

music compiler•27/1/67

top=7777

dimension rdb(140)

define error x

jsp er
flexo x

terminate

define complain x

jsp er1
flexo x

terminate

o/
beg,

jmp cln
lem
clcVlia
tyo
cli
law flexo af
arq
jmp nots
tyi
law 47
arq
jmp busy
law 51
arq i
jmp rt-2
jsp txx
text /busy/
jmp beg
cliVclavhltVlaiVclf 7
jmp rt-2

busy,

nots,

stp,

clcVlia
tyo
cli
law flexo af
arq
jmp nots
tyi
lai
sad (77
jmp rt-2
law i 51
arq
law i 20
jda fee
lac (600000
jda ppp
law i 520
jda fee
law i 47
arq
dsm
clcVhlt

cln,

pnt,	add . dac tne	/corrected for accidentals, key sig. a, /0=4th g below middle c /84=4th g above middle c /correct for up, down
	add t11 sub (10. spa jmp uat sub (63. sma jmp uat add (64. dac tne	/1=3rd f below middle c "bass b5" /63=3rd g above middle c "treble 17" /calculate articulation
tmt,	lac si spa jmp tms sza i jmp tmt sub (1 jmp .+2 lac ss dac sv lac et spa jmp tme sza jmp emb lac sv ral 6s jmp .+2	/get implicit articulation /1=0, s=1, e=2, h=3, q=4 /check for embellishments
s70,	cla ior tne ral 9s ior nft jda nc lac nl dac ltn	/last note
s72,	lac nft adm mm lac si dac sid jmp out	/no. of 192nds used in measure /explicit articulation of last note
cpp,	lac tne sza i jmp rr lac ton jmp s2v	/copy pitch from previous note /was a rest
rr,	dzm tne lac sh ior na ior fl ior si ior et sza i jmp s70 error air jmp s70	

eor, error eor
jmp s70
eit, error eit
jmp s70
tme, error tme
jmp s70
tms, error tms
jmp tmt
uat, error uat
dzm tne
jmp s70
tmf, error tmf
del, cla
dap mts
jmp out
tff, error tff
jmp del
ert, error ert
jmp del
unc, error unc
jmp del
dtu, error dtu
jmp del
tic, error tic
jmp del

up, jsp read
sza i
jmp up
lac nf1
up1, dac tll
jmp out

down, jsp read
sza i
jmp down
lac nf1
cma
jmp up1

bass, law 13.
jda clef

tenor, law 17.
jda clef

alto, law 19.
jda clef

treble, law 25.
jda clef

clef, 0
jmp out
units, jsp read
sza i
jmp units
lac nf1
ral 1s

/drop a note, suppress mts error

```
add nf1
ral 1s
dac 3u
clc
dac irl
jmp out
```

```
l,   cla
s,   jda ss
     law 1
e,   jda ss
     law 2
h,   jda ss
     law 3
q,   jda ss
     law 4
     jda ss
ss,  0
     jmp out
```

emb,

law i 49.
add ton
sma
jmp eor
add (49.
spq
jmp eor
add (mt-1
dap pmt
add (nt-mt
dap pnt
add (kt-nt
dap pkt
lac i pmt
add i pnt
add i pkt
add tll
add lwr
sub (10.
spa
jmp eor
sub (63.
sma
jmp eor
add (64.
dac tnd
law 2
adm pmt
law 2
adm pnt
law 2
adm pkt
lac i pmt
add i pnt
add i pkt
add tll
add upr
sub (10.
spa
jmp eor
sub (63.
sma
jmp eor
add (64.
dac tnf

