



jms = 0100000  
"  
q:0  
fmp = jms q

heps

lac i q  
gars  
isz q  
=1  
tad aexp  
tad hexp  
dac aexp  
lac ans  
lmq  
lac ams  
sna cll  
jmp 2f  
lis 1  
dac 3f  
dac 4f  
lacq  
dac 1f  
lac hls  
lmq  
lac hms  
sna cll  
jmp 2f  
lis 1  
dac hms  
lacq  
dac hls  
lac hms  
mul

1:0

dac ans  
lacq  
dac ce10  
lac hls  
mul

3:0

dac ams  
lacq  
tad ce10  
glk  
dzm ce10  
tad ams  
szl cll  
isz ce10  
tad ans  
szl cll  
isz ce10  
dac ans  
lac hms  
mul

4:0

dac ams  
lacq  
tad ans  
szl cll  
isz ce10  
lmq  
lac ce10

tad ams  
sma  
jmp 5f  
isz aexp  
nop  
lrs 1  
5:xor rsign  
dac ams  
lacq  
dac ans  
jmp i q  
2:dm aexp  
dm ams  
dm ans  
jmp i q

"

q:0  
fdy = jms q  
lac i q  
garg  
isz q  
lac hms  
sna  
sys save  
rai  
dac 2f  
dac 3f  
dac 4f  
=1  
tad hexp  
cma  
tad aexp  
dac aexp  
lac ans  
lmq  
lac ams  
sna cl1  
jmp 8f  
div

2:0

s21

sys save  
dac ce10 → REM

lacq

dac 5f

lac ce10

frdiv

3:0

Deep

HMS

Rem  $\left( \frac{\text{Ams}}{\text{Hms}} \right)$

$\frac{\text{Hms}}{\text{Hms}}$

HMS

| A + B |

| C + D |

4:0

s21

sys save

lacq

dac ce10

lac hls

and o377777

frdiv

5:0

s21

sys save

lacq

dac 2b →

sna cl1

X  
AB

-1  
tad 2b  
cli  
mul @  
5:0

dzm 2b  
spa  
isz 2b  
lls 1  
dac 3b  
lacq  
spa  
isz 3b  
skp  
isz 2b  
lac ce10  
lmg  
lac 3b  
sna  
jmp 6f  
cma  
tad d1  
stl  
tad ce10  
lmg  
szl  
isz 2b  
6: lac 2b

sma  
tad d1  
tad 5b  
sma cli  
jmp 7f  
lrs 1  
isz aexp  
nop  
7:xor rsign  
dac ams  
lacq  
dac ans  
jmp i q  
8:dzm aexp  
dzm ams  
jmp i q

9:0  
fad = jms q  
lac i q  
garg  
isz q  
lac hms  
sna  
jmp 4f  
lac ams  
sna  
jmp 8f  
7: lac aexp  
cma  
tad hexp  
sma  
jmp 5f

dac ce10  
tad d34  
spa cla  
jmp 0f  
lac ce10  
cma  
tad d1  
xor o640500  
dac 1f  
lac hls  
lmc  
lac hms  
c11  
1:lrs 0  
dac hms  
lacq  
dac hls  
lac rsign  
sma  
jmp 2f  
lac hls  
c11 cma  
tad d1  
dac hls  
lac hms  
szl cma  
tad d1  
dac hms  
2: lac ams  
rcr  
dac ams  
lac ans  
rar  
c11  
tad hls  
dac ans  
glk  
tad ams  
tad hms  
dac ams  
sma  
jmp 3f  
lac ans  
cma c11  
tad d1  
dac ans  
lac ams  
szl cma  
tad d1  
dac ams  
lac o400000  
3: lsz aexp  
nop  
0: xor asign  
and o400000  
dac rsign  
fno  
4: lac ams  
xor rsign  
dac ams  
jmp i q

5:jms 6f  
lac rsign  
xor asign  
dac asign  
jmp 7b  
8:jms 6f  
jmp 4b  
6:0  
lac ans  
lmq  
lac his  
dac ans  
lacq  
dac his  
lac ams  
lmq  
lac hms  
dac ams  
lacq  
dac hms  
lac hexp  
lmq  
lac aexp  
dac hexp  
lacq  
dac aexp  
jmp i 6b  
"  
q:0  
fno = jms q  
lac ans  
sad ams  
sza cll  
jmp 1f  
dzm aexp  
dzm rsign  
jmp i q  
"  
1:lmq  
lac ams  
and o200000  
sza  
jmp i q  
lac ams  
c1l  
norm 36  
dac ams  
lacq  
dac ans  
lacs  
tad o777743  
cma  
tad aexp  
10 dac aexp  
9 jmp i q  
"  
7 q:0  
6 fcp = jms q  
5 lac i q  
4 garg  
3 lsz q  
2

```
lac rsign
spa
jmp 1f
lac ams
dac 5f
xor asign
dac ams
sna
jmp 2f
lac hms
sna cma
jmp 3f
lac hexp
cma
tad d1
tad aexp
sza
jmp 4f
2: lac hms
cma
3:tad d1
tad 5f
sza
jmp 4f
lac hls
cma
tad d1
tad ans
sza
4:xor asign
jmp i q
1: lac ams
xor d1
jmp i q
5: 0
```

```
q: 0
garg = jms q
tad dm1
dac 8
lac i 8
dac hexp
lac i 8
lmc
and o377777
dac hms
lac i 8
dac hls
lacq
xor ams
and o400000
dac rsign
lac ams
and o400000
dac asign
lac ams
and o377777
dac ams
jmp i q
q: 0
```

sfmp = jms q  
lac i q  
garg  
isz q  
=1  
tad aexp  
tad hexp  
dac aexp  
lac ams  
sna rcl  
jmp 2f  
lmq  
lac hms  
sna rcl  
jmp 2f  
dac .+2  
0641122; 0  
sma  
jmp 1f  
rcr  
xor rsign  
dac ams  
isz aexp  
jmp i q  
jmp i q

1:

xor rsign  
dac ams  
jmp i q

2:

dzm aexp  
dzm ams  
jmp i q

q: 0  
sfdv = jms q  
lac i q  
garg  
isz q  
lac hexp  
cma  
tad aexp  
tad d1  
dac aexp  
lac hms  
sna ral cll  
sys save  
dac 1f  
lac ams  
frdiv; 1: 0  
szl  
sys save  
lacq  
spa  
jmp 1f  
xor rsign  
dac ams  
jmp i q

1:

rcr  
xor rsign

9  
8  
7  
6  
5  
4  
3  
2  
1  
0  
dac ams  
isz aexp  
jmp i q  
jmp i q  
sfad = jms q  
-1  
tad i q  
isz q  
dac 8  
lac i 8  
dac hexp  
lac i 8  
sma  
jmp 1f  
xor o377777  
tad d1

1:  
irss 1  
dac hms  
lac ams  
sma  
jmp 1f  
xor o377777  
tad d1

1:  
irss 1  
dac ams  
lac hexp  
cma  
tad aexp  
tad d1  
sma  
jmp 1f  
cma  
tad d1  
dac tmp  
lac ams  
lmq  
lac hms  
dac ams  
lacq  
dac hms  
lac hexp  
dac aexp  
lac tmp

1:  
tad dm18  
sma  
jmp 3f  
tad o660522  
dac 1f  
lac hms

1:  
irss 0  
dm rsign  
tad ams  
cll sma  
jmp 1f  
lmq  
and o400000

dac rsign  
lacq  
cma  
tad d1  
cll sma  
jmp 1f  
isz aexp  
nop  
rar

1:  
sna  
jmp 1f  
norm 18  
xor rsign  
dac ams  
lac  
tad om60  
cma  
tad aexp  
dac aexp  
jmp i q

1:  
dzm aexp  
dzm ams  
jmp i q

3:  
lac ams  
rcl  
sma  
jmp 1f  
cma  
tad d1  
xor o400000

1:  
dac ams  
jmp i q  
q: 0  
fld = jms q  
-1  
tad i q  
dac 8  
lac i 8  
dac aexp  
lac i 8  
dac ams  
lac i 8  
dac ans  
isz q  
jmp i q

q: 0  
fst = jms q  
-1  
tad i q  
dac 8  
lac aexp  
dac i 8  
lac ams  
dac i 8  
lac ans  
dac i 8

isz q  
jmp i q

q: 0  
fnq = jms q  
lac ams  
sza  
xor o400000  
dac ams  
jmp i q

q: 0  
fix = jms q  
lac aexp  
spa sna  
jmp 1f  
tad dm18  
sma  
jmp 3f  
cma  
tad o660500  
dac 2f  
lac ams  
sma  
jmp 2f  
xor o377777  
tad d1

2:  
lrss 0  
jmp i q  
1:  
lac ams  
lrss 18  
jmp i q  
3:  
lac ams  
and o400001  
sma  
lac o377777  
jmp i q

q: 0  
fit = jms q  
dac tmp  
dzm ans  
sma  
jmp 1f  
cma  
tad d1  
spa  
cla

1:  
sza  
jmp 1f  
dzm aexp  
dzm ams  
jmp 2f

1:  
clq  
norm 36  
dac ams

lacs  
tad om56  
cma  
dac aexp  
2:  
lac tmp  
and o400000  
xor ams  
dac ams  
jmp i q  
  
tmp: 0  
stmp: 0  
ce10: 0  
asign: 0  
aexp: 0  
ams: 0  
ans: 0  
hexp: 0  
hms: 0  
hls: 0  
  
q: 0  
sin = jms q  
lac ams  
and o400000  
dac sign  
lac ams  
and o377777  
dac ams  
fst; fttmp1  
fdv; fpi  
fix  
dac stmp  
and d1  
sna  
jmp 1f  
lac o400000  
xor sign  
dac sign  
1:  
lac stmp  
flt  
fmp; fpi  
fng  
fad; fttmp1  
fst; strm  
fst; sres  
fst; fttmp2  
fld; fp1  
fst; sfac  
=6  
dac scnt  
11  
bsin:  
fld; sfac  
fad; fp1  
fst; fttmp1  
fad; fp1  
fst; sfac  
fld; strm

fmp; ftmp2  
fmp; ftmp2  
fdv; sfac  
fdv; ftmp1  
fnq  
fst; strm  
fad; sres  
fst; sres  
isz scnt  
jmp 1b  
lac ams  
xor sign  
dac ams  
jmp i q

q: 0  
sqrt = jms q  
lac aexp  
tad d1  
llss 0  
rar  
dac aexp  
lac ans  
lmq  
lac ams  
spa  
sys save  
dac 1f  
snl  
jmp 5f  
lls 1  
dac ams  
lacq  
dac ans

5:  
lac 1f  
sna  
jmp q i  
snl cll  
xor o200000  
xor o400000  
dac 1f  
lac ams.  
frdiv; 1:.,  
szl  
clq  
lacq  
tad 1b  
rar  
cll  
dac 2f  
lac ams  
frdiv; 2:.,  
szl  
clq  
lacq  
tad 2b  
rar  
dac 3f  
dac 4f  
lac ans

1mq  
lac ams  
c11  
div; 3...  
szl  
clq ecla  
dac 1b  
iacq  
tad 3b  
clq lrs 1  
c11  
lrs 1  
dac ams  
iacq  
dac 2b  
lac 1b  
frdiv; 4...  
szl  
sys save  
iacq  
lrs 2  
tad 2b  
dac ans  
jmp q i

sfac: 0;0;0  
ftmp1: 0;0;0  
ftmp2: 0;0;0  
strm: 0;0;0  
scnt: 0  
sres: 0;0;0  
rsign: 0  
sign: 0

fp1: 1;0200000;0

o400000: 0400000  
o640500: 0640500  
o200000: 0200000  
d34: 34  
o777743: 0777743  
o2: 02  
o377777: 0377777  
dm18: -18  
o377777: 0377777  
om60: -060  
o660522: 0660522  
o660500: 0660500  
o400001: 0400001  
dm1: -1  
om56: -056  
fp1: 2;0311037; 0552421  
fpid2: 1; 0311037;0552421

buf:  
cgarg = gargs  
cfmp = fmp=jms  
cfav = fav=jms  
cfad = nad=jms  
cfno = fno=jms  
cfcp = fcp=jms  
csfmp = sfmp=jms

csfdv = sf<sup>d</sup>v=jms  
csfad = sfad=jms  
cfla = fl<sup>a</sup>=jms  
cfst = fst=jms  
cfng = fng=jms  
cfix = fix=jms  
cflt = flt=jms  
csin = sin=jms

C

11

10

9

8

7

6

5

4

3

2



" ln

```
lac 017777 i  
sad d4  
jmp error  
lac 017777  
tad d1  
dac dirn  
lac dirn i  
sad qli  
jmp clink  
lac 017777  
tad d5  
dac dirn  
dac name
```

loop:

```
lac 017777 i  
sad d8  
sys exit  
tad dm4  
dac 017777 i  
lac name  
tad d4  
dac name  
dac name1  
sys link; dien; 0; name; 0; name1; 0  
sma  
jmp loop  
lac name  
dac f1  
lac d1  
sys write; 1; 0; 4  
lac d1  
sys write; errmes; 2  
jmp loop
```

clink:

```
lac 017777 i  
sad d8  
jmp arg1  
sad d12  
jmp arg2  
sad d16  
jmp arg3
```

error:

```
lac d1  
sys write; errmes+1; 1  
sys exit
```

arg1:

```
lac 017777  
tad d5  
dac larg+1  
dac larg+2  
jmp dlink
```

arg2:

```
lac 017777  
tad d5  
dac larg  
tad d4
```

```
    dac larg+1
    dac larg+2
    jmp dlink
arg3:
    lac 017777
    tad d5
    dac larg
    tad d4
    dac larg+1
    tad d4
    dac larg+2
dlink:
    sys link; larg; defdir$0;0
    sma
    sys exit
    lac larg
    dac 1f
    lac larg+1
    dac 2f
    lac larg+2
    dac 3f
    lac d1
    sys write; 1t.;; 4
    lac d1
    sys write; 2t.;; 4
    lac d1
    sys write; 3t.;; 4
    lac d1
    sys write; errmes; 2
    sys exit
```

```
errmes:
    040; 077012
d1: 1
qli: <li>
d12: 12
d16: 16
defdir: <sy>;<st>;<em>;040040
d4: 4
d8: 8
dm4: -4
d5: 5
```

63

" ls  
" list

lac 017777  
tad d1  
dac name  
lac name i  
sad ali  
dac longflg  
lac 017777 i  
sad d4  
skp  
jmp loop  
law dd  
dac name  
jmp 2f

loop:  
lac 017777 i  
sad d4  
jmp done  
tad dm4  
dac 017777 i  
lac name  
tad d4  
dac name

2:  
law stbuf  
sys status; dd# name;,i  
spa  
jmp badfile  
lac s,flags  
and o20  
sna  
jmp badfile  
lac name  
dac Of  
sys open; Of,,i 0  
spa  
jmp badfile  
dac fi  
jms readdir  
lac fi  
sys close

1:  
lac o200000  
dac maxfn  
jms findf  
lac maxfn  
sad o200000  
jmp loop

11  
10 lac longflg  
9 sza  
8 jms longout  
7 law maxfn  
6 jms putfn  
5 law O12  
4 jms putc  
3 lac o200000  
dac maxfp i  
jmp 1b

longout: 0  
lac name  
dac Of  
law stbuf  
sys status; 0;,, maxfn  
lac s,i  
jms octal; -3  
lac s,flags  
jms octal; -2  
lac s,uid  
jms octal; -2  
-1  
tad s,nlinks  
cma  
jms octal; -2  
lac s,size  
jms octal; -5  
jmp longout i

octal: 0  
lmg  
lac d5  
tad octal i  
cma  
dac t

1:  
llss 3  
isz t  
jmp 1b  
lac octal i  
dac t  
1:  
ecia llss 3  
tad c60  
jms putc  
isz t  
jmp 1b  
law 040  
jms putc  
isz octal  
jmp octal i

toobig:  
law 076  
jms putc  
law 040  
jms putc

badfile:  
lac name  
jms putfn  
law 040  
jms putc  
law 077  
jms putc  
law 012  
jms putc  
jmp loop

putfn: 0

```
    dac t
    =4
    dac t1
1:
    lac t i
    lrss 9
    sad o40
    jmp putfn i
    jms putc
    lac t i
    and o177
    sad o40
    jmp putfn i
    jms putc
    isz t
    isz t1
    jmp 1b
    jmp putfn i

done:
    lac noc
    sna
    sys exit
    and d1
    sna cla
    jmp 1f
    jms putc
    jmp done
1:
    lac noc
    rcr
    dac 1f
    lac fo
    sys write; iopt+1; 11;
    sys exit

readdir: 0
    lac fi
    sys read; buf; 2048
    sad .=1
    jmp toobig
    lrss 3
    cma
    tad d1
    dac ndir
    jmp readdir i

findfi: 0
    law buf
    dac t
    lac ndir
    dac t1
1:
    lac t i
    sna
    jmp 2f
    isz t
    lac t i
    cma
    tad maxfn
    spa
```

```
    jmp 2f+1
    lac t
    dac maxfp
    lac t i
    dac maxfn
    skp
2:
    isz t
    lac t
    tad d7
    dac t
    isz t1
    jmp 1b

    lac maxfp
    dac 8
    lac 8 i
    dac maxfn+1
    lac 8 i
    dac maxfn+2
    lac 8 i
    dac maxfn+3
    jmp findf i

putc: 0
    and o177
    dac 2f+1
    lac opt
    dac 2f
    add o400000
    dac opt
    spa
    jmp 1f
    lac 2f i
    xor 2f+1
    jmp 3f
1:
    lac 2f+1
    alss 9
3:
    dac 2f i
    isz noc
    lac noc
    sad d128
    skp
    jmp putc i
    lac fo
    sys write; iopt+1; 64
    lac iopt
    dac opt
    dzm noc
    jmp putc i
10:
    jmp putc i
2: 0;0
fi: 0
opt: ,+2
iopt: ,+1; ,=, +64
noc: 0
fo: 1
longflg: 0
d1: 1
```

d5: 5  
o60: 060  
o40: 040  
o20: 020  
d7: 7  
o400000: 0400000  
d128: 128  
d4: 4  
dm4: =4  
o177: 0177  
o200000: 0200000  
dd: 056056;040040;040040;040040  
ali: <li>

t: ,=,+1  
t1: ,=,+1  
maxfn: ,=,+4  
maxfp: ,=,+1  
ndir: ,=,+1  
stbuf:  
    s.flags: ,=,+8  
    s.uid: ,=,+1  
    s.nlinks: ,=,+1  
    s.size: ,=,+2  
    s.in: ,=,+1  
buf:

64

" moo

jmp 1f  
reset:  
jms messg; <es>;<et>;012;0

1:  
jms messg; <vr>; <u 077; 040; 0  
sys open; moostat; 0  
spa  
sys exit  
dac fi  
sys open; moostat; 1  
spa  
sys exit  
dac fo  
jms readline  
dzm user  
skp

1:  
isz user  
jms getentry  
jmp noentry  
lac name  
sad u.name  
skp  
jmp 1b  
lac name+1  
sad u.name+1  
skp  
jmp 1b  
lac name+2  
sad u.name+2  
skp  
jmp 1b  
lac name+3  
sad u.name+3  
skp  
jmp 1b

gloop:  
jms messg; <re>; <ad>; <y 077; 040; 0  
law rqname=1  
dac 8  
law rqhand=1  
dac 9  
=nrq  
dac 2f  
jms readch

1:  
sad 8 i  
jmp 9 i  
isz 9  
isz 2f  
jmp 1b  
jms messg; 077012; 0  
jmp gloop

