

ddt 20 august 1966

6000/

xx=0

nsy=113

low=-nsy-1

tst=-2

est=-1

low/

char l+char ma	ac
char l+char mi	io
char l+char mm	msk
char li 10000	

flex and	020000
flex ior	040000
flex xor	060000
flex xct	100000
flex jfd	120000
flex cal	160000
flex jda	170000
flex lac	200000
flex lio	220000
flex dac	240000
flex dap	260000
flex dip	300000
flex dio	320000
flex dzm	340000
flex add	400000
flex sub	420000
flex idx	440000
flex isp	460000
flex sad	500000
flex sas	520000
flex mus	540000
flex dis	560000
flex jmp	600000
flex jsp	620000

flex skp	640000
flex szs	640000
flex szf	640000

flex sza	640100
flex spa	640200
flex sma	640400
flex szo	641000
flex spi	642000

flex ral	661000
flex ril	662000
flex rcl	663000
flex sal	665000
flex sil	666000
flex scl	667000
flex rar	671000
flex rir	672000
flex rcr	673000
flex sar	675000
flex sir	676000
flex scr	677000

flex law	700000
flex iot	720000
flex tyi	720004
flex rrb	720030
flex cks	720033
flex lsm	720054
flex esm	720055

flex cbs	720056
flex eem	724074
flex lem	720074
flex rpa	730001
flex rpb	730002
flex tyo	730003
flex ppa	730005
flex ppb	730006
flex dpy	730007

flex clf	760000
flex nop	760000
flex opr	760000

flex lap	760500
flex stf	760010
flex cla	760200

flex hlt	760400
flex cma	761000
flex clc	761200
flex lat	762200
flex cli	764000

est,

low

```
lap=cla 100
loh=iot 1
clo=651600
spq=650500
szm=640500
```

```
define
    senseswitch a
    szs 10xa
    term
```

```
define
    initialize a,b
    law b
    dap a
    term
```

```
define
    index a,b,c
    idx a
    sas b
    jmp c
    term
```

```
define
    listen
    cla+cli+clf 1-opr-opr
    szf 1 1
    jmp .-1
    tyi
    term
```

```
define
    swap
    swp
    term
```

```
define
    load a,b
    lio (b
    dio a
    term
```

```
define
    setup a,b
    law 1 b
    dac a
    term
```

```
define
    count a,b
    isp a
    jmp b
    term
```

define

move a,b
lio a
dio b
term

define

clear a,b
init . 2, a
dzm
index .-1, (dzm b 1, .-1
term

define

dispatch lc,uc
[1000xuc]+lc-[1001x1se]
terminate

lis, lio bki
bki, dio ch /or break addr
jsp sbc

lse, jsp lcc
lss, clc
dac chi

lsp, dzm wrd
lac cun

ssn, 500e dip sgn
dzm dnm
dzm syl

n2, dzm sym
clc
dac let

lsr, lio skl
dio wea
20 init bax, lwt
listen

ps1, dio ch
law dtb
30 add ch
dap .+1
lac .

cas, xx /rar 9s or cli
and (777

cad, add t1s
dap lsx
sub ar1 /last no-eval routine

40 spq
jmp i lsx
law syl
lio let
spi i
jsp ev1
jmp ev4
lac (flex U
jda tys
jmp lsp

evl, dap evx
evc, lac est
dap ev2

```

ev2,      lac .
          sad sym
          jmp ev3          /match found
          idx ev2
          index ev2, evc, ev2
          idx evx
ev3,      idx ev2
evx,      jmp .

ev4,      dap sgn
          lac wrd
sgn,      xx              /operator and syllable addr.
          dac wrd
          lio chi
          spi
          lac lwt
lsx,      jmp .

n,        rir 5s          /number routine
          lac syl
          ral 3s
          spi i
cun,      lor ch
          dac syl
          lac dnm
          ral 2s
          add dnm
          ral 1s
          spi i
          add ch
          dac dnm
          jmp l1

l,        dzm let        /letter routine
l1,      lac sym
          ral 6s
          add ch
          dac sym
          dzm chi
          jmp lsr