lac i et
mul trl
scr 1s
lai
sub nft
szm
jmp eit
idx et
jmp i et

/next lower note

/next higher note

/minimum time required

/too short

/dispatch

m2,	2 jsp tr1 lio tne sma lio tnf	/m - trill
trn,	lai ral 9s ior t1 jda nc lac t1 cma adm t2 law i 1 adm t3 sza i jmp s72	/finish trill
et1,	lac t2 sal 1s add t3 and (-1 cliVswp div t3 hlt dac t1 lac t3 rar 1s jmp .	/done /calculate time for next note /number of notes remaining
tr3,	jmp .	
tr1,	dap tr3 lac nft dac t2 add tr1 and (-1 cliVswp div tr1 hlt sal 1s dac t3 jmp et1	/set up trill /total number of cycles
n2,	4 jsp tr1 lio tne sma lio tnf sad (1 lio tnd jmp trn	/n - trill with grace notes
k2,	2 law 3 dac t3 law m2+2 jmp f3	/o - mordent

w2,	2	/w - inverted trill
	jsp tr1	
	lio tne	
	spa	
	lio tnf	
	jmp trn	
f2,	4	/f - full turn
	law 5	
	dac t3	/split time 5 ways
	law n2+2	
f3,	lio nft	
	dio t2	
	dap tr3	
	jmp et1	
d2,	3	/d - inverted mordent
	lac tne	
	ral 9s	
	ior tr1	
	jda nc	
	lac tnd	
	ral 9s	
	ior tr1	
	jda nc	
	lac tr1	
	ral 1s	
etx,	cma	/use up remaining time
	add nft	
	ral 3s	ior sv
	ral 6s	
	ior tne	
	ral 9s	
	jda nc	
	jmp s72	
u2,	4	/u - turn
	lac tnf	
	ral 9s	
	ior tr1	
	jda nc	
	lac tne	
	ral 9s	
	ior tr1	
	jda nc	
	lac tnd	
	ral 9s	
	ior tr1	
	jda nc	
	lac tr1	
	ral 1s	
	add tr1	
	jmp etx	

p2,

```
3
lac tnf
rar 8s
ior trl
sar 1s
jda nc
lac tne
rar 8s
ior trl
sar 1s
jda nc
lac tnf
rar 8s
ior trl
sar 1s
jda nc
lac trl
sar 1s
dac t3
sal 1s
add t3
jmp etx
```

/p - mordent

o2,

```
3
lac nft
dac t2
and (-1
cliVswp
div trl
hlt
sal 1s
add (1
dac t3
law m2+2
dap tr3
jmp etl
```

/o - odd numbered trill

```
trill,    jsp read
          sza i
          jmp trill
          law 177
          and nf2
          sza
          sas nf2
          jmp ertt
          add (rcr-1
          dac qqt
          and nf2
          sza
          jmp ertt
          lac fu
          cli
qqt,      0
          szavsni i
          jmp dtut
          dac t1
          ral 1s
          lio c1
          sni
          add t1
          dac tr1
          jmp out
dtut,    error dtu
          jmp out
ertt,    error ert
          jmp out
```

```

key,      law kt
          dap .+3
          law mt
          dap .+2
          dzm .
          dzm .
          idx .-2
          idx .-2
          sas .+2
          jmp .-5
          dzm mt+49.
          dzm kt+49.
ke2,      jsp read
          sza i
          dzm nf1
          add sh
          add na
          add f1
          sza i
          jmp ke2
          lac na
          sza
          jmp out
          clf 2
          lac f1
          sza i
          stf 2
          sza i
          lac sh
          sub nf1
          spa
          cla
          add nf1
          dac nf2
          law i 1
          szf 2 i
          law i 5
          dac nf
ke3,      lac nf2
          sza i
          jmp out
          lac nf
          dac cn
ke4,      law 7
          adm cn
          sub (49.
          sma
          jmp ke5
          add (kt+49.
          dap .+4
          law i 1
          szf 2
          law 1
          adm .
          jmp ke4
ke5,      law 3
          szf 2
          law 4
          add nf