2: 0

noentry:  
jms messg; <na>; <me>; 040; <no>; <t 040; <fe>  
<un>; <d 073; 040; <en>; <te>; <r 077; 040; 0  
jms readch

sad ch,y  
skp  
sys exit  
jms newline  
lac name  
dac u,name  
lac name+1  
dac u,name+1  
lac name+2  
dac u,name+2  
lac name+3  
dac u,name+3  
dzm u,ngames  
dzm u,nguess  
dzm u,ntime  
dzm u,npenalty  
jms putentry  
jmp gloop

gstart1:  
jms messag; <es>; 012; 0

gstart:  
jms random  
dac a1

1:  
jms random  
sad a1  
jmp 1b  
dac a2

1:  
jms random  
sad a1  
jmp 1b  
sad a2  
jmp 1b  
dac a3

1:  
jms random  
sad a1  
jmp 1b  
sad a2  
jmp 1b  
sad a3  
jmp 1b  
dac a4  
dzm nguess

guessloop:  
jms readguess  
lac nguess  
sza  
jmp 1f  
sys time  
lacq  
rcr  
dac stime  
1:  
dzm nnull  
dzm ncov  
lac g1

sad a1  
isz nbull  
sad a2  
isz ncow  
sad a3  
isz ncow  
sad a4  
isz ncow  
lac g2  
sad a1  
isz ncow  
sad a2  
isz nbull  
sad a3  
isz ncow  
sad a4  
isz ncow  
lac g3  
sad a1  
isz ncow  
sad a2  
isz ncow  
sad a3  
isz nbull  
sad a4  
isz ncow  
lac g4  
sad a1  
isz ncow  
sad a2  
isz ncow  
sad a3  
isz ncow  
sad a4  
isz nbull  
lac nbull  
sad d4  
jmp gdone  
jms messg; 040040; 040040; 040040; <bc>; 075; 0  
lac nbull  
jms number  
lac ncov  
jms number  
jms newline  
isz nguess  
jmp guessloop

gdone:  
sys time  
lacq  
rcr  
cma  
tad stime  
cma  
spa  
tad o400000  
rcr  
dac stime  
lac u\_ntime  
tad stime  
dac u\_ntime

lac u\_nguess  
tad nguess  
dae u\_nguess  
isz u\_ngames  
jms putentry  
jms messg; 012; <g 075; 0  
lac nguess  
jms number  
jms messg; 040; <t 075; 0  
lac stime  
cli; idiv; 15  
laeq  
jms number  
jms newline  
jmp gloop

random: 0  
sys time  
laeq  
tad rand  
cli; mul; 78125  
laeq  
dac rand  
cli; idiv; 10  
jmp random i

newline: 0  
jms messg; 012; 0  
jmp newline i

number: 0  
lmg  
lav 2f+1  
dae 3f  
laeq

1:  
cli; idiv; 10  
tad o60  
dac 3f i  
isz 3f  
laeq  
sza  
jmp 1b

1:  
-1  
tad 3f  
dae 3f  
lac 3f i  
sna  
jmp number i  
dac .+2  
jms messg; .+3 0  
jmp 1b  
2: 0; .=.+10  
3: .=.+1

readguess: 0  
jms messg; 077040; 0  
jms cnum  
dae g1  
jms cnum

1  
dac g2  
jms cnum  
dac g3  
jms cnum  
dac g4  
jms readch  
sad o12  
jmp readguess i  
jmp readguess+1

cnum: 0  
jms readch  
tad om60  
spa  
jmp readguess+1  
dac 1f  
tad dm10  
sma  
jmp readguess+1  
lac 1f  
jmp cnum i  
1: 0

readch: 0  
cla  
sys read; 1f; 1  
lac 1f  
lrss 9  
jmp readch i  
1: 0

readline: 0  
-1  
dac r1f  
jms read2  
dac name  
jms read2  
dac name+1  
jms read2  
dac name+2  
jms read2  
dac name+3  
jmp readline i

read2: 0  
jms read1  
alss 9  
dac 1f  
jms read1  
xor 1f  
jmp read2 i  
1: 0

read1: 0  
lac r1f  
sna  
jmp 1f  
jms readch  
sad o12  
skp  
jmp read1 i

```
dzm rif
1:
lac o40
jmp read1 i
rif: 0

getentry: 0
lac user
cli; mul; 16
lacq
dac Of
lac fi
sys seek; 01.,; 0
lac fi
sys read; userdata; 16
sza
isz getentry
jmp getentry i

putentry: 0
lac user
cli; mul; 16
lacq
dac Of
lac fo
sys seek; 01.,; 0
lac fo
sys write; userdata; 16
jmp putentry i

average:
jms messg; <ve>;<r 012>; 0
jms paver
jmp gloop

standing:
jms messg; <ta>; <nd>; 012; 0
lac user
dac 2f
dzm user
skip
1:
isz user
jms getentry
jmp 1f
lac d1
sys writes; u.name; 4
jms paver
jmp 1b
1:
lac 2f
dac user
jms getentry
nop
jmp gloop
2: 0

paver: 0
lac u.ngames
sna
jmp 1f
```

```
jms messg; <n 075; 0
lac u.ngames
jms number
jms messg; 040; <g 075; 0
lac u.nguess
jms number
lac u.nguess
jms aver
jms messg; 040; <t 075; 0
lac u.ntime
cli; idiv; 15
lacq
jms number
lac u.ntime
cli; idiv; 15
lacq
jms aver
jms newline
jmp paver i
```

```
1: jms messg; <no>i 040; <ga>i <me>i <s 012; 0
jmp paver i
```

```
aver; 0
dac 1f
lac u.ngames
dac 0f
jms messg; 050; 0
lac 1f
cli; idiv; 0; .
dac 1f
lacq
jms number
jms messg; 056; 0
lac u.ngames
dac 0f
lac 1f
cli; mul; 100
cli; div; 0i; .
ecia; div; 10
dac 1f
lacq
jms number
lac 1f
jms number
jms messg; 051; 0
jmp aver i
1: .=,+1
```

```
messg; 0
-1
tad messg
dac 8
1:
lac 8 i
sna
jmp 8 i
dac 1f
lac d1
sys write; 1f; 1
jmp 1b
```

```
1: 0

rqname:
    y>
    q>
    a>
    s>
    r>
nreq = ,=rename
rqhandl:
    jmp gstart1
    jmp quit
    jmp average
    jmp standings
    jmp reset

quit:
    jms messg; <ui>; <t 012>; 0
    sys exit

d1: 1
0400000: 0400000
060: 060
0m60: -060
dm10: -10
d4: 4
040: 040
012: 012
ch,y: y>

m1: <wr>;<u 077>; 040
m1s = ,=m1
m2: <re>;<ad>;<y 077>; 040
m2s = ,=m2
m3: <na>;<me>;040;<no>;<t 040>;<fo>;<un>;<a 012
    <en>;<te>;<r 077>; 040
m3s = ,=m3
m5: 077040
m5s = ,=m5
m7: 012;<gu>;<es>;<s 075
m7s = ,=m7
m8: <av>;<g 075
m8s = ,=m8
m9: 057
m9s = ,=m9
m10: 040075; 040
m10s = ,=m10
moostat: <mo>;<os>;<ta>;<t 040

f1: ,=,+1
f0: ,=,+1
name: ,=,+4
nguess: ,=,+1
nbull: ,=,+1
ncov: ,=,+1
stime: ,=,+1
a1: ,=,+1
a2: ,=,+1
a3: ,=,+1
a4: ,=,+1
g1: ,=,+1
```

```
g2: s+=s+1  
g3: s+=s+1  
g4: s+=s+1  
rand: s+=s+1  
user: s+=s+1  
userdata:  
    u.name: s+=s+4  
    u.ngames: s+=s+1  
    u.nguess: s+=s+1  
    u.ntime: s+=s+1  
    u.npenalty: s+=s+1  
    u.hguess: s+=s+1  
    u.lguess: s+=s+1  
    u.htime: s+=s+1  
    u.ltime: s+=s+1  
    s = userdata+16
```

65

" nm  
lac 017777 i  
sad d4  
skp  
jmp 1f  
law n,out  
dac fname  
jmp 2f  
1:  
lac 017777  
tad d1  
dac fname

loop:  
lac 017777 i  
sad d4  
jmp done  
tad dm4  
dac 017777 i  
lac fname  
tad d4  
dac fname

2:  
law 012  
jms putc  
sys open; fname; 0; 0  
sma  
jmp 1f  
lac fname  
dac 2f  
lac d1  
sys write; 21 0; 4  
lac d1  
sys write; mes; 2  
jmp loop  
mes:  
040077; 012

1:  
lac d2  
sys read; buf; 3072  
cll; idiv; 6  
lacq  
dac size  
law o200000  
dac c2

print:  
lac o200000  
dac c2 i  
azm c2name  
-1  
tad size  
cma  
dac c1  
law buf  
dac c3  
lac o200000  
dac c2name  
1:  
lac c3 i

cma  
tad c2name  
spa  
jmp 2f  
lac c3 i  
dac c2name  
lac c3  
dac c2

2:  
law 6  
tad c3  
dac c3  
isz c1  
jmp 1b  
lac c2name  
sad c200000  
skp  
jmp 1f  
lac d2  
sys close  
jmp loop

1:  
lac c2  
tad d3  
dac t  
lac i t  
sna  
jmp print  
isz t  
lac i t  
sna  
jmp print  
-1  
tad c2  
dac 8  
-4  
dac c3

1:  
lac 8 i  
lrss 9  
jms putc  
llss 9  
jms putc  
isz c3  
jmp 1b  
lac i t  
sad d3  
jmp undef  
sna  
jmp 1f  
law 0162  
skp

1:  
7 law 0141  
6 dae type  
5 law 040  
4 jms putc  
3 isz t  
lac i t  
1mg

```
-6
dac c3
1:
    cla
    liss 3
    tad 060
    jms putc
    isz c3
    jmp 1b
    law 040
    jms putc
    lac type
    jms putc
    law 012
    jms putc
    jmp print
undef:
-8
dac c3
1:
    law 040
    jms putc
    isz c3
    jmp 1b
    law 0165
    jms putc
    law 012
    jms putc
    jmp print

done:
    lac noc
    sna
    sys exit
    and d1
    sna cla
    jmp 1f
    jms putc
    jmp done
1:
    lac noc
    rcr
    dac 1f
    lac fo
    sys write; iopt+1; 1,;
    sys exit

putc: 0
    and 0777
    dac 2f+1
    lac opt
    dac 2f
    add 0400000
    dac opt
    spa
    jmp 1f
    lac 2f i
    xor 2f+1
    jmp 3f
1:
    lac 2f+1
```

alss 9  
3:  
dac 2f i  
isz noc  
lac noc  
sad d128  
skp  
jmp putc i  
lac fo  
sys write; iopt+1; 64  
lae iopt  
dac opt  
dzm noc  
jmp putc i  
2: 0;0  
ipt: 0  
eipt: 0  
iipt: ,+1; ,=,+64  
fi: 0  
opt: ,+2  
iopt: ,+1; ,=,+64  
noc: 0  
fo: 1  
  
c1: 0  
c2: 0  
c3: 0  
t: 0  
size: 0  
c2name: 0  
type: 0  
  
d1: 1  
d128: 128  
o200000: 0200000  
o777: 0777  
o400000: 0400000  
d2: 2  
d4: 4  
dm4: -4  
o60: 060  
d3: 3  
d6: 6  
n.out:  
0156056;0157165;0164040;040040  
buf:

16

67

" op

sys = 0020000  
dac = 0040000  
jms = 0100000  
dzm = 0140000  
lac = 0200000  
xor = 0240000  
add = 0300000  
tad = 0340000  
xct = 0400000  
isz = 0440000  
and = 0500000  
sad = 0540000  
jmp = 0600000  
law = 0760000

nop = 0740000  
cma = 0740001  
cmi = 0740002  
oas = 0740004  
ral = 0740010  
rar = 0740020  
hit = 0740040  
sma = 0740100  
sza = 0740200  
snl = 0740400  
skp = 0741000  
spa = 0741100  
sna = 0741200  
szl = 0741400  
rtl = 0742010  
rtr = 0742020  
cil = 0744000  
stl = 0744002  
rel = 0744010  
rcr = 0744020  
cla = 0750000  
clc = 0750001  
las = 0750004  
gik = 0750010  
lrs = 0640500  
lrss= 0660500  
lls = 0640600  
llss= 0660600  
als = 0640700  
alss= 0660700  
norm= 0640444  
norms=0660444  
mul = 0653122  
muls= 0657122  
div = 0640323  
divs= 0644323  
idiv= 0653323  
idivs=0657323  
frdiv = 0650323  
frdivs = 0654323  
lacq = 0641002  
lacqs = 0641001  
clq = 0650000  
ads = 0644000  
gsm = 0664000

osc = 0640001  
omq = 0640002  
otq = 0642000  
cmq = 0640004  
lmg = 0652000  
ecla = 0641000

i = 0200000

save = 1  
getuid = 2  
open = 3  
read = 4  
write = 5  
creat = 6  
seek = 7  
tell = 8  
close = 9  
link = 10  
unlink = 11  
setuid = 12  
rename = 13  
exit = 14  
time = 15  
intrp = 16  
chdir = 17  
chmod = 18  
chowner = 19  
sysloc = 21  
capt = 23  
rele = 24  
status = 25  
sleep = 26  
smes = 27  
rmes = 28  
fork = 29

88

" p1  
t = 0  
las  
dac .sw  
law 13  
sys sysloc  
dac .pb  
jms ballinit  
lac nballp  
cl1; mul; bvsiz  
lacq  
tad listpm1  
dac 15  
tad d1  
sys capt  
law outline-1  
dac 8  
-outline  
dac 9f+t

1:  
lac 8 i  
dac 15 i  
isz 9f+t  
jmp 1b  
lac 15  
dac dispplist  
lac 0400000  
dac 15 i  
jms dump

loop:

" dump/restore

lac waitup  
sza  
jmp 2f  
lac .pb i  
als 5; ral  
snl  
jmp 1f  
jms dump  
jmp 3f

1:  
sma  
jmp 3f  
jms restore  
jmp 3f

2:  
lac .pb i  
als 5; ral  
szi  
jmp 3f  
spa  
jmp 3f  
dzm waitup

3:  
sys time  
lacq  
sad stime  
jmp loop

```
tad dm1
sad stime
jmp loop
tad d1
dac stime
lac .pb i
als 7
spa
sys exit

lac dispist
dac 15

-nball
dac 9f+t

" q and stick controls

jms getball; ball1; 9f+t
lac ball1+vx
lmg
lac ball1+vy
omq
sza
jmp 1f
jms stickcont
jms putball; ball1; 9f+t

1:
jms getball; ball1; 9f+t

" if in pocket, ignore

lac ball1+sflg
spa
jmp 4f
lac 9
dac 14

" update

lac ball1+vx
lrss 6
tad ball1+x
and o177777
dac ball1+x
lac ball1+vy
lrss 6
tad ball1+y
and o177777
dac ball1+y

" display
10
lac ball1+x
lrss 6
xor o142000 " setx
dac 15 i
lac ball1+y
lrss 6
xor o164000 " sety
```

```
    dac 15 i  
  
    law circle=1  
    dac 8  
    -circsize  
    dac 9f+t+1  
2:  
    lac 8 i  
    dac 15 i  
    isz 9f+t+1  
    jmp 2b  
  
    " degrade velocity  
    jms frict  
  
    " edge collision  
  
    lac lefttest  
    tad ball1+x  
    sma  
    jmp 2f  
    jms pocketlr  
    jmp 4f  
    lac ball1+vx  
    sma  
    jmp 2f  
    cma  
    tad d1  
    dac ball1+vx  
2:  
    lac hottest  
    tad ball1+y  
    sma  
    jmp 2f  
    jms pockettb  
    jmp 4f  
    lac ball1+vy  
    sma  
    jmp 2f  
    cma  
    tad d1  
    dac ball1+vy  
2:  
    lac righttest  
    tad ball1+x  
    spa  
    jmp 2f  
    jms pocketlr  
    jmp 4f  
    lac ball1+vx  
    spa  
    jmp 2f  
    cma  
10    tad d1  
9    dac ball1+vx  
8:  
2:  
7    lac toptest  
6    tad ball1+y  
5    spa  
4    jmp 2f  
3    jms pockettb
```

jmp 4f  
lac ball1+vy  
spa  
jmp 2f  
cma  
tad d1  
dac ball1+vy

2:

" ball/ball collision

lac 9f+t  
tad d1  
sma  
jmp 4f  
dac 9f+t+1

2:

lac 14 i  
dac ball2+0  
lac 14 i  
dac ball2+1  
lac 14 i  
dac ball2+2  
lac 14 i  
dac ball2+3  
lac 14 i  
dac ball2+4

lac ball2+sflg

spa  
jmp 3f  
lac ball1+x  
cma  
tad ball2+x  
cma  
lmg  
gsm  
dac .+3  
lacq

muls; ..  
dac 9f+t+3

lrss 4

sza  
jmp 3f  
lac ball1+y

cma  
tad ball2+y  
cma  
lmg  
gsm  
dac .+3

lacq  
muls; ..  
tad 9f+t+3

lrss 4

sza  
jmp 3f

jms ballball

jms putball; ball2; 9f+t+1

3:

isz 9f+t+1

jmp 2b  
4:  
jms putball; ball1; 9f+t  
isz 9f+t  
jmp 1b  
  
lac 0400000  
dac 15 i  
jmp loop  
t = t+3

11  
10  
9  
8  
7  
6  
5  
4  
3

```
" p2  
frict: 0  
las  
and d1  
sza  
jmp frict i  
lac ball1+vx  
c11; muls; frfac  
dac ball1+vx  
lac ball1+vy  
c11; muls; frfac  
dac ball1+vy  
gsm  
dac 9f+t  
lac ball1+vx  
gsm  
tad 9f+t  
tad minvx  
sma  
jmp frict i  
dzm ball1+vx  
dzm ball1+vy  
jmp frict i  
t = t+1
```

```
ballball: 0  
llss 15 " x**2+y**2 in q  
[ cma  
tad 0300000  
dac 1f; dac 2f  
lac ball2+y  
cma  
tad ball1+y  
cma  
c11; muls; 1:.,; llss 6  
dac sin  
lac ball2+x  
cma  
tad ball1+x  
cma  
c11; muls; 2:.,; llss 6  
dac cos
```

```
" calculate closing velocities
```

```
lac ball1+vx  
gsm  
dac .+3  
lac cos  
muls; .,.; llss 3  
dac 9f+t  
lac ball1+vy  
gsm  
dac .+3  
lac sin  
muls; .,.; llss 3  
tad 9f+t  
dac vp1  
  
lac ball2+vx  
gsm
```

```
    dac .+3
    lac cos
    muls; .,; llss 3
    dac 9f+t
    lac ball2+vy
    gsm
    dac .+3
    lac sin
    muls; .,; llss 3
    tad 9f+t
    dac vp2
    cma
    tad vp1
    cma
    sma
    jmp ballball i

" calculate tangential velocities

    lac ball1+vx
    gsm
    dac .+3
    lac sin
    muls; .,; llss 3
    dac 9f+t
    lac ball1+vy
    gsm
    dac .+3
    lac cos
    muls; .,; llss 3
    cma
    tad 9f+t
    cma
    dac vt1

    lac ball2+vx
    gsm
    dac .+3
    lac sin
    muls; .,; llss 3
    dac 9f+t
    lac ball2+vy
    gsm
    dac .+3
    lac cos
    muls; .,; llss 3
    cma
    tad 9f+t
    cma
    dac vt2

" recalculate x,y velocities
" with interchanged closing components
10
9   lac vp2
8   gsm
7   dac .+3
6   lac cos
5   muls; .,; llss 3
4   cma
3   dac 9f+t
```

lac vt1  
gsm  
dac .+3  
lac sin  
muls; .,; llss 3  
tad 9f+t  
cma  
dac ball1+vx

lac vp2  
gsm  
dac .+3  
lac sin  
muls; .,; llss 3  
dac 9f+t  
lac vt1  
gsm  
dac .+3  
lac cos  
muls; .,; llss 3  
tad 9f+t  
dac ball1+vy

lac vp1  
gsm  
dac .+3  
lac cos  
muls; .,; llss 3  
cma  
dac 9f+t  
lac vt2  
gsm  
dac .+3  
lac sin  
muls; .,; llss 3  
tad 9f+t  
cma  
dac ball2+vx

lac vp1  
gsm  
dac .+3  
lac sin  
muls; .,; llss 3  
dac 9f+t  
lac vt2  
gsm  
dac .+3  
lac cos  
muls; .,; llss 3  
tad 9f+t  
dac ball2+vy

11  
10       jmp ballball i  
9        t = t+1  
8  
7       dump: 0  
6  
5       lac o17  
4       sys creat; dmpname  
3       spa  
2       sys save

```
    dac waitup
    sys write; qsin; 1
    lac waitup
    sys write; qcos; 1
    lac nbailp
    cll; mul; bvsiz
    lacq
    dac Of
    lac waitup
    sys write; list; 0;,
    lac waitup
    sys close
    jmp dump i

    restore: 0
    sys open; dmpname; 0
    spa
    sys save
    dac waitup
    sys read; qsin; 1
    lac waitup
    sys read; qcos; 1
    lac nbailp
    cll; mul; bvsiz
    lacq
    dac Of
    lac waitup
    sys read; list; 0;,
    sad Ob
    skp
    sys save
    lac waitup
    sys read; dump; 1
    sza
    sys save
    lac waitup
    sys close
    jmp restore i
```