```

```

uc,      lio rc          /upper case
         jmp .+2
lc,      lio ps1        /lower case
         dio cas
30      jmp lsr

sqo,     lac dnm        /' means take decimal number
         jmp n1+1

quo,     lac sym        /" means take as flexo codes
         jmp n1

a,       law ac         /A means accumulator
         jmp n1

ir,      law io         /I means i-o
40      jmp n1

m,       law msk        /M means mask register
         jmp n1

q,       lac lwt        /Q means last quantity
         jmp n1

f,       law est        /F means lowest register
n1,      dzm chi
         dac syl
50      jmp n2

err,     lac (743521    /?
eri,     jda tys
         law 7234      /lc, blk
         jda tys
         jmp lsr

daq,     law 7777        /D defines sym as address of Q
         and lwt
60      jmp .+2

com,     lac loc        /comma defines sym as loc
         dac df1

```

```

def,      lac let          /define symbol
sk1,      sza
          jmp err
          law pn2

de,       dap dex
          lio df1
          jsp ev1
          jmp df2
          law i 1
          add est
          dap est
          dio i est
          sub one
          52ac dap est
          lio sym
          dio i est
          jmp dex

df2,      dio i ev2
dex,      jmp .
del,      jmp pn2          /end of no-eval routines, delete

val,      dac df1
          10 jmp lss       /open paren, sets up value for define

eql,      dac lwt          /print octal integer
          jsp lct
          jda opt

pn2,      jsp lct
          jmp lss

arw,      dac lwt          /print as instruction
          jsp lct
          jda pi

ar1,      jmp del

oct,      law odv          /octal-decimal switch setup
          jmp .+2

dec,      law ddv
          dap ops
          jmp lse

smb,      law pi
          20 jmp .+2

cns,      law opt          /symbolic-constant switch setup
          dap pns
          jmp lse

oad,      law pv1
          jmp .+2

rad,      law pev          /octal-relative switch setup
          dap pa1

tls,      jmp lse

```



```

pls,      lac cad
           jmp ssn

min,      lac csu
           jmp ssn

uni,      jmp ssn-1

isc,      lac can
           jmp ssn

dot,      lac loc
           jmp n1

tab,      spi i           /tab
tas,      dac ch

ta3,      dac lwt
           jsp lcc
           jda pad
           law 7221
           jda tys

ta5,      dzm loc
           dap loc

ta6,      dap tas
           jsp lct
           lac i tas
           dac lwt

bax,      jda .           /p1, opt or lwt
           jmp pn2

bs,       spi i           /backspace
bs1,      dac i tas       /used as dac 1
           idx loc
           jmp ta3

fs,       spi i           /arrow up (forward space)
           dac i tas
           law i 1
           add loc
           dap loc
           jmp ta3

bac,      law opt         /open bracket (bar-constant)
           jmp .+2

bas,      law p1          /closed bracket (bar-symbolic)
           dap bax

bar,      lac lwt
           spi
           jmp ta6
           lac wrd
           jmp ta5

uc8,      spi i           /> means make corr. and open register
           dac i tas
           jmp ta6

```

```
cr,      spi 1
         dac 1 tas
         dac lwt
         law 72
         jda tys
         init tas, ch
         jmp lss

bk,      spi
         init bki, ch          /break
         jmp lse

tr,      0
         dap prc
         dap prd
         idx prd
         lac tr
         dac ac
         isp ch
         jmp pr2
         jsp tr1

tr2,     dap pra
         law 1 1
         add prc
         jda pad              //print trap addr
         law 55
         jda tys
         law ac
         jmp ta5

tr1,     dac ovf
         dio io
         jsp sbc
         dzm fl1
         szf 1
         dac fl1
         move bki, 1 bki
         lac bki
         jmp 1 ovf

xe1,     xx
         nop
         dac ac
         jsp tr1
         jmp lss
```

```

pra,      lio .
          dio bix
          lio chi
          spi
pr1,      law 0
          cma
          dac ch
          jsp lcc
          cks
          ril 2s
          spi 1
          jmp .-3
          lac sbi
          lot 56
          sza
          esm
pr3,      lac fl1
          sza 1
          clf 1
          clo
          lac ovf
          add ovf
          lio 1 bk1      /get instr. at new brk addr.
          dio bk1
          lio (jda tr
          dio 1 bk1
          lio 10

pr2,      lac ac
bix,      xx
prc,      jmp .
prd,      jmp .

xec,      dac xe1      /execute
          law xe1

bgn,      spi          /begin
          jmp err
          dap bix
          lac prc
          dip bix
          jmp pr1

eas,      law ea1      /effective address search
          jmp ws

nws,      lac sk2      /not word search
          dac wea

wds,      law ws1      /word search
ws,       spi
          jmp err
          dap ws2
          jsp lcc
          dzm t2
          lac ll
          dac t