```

/sharp

sma
sub (7
dac nf
law i 1
adm nf2
jmp ke3

nc,	0	/deposit note
	dap cnx	
	idx nl	
	sad bc	
	jmp tof	
	lac nc	
	dac i nl	
cnx,	jmp .	
sbc,	0	/deposit bar
	dap sbx	
	law i 1	
	adm bc	
	sad nl	
	jmp tof	
	lac sbc	
	dac i bc	
	idx b11	
	idx b12	
sbx,	jmp .	
fee,	0	/feed
	dap fex	
	szs 10	
	jmp fex	
	cli	
	ppa	
	isp fee	
	jmp fee+2	
fex,	jmp .	
ppp,	0	/punch binary word
	dap ppx	
	szs 10	
	jmp ppx	
	lio ppp	
	ppb	
	ril 6s	
	ppb	
	ril 6s	
	ppb	
ppx,	jmp .	

```

in,      dap inx                /read character
in0,     cks
          ril 1s
          spi i
          jmp .-3
          rrb
          iot 1
          sni
          jmp in0
          rir 7s
          spi
          jmp in0
          ril 7s
          lai
          ior (rar
          dac .+2
          law 2525
          0
          sma
          jmp ilp
          law 77
          and .-4
inx,     jmp .
ilp,     jsp txx
          text /ilp
          jmp in0
er,      clf 3                /error - print syllable in r
          jmp .+2
er1,     stf 3                /error
          dap erx
          szf 6
          jmp er3            /entire line already in
          lac poi            /read in rest of line
          dap f12
          lac ch
erk,     sad (77
          jmp er3
          idx f12
          sad (dac rdb+140
          jmp bof
          jsp in
          dac i f12
          jmp erk
er3,     skp .
          jmp erq
          jsp red
          jsp txx
          text /

```

To err is human---to forgive, divine.

	cla	
	dap er3	
erq,	jsp blk	
	szf 5	
	jmp erm	/syllable already printed
	szf 3	
	szf 6 i	
	jmp .+2	
	jmp erm	/call to er1 and line has been printed
	law rdb	
	dap f12	
er1,	szf 3	
	jmp ern	
	lac f12	
	sad f11	
	jsp red	
	lac f12	
	sad poi	
	jsp blk	
ern,	lio i f12	
	tyo	
	idx f12	
	lai	
	sas (77	
	jmp er1	
erm,	stf 5	
	stf 6	
	lio (36	
	tyo	
	szf 3	
	jsp red	
	szf 3 i	
	jsp blk	
erx,	lac .	
	lio .+3	
	rcl 6s	
	tyo	
	sas .	
	jmp .-3	
	clcvlia	
	tyo	
	idx erx	
	jsp blk	
	jmp i erx	

bof, jsp txx
 text /
Source line has too many characters.

law rdb
dap poi
lio i poi
tyo
idx poi
sas (dac rdb+100
jmp .-3
clcvlia
tyo
jmp beg

tof, jsp txx
 text /
Table overflow. Subdivide source text.

jmp beg

red, dap rx
 law 35
 jmp .+3

blk, dap rx
 law 34
 sad clr

rx, jmp .
 dac clr
 lia

clr, jmp rx
 34

txx, dap txy
 lio i txy
 idx txy
 lac (607600
 rcl 6s
 sad (lai

txy, jmp .
 sad .+2
 jmp txx+1
 swp
 tyo
 lia
 jmp txy-3

read,	dap rex	/read syllable and terminat
	law 77	
	sas ch	
	jmp re1	
	law rdb-1	/if new line, reset buffer
	dap poi	
	clf 6	/on if line already in buffer
re1,	law 1	
	add poi	
	dap fl1	/points to first character of syllable
	law 40	/current value of dot
	dac fc	
	law 100	/dots accumulated in fu
	dac fu	
	dzm tf1	
	dzm tf2	
	law tf1	
	dap tp	
	dzm cn	/counts non-numeric characters
	law nf1	
	dap np	
	dzm a1	/clear all counters
	dzm b1	
	dzm c1	
	dzm g1	
	dzm r1	
	dzm cm	
	dzm fl	
	dzm na	
	dzm sh	
	dzm si	
	dzm upr	
	dzm lwr	
	dzm et	
	clf 4	/on if reading comment
	clf 5	/on if number partially entered
rlp,	idx poi	
	sad (dac rdb+140	
	jmp bof	
	szf 6	
	jmp in1	/character already in
	jsp in	
poi,	dac .	
	sad (72	
	clf 4	
	szf 4	
	jmp rlp	/comment
	dac ch	
	add (dtb	
	dap dtp	