" p3  
rad = 02000  
diam = rad+rad  
diam3 = diam-0200  
middle: 0100000  
high: 0177700-010000  
low: 010000  
ballinit: 0  
-nball  
dac 9f+t

" Q ball

lac middle  
dac ball1+x  
lac low  
dac ball1+y  
dzm ball1+vx  
dzm ball1+vy  
dzm ball1+sflg  
jms put

" top row

lac d1  
dac ball1+sflg  
lac high  
dac ball1+y  
-diam-diam-diam  
tad ball1+x  
dac ball1+x  
jms put  
jms put  
jms put  
jms put  
jms put

" second row

-diam3  
tad ball1+y  
dac ball1+y  
-diam-diam=diam  
tad ball1+x  
dac ball1+x  
-diam-rad  
tad ball1+x  
dac ball1+x  
jms put  
jms put  
jms put  
jms put

" third row

-diam3  
tad ball1+y  
dac ball1+y  
-diam-diam=diam-rad  
tad ball1+x  
dac ball1+x

```
jms put  
jms put  
jms put  
  
" fourth row  
  
-diam3  
tad ball1+y  
dac ball1+y  
-diam-diam-rad  
tad ball1+x  
dac ball1+x  
jms put  
jms put  
  
" last row  
  
-diam3  
tad ball1+y  
dac ball1+y  
-diam-rad  
tad ball1+x  
dac ball1+x  
jms put  
  
jmp ballinit i  
  
put: 0  
jms putball; ball1; 9f+t  
-diam-1  
cma  
tad ball1+x  
dac ball1+x  
isz 9f+t  
jmp put i  
jmp put i  
t = t+1  
  
getball: 0  
-1  
tad getball i  
dac 8  
isz getball  
lac getball i  
dac 9f+t  
lac nbailp  
tad 9f+t i  
cl1; mul; bvsiz  
lacq  
tad listpm1  
dac 9  
-bvsiz  
dac 9f+t  
1:  
9: lac 9 i  
8: dac 8 i  
7: isz 9f+t  
6: jmp 1b  
5: isz getball  
4: jmp getball i  
3:
```

```
putball: 0
    -1
    tad putball i
    dac 8
    isz putball
    lac putball i
    dac 9f+t
    lac nballp
    tad 9f+t i
    cll; mul; bvsiz
    lacq
    tad listpm1
    dac 9
    -bvsiz
    dac 9f+t
```

```
1:
```

```
    lac 8 i
    dac 9 i
    isz 9f+t
    jmp 1b
    isz putball
    jmp putball i
    t = t+1
```

```
11
```

```
10
```

```
9
```

```
8
```

```
7
```

```
6
```

```
5
```

```
4
```

```
3
```

```
" p4  
stickcont; 0  
  
" display stick  
  
    lac ball1+x  
    lrss 6  
    xor o142000 " setx  
    dac 15 i  
    lac ball1+y  
    lrss 6  
    xor o164000 " sety  
    dac 15 i  
  
    lac ccos  
    lrss 8  
    sma  
    jmp 1f  
    cma  
    tad d1  
    xor o2000 " minus  
1:  
    xor o100000 " vecx hold  
    dac 15 i  
  
    lac qsin  
    lrss 8  
    sma  
    jmp 1f  
    cma  
    tad d1  
    xor o2000 " minus  
1:  
    xor o124000 " vecy vis  
    dac 15 i  
  
" rotation  
  
    lac .pb i  
    rtl  
    sma rar  
    jmp 2f  
  
" coarse rotation  
  
    sma  
    jmp 1f  
    szl  
    jmp 3f  
    jms rotate; mcsin; ccos  
    jmp 3f  
1:  
    snl  
    jmp 3f  
    jms rotate; csin; ccos  
    jmp 3f  
  
" fine rotation  
2:  
    sma
```

```
jmp 1f
szl
jmp 3f
jms rotate; mfsin; fcov
jmp 3f
1:
snl
jmp 3f
jms rotate; fsin; fcov
3:
" strike

lac .pb i
rtl; ral
sma ral
jmp 1f

lac qcov
dac ball1+vx
lac qsin
dac ball1+vy
jmp stickcont i

1: sma
jmp stickcont i
lac qcov
lrss 1
dac ball1+vx
lac qsin
lrss 1
dac ball1+vy
jmp stickcont i

rotate: 0
lac rotate i
dac 9f+t+1
lac 9f+t+1 i
dac 9f+t
isz rotate
lac rotate i
dac 9f+t+1
lac 9f+t+1 i
dac 9f+t+1
isz rotate

lac qsin
gsm
dac .+3
lac 9f+t+1
muls; .,; llss 2
dac 9f+t+2
lac qcov
gsm
dac .+3
lac 9f+t
muls; .,; llss 2
tad 9f+t+2
dac 9f+t+3

lac qsin
```

```
gsm  
dac ,+3  
lac 9f+t  
muls; .,; llss 2  
cma  
dac 9f+t+2  
lac qcos  
gsm  
dac ,+3  
lac 9f+t+1  
muls; .,; llss 2  
tad 9f+t+2  
dac qcoss  
lac 9f+t+3  
dac qsin  
jmp rotate i  
t = t+4
```

```
prad: 02000  
pocketlr: 0  
    lac ball1+sflg  
    sna  
    jmp 2f  
    -1  
    tad prad  
    cma  
    tad botttest  
    tad ball1+y  
    spa  
    jmp 3f  
    lac prad  
    tad toptest  
    tad ball1+y  
    sma  
    jmp 3f  
    -1  
    tad middle  
    cma  
    tad ball1+y  
    sma  
    cma  
    tad prad  
    sma  
    jmp 3f
```

```
2:  
    isz pocketlr  
    jmp pocketlr i  
3:  
    -1  
    dac ball1+sflg  
    jmp pocketlr i
```

```
pockettb: 0  
    lac ball1+sflg  
    sna  
    jmp 2f  
    -1  
    tad prad  
    cma  
    tad lefttest  
    tad ball1+x
```

spa  
jmp 3f  
lac prad  
tad righttest  
tad ball1+x  
sma  
jmp 3f  
2:  
isz pockettb  
jmp pockettb i  
3:  
-1  
dac ball1+sflg  
jmp pockettb i

11  
10  
9  
8  
7  
6  
5  
4  
3

" p5  
frfac = 0774000 " loss in vel to friction  
toptest: 02000+01000-0177700  
bottest: -02000-01000  
righttest: 02000+040400-0177700  
lefttest: -02000-040400  
minvx: -02000 " vel below whitch vel is set zero  
fsin: 0203 " sine of fine rotation angle  
csin: 02534 " sine of coarse rotation angle  
mfsin: -0203 " negative of fsin  
mcisin: -02534 " negative of csin  
fcos: 0177777 " cosine of fine rotation angle  
ccos: 0177761 " cosine of coarse rotation angle

nball = 16

d1: 1  
o17: 017  
o300000: 0300000  
dm1: -1  
o142000: 0142000  
o164000: 0164000  
o400000: 0400000  
o177777: 0177777  
o2000: 02000  
o100000: 0100000  
o124000: 0124000  
qsin: 0177777  
qcos: 0  
dmpname: <pd>;<um>;<p 040; 040040  
nballp: nball  
listpm1: list=1

outline:  
0065047  
0140000 0400  
0164000 04  
0220414  
0124000 01000-010-020  
0225010  
0221010  
0124000 0777-010-020  
0224414  
0221444  
0120000 0777-010-020  
0221404  
0224454  
0126000 0777-010-020  
0221050  
0225050  
0126000 01000-010-020  
0220454  
0225404  
0122000 0777-010-020  
0225444  
noutline = ,outline

circle:  
0212000  
0220002  
0220002



```
0220002
circsize = , -circle

.pb: , =, +1
.sw: , =, +1
stime: ., =, +1
waitup: ., =, +1
sin: , =, +1
cos: , =, +1
vp1: , =, +1
vp2: , =, +1
vt1: , =, +1
vt2: , =, +1
displist: ., =, +1
9: ., =, +t
ball1:
    sf1g = , -ball1
    , =, +1
    x = , -ball1
    , =, +1
    y = , -ball1
    , =, +1
    vx = , -ball1
    , =, +1
    vy = , -ball1
    , =, +1
bysize = , -ball1
ball2:
    , =, +bysize
list:
```

69

```
" pd

    sys open; dotdot; 0
    spa; jms error
    dac df
    law dir=1
    dac 8
1:
    lac df
    sys read; tbuf; 8
    spa; jms error
    sna
    jmp 1f
    lac tbuf
    sna
    jmp 1b
    -8
    dac c1
    law tbuf=1
    dac 9
2:
    lac 9 i
    dac 8 i
    issz c1
    jmp 2b
    jmp 1b

1:
    lac df
    sys close
    law 017
    sys creat; dotdot
    spa; jms error
    dac df
    law dir=2
    cma
    tad 8
    dac .+4
    lac df
    sys write; dir; ..
    spa; jms error
    lac df
    sys close
    sys exit

error: 0
-1
tad error
hlt
sys save

dotdot:
10      056056; 040040; 040040; 040040

c1: .=,+1
df: .=,+1
tbuf: .=,+8
dir:
```

019

```

" psych

lac d1
sys close
lac d13 "pushbuttons
sys sysloc
dac .pbp
tad d1
dac .pbp1
restart:
fld; arg1
fst; g
fld; arg2
fst; fact
fld; fp01
fst; d
jms capture
-100
dac i
fld; fp0
fst; z
fst; oldx
fst; oldy
lac o400000
dac i 11
dac i 11
lac setx
dac i 10
lac sety
dac i 10

loop:
sys time
fld; z
fad; fm110
fdv; radians
fst; b
fld; z
fad; g
fst; z
fld; g
fneg
fmp; fact
fst; g
fld; fp1
fdv; fact
fst; fact
fld; b
fad; fpid2
sin
fmp; d
fmp; fp500
fst; xx
jms in
jmp done
fld; b
sin
fmp; d
fmp; fp500
fst; yy
jms in
jmp done

```

$$g = \text{arg 1}$$

$$\text{fact} = \text{arg 2}$$

$$d = 1.0$$

$$z = 0$$

$$\text{oldx} = 0$$

$$\text{oldy} = 0$$

loop:

$$b = \frac{z + 110}{\text{radians}}$$

$$z = z + g$$

$$g = -g * \text{fact}$$

$$\text{fact} = 1/\text{fact}$$

$$xx = 500 * d * \sin(b + \pi/2)$$

in()

$$yy = 500 * d * \sin(b)$$

in()

goto loop

```
lac o400000
dac i 11
dac i 11
fld; oldx
fng
fad; xx
fix
spa
xor o775777
tad vecx
dac i 10
fld; oldy
fng
fad; yy
fix
spa
xor o775777
tad vecy
dac i 10
fld; xx
fst; oldx
fld; yy
fst; oldy
isz j
skp
jmp done
fld; a
fdv; fp90000
fad; d
fst; d
jmp loop
```

done:

```
-1000
dac 9f
sys time
isz 9f
jmp .-2
skp
```

9f:0

```
dzm char
lac auto
sza
jmp doauto
```

1:

```
sys time " swap
lac i .php
sna
jmp 1b
spa ral
jmp 1f
spa ral
jmp 2f
spa ral
jmp 3f
spa ral
jmp 4f
spa ral
jmp 5f
spa ral
jmp 6f
```

spa ral  
jmp 7f  
jms release  
sys exit

7:  
cla  
sys read; tmp; 1  
lac tmp  
sad o141000  
skp  
jmp 0f  
dac auto  
dzm i ,pbp1  
jmp doauto

0:  
sad o12000  
jmp restart  
dzm g

9:  
cla  
sys read; char; 1  
lac char  
sad o12000  
jmp 9f  
lrss 9  
dac char  
lac g  
alss 3  
tad char  
tad om60  
dac g  
jmp 9b

9:  
lac tmp  
sad o61000  
jmp 9f  
lac g  
dac arg2  
jmp restart

9:  
lac g  
dac arg1  
jmp restart

5:  
isz fp500  
nop  
jmp restart

6:  
-1  
tad fp500  
dac fp500  
jmp restart

4:  
10 lac o400000  
9 dac char

3:  
7 fld; arg1  
6 fmp; fm056  
5 lac char  
4 spa  
3 fng

fad; arg1  
fst; arg1  
jmp restart  
2:  
lac o400000  
dac char  
1:  
fld; arg2  
fmp; fm056  
lac char  
spa  
fnr  
fad; arg2  
fst; arg2  
jmp restart

doauto:  
lac i .pbp1  
sma  
jmp 1f  
dzm auto  
jmp done  
1:  
sys time  
omq  
tad 0  
tad stick  
cli  
mul  
78625  
lls 9  
dac stick  
dzm aexp  
and o177777  
xor o200000  
dac ams  
fad; fmhalf  
fmp; fp128  
lac stick  
and d1  
sna  
jmp 1f  
fst; arg1  
jmp restart  
1:  
fst; arg2  
jmp restart

in: 0  
fix  
tad d500  
spa  
jmp i in  
tad dm1000  
spa  
isz in  
jmp i in

capture: 0  
law buf=1  
dac 10

11  
10      dac 11  
10      lac o400000  
10      dac i 11  
10      law buf  
10      sys capt  
10      jmp i capture  
  
10      release: 0  
10      sys rele  
10      jmp i release  
  
10      char: 0  
10      stick: 0  
10      xx: 0; 0; 0  
10      oldx: 0; 0; 0  
10      yy: 0; 0; 0  
10      oldy: 0; 0; 0  
10      j: 0  
10      auto: 0  
  
10      setx: 0142000 +512  
10      sety: 0146000 +512  
10      vecx: 0100000  
10      vecy: 0124000  
10      o775777: 0775777  
10      o177777: 0177777  
10      o12000: 012000  
10      o61000: 061000  
10      om60: -060  
10      d13: 13  
10      d500: 500  
10      o141000: 0141000  
10      fp128: 0240000; 0  
10      fmhalf: 0; 0600000; 0  
10      dm1000: -1000  
  
10      fact: 0;0;0  
10      g: 0;0;0  
10      z: 0;0;0  
10      b: 0;0;0  
10      d: 0;0;0  
  
10      radians: 6;0345136;0  
10      fm110: 7;0734000;0  
10      arg1: 7;0234167;0  
10      arg2: 1;0275531;0  
10      fp0: 0;0;0  
10      fp01: -6;0243656;0  
10      fm056: -4;0745301;0  
10      fp500: 11;0372000;0  
10      fp90000: 021;0257620;0  
10      ,ppp: .+.+1  
10      ,ppp1: 0  
10      buf:  
9  
8  
7  
6  
5  
4  
3

113

" rm  
lac 017777  
tad d1  
dac 2f  
1:  
lac 017777 i  
sad d4  
sys exit  
tad dm4  
dac 017777 i  
lac 2f  
tad d4  
dac 2f  
sys unlink; 2: 0  
sma  
jmp 1b  
lac 2b  
dac 2f  
lac d1  
sys write; 2: 0; 4  
lac d1  
sys write; 1f; 2f-1f  
jmp 1b

1:  
040077;012000

2:

d1: 1  
d4: 4  
dm4: -4



" rn  
lac 017777  
tad d1  
dac name2  
loop:  
lac 017777 i  
sad d4  
sys exit  
sad d8  
jmp unbal  
tad dm8  
dac 017777 i  
lac name2  
tad d4  
dac name1  
tad d4  
dac name2  
sys unlink; name2; 0  
lac name2  
dac 1f  
sys rename; name1; 0; 1; 0  
sma  
jmp loop  
lac name1  
dac 1f  
lac d1  
sys write; 1; 0; 4  
lac d1  
sys write; mes; 1  
lac name2  
dac 1f  
lac d1  
sys write; 1; 0; 4  
lac d1  
sys write; mes; 2  
jmp loop

mes:  
040000;077012

unbal:  
lac name2  
tad d4  
dac 1f  
lac d1  
sys write; 1; 0; 4  
lac d1  
sys write; mes; 2  
sys exit

d1; 1

d4; 4

d8; 8

dm8; -8

9

8

7

6

5

4

3



```
" roff

    lac i 017777
    sad d4
    sys exit
    lac 017777
    tad d1
    dac fname
    jms nextfile
    lac obufp
    dac otal

" main i/o loop
1:
    jms readline
    jmp 3f
    lac rawchar
    sad cc
    jmp 2f
    jms text
    jmp 1b

2:
    jms control
    jmp 1b

3:
    jms break
    jms eject
    lac otal
    sma
    jmp 1f
    cla
    jms putsc; otal

1:
    -1
    tad obufp
    cma
    tad otal
    dac 1f
    lac output
    sys write; obuf; 1: 0
    sys exit

" read line routine
readline: 0
    law rawchar-1
    dac 8

1:
    jms getchar
    dac i 8
    sad o12
    skp
    jmp 1b
    isz readline
    jmp i readline

10
9   " read character routine
8   getchar: 0
7   lac ital
6   sad eibufp
5   skp
4   jmp 1f
3   -64
```

```
dac 3f  
law ibuf-1  
dac 15  
2:  
dzm i 15  
isz 3f  
jmp 2b  
lac input  
sys read; ibuf; 64  
sna  
jms nextfile  
lac ibufp  
dac ital  
1:  
jms getsc; ital  
sza  
jmp i getchar  
jmp getchar+1  
3: 0
```

```
putchar: 0  
jms putsc; otal  
lac otal  
sad eobufp  
skp  
jmp i putchar  
lac output  
sys write; obuf; 64  
lac obufp  
dac otal  
jmp i putchar
```

```
laci: 0  
dac 1f  
lac i 1f  
jmp i laci  
1: 0
```

```
nextfile: 0  
lac i 017777  
sad d4  
jmp i readline  
tad dm4  
dac i 017777  
lac fname  
tad d4  
dac fname  
lac input  
sys close  
sys open; fname: 0; 0  
sma  
jmp 1f  
lac fname  
dac 2f  
lac d1  
sys write; 2: 0; 4  
lac d1  
sys write; 2f; 2  
sys exit  
2: 040077;012  
1:
```

```
    dac input
    lac eibufp
    dac ital
    jmp i nextfile

    getsc: 0
    lac i getsc
    dac sctalp
    isz getsc
    lac i sctalp
    dac sctal
    add o400000
    dac i sctalp
    ral
    lac i sctal
    szl
    lrss 9
    and o177
    jmp i getsc

    putsc: 0
    and o177
    lmq
    lac i putsc
    dac sctalp
    isz putsc
    lac i sctalp
    dac sctal
    add o400000
    dac i sctalp
    sma cla
    jmp 1f
    llss 27
    dac i sctal
    lrss 9
    jmp i putsc
    1:
    lac i sctal
    omq
    dac i sctal
    lacq
    jmp i putsc

    " control card decoder
control: 0
    law 2f-1
    dac 8
    -ncase
    dac c
    lac rawchar+1
    alss 9
    xor rawchar+2
    1:
    10:   sad i 8
    9:    jmp i 8
    8:    isz 8
    7:    isz c
    6:    jmp 1b
    5:    jmp i control
    4:
    3:    ncase = 0
```

```
<ad>; jmp casead; ncase = ncase+1
<bp>; jmp casebp; ncase = ncase+1
<br>; jmp casebr; ncase = ncase+1
<cc>; jmp casecc; ncase = ncase+1
<ce>; jmp casece; ncase = ncase+1
<ds>; jmp caseds; ncase = ncase+1
<fi>; jmp casefi; ncase = ncase+1
<in>; jmp casein; ncase = ncase+1
<li>; jmp caseli; ncase = ncase+1
<l1>; jmp casell; ncase = ncase+1
<ls>; jmp casels; ncase = ncase+1
<na>; jmp casena; ncase = ncase+1
<ne>; jmp casene; ncase = ncase+1
<nf>; jmp casenf; ncase = ncase+1
<pl>; jmp casepl; ncase = ncase+1
<sp>; jmp casesp; ncase = ncase+1
<ss>; jmp casess; ncase = ncase+1
<ti>; jmp caseti; ncase = ncase+1
<ul>; jmp caseul; ncase = ncase+1
<un>; jmp caseun; ncase = ncase+1
```