```

```

ws4,      dzm sym
          dap t2
          lac 1 t2
ws2,      jmp .                /ea1 or ws1

ea1,      and c1
          sza
          jmp ea2
          law 7777
          and 1 t2

ws1,      xor wrd
can,      and msk            /used as and
wea,      xx                /sza or sza 1
          jmp ws3

ws6,      law lcc

pac,      dap pax
          lac t
          jda pad
          law 2136
          jda tys
          lac 1 t
          jda lwt
pax,      jsp .

ws3,      idx t                /index and skip over pgm
          sub ul
          szm
          jmp lse
          add ul
          sub est
          sma
          jmp lse
          lac t
          jmp ws4

ea2,      idx sym
          sad c77
          jmp ws3
          lac 1 t2
          jmp ws4+1

pbx,      dac lwt
          jsp lct            /print as bcd
          jda tys
          jmp pn2

```

```

vfy,      jsp lcc
          lac rb2
          jmp .+2
rd,       lac bs1
          dip vf4
          jsp soi

vf1,      lac t
          sub ll
          sub (dio
          spa
          jmp vf2
          add ll
          sub ul
          szm
          jmp vf2
vf4,      lac i la
          t           /dac 1 or sad 1
          jmp vf2

vf3,      jsp pac
          jsp let
          lac i la
          jda lwt
          jsp lcc

vf2,      idx t
          idx la
          sad rb1
          jsp rbk
          jmp vf1

lwt,      0
          dap pnx
          lac lwt

pns,      jda p1
pnx,      jmp .           /pi or opt

kil,      law low
          dac est
          jmp lse

```

```

tbl,      jsp soi          /symbol table reader
tbl,      lac 1 la
          and (202020      /permute zones
          ral 1s
          xor 1 la
          6600 - xor c4
          cli
          rcl 6s
          sza
          jmp .-2
          idx la
          sad rb1
          jmp tbn
          lac 1 la
          dac df1
          dio sym
          law 1 1700
          and sym          /delete symbols of form 1s, 2s,...9s
          sas (char rs
          jsp de
          idx la
          sad rb1
          jsp rbk
          jmp tbl

tbn,      jsp lct
          lac est
          jda opt

tbn,      jsp rbk          /skips rest of  tape
          jmp tbn

          define
          feed n
          law 1 n
          jda fee
          terminate

ttl,      jsp lcc          /title punch and punch format setup
          listen
          rcr 9s
rc,       rar 9s
          sad c77
          jmp pir
          sad (36
          jmp pri
          sad (75
          jmp pi2
          ral 1s
          add (ftp
          jda tt1
          idx tt1
          law tt1+1
          jmp tt1+1

```

```

jbc,      spi                /jump block
          jmp err
          add cj
          dac lwt
          feed 40
          lio lwt
          jsp pbw
          feed 240
          jmp lse

pul,      dap fa             /punch lower limit setup
          jmp lss

pwd,      spi i              /punch word
          dac i tas
          dac lwt
          lac tas
          dap fa

pun,      dap la             /punch any length block

pb5,      lac fa
          lor c77
          dac t
          sub la
          sma
          jmp pb6             /next hundred too high
          idx t

pb4,      jsp pbb            /pbb or pur
          lac t
          dap fa
          jmp pb5

pb6,      lac la
          dac t
          idx t
          xct pb4
          jmp pn2

