU_NORMAL);
121 0248 3
122 0249 3 IF .FS_LENGTH (OUTPUT_STRING) EQL .FS_MAXSIZE (OUTPUT_STRING)
123 0250 3 THEN
124 0251 3     (WHILE LETTER (.KHAR) DO (KCNS ()); RETURN GSLU_TOO_LONG); !Throw away excess letters.
125 0252 3
126 0253 3 FS_WCHAR (OUTPUT_STRING, !Convert to upper case and output the letter.
127 0254 3     (IF UPPER_LETTER (.KHAR)
128 0255 3     THEN
129 0256 3     .KHAR ELSE UPPER_CASE (.KHAR)));
130 0257 4 KCNS () !Get next character.
131 0258 3 END !end of processing loop.
132 0259 3
133 0260 1 END; !End of GSLU

```

					.TITLE	GSLU			
					.IDENT	\V04-000\			
					.EXTRN	KHAR, RINTES			
					.PSECT	\$CODE\$,NOWRT,2			
					.ENTRY	GSLU, Save R2,R3,R4,R5,R6,R7,R8		0205	
	58	00000000G	EF	9E	00002	MOVAB	KHAR, R8	0249	
	54	08	AC	D0	00009	MOVL	OUTPUT_STRING, R4	0256	
	57	08	AC	D0	0000D	MOVL	OUTPUT_STRING, R7	0257	
	52	08	AC	D0	00011	MOVL	OUTPUT_STRING, R2		
	55	04	AC	D0	00015	MOVL	IRA, R5		
	56	0C	A5	9E	00019	MOVAB	12(R5), R6		
	50		68	D0	0001D	1\$:	MOVL	KHAR, R0	0244
00000041	8F		50	D1	00020		C MPL	R0, #65	
			09	19	00027		BLSS	2\$	
0000005A	8F		50	D1	00029		C MPL	R0, #90	
			23	15	00030		BLEQ	5\$	
00000061	8F		50	D1	00032	2\$:	C MPL	R0, #97	
			09	19	00039		BLSS	3\$	
0000007A	8F		50	D1	0003B		C MPL	R0, #122	
			11	15	00042		BLEQ	5\$	
	50	08	AC	D0	00044	3\$:	MOVL	OUTPUT_STRING, R0	0246
		0C	A0	D5	00048		T STL	12(R0)	
			04	12	0004B		B NEQ	4\$	
	50		02	D0	0004D		MOVL	#2, R0	
				04	00050		RET		
	50		01	D0	00051	4\$:	MOVL	#1, R0	
				04	00054		RET		
08	A7	0C	A4	D1	00055	5\$:	C MPL	12(R4), 8(R7)	0249
			4B	12	0005A		B NEQ	11\$	
	50	04	AC	D0	0005C		MOVL	IRA, R0	0251
	53	0C	A0	9E	00060		MOVAB	12(R0), R3	
	51		68	D0	00064	6\$:	MOVL	KHAR, R1	
00000041	8F		51	D1	00067		C MPL	R1, #65	
			09	19	0006E		BLSS	7\$	
0000005A	8F		51	D1	00070		C MPL	R1, #90	
			12	15	00077		BLEQ	8\$	
00000061	8F		51	D1	00079	7\$:	C MPL	R1, #97	
			21	19	00080		BLSS	10\$	
0000007A	8F		51	D1	00082		C MPL	R1, #122	
			18	14	00089		B GTR	10\$	
			63	D5	0008B	8\$:	T STL	(R3)	
			09	14	0008D		B GTR	9\$	
	68	00G	8F	9A	0008F		MOVZBL	#RINTES, KHAR	
	63		01	CE	00093		M NEGL	#1, (R3)	
			CC	11	00096		BRB	6\$	
	68	04	B0	9A	00098	9\$:	MOVZBL	24(R0), KHAR	
		04	A0	D6	0009C		INCL	4(R0)	
			63	D7	0009F		DECL	(R3)	
			C1	11	000A1		BRB	6\$	
	50		03	D0	000A3	10\$:	MOVL	#3, R0	
				04	000A6		RET		
	50		68	D0	000A7	11\$:	MOVL	KHAR, R0	0256
00000041	8F		50	D1	000AA		C MPL	R0, #65	

```

0000005A 8F          09 19 000B1    BLSS 12$
          50 D1 000B3    CMPL R0, #90
          03 15 000BA    BLEQ 13$
          04 50        20 C2 000BC 12$:  SUBL2 #32, R0
          B2        50 90 000BF 13$:  MOVB  R0, @4(R2)
          04 A2 D6 000C3    INCL 4(R2)
          0C A4 D6 000C6    INCL 12(R4)
          66 D5 000C9    TSTL (R6)
          09 14 000CB    BGTR 14$
          68 00G 8F 9A 000CD  MOVZBL #RINTES, KHAR
          66 01 CE 000D1    MNEGL #1, (R6)
          68 04 B5 9A 000D6 14$:  BRB 15$
          04 A5 D6 000DA    MOVZBL @4(R5), KHAR
          66 D7 000DD    INCL 4(R5)
          FF3B 31 000DF 15$:  DECL (R6)
          BRW 1$
    
```

0257

; Routine Size: 226 bytes, Routine Base: \$CODE\$ + 0000

```

: 134      0261 1
: 135      0262 1 END
: 136      0263 0 ELUDOM
    
```

!End of module

PSECT SUMMARY

Name	Bytes	Attributes
\$CODE\$	226	NOVEC, NOWRT, RD, EXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)

Library Statistics

File	Total	Symbols Loaded	Percent	Pages Mapped	Processing Time
\$255\$DUA28:[SYSLIB]XPORT.L32;1	590	0	0	252	00:00.1
\$255\$DUA28:[RUNOFF.SRC]DSRLIB.L32;1	1248	17	1	86	00:00.3

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD, INITIAL, OPTIMIZE)/LIS=LIS\$:GSLU/OBJ=OBJ\$:GSLU MSRC\$:GSLU/UPDATE=(ENH\$:GSLU)

; Size: 226 code + 0 data bytes

GSLU
V04-000

Module Level Declarations

E 9
16-Sep-1984 00:43:09

VAX-11 Bliss-32 V4.0-742

Page 7

HE
VO

: Run Time: 00:04.7
: Elapsed Time: 00:14.0
: Lines/CPU Min: 3386
: Lexemes/CPU-Min: 15733
: Memory Used: 70 pages
: Compilation Complete