" control cases

```
casead:
    jms break
    =1
    dac ad
    jmp i control
```

```
casebp:
    jms break
    jms eject
    jmp i control
```

```
casebr:
    jms break
    jmp i control
```

```
casecc:
    jms skipcont
    lac i 8
    sad o12
    jmp i control
    dac cc
    jmp i control
```

```
casece:
    jms break
    jms number; d0
    spa
    cla
    dac ce
    jms need; ce
    jmp i control
```

```
10
9   caseds:
8     jms break
7     lac d2
6     dac ls
5     jmp i control
4
3   casefi:
```

jms break  
-1  
dac fi  
jmp i control

casein:  
jms number; in  
cma  
tad d1  
sma  
cla  
dac in  
dac un  
jmp i control

caseli:  
jms number; d0  
cma  
tad d1  
sma  
jmp i control  
dac 2f

1:  
jms readline  
jmp i control  
jms text  
isz 2f  
jmp 1b  
jmp i control

2: 0

casell:  
jms number; l1  
spa  
cla  
dac l1  
jmp i control

casels:  
jms number; d0  
sza; spa  
lac d1  
dac ls  
jmp i control

casena:  
jms break  
dzm ad  
jmp i control

casene:  
jms number; d0  
spa  
cla  
dac c  
jms need; c  
jmp i control

casenf:  
jms break  
dzm fi

```
    jmp i control

casepl:
    jms number; pl
    spa
    cla
    dac pl
    jms topbot
    jmp i control

casesp:
    jms break
    jms number; d0
    cma
    tad d1
    sma
    jmp i control
    dac c
1:
    jms nline
    isz c
    jmp 1b
    jmp i control

casess:
    jms break
    lac d1
    dac ls
    jmp i control

caseti:
    jms break
    jms number; in
    spa
    cla
    dac un
    jmp i control

caseul:
    jms number; d0
    spa
    cla
    dac ul
    jmp i control

caseun:
    jms number; d0
    tad in
    sma
    cla
    dac un
    jmp i control

11
10 " selected short routines
skipcont: 0
9     law rawchar-1
8     dac 8
7
6 1:
5     lac i 8
4     sad o40
3     jmp 1f
```

```
sad o12
jmp 2f
jmp 1b
1:
lac i 8
sad o12
jmp 2f
sad o40
jmp 1b
2:
-1
tad 8
dac 8
jmp i skipcont

break: 0
lac nc
sna
jmp i break
-2
tad ls
cma
sma
jmp 2f
dac c
1:
jms nline
isz c
jmp 1b
2:
lac nl
sad bl
jms eject
lac nl
sza
jmp 2f
-5
dac c
1:
lac o55
jms putchar
isz c
jmp 1b
lac ma1
dac c
1:
jms newline
isz c
jmp 1b
2:
law char-1
dac 8
lac un
sma
jmp 1f
dac c
2:
lac o40
jms putchar
isz c
jmp 2b
```

*-10x2*

*2(1-f)*

1:  
lac i 8  
jms putchar  
isz nc  
jmp 1b  
jms newline  
dzm nwd  
dzm ne  
lac in  
dac un  
jmp i break

newline: 0  
lac o12  
jms putchar  
isz nl  
jmp i newline

nline: 0  
lac nl  
sna  
jmp i nline  
sad bl  
jmp i nline  
jms newline  
jmp i nline

number: 0  
dzm num  
dzm sign  
-1  
dac any  
jms skipcont

1:  
lac i 8  
sad o12  
jmp 3f  
sad o53  
jmp 2f  
sad o55  
jmp 2f  
tad om72  
sma  
jmp 1b  
tad o12  
spa  
jmp 1b  
dac any  
lac num  
cl1; mul; 10  
lacq  
tad any  
dac num  
jmp 1b

2:  
dac sign  
jmp 1b

3:  
lac any  
sma  
jmp 1f

```
lac d1
isz number
jmp i number
1:
lac sign
sza
jmp 1f
lac num
isz number
jmp i number
1:
sad o53
jmp 1f
lac i number
jms laci
cma
tad num
cma
isz number
jmp i number
1:
lac i number
jms laci
tad num
isz number
jmp i number
```

```
eject: 0
lac pl
sna
jmp i eject
lac nl
sna
jmp i eject
1:
sad pl
jmp 1f
jms newline
lac nl
jmp 1b
1:
dzm nl
jmp i eject
```

```
storechar: 0
lmc
lac nc
sza
jmp 1f
law char-1
dac 10
1:
```

```
11
10 lacq
9 dac i 10
8 jms width
7 cma
6 tad d1
5 tad ne
4 dac ne
3 -1
2 tad nc
```

dac nc  
jmp i storechar

getword: 0  
law word-1  
dac 8

1:  
dzm wne  
dzm wch  
lac i 11  
sad o12  
jmp i getword.

2:  
dac i 8  
lmc  
jms width  
cma  
tad d1  
tad wne  
dac wne  
-1  
tad wch  
dac wch  
lacq  
sad o40  
jmp 1b  
lac word  
sad o40  
jmp 1f  
lac o40  
dac word  
lacq  
jmp 2b

Sp [ word ]

1:  
lac i 11  
sad o12  
jmp 1f  
sad o40  
jmp 1f  
dac i 8  
jms width  
cma  
tad d1  
tad wne  
dac wne  
-1  
tad wch  
dac wch  
jmp 1b

1:  
-1  
tad 11  
dac 11  
isz getword  
jmp i getword

need: 0  
lac ls  
dac 1f  
lac i need  
jms laci

cli; mul; 1: 0  
lacq  
tad nl  
cma  
tad bl  
spa  
jms eject  
isz need  
jmp i need

" text line routine  
text: 0  
-1  
tad ul  
sma  
jms undline  
-1  
tad ce  
sma  
jms center  
law rawchar-1  
dac 11  
lac rawchar  
sad o12  
jmp 1f-1  
sad o40  
jmp 1f-1  
lac fi  
sza  
jmp 2f  
skp  
jms break  
1:  
lac i 11  
sad o12  
jmp 1f  
jms storechar  
jmp 1b  
1:  
lac nc  
sna  
jms nline  
jms break  
jmp i text  
2:  
jms getword; jmp i text  
lac wne  
tad ne  
tad un  
tad ll  
spa  
jms adjust  
law word-1  
dac 8  
lac nw8  
sza  
jmp 3f  
1:  
lac i 8  
sad o40  
skp

```
jmp 3f+1
isz wch
jmp 1b
3:
lac i 8
jms storechar
isz wch
jmp 3b
isz nwd
jmp 2b

" adjust routine
adjust: 0
lac nwd
sna
jmp i adjust ↗
law char-1
dac 8
law tchar-1
dac 9
dzm ndiv
dzm nrem
lac ad
sna
jmp 1f
-1
tad nwd
sna
jmp 1f
dac 2f
lac 11
tad ne
tad un
spa
jmp 1f
cll; idiv; 2: 0
dac nrem
lacq
dac ndiv
1:
lac i 8
sad o40
jms fill
dac i 9
isz nc
jmp 1b
lac o12
dac i 9
law tchar-1
dac 8
2:
lac i 8
sad o12
jmp 2f
jms storechar
jmp 2b
2:
jms break
jmp i adjust
fill: 0
```

lac nrem  
sna  
jmp 2f  
tad dm1  
dac nrem.  
lac d1

2:  
tad ndiv  
cma  
dac c  
lac o40  
2:  
dac i 9  
isz c  
jmp 2b  
2:

isz nc  
lac i 8  
sad o40  
skp  
jmp i fill  
dac i 9  
jmp 2b

" more routines

topbot: 0  
lac pl  
sza  
jmp 1f  
azm bl  
jmp i topbot

1:  
-11  
tad pl  
spa  
jmp 1f  
dac bl  
cma  
tad d1  
tad nl  
spa  
jmp i topbot  
lac bl  
dac nl  
jmp i topbot

1:  
lac d55  
dac bl  
dac nl  
tad d11  
dac pl  
jmp i topbot

10 undline: 0  
9 dac ul  
8 law rawchar-1  
7 dac 8  
6 law tchar-1  
5 dac 9  
4 1:  
3 lac i 8

mov ag, ro  
breq 1f  
inc ac  
mov \$1, ro  
||  
add mg, ro  
add \$1, ro  
mov ro, -(sp)

not if int small

```
    dac i 9
    sad o12
    jmp 1f
    sad o40
    jmp 1b
    lac o10
    dac i 9
    lac o137
    dac i 9
    jmp 1b
1:
    law tchar-1
    dac 8
    law rawchar-1
    dac 9
1:
    lac i 8
    dac i 9
    sad o12
    jmp i undline
    jmp 1b
center: 0
    dac ce
    law rawchar-1
    dac 8
    law tchar-1
    dac 9
    dzm wne
1:
    lac i 8
    dac i 9
    sad o12
    jmp 1f
    jms width
    tad wne
    dac wne
    jmp 1b
1:
    -1
    tad wne
    spa
    jmp i center
    cma
    tad 11
    tad in
    lrss 1
    cma
    tad d1
    sma
    -1
    dac c
    law tchar-1
    dac 8
    law rawchar-1
    dac 9
    lac o40
1:
    dac i 9
    isz c
    jmp 1b
```

1:  
  lac i 8  
  dac i 9  
  sad o12  
  jmp i center  
  jmp 1b  
  
width: 0  
  sad o10  
  jmp 1f  
  lac d1  
  jmp i width  
1:  
  -1  
  jmp i width

eibufp: ibuf+64  
ibufp: ibuf  
eobufp: obuf+64  
obufp: obuf  
input: 0  
output: 1  
ls: 1  
ce: 0  
in: 0  
un: 0  
ul: 0  
ma1: -5  
bl: 55  
ll: 50  
nwd: 0  
nl: 0  
nc: 0  
ne: 0  
pl: 66  
ad: -1  
fi: -1  
cc: ,>

o12: 012  
o40: 040  
o177: 0177  
o53: 053  
om72: -072  
o55: 055  
o400000: 0400000  
o10: 010  
d11: 11  
d55: 55  
o137: 0137  
d1: 1  
d2: 2  
d4: 4  
dm4: -4  
d0: 0  
dm1: -1

c: ,=,+1  
nrem: ,=,+1  
ndiv: ,=,+1  
num: ,=,+1

any: .=.,+1  
ital1: .=.,+1  
otali: .=.,+1  
sctal: .=.,+1  
sctalp: .=.,+1  
sign: .=.,+1  
wch: .=.,+1  
wne: .=.,+1  
word: .=.,+300  
char: .=.,+300  
tchar: .=.,+300  
rawchar: .=.,+300  
ibuf: .=.,+64  
obuf: .=.,+64

14

" salv

lac d1  
sys sysloc  
dac iget

lac d2  
sys sysloc  
dac inode

lac d4  
sys sysloc  
dac nxbblk  
tad d1  
dac nfbblk  
tad d1  
dac fblk

lac d5  
sys sysloc  
dac copy

lac d6  
sys sysloc  
dac copyz

lac d7  
sys sysloc  
dac between

lac d8  
sys sysloc  
dac dskrd

lac d10  
sys sysloc  
dac dskbuf

lac d15  
sys sysloc  
dac free

dzm indircnt  
dzm icnt  
dzm licnt  
dzm blcnt  
dzm curi  
jms copyz i; usetab; 500

iloop:  
isz curi  
-3400  
tad curi  
Sma  
jmp part2  
lac curi  
jms iget i  
jms copy i; inode: 0; linode; 12  
lac iflags  
Sma  
jmp iloop  
isz icnt

lac iflags  
and 040  
sza  
jmp iloop  
law idskps  
dac t1  
-7  
dac t2

1:  
lac i t1  
sza  
jms dupcheck  
isz t1  
isz t2  
jmp 1b  
lac iflags  
and 0200000  
sna  
jmp iloop

isz licnt  
law idskps  
dac t1  
-7  
dac t2

1:  
lac i t1  
sna  
jmp 3f  
jms dskrd i  
jms copy i; dskbuf; 0; ldkbuf; 64  
isz indircnt  
law ldkbuf  
dac t3  
-64  
dac t4

2:  
lac i t3  
sza  
jms dupcheck  
isz t3  
isz t4  
jmp 2b

3:  
isz t1  
isz t2  
jmp 1b  
jmp iloop

dupcheck; 0  
isz blcnt  
jms betwen i; d709; d6400  
jmp badadr  
dac t5  
lrss 4  
tad usetabp  
dac t6  
cla  
llss 4  
tad alsscom  
dac 2f

```
    lac d1
2: alss 0
    dac bit
    lac i t6
    and bit
    sza
    jmp dup
    lac i t6
    xor bit
    dac i t6
    jmp i dupcheck

badadr:
    jms print
    lac d1
    sys write; badmes; 3
    jmp i dupcheck
badmes:
    < b>;<ad>;<r 012

dup:
    lac t5
    jms print
    lac d1
    sys write; dupmes; 3
    jmp i dupcheck
dupmes:
    < d>;<up>;<a 012

print: 0
    lmg
    law prbuf=1
    dac 8
    -6
    dac t6
1:
    cla
    llss 3
    tad o60
    dac i 8
    isz t6
    jmp 1b
    lac d1
    sys write; prbuf; 6
    jmp i print

part2:
    lac licnt
    jms print
    lac d1
    sys write; m3; m3s
    lac licnt
    jms print
    lac d1
    sys write; m4; m4s
    lac indircnt
    jms print
    lac d1
    sys write; m5; m5s
    lac blcnt
    jms print
```

```
lac d1
sys write; m6; m6s
dzm blcnt
dzm nxfbblk i
dzm nfbblk i

lac d709
dac t1
1:
isz t1
lac t1
sad d6400
jmp part3
lrss 4
tad usetabp
dac t2
cla
llss 4
tad alsscom
dac 2f
lac d1
2: alss 0
dac bit
lac i t2
and bit
sza
jmp 1b
lac t1
jms free i
isz blcnt
jmp 1b

part3:
lac blcnt
jms print
lac d1
sys write; m7; m7s
sys exit

d1: 1
d2: 2
d4: 4
d5: 5
d6: 6
d7: 7
d8: 8
d10: 10
d15: 15
o60: 060
o4000000: 04000000
o400001: 0400001
o40: 040
o200000: 0200000
alsscom: alss 0
d709: 709
d6400: 6400

m3:
        040; <fi>; <le>; <s 012
m3s = ,m3
m4:
```

```
    040;<la>;<rg>;<e 012  
m4s = ,=m4  
m5:  
    040;<in>;<di>;<r 012  
m5s = ,=m5  
m6:  
    040;<us>;<ed>;012  
m6s = ,=m6  
m7:  
    040;<fr>;<ee>;012  
m7s = ,=m7  
m8:  
    040;<mi>;<ss>;<in>;<g 012  
m8s = ,=m8
```

```
usetabp: usetab  
curi: 0  
bit: 0  
blknt: 0  
indirent: 0  
icnt: 0  
licnt: 0  
t1: 0  
t2: 0  
t3: 0  
t4: 0  
t5: 0  
t6: 0
```

```
iget: 0  
nxfblik: 0  
nfbiks: 0  
fblks: 0  
copy: 0  
copyz: 0  
betwen: 0  
dskrdt: 0  
free: 0
```

```
laskbuf: ,=,+64  
linode: ,=,+12  
iflags = linode  
laskps = iflags+1  
usetabi: ,=,+500  
prbuf: ,=,+6
```

115

```
" sh  
clear:  
    jmp shell  
  
1:  
    dzm i 8  
    isz clear  
    jmp 1b  
    lacq  
    jmp 017771  
zerop: .=1  
  
comerr:  
    lac d1  
    sys write; errmes; 1  
  
shell:  
    lac d1  
    sys write; ready; 1  
shell1:  
    lac delim  
    sad newln  
    jms rline  
    jms getcom  
    lac narg  
    sna  
    jmp comretrn  
    lac args  
    sad chcom  
    skp  
    jmp 3f  
    lac args+1  
    sad spsp  
    jmp doch  
  
3:  
    sys fork  
        skp  
        jmp loadcom  
    spa  
    jmp comerr  
    lmg  
    lac delim  
    sad amper  
    jmp shell1  
    lacq  
    clq  
    sys smes  
comretrn:  
    lac delim  
    sad newln  
    jmp shell  
    jmp shell1  
  
loadcom:  
    sys open; args; 0  
    sma  
    jmp 1f  
    sys link; system; args; args  
    spa
```

parent "while"  
new to child

```
jmp 2f
-1
dac lnkflg
jmp loadcom

2:
lac lnkflg
sna
jmp 3f
sys unlink; args

3:
lac d1
sys write; args; 4
lac d1
sys write; errmes; 1
sys exit

1:
lac lnkflg
sna
jmp 1f
sys unlink; args

1:
lac in
sna
jmp 2f
cla
sys close
sys open; in; 0
sma
jmp 2f
lac d1
sys write; in; 4
lac d1
sys write; errmes; 1
sys exit

2:
lac out
sna
jmp 1f
lac d1
sys close
lac o17
sys creat; out
spa
sys exit

1:
lac narg
cma
dac t1
tad o17771
dac 017777
tad dm1
dac 8
and o7777
dac boot+2
cma
tad d7
dac clear
lac naesp
dac 9
```

2:  
lac i 9  
dac i 8  
isz t1  
jmp 2b  
lac bootp  
dac 9  
-6  
dac t1  
2:  
lac i 9  
dac i 8  
isz t1  
jmp 2b  
lac d2  
lmg  
lac zeroop  
dac 8  
jmp clear+1

boot:  
sys read 4096 ..  
lacq  
sys close  
jmp 4096

getcom: 0  
law args=1  
dac 10  
dzm in  
dzm out  
dzm narg  
dzm lnkflg

jms get  
nparm:  
sad gr  
jmp cgr  
sad ls  
jmp cls  
sad amper  
jmp endcom  
sad semic  
jmp endcom  
sad newln  
jmp endcom  
sad space  
jmp nparm-1  
lmg  
lac narg  
tad d4  
dac narg  
lacq  
jms getparm  
jmp nparm

endcom:  
dac delim  
jmp getcom i

cls:

```
jms get
jms getparm
dac t1
law in=1
jmp cpio
cgr:
jms get
jms getparm
dac t1
law out=1
cpio:
das 11
lac 10
tad dm4
dac 10
lmg
-4
dac c1
1:
lac i 10
dac i 11
isz c1
jmp 1b
lacq
dac 10
lac t1
jmp nparm

getparm:0
lmg
-8
dac c1
lacq
skip
1:
jms get
sad space
jmp 1b
jms checkd1m
jmp comerr
jmp 2f
1:
jms get
jms checkd1m
jmp fill1
2:
alss 9
isz c1
lmg
jms get
jms checkd1m
jmp fill
omg
dac i 10
isz c1
jmp 1b
1:
jms get
jms checkd1m
jmp i getparm
jmp 1b
```

fill:  
dac t1  
lac space  
omq  
dac i 10  
isz c1  
nop  
lac t1  
fill1:  
lmg  
lac c1  
spa  
jmp 1f  
laeq  
jmp i getparm

1:  
lac spsp  
dac i 10  
isz c1  
isz c1  
jmp 1b  
laeq  
jmp i getparm

checkdlim:0  
sad space  
jmp i checkdlim  
sad newln  
jmp i checkdlim  
sad amper  
jmp i checkdlim  
sad semic  
jmp i checkdlim  
isz checkdlim  
jmp i checkdlim

get: 0  
lac i 8  
sad slash  
skp  
jmp i get  
lac i 8  
sad newln  
skp  
jmp comerr  
laeq  
dac 1f  
jms rline  
lac 1f  
lmg  
lac space  
jmp i get