```

```

zro,      law 7777
          spi
          dac wrd
          and fa
          spi
          cla
          dac t

zr1,      sub est
          sma
          jmp lse
          add est
          sub wrd
          szm
          jmp lse
          dzm i t
          idx t
          jmp zr1

fee,t2,   0
          dap fex
          cli
          ppa
          isp fee
          jmp .-2
fex,      jmp .

6746 sbc,  dap sbx
          dac sbi
          cks
          ril 6s
          spi i
          dzm sbi
          lsm
sbx,      jmp .

```

/zero registers below ddt

/feed subroutine and temp storage.

/sequence break status check


```

pi,      xx          /print instruction
        dap px
        jsp pev
        sub ci
        spa
        jmp ppk
        dac pi
        law 72
        jda tys
        jsp tou
        law 71
        jda tys

ppk,     jsp tou
        law 72
        jda tys
        and (760000
        sad pr1      /law
        jmp plo
        rar 1s
        sza
csu,     sub (320000  /used as sub
        spa
        jmp plo

pvl,     lac pi
        jda opt
px,      jmp .        /exit

pev,     dap pex      /symbol lookup subr
        lac est
        dap ea
        clf 1

eal,     idx ea
ea,      lac .
        xor pi
        spa
        jmp eix
        lac pi
        sub i ea
        spa
        jmp eix
        szf i 1
        jmp psw
        lac i ea
        sub i ch
        szm
        jmp psw

```

```

ex,      index ea, evc, eal
        szf i 1
        jmp pvl
        lac pi
        sub i ch
        dac pi
        law i 1
        add ch
        dap ch
        lac i ch
        jda tys
        lac pi
sk2,     sza i
        jmp px
pex,     jmp .

pad,     0                /print address
        dap px
        law 7777
        and pad
        dac pi
pa1,     jsp pev          /pev or pvl
        lac (flexo +
        jda tys
        jmp pvl

7085 tys, 0                /type symbol, etc.
        dap tyx
        setup opt,3

tyl,     lac tys
        ral 6s
        dac tys
        and c77
        sza i
        jmp tyc
        sad (72
        jmp dns
        sad (74
        jmp ups
        swap
tyb,     jsp tou
tyc,     count opt, tyl
        lac lwt
        cli
tyx,     jmp .

```

```

dns,      lac ps1          /redundant case shift filter
          lio (72
dn1,      sad cas
          jmp tyc
          dac cas
          jmp tyb

ups,      lac rc
          lio (74
          jmp dn1

721 lcc,   dap lcx
          law 7277
          jmp lc1

lct,      dap lcx
          law 7236
lc1,      jda tys
lcx,      jmp .

so1,      rpb              /skip over input routine
so1,      rpb              /enter here
          spi 1
          jmp so1

rbk,      dap rbx          /read a block into buffer
          init rb1, buf
          dap la
          dzm chi
          rpb
          dio t2
          dio t
          spi
          jmp lse          /start block read
          rpb
          dio ch

rb0,      rpb
rb1,      dio .
          lac i rb1
          add chi
          dac chi
          idx rb1
          index t2, ch, rb0
          add chi
          add t
          rpb
          dio chi

rb2,      sad i .-1        /used as sad i

rbx,      jmp .
          hlt+clc-opr      /checksum error stop
          jmp rbk+1

```

```

tt1,      0                /title punch subroutine
          dap tt2
          lac 1 tt1
          repeat 3 cli     rcl 6s    ppa
tt2,      jmp .

pur,      dap pb2         /punch read-in mode blocks

pu1,      lio fa
          jsp pbw
          lio 1 fa
          jsp pbw
          index fa, t, pu1
          jmp pux

pbb,      dap pb2         /punch binary block format
          dzm t2
          lio fa
          jsp pbw
          lio t
          jsp pbw

pb1,      lio 1 fa
          jsp pbw
          index fa, t, pb1
          lio t2
          jsp pbw

pux,      feed 5
pb2,      jmp .

pir,      feed 40
          move 7754, t
          init fa, 7751
          jsp pur

pi2,      lio 7775         /jmp 7751
          jsp pbw
          law pbb

pi1,      dap pb4
          feed 30
          jmp lse