dtp,	lio . spi i jmp tp-1 xct i dtp jmp ema lio i dtp rir 2s law upr spi law lwr dap emu law 1 and i dtp sza i law i 1	/articulation or embellishment /i,j,y, or z
emu,	adm . jmp tp	
ema,	spa jmp em lio si dac si sni jmp tp clc dac si jmp tp	/too many
em,	lio et dap et sni jmp tp clc dac et jmp tp	/too many
tp,	idx i dtp lac . lio ch rcr 6s dac i tp idx cn sad (3 idx tp szf 5 idx np jmp rlp-1	/count /pack
in1,	lac i poi jmp poi+1	

dtb,	jmp sep	/space
	repeat 11, jmp n	
	jmp rlp	
	jmp rlp	/stop code
	jmp rlp	
	jmp rlp	
	jmp rlp	
	jmp n	/o
	jmp sep	/slash
	law 2	/s - stacatto
	jmp tp	/t
	law i 7777-u2	/u - turn
	jmp tp	
	law i 7777-w2	/w - inverted trill
	jmp xe	/x
	sni 3	/y - raise lower note
	sni 2	/z - depress lower note
	jmp rlp	
cm,	0	/comma
	jmp rlp	
	jmp rlp	
	jmp sep	/tab
	jmp rlp	
	jmp rlp	
	sni 0	/j - depress upper note
	law i 7777-k2	/k - mordent
	law 1	/l - legato
	law i 7777-m2	/m - trill
	law i 7777-n2	/n - trill with grace notes
	law i 7777-o2	/o - odd numbered trill
	law i 7777-p2	/p - shake
	law 5	/q - quarter
r1,	0	/r - rest
	jmp rlp	
	jmp rlp	
f1,	0	/- - flat
na,	0	/) - natural
	jmp rlp	
sh,	0	/(- - sharp
	jmp rlp	
a1,	0	/a - above
b1,	0	/b - below
c1,	0	/c - triplet
	law i 7777-d2	/d - inverted mordent
	law 3	/e - normal articulation
	law i 7777-f2	/f - full turn
g1,	0	/g - grace note
	law 4	/h - half
	sni 1	/i - depress upper note
	jmp rlp	
	jmp dot	/dot
	jmp upc	/upper case
	jmp rlp	
	jmp rlp	
	jmp sep	/car. ret.
dtb+100,		

tf1,	0	/text of pseudo-instruction
tf2,	0	
nf1,	0	/first numeric field
nf2,	0	
n,	lac .	/digit
	sad np	
	jmp np1	
	szf 5 i	
	dzm i np	
np,	lac .	
	ral 2s	
	add i np	
	ral 1s	
	lio ch	
	rir 5s	
	spi i	
	add ch	
	dac i np	
	dac nf2	
np1,	stf 5	
	jmp rlp	
upc,	stf 4	/upper case
	jmp rlp	
sep,	szf 5	/space, tab, c.r., or slash
	idx np	
	lac np	
	sub (lac nf1	
	dac nf	/number of numeric fields
	clf 5	/on if syllable is printed
rex,	jmp .	
dot,	lac fc	/dot
	sma	
	adm fu	
	spa	
	dzm fu	/dot error
xe,	lac fc	/x
	sma	
	rar 1s	
	dac fc	
	jmp tp	
f11,	dac .	/points to beginning of syllable
f12,	dac .	/pointer used by err
constants		