9  
8 1: 0  
rline:0  
7  
6 2:  
5 law lineb-1  
4 dac 15  
3 dac 8  
1: jms getcha

```
    dac i 15
    sad newln
    jmp i rline
    sad sharp
    jmp psharp
    sad atsign
    jmp 2b
    jmp 1b
    psharp:
        -1
        tad 15
        sad 2b
        jmp 2b
        tad d1
        dac 15
        jmp 1b

    getcha:
        lac char
        dzm char
        sza
        jmp i getcha
        iss nread
        jmp 1f
        cla
        sys read$ inbuf; 64
        spa sna
        jmp lgeut
        cma
        tad d1
        dac nread
        lav inbuf=1
        dac 14
    1:
        lac i 14
        lmq
        and o777
        dac char
        ecia liss 9
        jmp i getcha

    doch:
        lac naug
        liss 2
        cma
        tad d1
        dac naug
        lav args+4
        dac 1f
    2:
        iss naug
        skp
        jmp comreturn
        sys chdir; 1:0
        spa
        jmp cherr
        lac 1b
        tad d4
        dac 1b
        jmp 2b
    cherr:
```

```
lac 1b
dac ,+3
lac d1
sys write; ,; 4
jmp comerr

lgout:
    clq
    lac d1
    sys smes
    sys exit

d1: 1
dm1: -1
d4: 4
dm4: -4
d2: 2
d7: 7
o17: 017
o17771: 017771
o7777: 07777
o777: 0777
gr: 076
ls: 074
ampers: &>
semic: 073
space: 040
sharp: 043
atsign: 0100
newint: 012
slash: 057
in: 0;0;0
out: 0;0;0
errmes: 077012
chcom: <ch>
ready: 0100040
delim: 012
system: <sy>; <st>; <em>; spsp: 040040
nargp: narg=1
bootp: boot=1
char: 0
nread: -1
linebt: ,=,+128
inbuf: ,=,+64
c1: ,=,+1
t1: ,=,+1
lnkflg: ,=,+1
narg: ,=,+1
args:
```

bill

617

" space travel 1

t = 0

start:  
law 13  
sys sysloc  
dac .pbson  
lac pvv  
dac 1f  
dac 2f  
lac pw  
dac 3f  
-32  
dac cplan

4:  
fld; 1:0  
fmp; 2:0  
=1  
tad aexp  
dac aexp  
fng  
fad; fpone  
fst; 3:0  
lac 1b  
tad d3  
dac 1b  
dac 2b  
lac 3b  
tad d3  
dac 3b  
isz cplan  
jmp 4b  
law dspl-1  
dac clistp  
dac lanflg  
dzm erflg  
dzm goflg  
dzm ,pbsint  
dzm forflg  
dzm bacflg  
dzm dspflg  
dzm locflg  
dzm locpar  
lac d1  
dac par  
jms dispname  
fld; prsq+4  
sqrt  
fst; rpar  
fst; y  
fst; oy  
fld; fpone  
fst; sphi  
fst; stheta  
jms dpsc  
fld; fzzero  
fst; x  
fst; ox  
fst; cphi  
fst; ctheta  
law dispist

stuff for gravity

```
sys capt
jmp loop

loop:
    law dspl-1
    dac clistp
    jms contrl1
    lac par
    jms absxy
    jms shipxy
    lac goflg
    spa
    jmp loop3 —
    fld; fpzero
    fst; ax
    fst; ay
    fst; maxa
    lac nplan
    skp
loop1:
    lac cplan
    tad dm1
    spa
    jmp loop2
    dac cplan
    tad fppar
    dac fcplan
    lac i fcplan
    dac fcplan
    jms updacc
    jms displa
    lac cplan
    sza
    jms updpln
    lac cplan
    and o7
    sad o7
    jms contrl
    jmp loop1

loop2:
    lac lanflg
    sma
    jms updshp
loop3:
    fld; horizv
    lac scale
    cma
    tad vscale
    tad aexp
    dac aexp
    jms inscr
    jmp loop4
    tad o141577
    dac i clistp
    lac o164972
    dac clistp i
    cla
    jms dispplanet
Loop4:
    sys time "put delay here.....
```

→ dac ~~clistp~~ dsvel

dzm dispcl  
lac crflg  
sma  
jmp 1f  
lac o20714 "cl  
dac dispcl  
jmp 2f

1:  
lac lanflg  
sma  
jmp 2f  
lac o114 "l  
dac dispcl

2:  
lac dhalt  
dac i clistp  
jmp loop " check 2-display question

contr1: 0  
lac i .pbson  
xor .pbsint  
and .pbson i  
sna  
jmp noneon  
lmq  
spa ral  
sys exit  
sma  
jmp 1f  
dzm goflg  
dzm crflg

1:  
lacq  
als 6  
sma ral  
jmp 1f  
spa  
jmp noneon  
isz scale  
nop  
jms dpscra "uprange  
jmp noneon

1:  
sma  
jmp noneon  
=1  
tad scale  
dac scale  
jms dpscra "downrange

noneon:  
dzm forflg  
dzm bacflg  
lac i .pbson  
dac .pbsint  
als 2  
sma  
jmp 1f  
lac dhalt  
dac forflg  
lac goflg  
sma

dzm lanflg

1:  
lac i '.pbson  
als 3  
sma  
jmp 1f  
lac dhalt  
dac bacflg  
lac goflg  
sma  
dzm lanflg

1:  
lac i '.pbson  
als 4  
sma  
jmp 1f  
ral  
spa  
jmp i contrl  
dzm 9f+t  
jmp 2f

1:  
ral  
sma  
jmp i contrl  
lac dhalt  
dac 9f+t

2:  
fld; cphi  
fmp; sdphi  
lac 9f+t  
sma  
fnq  
fst; fttmp1  
fld; sphi  
fmp; cdphi  
fad; fttmp1  
fst; fttmp2  
fld; sphi  
fmp; sdphi  
lac 9f+t  
spa  
fnq  
fst; fttmp1  
fld; cphi  
fmp; cdphi  
fad; fttmp1  
fst; cphi  
fld; fttmp2  
fst; sphi  
jmp i contrl

t = t+1

shipxy; 0  
fld; absx  
fad; x  
fnq  
fst; shipx  
fld; absy  
fad; y

fng  
fst; shipy  
jmp i shipxy

" space travel 2

absv: 0  
dzm absx  
dzm absx+1  
dzm absx+2  
dzm absy  
dzm absy+1  
dzm absy+2

1:  
dac absi  
sna  
jmp i absv  
tad fppar  
dac 9f+t  
jms invert  
fld; ftmp1  
fng  
fad; absx  
fst; absx  
fld; ftmp2  
fng  
fad; absy  
fst; absy  
jms invert  
fld; ftmp1  
fad; absx  
fst; absx  
fld; ftmp2  
fad; absy  
fst; absy  
lac absi  
tad ppar  
dac 9f+t  
lac i 9f+t  
jmp 1b

invert: 0  
lac i 9f+t  
dac fcplan  
tad pww  
dac 1f  
dac 2f  
iac fcplan  
tad px  
dac 3f  
iac fcplan  
tad py  
dac 4f  
fld; 1::.  
fng  
fst; 2::.  
jms updpln  
fld; 3::.  
fst; ftmp1  
fld; 4::.  
fst; ftmp2  
jmp i invert

t = t+1

updpln: 0

lac cplan  
lmc  
tad px  
dac 1f  
dac 5f  
lacq  
tad py  
dac 3f  
dac 6f  
dac 0f  
lacq  
tad pw  
dac 2f  
dac 7f  
lacq  
tad pww  
dac 4f  
dac 8f

fld; 1...  
fst; ftmp1  
fmp; 2...  
fst; ftmp2  
fld; 3...  
fmp; 4...  
fng  
fad; ftmp2  
fst; 5...  
fld; 6...  
fmp; 7...  
fst; ftmp2  
fld; ftmp1  
fmp; 8...  
fad; ftmp2  
fst; 0...  
jmp updp1n i

updacc; 0  
lac cplan  
sad par  
jmp upda2  
jms absxy  
fld; absx  
fad; shlpz  
jmp 1f  
upda2:  
fld; x  
fng  
1:  
fst; absx  
fmp; absx  
fst; ftmp1  
lac cplan  
sad par  
jmp 1f  
fld; absy  
fad; shlpz  
jmp 2f  
1:  
fld; y  
fng

2:  
fst; absy  
fmp; absy  
fad; fttmp1  
fst; dttmp1  
sqrt  
fst; dpar  
lac cplan  
sad par  
skp  
jmp upda5  
fld; ox  
fnq  
fad; x  
fst; fttmp1  
fmp; y  
fst; horizv  
fld; y  
fnq  
fad; oy  
fst; fttmp2  
fmp; x  
fad; horizv  
fdv; dpar  
fst; horizv  
fld; dpar  
fcv; rpar  
sma  
jmp upda5  
lac lanflg  
spa  
jmp upda5  
fld; fttmp1  
fmp; fttmp1  
fst; fttmp1  
fld; fttmp2  
fmp; fttmp2  
fad; fttmp1  
fcv; crash  
spa  
jmp 1f  
lac dhalt  
dac qoflg  
dac crflg

1:  
lac dhalt  
dac lanflg  
fld; rpar  
fdv; dpar  
fst; fttmp1  
fmp; x  
fst; x  
fst; ox  
fld; fttmp1  
fmp; y  
fst; y  
fst; oy  
lac par  
jms absxy  
jms shpxy  
jmp upda2

upda5:  
  fcp; fardst  
  spa  
  jmp 1f  
  lac cplan  
  sna  
  jmp 1f  
  lac dhalt  
  dac grvflg  
  jmp i updacc

1:  
  dzm grvflg  
  lac fcplan  
  tad accol  
  dac 1f  
  fld; 1...  
  fdv; dtmp1  
  fcp; maxa  
  spa  
  jmp 2f  
  fst; maxa  
  lac cplan  
  dac maxj

2:  
  fdv; dpar  
  fst; ftmp1  
  fmp; absx  
  fad; ax  
  fst; ax  
  fld; ftmp1  
  fmp; absy  
  fad; ay  
  fst; ay  
  jmp i updacc

" space travel 3

```
updshp: 0
lac forflg
spa
jmp .+4
lac bacflg
sma
jmp 3f
fld; ascale
lac forflg
sma
jmp 1f
lac bacflg
sma
jmp 1f+1
fld; fzzero
jmp 2f
```

1:

```
fng
lac scale
tad aexp
dac aexp
lac forflg
sma
jmp .+3
```

2:

```
lac accflg
sma
jmp .+3
fad; maxa
fng
fst; ftimp1
fmp; ctheta
fad; ax
fst; ax
fld; ftimp1
fmp; sttheta
fad; ay
fst; ay
```

3:

```
fld; ox
fng
fad; ax
fst; ftimp1
fld; x
lac aexp
tad d1
dac aexp
fad; ftimp1
fst; ftimp1
fld; x
fst; ox
fld; ftimp1
fst; x
fld; oy
fng
fad; ay
fst; ftimp1
fld; y
lac aexp
tad d1
```

howto set?

```

dac aexp
fad; ftmp1
fst; ftmp1
fld; y
fst; oy
fld; ftmp1
fst; y
lac par
sad maxj
jmp i updshp
jms absxy
jms shippy
lac par
jms absy
fld; ox
fng
fad; x
fad; absx
fst; ox
fld; oy
fng
fad; y
fad; absy
fst; oy
lac maxj
dac par
jms absy
fld; ox
fng
fad; absx
fst; ox
fld; oy
fng
fad; absy
fst; oy
lac par
jms absxy
fld; absx
fad; shippy
fng
fst; x
fad; ox
fst; ox
fld; absy
fad; shippy
fng
fst; y
fadins;
fad; oy
fst; oy
lac par
tad fppar
dac 1f
lac i 1f
tad prsq
dac 1f
fld; 1;;
sqrt
fst; rpar
jms dssca
lac par

```

jms dispname  
jmp i updshp

inscr: 0  
fng  
fix  
tad d383  
spa  
jmp i inscr  
tad dm768  
sma  
jmp i inscr  
isz inscr  
jmp i inscr

absxy: 0  
sna  
jmp 7f  
lmq  
lac fldins  
dac 2f-1  
dac 4f-1  
lacq

1:  
dac absi  
sna  
jmp i absxy  
tad fppar  
dac 9f+t  
lac i 9f+t

5:  
tad px  
dac 2f  
fldins:  
fld; absx  
fad; 2...  
fst; absx  
lac i 9f+t

6:  
tad py  
dac 4f  
fld; absy  
fad; 4...  
fst; absy  
lac fadins  
dac 2b-1  
dac 4b-1  
lac absi  
tad ppar  
dac 9f+t  
lac i 9f+t  
jmp 1b

7:  
10 dzm absx  
9 dzm absx+1  
8 dzm absx+2  
7 dzm absy  
6 dzm absy+1  
5 dzm absy+2  
4 jmp i absxy  
3  
2

t = t+1

```
" space travel 4  
displa; 0  
lac locpar  
sad cplan  
skp  
jmp 2f  
lac locflg ← how to set?  
sma  
jmp 1f  
fld; cphi  
fmp; absy  
fst; fttmp1  
fld; sphi  
fmp; absx  
fad; fttmp1  
fdv; dpar  
fst; stheta  
fld; svhi  
fmp; absy  
fnr  
fst; fttmp1  
fld; cphi  
fmp; absx  
fad; fttmp1  
fdv; dpar  
fst; ctheta  
jmp 2f
```

} lock calculations

```
1:  
fld; sphi  
fst; stheta  
fld; cphi  
fst; ctheta
```

```
2:  
fld; absx  
sfmp; ctheta  
fst; fttmp1  
fld; absy  
sfmp; stheta  
fad; fttmp1  
lac aexp  
cma  
tad scale  
cma  
dac aexp  
fst; spy  
dzm inflg  
jms inscr  
jmp 1f  
tad o145777  
dac clistp i  
jms rotx  
lac dhalt  
dac inflg  
jms inscr  
jmp 1f  
tad o161577  
dac i clistp  
lac cplan  
jms dsplanet  
1:  
jms drcirc
```

jmp i displa  
rotx; 0  
fld; absx  
sfmp; sttheta  
fst; fttmp1  
fld; absy  
sfmp; ctheta  
fng  
fad; fttmp1  
lac aexp  
cma  
tad scale  
cma  
dac aexp  
fst; spx  
jmp i rotx

surf; 0  
=1  
tad setx  
cma  
dac tsetx  
lac setx  
tad o141577  
dac i clistp  
=1  
tad sety  
cma  
dac tsety  
lac sety  
tad o165777  
dac clistp i  
lac narcs  
dac nt  
fld; vx  
fst; twx  
fld; wy  
fst; twy  
fld; v  
fng  
fst; v

2:  
fld; v  
sfmp; twy  
fng  
fst; fttmp1  
fld; vv  
sfmp; twx  
fad; fttmp1  
fst; fttmp2  
fld; v  
sfmp; twx  
fst; fttmp1  
fld; vv  
sfmp; twy  
fad; fttmp1  
fst; twy  
fad; spy  
jms inscr  
jmp 1f

tad tsety  
dac dely  
cma  
tad d1  
tad tsety  
dac tsety  
fld; ftmp2  
fst; tvx  
fad; spx  
jms inscr  
jmp 1f  
tad tsetx  
dac delx  
cma  
tad d1  
tad tsetx  
dac tsetx  
lac delx  
sma  
jmp .+3  
cma  
tad o41  
alss 6  
dac delx  
lac dely  
sma  
jmp .+3  
cma  
tad o41  
tad delx  
tad o220000  
dac i clistp  
isz nt  
jmp 2b  
jmp i surf

1:  
isz surf  
jmp i surf

drcirc: 0  
lac ervflg  
spa  
jmp i drcirc  
lac fcplan  
tad prsq  
dac .+2  
fld; ..  
sqrt  
lac aexp  
cma  
tad scale  
cma  
dac aexp  
fst; dtmp1  
fcp; thrs  
spa  
jmp i drcirc  
fng  
lac dpar  
cma  
tad scale

cma  
dac dpar  
fad; dpar  
sfdiv; dpar  
fst; dtmp2  
sfmp; spy  
fst; vy  
jms inscr  
jmp i drcirc  
dac sety  
lac inflg  
sma  
jms rotx  
fld; dtmp2  
sfmp; spx  
fst; wx  
jms inscr  
jmp i drcirc  
dac setx  
fld; spy  
fng  
fad; vy  
fst; vy  
fld; spx  
fng  
fad; wx  
fst; wx  
fld; dtmp1  
sfmp; pid10  
fcp; f400  
spa  
jmp 1f  
lac d400  
dac narcs  
jmp 2f

1:  
fix  
tad dm20  
spa  
cla  
tad d20  
dac narcs  
flt

2:  
fst; dtmp1  
=1  
tad narcs  
cma  
dac narcs  
fld; f2pi  
sfdiv; dtmp1  
fst; v  
sfmp; v

11  
10  
9  
8  
7  
6  
5  
4  
3  
2:  
tad aexp  
dac aexp  
fng  
fad; fpone  
fst; vv  
lac o40004  
dac i clistp

jms surf  
jmp i drcirc  
jms surf  
jmp i drcirc  
jmp i drcirc

11

10

9

8

7

6

5

4

3

2

## space travel 5

```
dsplanet: 0
    tad points
    dac 9f+t
    lac i 9f+t
    dac i clistp
    lac o246256
    dac i clistp
    lac o253052
    dac i clistp
    lac o246036
    dac i clistp
    jmp dsplanet i
```

```
dispname: 0
    tad names
    dac 9f+t
    lac i 9f+t
    tad dm1
    dac 8
    law namedsp=1
    dac 9
    =10
    dac 9f+t
```

```
1:
    lac i 8
    sna
    jmp 2f
    dac 9f+t+1
    lrss 9
    dac i 9
    isz 9f+t
    skp
    jmp i dispname
    lac 9f+t+1
    and o177
    sna
    jmp 2f
    dac i 9
    isz 9f+t
    jmp 1b
    jmp i dispname
```

```
2:
    dzm i 9
    isz 9f+t
    jmp 2b
    jmp i dispname
```

t = t+2

```
dspscas: 0
    lac scale
    sma
    jmp 1f
    lac o55 "
    dac dssc
    =1
    tad scale
    cma
    jmp 2f
```

1:

2:

lac o53 "+  
dac dssca  
lac scale  
2:  
c11; idiv; 10  
tad o60  
dac dssca+2  
lacq  
tad o60  
dac dssca+1  
jmp i dssca

11

10

9

8

7

6

5

4

3

2

"space travel 6 -- tables

names: .+1  
sun  
earth  
ariel  
callisto  
moon  
deimos  
dione  
enceladus  
europa  
ganymede  
hyperion  
iapetus  
io  
jupiter  
mars  
mercury  
mimas  
miranda  
neptune  
nereid  
oberon  
phobos  
phoebe  
pluto  
rhea  
saturn  
tethys  
titan  
triton  
umbriel  
uranus  
venus

sun: <su>;<n  
earth: <ea>;<rt>;<h  
ariel: <ar>;<ie>; <l  
callisto: <ca>;<ll>;<is>;<to>;0  
moon: <mo>;<on>;0  
deimos: <de>;<im>;<os>;0  
dione: <di>;<on>;<e  
enceladus: <en>;<ce>;<la>;<du>;<s  
europa: <eu>;<ro>;<pa>;0  
ganymede: <ga>;<ny>;<me>;<de>;0  
hyperion: <hy>;<pe>;<ri>;<on>;0  
iapetus: <ia>;<pe>;<tu>;<s  
io: <io>;0  
jupiter: <ju>;<pi>;<te>;<r  
mars: <ma>;<rs>;0  
mercury: <me>;<cu>;<ry>;0  
mimas: <mi>;<ma>;<s  
miranda: <mi>;<ra>;<nd>;<a  
neptune: <ne>;<pt>;<un>;<e  
nereid: <ne>;<re>;<id>;0  
oberon: <ob>;<er>;<on>;0  
phobos: <ph>;<ob>;<os>;0  
phoebe: <ph>;<oe>;<be>;0  
pluto: <pl>;<ut>;<o  
rhea: <rh>;<ea>;0  
saturn: <sa>;<tu>;<rn>;0

tethys: <te>;<th>;<ys>;0  
titan: <ti>;<ta>;<n  
triton: <tr>;<it>;<on>;0  
umbriel: <um>;<br>;<ie>;<l  
uranus: <ur>;<an>;<us>;0  
venus: <ve>;<nu>;<s

points: .+1

br3

br2

br0

br2

br1

br0

br1

br0

br1

br1

br2

br0

br1

br1

br3

br2

br1

br0

br0

br3

br3

br1

br0

br0

br1

br1

br3

br1

br2

br1

br0

br3

br2

fppar: .+1  
0;3;6;9;12;15;18;21  
24;27;30;33;36;39;42;45  
48;51;54;57;60;63;66;69  
72;75;78;81;84;87;90;93

ppar: .+1

0;0;036;015;01;016;031;031  
015;015;031;031;015;0;0;0  
031;036;0;022;036;016;031;0  
031;0;031;031;022;036;0;0

prsq: .+1

016;0272245;075341  
1;0200000;0  
-07;0244122;0506362  
-02;0251477;0620663  
-03;0230761;0127762  
-025;0320300;054474  
-06;0324134;0124211