```

/combined octal-decimal print subroutine

```

opt,      0
           dap opx
ops,      init op1, odv      /odv or ddv
           setup op2, 6
           stf 1

opa,      dzm opd
           szf 1 1
           jsp tou
           jmp opc

opb,      clf 1
           dac opt
           idx opd
opc,      lac opt
           lio opt
op1,      sub .
           spi 1
           sma
           jmp opb

           lac opd
           lio opd
           sza 1
           lio ddv+1
           idx op1
           count op2, opa
           jsp tou
opx,      jmp .

ddv,      decimal 100000      10000      1000
           100          10          1          octal

odv,      100000      ci, 10000      1000
           100          10          one, 1

```

/dispatch table

```

72x dtb,      disp pls, pls      /0
              disp n, quo
              disp n, sqo
              disp n, pbx
              disp n, daq
              disp n, uni
              disp n, isc
              disp n, pul
              disp n, uc8
              disp n, fs
wr d,         0
sym,         0
chi,         0
let,         0
ch,          0
loc,         0
              disp n, arw
              disp bar, err
              disp l, smb
              disp l, tbl
              disp l, dec
              disp l, vfy
              disp l, wds
              disp l, xec
              disp l, rd
              disp l, zro
syl,         0
              disp com, eql
t,           0
la,          dio
              disp tab, tab
fa,          dio

```

```

disp pwd, err
disp l, jbk
disp l, kil
disp l, ttl
disp l, m
disp l, nws
disp l, oad
disp l, pra
disp l, q
disp l, rad
bki, opr
sbi, -0
disp min, pls
disp def, bas
disp err, err
disp val, bac
f11, 0
disp l, a
disp l, bk
disp l, cns
disp l, pun
disp l, eas
disp l, f
disp l, bgn
disp l, oct
disp l, ir
disp lc, lc
disp dot, del
disp uc, uc
disp bs, bs
df1, 0
disp cr, cr

```

/title punch table

```

7ABC ftp, 0 0 /space
004277 c4,400000 /1
625151 514600 /2
224145 453200 /3
141211 771000 /4
274545 453100 /5
364545 453000 /6
010171 050300 /7
324545 453200 /8
065151 513600 /9
tou, dap tox dio tot /typeout subroutine
cks ril 2s
spi 1 jmp .-3
lio tot tyo-1
tox, jmp . op2, 0
opd,dnm, 0 tot, 0

```

	364141	413600	/zero
c77,	000077	000000	//
	224545	453000	/s
	010177	010100	/t
	374040	403700	/u
	073060	300700	/v
	376014	602700	/w
	412214	224100	/x
	010274	020100	/y
	615141	454300	/z
plo,	jsp pev	jmp pa1+1	
	141414	141400	/=
pbw,	dap pby	ppb	/punch 1 word
	rcl 6s	ppb	
	rcl 6s	ppb	
	rcl 6s	add t2	
	dac t2	pby,	jmp .
	204040	403700	/j
	771014	224100	/k
	774040	404000	/l
	770214	027700	/m
	770214	207700	/n
	364141	413600	/o
	771111	110600	/p
	364151	215600	/q
	771111	314600	/r
psw,	lio ea	dio ch	
	stf 1	jmp eix	
	101010	101000	/-
	000041	221400	/)
	101074	101000	/+
	001422	410000	/(
pri,	law pur	jmp pi1	
	761111	117600	/a
	774545	453200	/b
	364141	412200	/c
	774141	413600	/d
	774545	414100	/e
	770505	010100	/f
	364151	513000	/g
	771010	107700	/h
	004177	410000	/i
	000001	030000	/close quote
	000060	cj,600000	/.
	000003	020000	/open quote

7626 buf, buf+100/
 ovf, 0
 ac, 0
 io, 0
 msk, -0
 ll, 0
 ul, 7777

7627 constants
 start lis