nt, /main pitch table
decimal

0 2
5 7
12 14
17 19
24 26
29 31
36 38
41 43
48 50
53 55
60 62

4
9 10
16
21 22
28
33 34
40
45 46
52
57 58
6465 67

0 7 69

72 74
77 79
84

76
81 82

octal

kt, .+50./
mt, .+50./
variables
not,
start beg

/key signature table
/status of current measure

	jmp ng+1	
	cma	
	adm nft	/rob it
	sub (6	
	sma	
	jmp ng	
	lac nft	/only 1/32nd left
	sub rob	
	sma	
	jmp ng	/ok anyway
	error itg	
	lac nfp	/unrob it
	dac nft	
ng,	dzm rob	
	lac r1	
	sza	
	jmp rr	/rest
	lac nf	
	sza i	
	jmp cpp	/copy pitch
	lac a1	
	sub b1	
	dac a1	
	ral 1s	
	add a1	
	ral 2s	
	add nf1	/pitch corrected for a,b
	add clef	
	dac ton	/0=7th leger line below bass clef
		/49=space above 7th leger line above tre
	spa	
	jmp uat	
	sub (50.	
	sma	
s2v,	jmp uat	
	law mt	
	add ton	
	dap pmt	
	add (nt-mt	
	dap pnt	
	add (kt-nt	
	dap pkt	
	lac sh	
	add fl	
	add na	
	sza i	
	jmp pmt	/no accidentals given
	cla	
	sas sh	
	sad fl	
	jmp s2w	
	error nor	
s2w,	jmp pmt	
	lac sh	
	sub fl	
pkt,	sub .	
	dac i pmt	/save it in mt for rest of measure
	jmp pmt+1	
pmt,	lac .	
	add i pkt	

16,	<pre> sza jmp gr law 177 and nf2 sza sas nf2 jmp ert add (rcr-1 dac qq and nf2 sza jmp ert lac fu cli </pre>	<pre> /grace note /check duration </pre>
qq,	<pre> 0 sza vsni i jmp dtu dac tim ral 1s lio c1 sni add tim dac nfp dac nft dio ccc lac g1 sza i jmp nr lac nft adm rob jmp ng+1 </pre>	<pre> /dot underflow /duration in 64ths with dots /duration in 192nds with dots and triple /save triplet status </pre>
gn,	<pre> lac c1 ior ccc lia lac tim spa jmp unc ral 1s sni add tim dac nfp dac nft law 100 sas fu jmp tic lac g1 sza jmp gr jmp ng+1 </pre>	<pre> /grace note </pre>
15,	<pre> lac c1 ior ccc lia lac tim spa jmp unc ral 1s sni add tim dac nfp dac nft law 100 sas fu jmp tic lac g1 sza jmp gr jmp ng+1 </pre>	<pre> /comma, get previous time /previous time in 64ths /undefined /no dots allowed /grace note with comma </pre>
gr,	<pre> law 1 sas nf jmp l6 lac (rcr 5s dac qq jmp qq-2 </pre>	<pre> /if two fields, use given duration /make it a 32nd </pre>
nr,	<pre> lac rob sza i </pre>	<pre> /not grace note or comma </pre>

	lac nf1	
	dac bl2	/permit alternate numbering
	law 600	
	jmp bb2	
ff1,	law 1	/a note
	sub cm	
	sma	
	jmp ff2	
	law 1	
	dac cm	
	error tmc	
ff2,	law 1	
	sub g1	
	sma	
	jmp ff3	
	error tmg	
	law 1	
	dac g1	
ff3,	law 1	
	sub r1	
	sma	
	jmp ff4	
	error tmr	
	law 1	
	dac r1	
ff4,	lac r1	
	and g1	
	sza	
	jmp tmf	
	law 2	
	sub cm	
	sub r1	
	sub nf	
	spa	
	jmp tmf	
	lac r1	
	sza i	
	jmp 17	
	lac sid	/rest
	szm	
	jmp 17	
	lac ltn	/no explicit articulation on last note
	spa	
	jmp 17	/last note was in this measure
	lio i ltn	
	cla	
	ril 3s	
	rcl 3s	
	dio i ltn	/make preceding note legato
17,	lio cm	
	sni i	
	jmp 15	
	law i 2	/no comma
	add r1	
	add g1	
	add nf	
	spa	
	jmp tff	/must have 2 of r,g,number
	lac g1	

```

out,    law 21
        sas ch
        jmp lup
        jsp dun
        jmp lu1

dun,    dap dux
        lio mm
        sni

dux,    jmp .
        lac rob
        sza i
        jmp du2

du2,    complain itg
        lac (600000
        jda nc
        lac lmb
        jda sbc
        lac nl1
        sza i
        jmp du1
        add (1
        dac nl3
        dzm nl1
        lac nl
        dac nl2
        lac bc
        dac bcs
        lac mm
        sub mms
        dac mms

du1,    law i not-1
        add nl
        dac lmb
        lac mm
        sub 3u
        sza i
        jmp dux
        spa
        jmp mts
        complain mtl

mts,    jmp dux
        skip .
        jmp dux
        complain mts

ff,     jmp dux
        lac cn
        sza
        jmp ff1
        lac nf1
        sad b11
        jmp bb1
        sad b12

bbs,    skip .
        jmp bb1

bb1,    dac b11
        cla

bb2,    dap bbs
        jmp out

bb1,    error bb1