-010;0335416;0541570  
-04;0371372;0  
-06;0247430;0  
-011;0311150;0  
-05;0302622;0  
-02;0256475;0  
07;0376733;0  
-01;0221530;0  
-02;0235142;0  
-010;0217266;0  
-011;0274361;0  
04;0365471;0  
-012;0227176;0  
-06;0342454;0  
-023;0326340;0  
-013;0326265;0  
-02;0323774;0  
-05;0255140;0  
07;0263573;0  
-06;0223174;0  
-02;0251477;0  
-02;0235142;0  
-05;0223060;0  
05;0206115;0  
0;0362406;0

acc1: +1  
0;0204365;0  
-023;0320324;0  
-036;0227207;0  
-017;0340500;0  
-030;0210041;0  
-063;0341666;0  
-034;0235122;0  
-037;0247531;0  
-031;0310316;0  
-027;0334427;0  
-041;0315203;0  
-033;0303403;0  
-030;0245752;0  
-017;0201414;0  
-026;0263753;0  
-026;0205241;0  
-040;0256464;0  
-041;0272051;0  
-017;0340566;0  
-043;0275073;0  
-034;0255345;0  
-060;0341650;0  
-044;0341552;0  
-020;0307762;0  
-033;0243712;0  
-014;0233053;0  
10 -035;0265543;0  
9 -027;0340500;0  
8 -027;0210344;0  
7 -037;0210777;0  
6 -017;0275653;0  
5 -023;0252667;0  
4  
3 px: +1  
2

0;000000;0  
015;0620356;0  
005;0360005;0  
010;0666214;0  
005;0704053;0  
002;0347600;0  
005;0310506;0  
006;0220622;0  
007;0310473;0  
004;0370065;0  
006;0304101;0  
011;0676631;0  
006;0653020;0  
020;0317202;0  
017;0644356;0  
013;0206414;0  
005;0245346;0  
004;0222264;0  
023;0261234;0  
001;0372225;0  
007;0646102;0  
001;0212446;0  
013;0773152;0  
024;0274557;0  
004;0227474;0  
020;0263122;0  
004;0333254;0  
010;0216672;0  
006;0231142;0  
006;0212701;0  
023;0650051;0  
016;0233751;0

py: +1  
000;0000000;0  
017;0664054;0  
002;0662035;0  
010;0350757;0  
006;0334771;0  
000;0267000;0  
006;0726770;0  
004;0225752;0  
006;0201346;0  
010;0247536;0  
010;0343277;0  
011;0314411;0  
006;0712237;0  
021;0334656;0  
017;0342324;0  
016;0615151;0  
005;0644257;0  
005;0622456;0  
024;0224063;0  
10 012;0331314;0  
9 006;0640034;0  
8 001;0200024;0  
7 011;0243161;0  
6 023;0772355;0  
5 007;0644432;0  
4 022;0726324;0  
3 006;0260740;0  
2

010;0600213;0  
006;0237476;0  
005;0674734;0  
023;0616334;0  
016;0327155;0

$$p \vee q = 1$$

PWWS: .+1  
0000;0000000;0000000  
-054;0663265;0376074  
-036;0743326;0460356  
-043;0647730;0444215  
-045;0767246;0341205  
-034;0745027;0221674  
-036;0702670;0530661  
-034;0702340;0273047  
-037;0747771;0100452  
-041;0743411;0732756  
-044;0716603;0200021  
-050;0755641;0517072  
-035;0751602;0332677  
-072;0631066;0300145  
-056;0712656;0015171  
-050;0701167;0507203  
-033;0715124;0202507  
-034;0665024;0365605  
-073;0730064;0776551  
-054;0667362;0431776  
-043;0775760;0312631  
-030;0740613;0034530  
-055;0635657;0370276  
-074;0677234;0315321  
-037;0617102;0655555  
-066;0723370;0773672  
-035;0714526;0724272  
-043;0667633;0402706  
-040;0636053;0440472  
-037;0650025;0224325  
-071;0717663;0701773  
-052;0754612;0304722

" space travel 7

clistp = 017  
br0 = 040004  
br1 = 040005  
br2 = 040006  
br3 = 040007

d10: 10  
d3: 3  
o246256: 0246256  
o253052: 0253052  
o246036: 0246036  
o177: 0177  
nplan: 32  
d1: 1  
d20: 20  
dm1: -1  
o141577: 0141577  
o161577: 0161577  
o164372: 0164372  
o114: 0114  
o20714: 020714  
d383: 383  
dm768: -768  
o145777: 0145777  
o165777: 0165777  
o7: 7  
o60: 060  
o55: 055  
o53: 053  
o41: 041  
o220000: 0220000  
d400: 400  
dm20: -20  
o40004: 040004

fardst: 020;0200000; 0  
f2pi: 03;0311037;552421  
pid10: -01;0240662;756647  
thrs: 02;0200000;0  
f400: 011;0310000;0  
crash: -028;0200000;0  
stheta: 01;0200000;0  
ctheta: 0;0;0  
fpzero: 0;0;0  
scale: 0  
vscale: 6  
ascale: -1  
sdphi: -05;0253436;0700177  
cdphi: 000;0377743;0201725  
dhalt: 0400000  
fpone: 01;0200000;0

11  
10 9: ,=,+t  
9 horizv: ,=,+3  
8 ,pbson: ,=,+1  
7 ,pbsint: ,=,+1  
6 ,dspflg: ,=,+1  
5 par: ,=,+1  
4 absi: ,=,+1  
3 absx: ,=,+3  
2

absy: .=,+3  
v: .=,+3  
vv: .=,+3  
spx: .=,+3  
spy: .=,+3  
wx: .=,+3  
wy: .=,+3  
tx: .=,+3  
ty: .=,+3  
setx: .=,+1  
sety: .=,+1  
narcs: .=,+1  
nt: .=,+1  
inflg: .=,+1  
grvflg: .=,+1  
dtmp1: .=,+3  
dtmp2: .=,+3  
delx: .=,+1  
dely: .=,+1  
tsetx: .=,+1  
tsety: .=,+1  
accflg: .=,+1  
locpar: .=,+1  
crflg: .=,+1  
rpar: .=,+3  
dpar: .=,+3  
ax: .=,+3  
ay: .=,+3  
maxa: .=,+3  
maxj: .=,+1  
dcplan: .=,+1  
fcplan: .=,+1  
cplant: .=,+1  
shipx: .=,+3  
shipy: .=,+3  
xi: .=,+3  
yi: .=,+3  
ox: .=,+3  
oy: .=,+3  
lanflg: .=,+1  
goflg: .=,+1  
forflg: .=,+1  
bacflg: .=,+1  
sphi: .=,+3  
cphi: .=,+3  
ftmp1: .=,+3  
ftmp2: .=,+3  
locflg: .=,+3

dsetx = 0140000  
dsety = 0164000  
vecx = 0120000  
vecy = 0124000  
m = 02000  
displist:  
075057 "scale 1 intens 3 blink on lp 0 sym 0  
dsetx 800  
dsety 20  
dispcl:  
0  
060004 "intens 0 blink off

```
dsetx 0  
dsety 20  
namedsp:  
:=,+10  
dsetx 400  
dsety 20  
dssca:  
:=,+3  
040040 "scale 0  
dsetx 127  
dsety 250  
vecx 768  
dsetx 895  
dsety 255  
vecy 768  
dsetx 895  
dsety 1023  
vecx m 768  
dsetx 127  
dsety 1023  
vecy m 768  
dsetx 127  
dsety 255  
vecx 768  
dsetx 511  
dsety 255  
vecy 767  
dsetx 127  
dsety 639 -  
vecx 767  
dspl:
```

0400000

6A3

" stat  
arg = 017777  
lac arg  
tad d1  
dac name  
  
loop:  
lac arg i  
sad d4  
sys exit  
tad dm4  
dac arg i  
lac name  
tad d4  
dac name  
  
law statbuf  
sys status; dotdot; name: ..  
spa  
jmp error  
law obuf=1  
dac 12  
lac ii  
lmg  
-3  
jms octal  
lac iflags  
lmg  
-2  
jms octal  
lac iuid  
lmg  
-2  
jms octal  
-1  
tad inlks  
cma  
lmg  
-2  
jms octal  
lac isize  
lmg  
-5  
jms octal  
lac o12  
dac obuf+18  
lac d1  
sys write; obuf; 19  
jmp loop

error:  
lac name  
dac 1f  
lac d1  
sys write; 1;,,; 4  
lac d1  
sys write; 1f; 2  
jmp loop  
1: 040077; 012

```
octal: 0
dac c
law tbuf=1
dac 8
lac o40
dac 8 i
1:
lacq
and o7
tad o60
dac 8 i
lrs 3
iss c
jmp 1b
lac 8
dac c
1:
lac c i
dac 12 i
sad o40
jmp octal i
=1
tad c
dac c
jmp 1b
```

```
dotdot:
056056; 040040; 040040; 040040
```

```
d1: 1
d2: 2
d3: 3
d8: 8
d14: 14
d12: 12
o12: 012
o60: 060
o40: 040
o7: 7
d4: 4
dm4: -4
```

```
c: ,=,+1
```

```
statbuf:
iflags: ,=,+1 "
,=,+7           "
iuid: ,=,+1   "
inikst: ,=,+1   "
isize: ,=,+1   "
,=,+1
iif: ,=,+1   "
```

```
obuf: ,=,+19
tbuf: ,=,+10
```

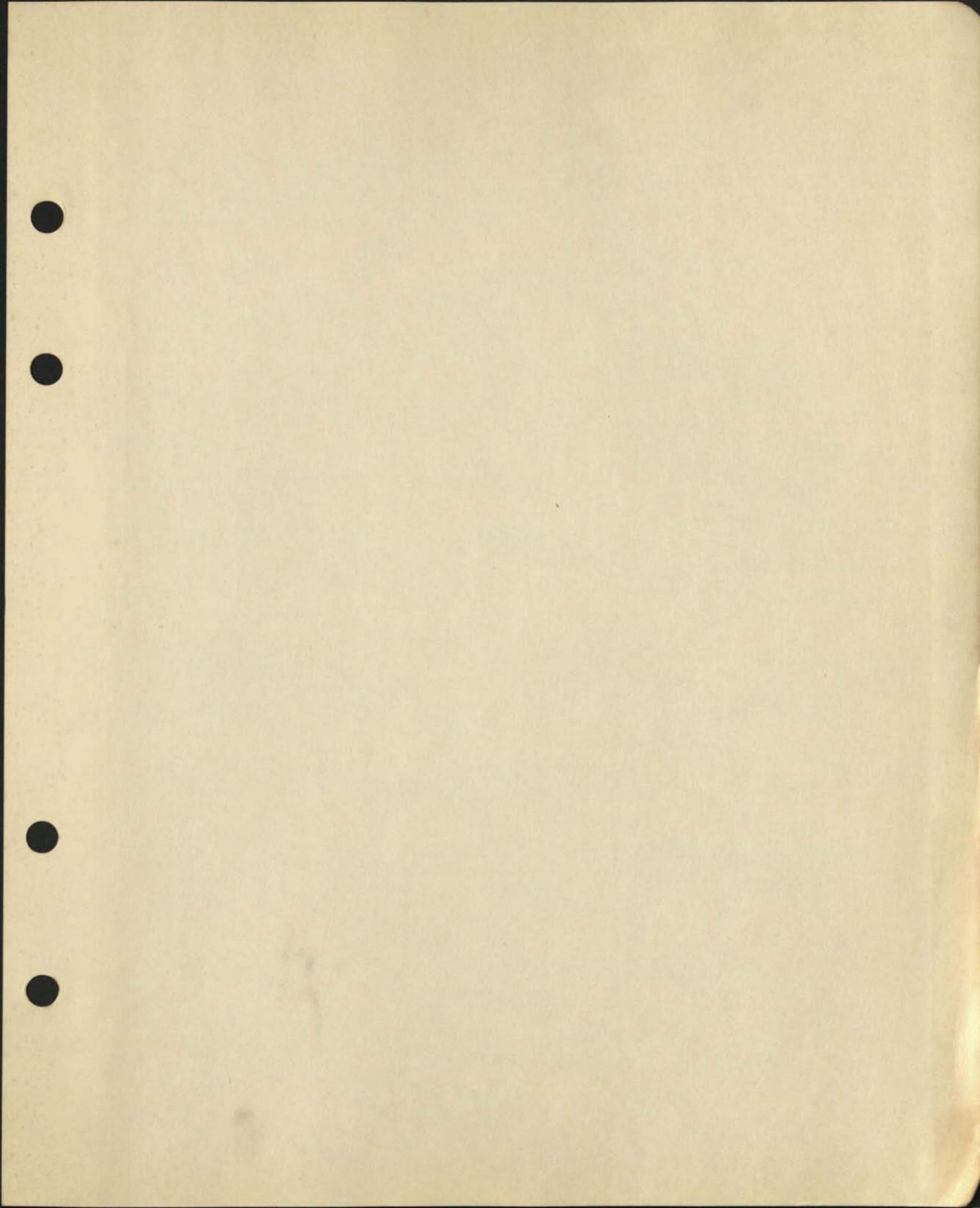
246

" tm  
sys time  
cli  
div  
216000  
dac t1  
lacq  
cli  
idiv  
10  
tad o60  
dac buf+1  
lacq  
cli; idiv; 10  
tad o60  
dac buf  
lacq  
sna  
jmp 1f  
tad o60  
alss 9  
xor buf  
dac buf

1:  
lac t1  
cli  
idiv  
6  
lacq  
cli  
idiv  
10  
tad o56060  
dac buf+6  
lacq  
cli  
idiv  
10  
tad o60  
dac buf+5  
lacq  
cli  
idiv  
6  
tad o72060  
dac buf+4  
lacq  
cli  
idiv  
10  
tad o60  
dac buf+3  
lacq  
cli  
idiv  
6  
tad o72060  
dac buf+2  
lac d1  
sys write; buf; 8  
sys exit

d1: 1  
d13: 13  
d2: 2  
o60: 060  
o56060: 056060  
o72060: 072060  
t1: 0  
t2: 0  
buf: ,=,+7; 012

PA20



t=0  
main:  
    lac 017777 i  
    sad d4  
    jmp 2f  
    sad d8  
    jmp 1f  
    law 9  
    tad 017777  
    dac 0f  
    law 017  
    sys creat;0;0  
    dac output  
1:law 5  
    tad 017777  
    dac 0f  
    sys open;0;0;0  
    dac input  
    spa  
    sys exit  
2:  
    jms advance; jmp 1f  
    jmp rinterp  
  
0:lac 2f  
    jms obuild  
1:lac owrite  
    spa  
    jmp 0b  
    jms oflush  
    las  
    spa  
    sys exit  
    sys save  
2:o777

"special machine language code

"puts out and octal string from symtab entry

symoect:  
    lac equ  
    add equbot  
    dac 9f+t  
    lac 1f  
    jms twoktab; lac 9f+t i  
1:gf ,+1 x  
    lac ii  
    dac 8  
    lac 8 i  
    add symbot  
    dac 9f+t

10     2:lac 9f+t i  
9     jms putoect  
8     lac onenl  
7     jms obuild  
6     lac 9f+t i  
5     and o600600  
4     sza    "skip unless word contained eof (0777)  
3     jmp ggoon

isz 9f+t  
jmp 2b  
t=t+1  
0600600:0600600

" recognition stack frame advance

advance:0  
lac frame  
dac 8  
lac advance i  
dac 8 i  
lac ii  
dac 8 i  
lac ignore  
dac 8 i  
lac j  
dac 8 i  
lac k  
dac 8 i  
lac frame  
dac nframe i  
lac nframe  
dac frame  
add dffrmsz  
dac nframe  
dac nframe  
jms between; add rbot; add rtop  
jms halt  
isz advance  
jmp advance i

retreat:

dzm junk  
lac gflag  
sza  
jmp 1f  
jms bundlep  
dac junk  
1: lac frame  
dac nframe  
lac frame i  
dac frame  
dac 8  
lac 8 i  
dac 3f "retrun address  
lac 8 i  
dac ii  
lac 8 i  
dac ignore  
lac fflag  
sna  
jmp 2f  
lac 8 i "restore j and k on failure  
dac j  
lac 8 i  
dac k  
2: lac junk  
sna  
jmp 3f  
dac nframe i "stass reslts  
isz nframe  
3: jmp

" bundle up results and return single pointer to them in ac  
" return 0 if no results

```
bundlep:0
    lac fflag
    sza
    jmp 2f    "no results on failure
    jms nframe0
    dac 9f+t
    cma
    tad nframe
    cma
    dac 9f+t+1
    sma
    jmp 2f
    sad m1
    jmp 3f    "only one result, no bundling necessary
    lac 9f+t
    tad m1
    dac 8

1: lac 8 i
    jms kput
    isz 9f+t+1
    jmp 1b
    lac k    "make up result pointer
    add l.gk
    jmp bundlep i

2: cla
    jmp bundlep i
3: lac 9f+t i
    jmp bundlep i
t=t+1    "where to find results
t=t+1    "negative of result count

    " the main interpreter loop
    " locate original value of nframe for present stack level,
nframe0:0
    jms s1get; add d,ii
    dac junk
    lac junk i
    dac junk
    lac junk i
    and opmask
    sad l.rw
    jmp 1f
    lac refrsz
    jmp 2f
1: lac junk i
    and o17777
2: add frame
    jmp nframe0 i

halt:0
    lac 1f
10   jms obuild
    lac halt
9    jms putoct
8    lac onenl
7    jms obuild
6    xct rstack+1
5    1:+1;012077;end
4
3
```

rinterp:  
    las "trace check  
    and d5  
    sna  
    jmp ,+3  
    lac bugr  
    jms bug

    lac fflag  
    ral  
    lac ii i  
    and opmask  
    sad l.ra  
    jmp rera  
    sad l.rb  
    jmp rerb  
    szl  
    jmp retreat  
    lrs 14  
    and o17  
    add rbranch  
    dac ,+1  
    jmp

rbranch:  
    jmp ,+1 i

    reno  
    rerk  
    rerc  
    regc  
    rerf  
    rerv  
    rera  
    reuu  
    rero  
    rerm  
    rers  
    rerv  
    reuu  
    reuu  
    reuu  
    reuu

reuu:  
    jms halt

rerb:  
    cml  
rera:  
    dzm fflag  
    snl  
    jmp goon  
    jms aget  
    dac ii  
    jmp rinterp

backup:  
    lac jsav  
    dac j  
nuts:  
    law

11  
10  
9  
8  
7  
6  
5  
4  
3  
2  
1  
0

```
dac fflag

reno:
goon:
    lac ii i
    isz ii
    and exitmask
    sza
    jmp retreat
    jmp rinterp

rerv:
    jms aget
    add frame
    dac nframe
    jmp goon

rerc:
    jms advance; jmp goon
    jms aget
    dac ii
    jmp rinterp

gegf:
rerf:
    jms aget
    add ljmp
    dac .+1
    jmp

rerx:
    lac j
    dac jsav
    jms aget
1: dac 9f+t
    jms lchar
    sad o777
    jmp goon
    dac 9f+t+1
    jms getj
    sad 9f+t+1
    jmp 2f
    jmp backup
2: lac 9f+t
    add o400000
    jmp 1b
t=t+1 "address of next comparison char
t=t+1 "character itself

aget:0
    lac ii i
    and o17777
    jmp aget i

regc:
    lac ii i
    and o757777
    xor exitmask
    dac nframe i
```

```
isz nframe
jmp goon

kput:0
isz k
dac junk1
lac k
jms between; add d0; add kmax
jms halt
add kbot
dac junk
lac junk1
dac junk i
jmp kput i

s0get:0
lac frame
xct s0get i
dac junk
lac junk i
isz s0get
jmp s0get i

s1get:0
lac frame i
xct s1get i
dac junk
lac junk i
isz s1get
jmp s1get i

s0put:0
lmq
lac frame
xct s0put i
dac junk
lacq
dac junk i
isz s0put
jmp s0put i
" here is the generaion interpreter
" the k table cant move while its active

geno:
ggoon:
lac ii i
isz ii
and exitmask
sza
jmp retreat
ginterp:
las "trace check
and d6
sna
jmp .+3
lac bugg
jms bug

lac ii i
lrss 14
and o7
```

```
add gbranch
dac ,+1
jmp

gbranch:
jmp ,+1 i
geno
gegx
geuu
gegc
gegf
gegk
gegp
gegq

geuu:
jms halt

gegx:
lac ii i
and o417777
jms obuild
jmp ggoon

gegq:
jms advance; jmp ggoon
lac env
add d.ii
dac junk
jms aget
add junk i
dac junk
lac junk i
dac ii
lac env
add d.env
dac junk
lac junk i
dac env
jmp ginterp

gegp:
jms advance; jmp ggoon
lac env
add d.ii
dac junk
lac frame i
dac env
jms aget
cma
add junk i
dac ii
jmp ginterp

gegk:
lac ii i
jms aget
add kbot
dac ii
jms s0put; add d.ii
lac frame
```

dac env  
jmp ginterp

gegc:  
jms advance; jmp ggoon  
jms aget  
dac ii  
jmp ginterp

bug:0  
dac 1f+2  
lac onenl  
jms obuild  
lac ii  
jms putoct  
lac ii i  
lrs 14  
and o17  
add 1f+2  
dac 1f+2  
lac 1f+2 i  
dac 1f+2  
lac 1f  
jms obuild  
lac ii i  
jms putoct  
las  
and d4  
sza  
jms halt  
jmp bug i

1:04000000 ,+1: 040; 0; 040777

move: 0  
dac 9f+t  
lac move i  
1:dac 2f  
lac 9f+t  
jms lchar  
dac 9f+t+1  
jms dchar  
2: 0  
lac 9f+t+1  
sad o777  
jmp 3f  
lac o400000  
add 9f+t  
dac 9f+t  
lac o400000  
add 2b  
jmp 1b

13

3:isz move  
lac 2b  
jmp move i  
t=t+1 "source address  
t=t+1 "source character

lchar:0  
dac junk  
ral  
lac junk i  
snl  
lrs 9  
and o777  
jmp lchar i

dchar:0  
lmq  
lac dchar i  
dac junk  
spa  
jmp 1f  
llss 9  
lac o777  
jmp ,+2  
1:0777000  
and junk i  
omq  
dac junk i  
isz dchar  
jmp dchar i

" gets designated character from input

jget:0  
1: lac j  
10: jms cbetween; add jmin; add jmax  
9: jmp jmore  
8: cma  
7: add jmin  
6: cma  
5: add jbot  
4: jms lchar  
3: jms class; add ignore

```
jmp jget i
lac j
add o400000
dac j
jmp 1b

" read more input - filthy code, enough to make disk &
" terminal input work. These only deliver full count
" except at eof or 1 word

jmore:
and o377700
dac jmin
add ljsiz
dac 9f+t
lac jmax
jms cbetween; add jmin; add 9f+t
lac jmin
dac jmax
dac 1f
cma
add jmin
cma
add jbot
dac 2f
lac input
sys seek
1:0;j0
-1
dac 2f i
lac input
sys read
2:0;jlsiz
sna
lac d1
add jmax
dac jmax
jmp jget+1
t=t+1

"      gets next character from input

getj:0
lac j
jms jget
dac junk
lac j
add o400000
dac j
lac junk
jmp getj i

" compare two strings - assume both left justified

comp:0
dac 9f+t
lac comp i
dac 9f+t+1
isz comp
1: lac 9f+t i
sad 9f+t+1 i
```

```
jmp 3f
and 9f+t+1 i    "do both start with eof?
spa
2:isz comp
jmp comp i
3:and o600600 "is there an eof?
sza
jmp 2b
isz 9f+t
isz 9f+t+1
jmp 1b
t=t+1    "address of string 1
t=t+1    "address of string 2
```

```
obuild:0
lmc
lac owrite
add obot
dac 2f
lacq
1:jms move
2:0
cma
add obot
cma
dac owrite
jms cbetween; add d0; add omax
skip
jmp obuild i
```

```
lac lochunk
jms oflush
lac obot
dac 2b
add lochunk
jmp 1b
```

```
oflush:0
dac 2f
lac obot
dac 1f
lac output
sys write
1:0
2:0
jmp oflush i
```

```
" outputs octal string from designated value
```

```
octal:
isz ii
lac ii i
dac 2f
lac 1f
jms twoktab
lac 2f i
1:gf geoctal x
2:0
```

```
geoctal:  
    lac ii  
    dac 8  
    lac 8 i  
    jms putoct  
    jmp ggoen
```

" converts word in ac into octal on output stream

```
putoct:0  
    dac 9f+t  
    lac 7f  
    jms obuild  
    dzm 9f+t+2  
    -6  
    dac 9f+t+1  
  
1: lac 9f+t  
    c11  
    lrs 15  
    add o60  
    dac 8f+1  
    lls 18  
    dac 9f+t  
    lac 9f+t+2    "have nonzero digits been seen?  
    sza  
    jmp 2f  
    lac 8f+1    "no, is this nonzero?  
    sad o60  
    jmp 3f    "no  
2: lac 8f  
    jms obuild  
    law  
    dac 9f+t+2  
3: isz 9f+t+1  
    jmp 1b  
    jmp putoct i  
    t=t+1    "value to convert  
    t=t+1    "digit count  
    t=t+1    "nonzero digit flag  
7: ,+1; 060777  
8: 04000000 ,+1; 0; end
```

```
eof:  
    lac j  
    dac jsav  
    jms jget  
    sad o777  
    jmp goon  
    jmp backup  
class:0  
10    dac junk1  
11    lrss 7  
12    sza  
13    jmp 2f  
14    lls 3  
15    xct class i  
16    isz class  
17    dac junk
```

c1a  
llss 4  
add l,llss  
dac 1f  
lac junk i  
1:llss  
spa  
2:isz class  
lac junk1  
jmp class i

" put symbuf symbol into table

table:  
lac equwrit  
dac equ  
jms between; add d0; add equmax  
jms halt  
add delta  
dac equwrit  
lac equ  
add equbot  
tad m1  
dac 8  
lac symwrit  
dac 8 i  
add symbot  
dac 2f  
lac mdelta  
1:dzm 8 i  
tad d1  
spa  
jmp 1b  
lac sbbot  
jms move  
2:0  
add o400000  
cma  
add symbot  
cma  
add o400000  
and o17777  
dac symwrit  
jms between; add d0; add symmax  
jms halt  
jmp goon

" find occurrence of symbuf symbol in equatab

prev:  
lac equ  
jmp find+1

find:  
lac equwrit  
dac 9f+t  
lac o777  
jms sbput  
lac sbbot  
dac 2f  
1: lac 9f+t  
tad mdelta  
dac 9f+t  
spa  
jmp nuts  
add equbot  
dac junk  
lac junk i  
add symbot  
jms comp  
2:0  
jmp 1b  
lac 9f+t  
dac equ

```
jmp goon
t=t+1 "next equtab location to test

sbput:0
  lmq
  lac sbwrite
  add shbot
  dac 1f
  lacq
  jms dchar
1:0
  lac sbwrite
  add o400000
  dac sbwrite
  jms chbetween; add d0; add sbmax
  jms halt
  jmp sbput i

getname:
  lac equ
  add equbot
  dac 9f+t
  lac 1f
  jms twoktab; lac 9f+t i

1:gf .+1 x
  lac ii
  dac 8
  lac 8 i
  add symbot
  and o17777
  jms obuild
  jmp goon
t=t+1  "equatable entry

" puts double word entries in ktab and gives
"pointer to first as result

twoktab:0
  jms kput
  lac 1.gk
  add k
  dac nframe i
  isz nframe
  xct twoktab i
  jms kput
  jmp goon
```

f5

```
char:
    lac j
    dac jsav
    isz ii
    jms ctest
    jmp backup
    jmp goon

string:
    isz ii
    jms ctest
    jmp goon
    jmp string+1

ctest:0
    jms jget
    jms class; add ii i
    jmp ctest i
    jms sbput
    lac j
    add o400000
    dac j
    isz ctest
    jmp ctest i

mark:
    jms jget
    dzm shwrite
    jmp goon

parsedo:
    isz ii
    jms advance; jmp 3f
    jms advance; jmp 1f
    jms aget
    dac ii
    jmp rinterp

1: lac frame
    add refsz
    dac ii
    sad nframe
    jmp retreat
    dac gflag
    lac gefrsz
    dac dffrmsz
    jms advance; jmp 2f
    jmp ginterp

2: lac refsz
    dac dffrmsz
    add frame
    dac nframe
    dzm gflag
    jmp retreat

3: jms s0get; add d,k
    dac k
    jmp goon

bundle:
    jms bundlep
```

```
dac 9f+t  
sna  
jmp goon  
jms nframe0  
dac nframe  
lac 9f+t  
dac nframe i  
isz nframe  
jmp goon  
t=t+1
```

" jms between; add a; add b; skip if a<=ac< b

```
between:0  
dac 9f+t  
cma  
xct between i  
isz between  
sma  
jmp 1f  
lac 9f+t  
cma  
xct between i  
isz between  
sma  
1:isz between  
lac 9f+t  
jmp between i  
t=t+0 "shared with next temporary
```

" jms cbetween; add a; add b; skip if a<=ac< b where ac  
" contains a character address

```
cbetween:0  
dac 9f+t  
cma  
xct cbetween i  
isz cbetween  
ral  
sma  
jmp 1f  
lac 9f+t  
cma  
xct cbetween i  
isz cbetween  
ral  
sma  
1:isz cbetween  
lac 9f+t  
jmp cbetween i  
t=t+1 "ac contents
```

rerm;  
jms aget  
jmp 1f  
rers:  
jms aget  
add frame  
1:dac holdlv  
lac holdlv i  
jmp 2f  
rerv:  
jms aget  
lls 6  
lrs 6  
2:dac nframe i  
isz nframe  
jmp goon

j6

rero:  
jms decnf  
dac rand1  
jms aget  
and o77  
add obranch  
dac 2f  
cma  
tad unary  
spa  
jmp 1f  
jms decnf  
dac rand2  
1: lac rand1  
2:xct  
jmp result

obranch:  
xct .+1  
opr ;op=04000000+1  
jmp rorel ;le=op;op=op+1  
jmp rorel ;ne=op;op=op+1  
jmp rorel ;lt=op;op=op+1  
jmp rorel ;ge=op;op=op+1  
jmp rorel ;eq=op;op=op+1  
jmp rorel ;gt=op;op=op+1  
tad rand2 ;ad=op;op=op+1  
jms sub ;sb=op;op=op+1  
and rand2 ;an=op;op=op+1  
jmp roor ;or=op;op=op+1  
xor rand2 ;xo=op;op=op+1  
jmp rosr ;sr=op;op=op+1  
jmp ros1 ;sl=op;op=op+1  
jmp romn ;mn=op;op=op+1  
jmp romx ;mx=op;op=op+1  
lac rand2 ;as=op;op=op+1

11  
10 opr ;pl=op;op=op+1  
9 jmp romi ;mi=op;op=op+1  
8 cma ;cm=op;op=op+1  
7 jmp roindir;indir=op;op=op+1  
6 lac holdlv ;addr=op;op=op+1  
5 unary:xct obranch+1+pl  
4  
3 rorel: "<= 001

jms sub "!= 010  
sna "< 011  
jmp 2f ">= 100  
spa "= 101  
jmp 1f "> 110  
lac d1 "a>b, code 001  
jmp 3f  
1: lac d2 "a<b, code 100  
2: add d2 "a=b, code 010  
3: and ii i  
sza  
-1  
cma  
jmp result

subi0  
cma  
tad rand2  
cma  
jmp sub i

roor:  
lmq  
lac rand2  
omq  
jmp result

rosr:  
lac rand2  
add l.lrs  
cll  
jmp 1f  
rosi:  
lac rand2  
add l.lls  
clq  
1:dac ,+2  
lac rand1  
0  
jmp result

romn:  
jms sub  
cma  
jmp ,+2  
romx:  
jms sub  
ral  
lac rand1  
sz1  
lac rand2  
jmp result

romi:  
cma  
tad d1  
jmp result

reindir:  
dac holdlv  
lac holdlv i

```
jmp result

result:
dac junk
dac nframe i
isz nframe
lac ii i
and stbit
sna
jmp exprtest
lac junk
dac holdlv i
lac ii i
and fibit
sza
jms decnf
jmp goon
```

```
exprtest:
lac ii i
and fibit
sna
jmp goon
jms decnf
lac nframe i
sza
-1
cma
dac fflag
jmp goon
```

```
decnf:0
-1
tad nframe
dac nframe
lac nframe i
jmp decnf i
```

9: , = , + t

end = -1

rx = 0000000  
rx = 0040000; gx = rx  
rc = 0100000  
rt = 0140000; gc = rt  
rf = 0200000; gf = rf  
rw = 0240000; gk = rw  
ra = 0300000; gp = ra  
rb = 0340000; qq = rb  
ro = 0400000  
rm = 0440000  
rs = 0500000  
rv = 0540000

+ 7

ljmp:jmp  
l,llss:llss  
l,lrs:lrs  
l,lls:lls  
l,ra:ra  
l,rb:rb  
l,rw:rw  
l,gk:gk  
l,gcx:gc x

x = 020000  
st = 0100  
fi = 0200  
opmask:0740000  
exitmask: x

m1:0=1  
d0:00:0  
d1:01:1  
d2:02:2  
d3:03:3  
d4:04:4  
d5:05:5  
d6:06:6  
d7:07:7  
d8:010:8  
asciisp:040  
asciinl:012  
nl:012777  
onenl:nl  
bugr:,+1;<rn>;<rx>;<rc>;<rt>;<rf>;<rw>;<ra>;<rb>  
<ro>;<rm>;<rs>;<rv>  
bugg:,+1;<gn>;<gx>;<gz>;<gc>;<gf>;<gk>;<gp>;<qq>

o17:017  
o60:060  
o77:077  
o777:0777  
  
o417777:0417777  
o400000:0400000  
o740000:0740000  
o377700:0377700  
o17777:017777

0757777:0757777  
0600600:0600600

junk:0  
junk1:0

output:1  
input:0

stbit:st  
fibit:fi  
holdlv:0  
rand1:0  
rand2:0

symwrite:0  
symbol: symtab  
symsiz = 500  
symmax:symsiz

equwrite:0  
equread:0  
equ = equread  
equbot: equtab  
equsiz = 500  
equmax: equsiz  
delta: 2  
mdelta:0-2

sbsiz=50  
sbmax:sbsiz  
sbwrite:0  
sbbot:sbbuf

fflag:0  
gflag:0  
ignore:.+1;0400000;0;0;0#0;0;0;4  
frame:rstack  
nframe:rstack+6  
env = ignore  
d,ii = d2  
d,enry = d3  
d,blkmod = d3  
d,j = d4  
d,k = d5  
dffrmsz:6  
framsiz:4  
refrsz = d6  
gefesz = d4  
ii: start  
k:0

10 rsiz = 500  
9 rmax: rsiz  
8 rbot:rstack  
7 rtop:rstack+rsiz  
6

5 owrite:0  
4 obot:obuf  
3 osiz=64

o<sub>max</sub>:o<sub>siz</sub>  
o<sub>chunk</sub>=64  
lo<sub>chunk</sub>:o<sub>chunk</sub>

j<sub>sav</sub>:0  
j<sub>read</sub>:0  
j:0  
j<sub>min</sub>:0  
j<sub>max</sub>:0  
j<sub>siz</sub> = 64  
l<sub>jsiz</sub>:j<sub>siz</sub>  
m<sub>jsiz</sub>:-j<sub>siz</sub>-1  
j<sub>bot</sub>:j<sub>buf</sub>

k<sub>siz</sub> = 200  
k<sub>max</sub>:k<sub>siz</sub>  
k<sub>bot</sub>:k<sub>tab</sub>

e<sub>qutab</sub>:.=,+e<sub>qusiz</sub>  
r<sub>stack</sub>:.=,+r<sub>siz</sub>  
,.=+8 "overflow protection  
o<sub>buf</sub>:.=,+o<sub>siz</sub>+o<sub>chunk</sub>+o<sub>chunk</sub>  
s<sub>bbuf</sub>:.=,+s<sub>bsiz</sub>  
j<sub>buf</sub>:.=,+j<sub>siz</sub>+1  
s<sub>yntab</sub>:.=,+s<sub>ymsiz</sub>  
k<sub>tab</sub>:.=,+k<sub>siz</sub>

" op.s is the op code defination file  
" for the assembler  
" no iot ops are defined  
" system cal entries are defined

f 8

"  
dac = 0040000  
jms = 0100000  
dzm = 0140000  
lac = 0200000  
xor = 0240000  
add = 0300000  
tad = 0340000  
xct = 0400000  
isz = 0440000  
and = 0500000  
sad = 0540000  
jmp = 0600000  
eae = 0640000  
i = 020000  
opr = 0740000  
law = opr i  
cma = opr 1  
cml = opr 2  
ral = opr 010  
rar = opr 020  
sma = opr 0100  
sza = opr 0200  
snl = opr 0400  
skp = opr 01000  
spa = opr 01100  
sna = opr 01200  
szl = opr 01400  
cll = opr 04000  
cla = opr 010000  
las = opr 010004  
lrs = eae 0500  
lrss = i lrs  
lls = eae 0600  
llss = i lls  
als = eae 0700  
alss = i als  
lacc = eae 01002  
lacs = eae 01001  
clq = eae 010000  
omq = eae 2  
cmq = eae 4  
lmq = eae 012000

sys = i  
save = 1  
open = 3  
read = 4  
write = 5  
creat = 6  
seek = 7  
close = 9  
exit = 14

7

6

5

4

3

21



```
" ttt1.s

t = 0
lac 017777
tad d1
dac 9f+t
lac 9f+t i
sad qdt
skp
jmp loop
dac dspflg
law sbuf-1
dac blk1
dac blk2
jms dsboard
law dbuf
sys capt
law 16
sys sysloc
tad d1
dac lpadr
law 13
sys sysloc
dac pbadr
dzm lpadr i
loop:
jms move
jms must; jmp loop
lac imaxin
dac maxin
law stack-1
dac 10
jms try; jmp unwind
jms heur; jmp loop
```

```
unwind:
lac 12
dac 9f+t
1:
lac 9f+t i
jms mark; 1
-1
tad 9f+t
dac 9f+t
lac 9f+t i
spa
jmp done
jms mark; 512
-1
tad 9f+t
dac 9f+t
lac dspflg
```

10
9
8
7
6
5
4
3
2
1
0

```
sna
jmp 1b
jms getpb
sza
jmp ,-2
jms getpb
sna
jmp ,-2
jmp 1b
```

```
t = t+1
```

move: 0  
lac dspflg  
sza  
jmp dspmove  
jms messg; M>po>iv>;e>70  
dzm 9f+t

1:  
jms getc  
sad o12  
jmp 1f  
tad om60  
spa  
jmp move+1  
dac 9f+t+1  
and o3  
sad 9f+t+1  
skp  
jmp move+1  
lac 9f+t  
alss 2  
xor 9f+t+1  
dac 9f+t  
jmp 1b