```

/other characters present

/bar label

/current bar

/alternate bar

/skip if alternate bar permitted

```

tempo,      jmp lu1
            jsp read
            sza i
            jmp tempo
            lac nf1
            ior (700000)
            jda nc
            jmp out

pp,         cla
            jmp f+1
p,          law 1
            jmp f+1
mf,        law 2
            jmp f+1
f,         law 3
            ior (500000)
            jda nc
            jmp out

end,        jsp dun
            lac (600000)
            dac i nl      /make sure notes end with 60
            cla
            jda nc      /note/bar marker
            lac bc
            sad (top+1
            jmp stp     /no music
            law not
            dac t3      /punch pointer
            idx nl
            sub bc
            add (top-not+1      /total size of i
p41,       jda ppp
            law i 6
            jda fee
            law i 104
            dac t1      /block count
            dzm t2      /chacksum
p42,       lac i t3
            jda ppp
            lac i t3
            adm t2
            idx t3
            sad (top+1
            jmp p43     /done
            sad nl
            lac bc      /change from notes to bars
            dac t3
            isp t1
            jmp p42
            lac t2
            jmp p41     /punch checksum
p43,       lac t2
            jda ppp     /punch last checksum
            law i 20
            jda fee
            jmp stp

```

copy,

```
jda nc
lac (600000)
jda nc
jmp irh
jsp dun
jsp read
sza i
jmp .-2
cla
```

/forbid alternate labels

```
dap bbs
law i 1
add bl1
sub nf1
spa
jmp blc
add bc
dac cbh
lac nf1
dac t1
jsp read
sza i
jmp .-2
law i 2
add bc
add bl1
sub nf1
dac nf2
law 1
add nf1
sub t1
spq
```

```
jmp brc
law i not-1
add nl
sad lmb
jmp cl
```

/stray tempo or loudness com

```
law not
add i cbh
dap .+1
lac .
```

cl,

```
jda nc
sad (600000)
jmp .+3
idx .-4
jmp .-5
lac lmb
jmp cl+1
lac i cbh
```

blc,

```
jda sbc
law i 1
adm cbh
sas nf2
jmp cl
```

brc,

```
jmp lu0
error blc
jsp read
sza i
jmp .-2
jmp lu1
error brc
```

ttb,

```
text /s // jmp s
text /e // jmp e
text /l // jmp l
text /h // jmp h
text /q // jmp q
text /sert // jmp rest
text /yek // jmp key
text /pocy // jmp copy
text /inust // jmp units
text /pp // jmp pp
text /p // jmp p
text /fm // jmp mf
text /f // jmp f
text /metop // jmp tempo
text /dne // jmp end
text /irtll // jmp trill
text /sabs // jmp bass
text /ertelb // jmp treble
text /netro // jmp tenor
text /tlao // jmp alto
text /pu // jmp up
text /wodn // jmp down
text /epon // jmp open
text /olces // jmp close
```