1:  
lac 9f+t  
spa  
jmp move+1  
and o77  
sad 9f+t  
skp  
jmp move+1  
tad boardp  
dac 9f+t  
lac 9f+t i  
sza  
jmp move+1  
lac 9f+t  
jms mark; 512  
jmp move i

dspmove:  
jms getpb  
sza  
jmp pbhit  
lac lpadr i  
sna  
jmp dspmove

lphit:  
1mq  
lac blink  
sna  
jmp 1f  
cma  
tad boardp  
cma  
alss 4  
tad sbufp  
dac 9f+t  
lac noblink

```
    dac blk1 i  
    dac blk2 i  
    dac 9f+t i  
  
1:  
    dzm lpadr i  
    lacq  
    cma  
    tad sbufp  
    cma  
    lrss 4  
    dac 9f+t  
    and o77  
    sad 9f+t  
    skp  
    jmp dspmove  
    tad boardp  
    dac blink  
    lac 9f+t  
    alss 4  
    tad sbufp  
    dac 9f+t  
    lac blinkpar  
    dac 9f+t i  
    jmp dspmove  
  
pbhit:  
    lac blink  
    sna  
    jmp dspmove  
    jms mark; 512  
    dzm blink  
    jmp move i  
t = t+2
```

```
must: 0
```

```
" check for 3g,4g,4h
```

```
    law line=1  
    dac 8  
    -76  
    dac 9f+t  
    dzm 9f+t+1  
    dzm 9f+t+2  
  
1:  
    cla  
    xct 8 i  
    xct 8 i  
    xct 8 i  
    xct 8 i  
    sad o4  
    jmp done  
    sad o4000  
    jmp done  
    sad o3  
    dac 9f+t+2  
    sad o3000  
    skp  
    jmp 2f  
    -4  
    tad 8
```

```
    dac 9
    cla
    xct 9 i
    sza cla
    jmp .=2
    =1
    tad 9
    dac 9
    lac 9 i
    dac 9f+t+1
2:
    isz 9f+t
    jmp 1b
    lac 9f+t+2
    sza
    jmp 1f
    lac 9f+t+1
    sna
    jmp 1f
    jms mark; 1
    jmp must i
1:
    isz must
    jmp must i
    t = t+3
done:
    lac dspflg
    sza
    jmp 1f
    jms messg; e>;x>;i>;t>;0
    sys exit
1:
    law sbuf=1
    dac blk1
    dac blk2
    jms dsboard
    law line=1
    dac 8
    =76
    dac 9f+t
1:
    cla
    xct 8 i
    xct 8 i
    xct 8 i
    xct 8 i
    sad o4
    jmp 1f
    sad o4000
    jmp 1f
    isz 9f+t
    jmp 1b
    sys exit
1:
    =4
    tad 8
    dac 9
1:
    lac 9 i
    cma
```

tad boardp  
cma  
alss 4  
tad sbufp  
dac 9f+t  
lac blinkpar  
dac 9f+t i  
lac 8  
sad 9  
skp  
jmp 1b  
jms getpb  
sza  
jmp .-2  
jms getpb  
sna  
jmp .-2  
sys exit  
t = t+1

mark; 0  
dac 9f+t  
lac mark i  
dac 9f+t i  
isz mark  
lac dspflg  
sna  
jmp 1f  
lac blk1  
dac blk2  
lac 9f+t  
cma  
tad boardp  
cma  
alss 4  
tad sbufp  
dac blk1  
jms dsboard  
jmp mark i

1:  
lac 9f+t i  
sad d1  
skp  
jmp mark i  
lac 9f+t  
cma  
tad boardp  
cma  
dac 9f+t  
lrs 4  
and o3  
tad o60  
dac 0f  
lac 9f+t  
lrs 2  
and o3  
tad o60  
dac 0f+1  
lac 9f+t  
and o3  
tad o60

```
    dac 0f+2  
    jms messg; 0:0:0:0:0  
    jmp mark i  
t = t+1
```

```
try: 0
```

```
" check 3g or 3b
```

```
    law line=1  
    dac 9  
    -76  
    dac 9f+t+3  
    dzm 9f+t+4
```

```
1:
```

```
    cla  
    xct 9 i  
    xct 9 i  
    xct 9 i  
    xct 9 i  
    sad o3000  
    dac 9f+t+4  
    sad o3  
    skp  
    jmp 2f  
    lac 10  
    dac 12  
    -1  
    dac 12 i  
    -4  
    tad 9  
    dac 9  
    cla  
    xct 9 i  
    sza cla  
    jmp ,=2  
    -1  
    tad 9  
    dac 9  
    lac 9 i  
    dac 12 i  
    lac dspflg  
    sza  
    jmp try i  
    jms messg; i>;040;w>;i>;n>;0  
    jmp try i
```

```
2:
```

```
    isz 9f+t+3  
    jmp 1b  
    lac 9f+t+4  
    sna  
    jmp 1f  
    isz try  
    jmp try i
```

```
" Save
```

```
1:
```

```
    isz maxin  
    jmp 1f  
    -1
```

```
    dac maxin  
    isz try  
    jmp try i  
1:  
    lac try  
    dac 10 i  
    lac 8  
    dac 10 i  
    lac 9f+t  
    dac 10 i  
    lac 9f+t+1  
    dac 10 i  
    lac 9f+t+2  
    dac 10 i  
  
" check 2g  
  
    law line=1  
    dac 8  
    =76  
    dac 9f+t  
1:  
    cla  
    xct 8 i  
    xct 8 i  
    xct 8 i  
    xct 8 i  
    sad o2  
    skp  
    jmp 2f  
    =4  
    tad 8  
    dac 9  
    cla  
    xct 9 i  
    sza cla  
    jmp .=2  
    lac 9  
    dac 9f+t+1  
    lac 9f+t+1 i  
    dac 9f+t+1  
    cla  
    xct 9 i  
    sza cla  
    jmp .=2  
    lac 9  
    dac 9f+t+2  
    lac 9f+t+2 i  
    dac 9f+t+2  
  
" recurse  
10  
9  
8  
7  
6  
5  
4  
3  
lac d1  
dac 9f+t+1 i  
lac o1000  
dac 9f+t+2 i  
jms try; jmp prnt  
lac d1  
dac 9f+t+2 i  
lac o1000  
dac 9f+t+1 i
```

```
jms try; jmp prnt  
dzm 9f+t+1 i  
dzm 9f+t+2 i  
2:  
isz 9f+t  
jmp 1b
```

" restore

```
=5  
tad 10  
dac 10  
dac 9  
lac 9 i  
dac try  
lac 9 i  
dac 8  
lac 9 i  
dac 9f+t  
lac 9 i  
dac 9f+t+1  
lac 9 i  
dac 9f+t+2  
isz try  
jmp try i
```

prnt:

```
lac 9f+t+1 i  
sad d1  
jmp 1f  
lac 9f+t+1  
dac 12 i  
lac 9f+t+2  
jmp 2f
```

1:

```
lac 9f+t+2  
dac 12 i  
lac 9f+t+1
```

2:

```
dac 12 i  
dzm 9f+t+1 i  
dzm 9f+t+2 i  
=5  
tad 10  
dac 10  
dac 9  
lac 9 i  
dac try  
lac 9 i  
dac 8  
lac 9 i  
dac 9f+t  
lac 9 i  
dac 9f+t+1  
lac 9 i  
dac 9f+t+2  
jmp try i  
t = t+5
```

```
heur: 0  
jms addpri
```

```
-2
tad force
dac lforce
=1000
dac lpri
=64
dac 9f+t
lac boardp
dac 9f+t+1
1:
lac 9f+t+1 i
sza
jmp 3f
lac d1
dac 9f+t+1 i
jms addpri
lac force
sad lforce
jmp 4f
lac pri
cma
tad lpri
sma cma
jmp 3f=1
sza
jmp 2f
isz prob
=1
cll; idiv; prob;.,
lacq
lrss 6
dac force
sys time
lacq
tad rand
cll; mul; 37111
lacq
dac rand
cll; lrs 6
cma
tad force
spa
jmp 3f=1
jmp 2f+2
2:
lac d1
dac prob
lac pri
dac lpri
lac 9f+t+1
dac lmov
dzm 9f+t+1 i
3:
isz 9f+t+1
isz 9f+t
jmp 1b
lac lmov
jms mark1 1
jmp heur i
4:
lac 9f+t+1
```

jms mark; 1  
jmp heur i  
t = t+2

addpri: 0  
clq  
law line=1  
dac 8  
=76  
dac 9f+t  
dzm force

1:  
cla  
xct 8 i  
xct 8 i  
xct 8 i  
xct 8 i  
sad o2000  
isz force  
dac 9f+t+1  
rtr,rtr,rtr,rar  
xor 9f+t+1  
and o77  
tad 2f  
dac ,+2  
lacq  
tad ,  
lme  
isz 9f+t  
jmp 1b  
law plane=1  
dac 8  
=18  
dac 9f+t

1:  
=16  
dac 9f+t+1  
cla

0:  
xct 8 i  
isz 9f+t+1  
jmp 0b

pstrat:  
jms 3f; 04002; 1  
jms 3f; 03001; 15  
jms 3f; 04001; 20  
jms 3f; 1; 1  
isz 9f+t  
jmp 1b

lacq  
dac pri  
jmp addpri i  
2: tad pritab  
3: 0

sad 3b i  
jmp 1f  
isz 3b  
isz 3b  
jmp 3b i

1:  
isz 3b

```
lacq
tad 3b i
lmq
cla
isz 3b
jmp 3b i
t = t+2

dsboard: 0
=64
dac 9f+t
dzm 9f+t+2
law board=1
dac 8
law sbuf=1
dac 11
```

```
8:
lac noblink
dac 11 i
lac 9f+t+2
and o3
alss 6
dac 9f+t+3
alss 1
tad 9f+t+3
dac 9f+t+3
lac 9f+t+2
and o14
alss 4
tad 9f+t+3
xor setxv
dac 11 i
lac 9f+t+2
and o74
alss 4
xor setyy
dac 11 i
lac 8 i
sna
jmp 4f
sad d1
jmp 3f
law ex=1
jmp 2f
3: law oh=1
jmp 2f
4: law dot=1
2: dac 9
=12
dac 9f+t+1
```

```
1:
lac 9 i
dac 11 i
isz 9f+t+1
jmp 1b
lac noblink
dac 11 i
isz 9f+t+2
isz 9f+t
jmp 8b
```

```
-1
dac 11 i
lac blinkpar
dac blk1 i
dac blk2 i
jmp dsboard i
t = t+4

getc: 0
cla
sys read; 9f+t; 1
sna spa
sys save
lac 9f+t
lrss 9
jmp getc i

messg: 0
-1
tad messg
dac 9

1:
lac 9 i
sna
jmp 1f
dac 9f+t
lac d1
sys write; 9f+t; 1
jmp 1b

1:
lac d1
sys write; 012; 1
jmp 9 i
t = t+1

getpb: 0
sys time
lac pbaddr i
and o2000
sza
sys exit
lac pbaddr i
jmp getpb i
```

" ttt2  
boardp: board  
shufp: shuf  
qdt: <dt>  
imaxin: -4000  
noblink: 060000  
blinkpar: 070000  
setxv: setx  
setyv: sety  
o12: 012  
o14: 014  
o74: 074  
o3: 03  
o2000: 02000  
o60: 060  
o4: 4  
o3000: 03000  
o4000: 04000  
o2: 2  
o1000: 01000  
d1: 1  
om60: -060  
o77: 077  
line:

tb = tad boardm1  
tb+52; tb+56; tb+60; tb+64  
tb+1; tb+6; tb+11; tb+16  
tb+2; tb+6; tb+10; tb+14  
tb+2; tb+18; tb+34; tb+50  
tb+2; tb+22; tb+42; tb+62  
tb+3; tb+7; tb+11; tb+15  
tb+3; tb+23; tb+43; tb+63  
tb+3; tb+19; tb+35; tb+51  
tb+4; tb+7; tb+10; tb+13  
tb+4; tb+8; tb+12; tb+16  
tb+4; tb+19; tb+34; tb+49  
tb+4; tb+20; tb+36; tb+52  
tb+4; tb+23; tb+42; tb+61  
tb+61; tb+62; tb+63; tb+64  
tb+4; tb+24; tb+44; tb+64  
tb+5; tb+6; tb+7; tb+8  
tb+5; tb+21; tb+37; tb+53  
tb+5; tb+22; tb+39; tb+56  
tb+6; tb+22; tb+38; tb+54  
tb+7; tb+23; tb+39; tb+55  
tb+8; tb+23; tb+38; tb+53  
tb+8; tb+24; tb+40; tb+56  
tb+9; tb+10; tb+11; tb+12  
tb+9; tb+26; tb+43; tb+60  
tb+9; tb+25; tb+41; tb+57  
tb+12; tb+28; tb+44; tb+60  
tb+13; tb+14; tb+15; tb+16  
tb+13; tb+25; tb+37; tb+49  
tb+13; tb+26; tb+39; tb+52  
tb+13; tb+29; tb+45; tb+61  
tb+13; tb+30; tb+47; tb+64  
tb+14; tb+26; tb+38; tb+50  
tb+14; tb+30; tb+46; tb+62  
tb+15; tb+27; tb+39; tb+51  
tb+53; tb+54; tb+55; tb+56  
tb+15; tb+31; tb+47; tb+63

16-

tb+16; tb+28; tb+40; tb+52  
tb+16; tb+27; tb+38; tb+49  
tb+16; tb+31; tb+46; tb+61  
tb+16; tb+32; tb+48; tb+64  
tb+17; tb+21; tb+25; tb+29  
tb+17; tb+18; tb+19; tb+20  
tb+17; tb+22; tb+27; tb+32  
tb+18; tb+22; tb+26; tb+30  
tb+19; tb+23; tb+27; tb+31  
tb+20; tb+23; tb+26; tb+29  
tb+20; tb+24; tb+28; tb+32  
tb+21; tb+22; tb+23; tb+24  
tb+25; tb+26; tb+27; tb+28  
tb+29; tb+30; tb+31; tb+32  
tb+33; tb+34; tb+35; tb+36  
tb+33; tb+38; tb+43; tb+48  
tb+33; tb+37; tb+41; tb+45  
tb+34; tb+38; tb+42; tb+46  
tb+35; tb+39; tb+43; tb+47  
tb+36; tb+39; tb+42; tb+45  
tb+36; tb+40; tb+44; tb+48  
tb+37; tb+38; tb+39; tb+40  
tb+41; tb+42; tb+43; tb+44  
tb+45; tb+46; tb+47; tb+48  
tb+49; tb+50; tb+51; tb+52  
tb+49; tb+53; tb+57; tb+61  
tb+12; tb+27; tb+42; tb+57  
tb+49; tb+54; tb+59; tb+64  
tb+50; tb+54; tb+58; tb+62  
tb+51; tb+55; tb+59; tb+63  
tb+52; tb+55; tb+58; tb+61  
tb+57; tb+58; tb+59; tb+60  
tb+11; tb+27; tb+43; tb+59  
tb+10; tb+26; tb+42; tb+58  
tb+1; tb+2; tb+3; tb+4✓  
tb+1; tb+5; tb+9; tb+13✓  
tb+1; tb+17; tb+33; tb+49✓  
tb+1; tb+18; tb+35; tb+52✓  
tb+1; tb+21; tb+41; tb+61✓  
tb+1; tb+22; tb+43; tb+64✓

planet:

tb+1; tb+2; tb+3; tb+4; tb+5; tb+6; tb+7; tb+8  
tb+9; tb+10; tb+11; tb+12; tb+13; tb+14; tb+15; tb+16

tb+17; tb+18; tb+19; tb+20; tb+21; tb+22; tb+23; tb+24  
tb+25; tb+26; tb+27; tb+28; tb+29; tb+30; tb+31; tb+32

tb+33; tb+34; tb+35; tb+36; tb+37; tb+38; tb+39; tb+40  
tb+41; tb+42; tb+43; tb+44; tb+45; tb+46; tb+47; tb+48

tb+49; tb+50; tb+51; tb+52; tb+53; tb+54; tb+55; tb+56  
tb+57; tb+58; tb+59; tb+60; tb+61; tb+62; tb+63; tb+64

tb+13; tb+14; tb+15; tb+16; tb+29; tb+30; tb+31; tb+32  
tb+45; tb+46; tb+47; tb+48; tb+61; tb+62; tb+63; tb+64

tb+9; tb+10; tb+11; tb+12; tb+25; tb+26; tb+27; tb+28  
tb+41; tb+42; tb+43; tb+44; tb+57; tb+58; tb+59; tb+60

tb+5; tb+6; tb+7; tb+8; tb+21; tb+22; tb+23; tb+24  
tb+37; tb+38; tb+39; tb+40; tb+53; tb+54; tb+55; tb+56

tb+1; tb+2; tb+3; tb+4; tb+17; tb+18; tb+19; tb+20  
tb+33; tb+34; tb+35; tb+36; tb+49; tb+50; tb+51; tb+52

tb+1; tb+5; tb+9; tb+13; tb+17; tb+21; tb+25; tb+29  
tb+33; tb+37; tb+41; tb+45; tb+49; tb+53; tb+57; tb+61

tb+2; tb+6; tb+10; tb+14; tb+18; tb+22; tb+26; tb+30  
tb+34; tb+38; tb+42; tb+46; tb+50; tb+54; tb+58; tb+62

tb+3; tb+7; tb+11; tb+15; tb+19; tb+23; tb+27; tb+31  
tb+35; tb+39; tb+43; tb+47; tb+51; tb+55; tb+59; tb+63

tb+4; tb+8; tb+12; tb+16; tb+20; tb+24; tb+28; tb+32  
tb+36; tb+40; tb+44; tb+48; tb+52; tb+56; tb+60; tb+64

tb+1; tb+6; tb+11; tb+16; tb+17; tb+22; tb+27; tb+32  
tb+33; tb+38; tb+43; tb+48; tb+49; tb+54; tb+59; tb+64

tb+4; tb+7; tb+10; tb+13; tb+20; tb+23; tb+26; tb+29  
tb+35; tb+39; tb+42; tb+45; tb+52; tb+55; tb+58; tb+62

tb+1; tb+2; tb+3; tb+4; tb+21; tb+22; tb+23; tb+24  
tb+41; tb+42; tb+43; tb+44; tb+61; tb+62; tb+63; tb+64

tb+13; tb+14; tb+15; tb+16; tb+25; tb+26; tb+27; tb+28  
tb+37; tb+38; tb+39; tb+40; tb+39; tb+50; tb+51; tb+52

tb+1; tb+5; tb+9; tb+13; tb+18; tb+22; tb+26; tb+30  
tb+35; tb+39; tb+43; tb+47; tb+52; tb+56; tb+60; tb+64

tb+4; tb+8; tb+12; tb+16; tb+19; tb+23; tb+27; tb+31  
tb+34; tb+38; tb+42; tb+46; tb+49; tb+53; tb+57; tb+61

pritable

0;10;22;0

0;10;20;0

0;30;30;0

0;0;0;0;0

ex:

vx 0100

vy 020 iv

vx 0200

vy 040 v

vx m 0140

vy 0 iv

vx 0100

vy m 040 v

dneP

dneP

dneP

dneP

oh:

vx 0100

vy 020 iv

vx 0140

vy 0 v

vx 040

vy 040 v

vx m 0140

vy 0 v

vx m 040  
vy m 040 v  
dneop  
dneop  
dot:  
lpenb  
vx 0200  
vy 040 iv  
vx 05  
vy 0 v  
vx 0  
vy m 05 v  
vx m 05  
vy 0 v  
vx 0  
vy 05 v  
lpdis  
dbuf:  
lpdis  
scale 0  
setx 0  
sety 0  
"  
vx 01400 v  
vx 0377  
vy 0374 v  
vx m 01377 v  
vx m 0400  
vy m 0374 v  
"  
setx 0  
sety 0400  
vx 01400 v  
vx 0377  
vy 0374 v  
vx m 01377 v  
vx m 0400  
vy m 0374 v  
"  
setx 0  
sety 01000  
vx 01400 v  
vx 0377  
vy 0374 v  
vx m 01377 v  
vx m 0400  
vy m 0374 v  
"  
setx 0  
sety 01400  
vx 01400 v  
vx 0377  
vy 0374 v  
vx m 01377 v  
vx m 0400  
vy m 0374 v  
"  
dneop  
sbuf: .=,+1050  
board: .=,+64  
maxint: .=,+1  
force: .=,+1

pri: ,=,+1  
lpri: ,=,+1  
lforce: ,=,+1  
rand: ,=,+1  
lmovi: ,=,+1  
dspflg: ,=,+1  
lpadr: ,=,+1  
pbadr: ,=,+1  
blink: ,=,+1  
blk1: ,=,+1  
blk2: ,=,+1  
9: ,=,+t  
stack:

dnop = 040040  
setx = 0140000  
sety = 0164000  
m = 02000  
scale = 0040040  
v = 020000  
iv = 030000  
vx = 0100000  
vy = 0104000  
lpdis = 0044000  
lpenb = 0046000  
wbl = 0704424

823

" un

```
sys open; n,out; 0
spa
jmp error
sys read; buf; 3072
cli; idiy; 6
lacq; ema; tad d1
dac c1
1:
lac t1
tad d4
dac t2
lac i t2
sad d3
skp
jmp 2f
lac d1
sys write; t1; buf; 4
lac d1
sys write; mes+1; 1
2:
lac t1
tad d6
dac t1
isz c1
jmp 1b

sys exit

error:
lac d1
sys write; mes; 2
sys exit

mes:
077; 012
n,out:
<n,>;<ou>;<t 040;040040

d1: 1
d6: 6
d3: 3
c1: 0
t2: 0
d4: 4

buf:
```