ttn,
rest,

```
jsp dun
jsp read
sza i
jmp .-2
lac nf1
sza i
jmp lu1
cma
dac nf1
lac irl
sma
jmp irm /rest already exists
law i not-1
add nl
dac irl /location of rest
lac 3u
jda nc
lac (600000
jda nc
lac lmb
jmp irn+1
law i not-1
add nl
sas lmb
jmp irk
lac irl
jda sbc
isp nf1
jmp irn
jmp lu0
law not /stray tempo or loudness com
add irl
dap .+1
lac .
```

irh,

irm,

irn,

irk,

c13,	lac (600000
	jda nc
	lac lmb
c14,	jda sbc
	law not
	add lmb
	dac t1
	dzm nl2
	law i not-1
	add nl
	dac lmb
c15,	lac t1
	sad t2
	jmp c16
	lac i t1
	jda nc
	idx t1
c16,	jmp c15
	lac bcs
	sad t3
	jmp out
	law i 1
	adm bcs
	lac i bcs
	jda sbc
c11,	jmp c16
	lac nl3
	sas nl2
	jmp rtc
cls,	jmp c14
	lac nl1
	sza
	jmp rtc
	error nop
	dzm nl1
rtc,	jmp out
	error rtc
	jmp out
clt,	spa
	jmp .+4
	error rtl
	jmp c12
	xct mts
	jmp c12
	error rts
	jmp c12

open,

clc
dac ltn
lac nl1
ior nl2
sza
jmp tmo
law i not-1

add nl
sas lmb
jmp op1
lac nl
dac nl2
dac nl3
lac bc
dac bcs
dzm mms

/at beginning of measure

op1,

jmp out
lac nl
dac nl1
lac mm
dac mms
jmp out
error tmo
jmp out

tmo,

close,

clc
dac ltn
lac nl2
sza i
jmp cls
lac bc
dac t3
law 1
add nl
dac t2
lac mm
add mms
sza i
jmp cl1
sub 3u
sza

cl2,

jmp clt
lac nl3
sad nl2
jmp cl3
lac i nl3
jda nc
idx nl3
jmp cl2

sp,

ng1,

nps,

```
sma
jmp nps
law ttb
dap sp
lac .
sas tf1
jmp ng1
idx sp
lac i sp
sas tf2
jmp ng1+1
idx sp
jmp i sp
idx sp
idx sp
idx sp
sas (lac ttn
jmp sp
error nps
jmp out
```

```

law 600
dap er3
cla
dap bbs /on if alternate bar label p
clc
dac tim /previous time=undefined
law 1
dac bl1
dac bl2
law 77
dac ch /initial separator=c.r.
lac (top+1
dac bc
law not-1
dac nl
law 32.x6
dac 3u /units=32
clc
dac r1 /location of a rest
law 6
dac tr1 /trill time=32nd
law 25.
dac clef /clef=treble
law 2
dac ss /normal articulation=e
dzm n1 /repeat indicators
dzm n2
dzm t1 /up/down=0
law kt /key=)
dap .+1
dzm .
idx .-1
sas .+2
jmp .-3
dzm kt+49.
lu0, law i not-1 /set lmb to poin
add nl
lu1, dac lmb
dzm nm /time used in this measure
clc
dac ltn /last note in this measure
law 600
dap mts
law mt /clear accidentals
dap .+1
dzm .
idx .-1
sas .+2
jmp .-3
dzm mt+49.
lup, dzm rob /time robbed for grace notes
jsp read
add cm
sza
jmp ff /numeric fields or comma pre
lac cn
sza i
jmp out /nothing
sub (7 /pseudo-instruction

```

rt,

rrb
law rdb
dap poi
dap fl2
jsp in
sad (13
jmp .-2
dac ch
dac i poi
idx poi
sad (dac rdb+140
jmp rt1
law 21
sas ch
jmp rt+3
lio i fl2
tyo
idx fl2
sas poi
jmp rt1
law 21
sas ch
jmp rt
clevla
tyo
law i 6301
arq

iot 1

rt1,