

spacewar 4.2 5/11/63 ddp . pt 1

```
3/      jmp sbf          / ignore seq. break
      jmp a40
      jmp a1      / use test word for control, not iot 11 co
```

/ interesting and often changed constants

/symb loc usual value (all instructions are executed,
/ and may be replaced by jda or jsp)

```
tno, 6, law i 41          / number of torps + 1
tv1, 7, sar 4s      / torpedo velocity
rlt, 10, law i 20 / torpedo reload time
tlf, 11, law i 140 / torpedo life
foo, 12, -20000 / fuel supply
maa, 13, 10 / spaceship angular acceleration
sac, 14, sar 4s / spaceship acceleration
str, 15, 100 / star capture radius
me1, 16, 6000 / collision "radius"
me2, 17, 3000 / above/2
ddd, 20, -0 / 0 to save space for ddt
the, 21, sar 9s / amount of torpedo space warpage
mhs, 22, law i 10 / number of hyperspace shots
hd1, 23, law i 40 / time in hyperspace before breakout
hd2, 24, law i 100 / time in hyperspace breakout
hd3, 25, law i 200 / time to recharge hyperfield generators
hr1, 26, scl 9s / scale on hyperspatial displacement
hr2, 27, scl 4s / scale on hyperspatially induced velocity
hur, 30, 40000 / hyperspatial uncertainty
ran, 31, 0 / random number
grv, 32, sar 6s / gravitational constant
```

/ place to build a private control word routine.
/ it should leave the control word in the io as follows.
/ high order 4 bits, rotate ccw, rotate cw, (both mean hyperspace)
/ fire rocket, and fire torpedo. Low order 4 bits, same for
/ other ship. Routine is entered by jsp cwg.

40/

```
cwr,      jmp mg1 / normally iot 11 control
. 20/     / space
```

/ routine to flush sequence breakes, if they occur.

```
sbfi,      tyi
           lio 2
           lac 0
           lsm
           jmp i 1
```

```
           define
xincr X,Y,INS
           lac Y
           INS ssn
           dac Y
           lac X
           INS scn
           dac X
           term
```

```
           define
yincr X,Y,INS
           lac Y
           INS scn
           dac Y
           lac X
           -INS+add+sub ssn
           dac X
           terminate
```

```
           define
dispatch
           add (. 3
           dap . 1
           jmp .
           term
```

```
           define
dispt A,Y,B
           repeat 6  B=B+B
           lio Y
           dpy-A+B
           term
```

```
           define
scale A,B,C
           lac A
           sar B
           dac C
           term
```

```
define
diff V,S,SF
    add i V
    dac i V
    xct SF
    add i S
    dac i S
    term
```

```
define
random
    lac ran
    rar 1s
    xor (355670
    add (355670
    dac ran
    term
```

```
define
ranct S,SS,C
    random
    S
    SS
    sma
    cma
    dac C
    terminate
```

define

```
varsft
dzm  $\overline{xys}$ 
dac  $\overline{t1}$ 
idx  $\overline{xys}$ 
v2, idx  $\overline{xys}$ 
lac  $\overline{t1}$ 
scr 2s
dac  $\overline{t1}$ 
sza
jmp v2+R
scr 2s
swap
terminate
```

define

```
undosft
dac t1
dio  $\overline{t2}$ 
lac  $\overline{xys}$ 
add sft
dap .+1
lac .
dac .+6
dac .+6
xor (10000
dac  $\overline{xyt}$ 
lac  $\overline{t1}$ 
dio  $\overline{t2}$ 
scr .
scr .
terminate
```

/ change scr to scl or scl to scr.

define

```
integrate A,B
cli
lac i A
scr 9s
scr 1s
div  $\overline{t1}$ 
hlt
cma+cli-opr
xct  $\overline{xyt}$ 
xct grv
dac B
terminate
```

sft,

```
lac .-1
scr 7s
scr 6s
scr 5s
scr 4s
scr 3s
scr 2s
scr 1s
scr
scl 1s
```

/sine-cosine subroutine•Adams associates
/calling sequence= number in AC, jda sin or jdacos.
/argument is between +2 pi, with binary point to right of bit 3.
/answer has binary point to right of bit 0. Time = 2.35-? ms.
/changed for auto-multiply , ddp 1/19/63

```
cos,      0
          dap csx
          lac (62210
          add cos
          dac sin
          jmp .+4

sin,      0
          dap csx
          lac sin
          spa
si1,      add (311040
          sub (62210
          sma
          jmp si2
          add (62210

si3,      ral 2s
          mul (242763
          dac sin
          mul sin
          dac cos
          mul (756103
          add (121312
          mul cos
          add (532511
          mul cos
          add (144417
          mul sin
          scl 3s
          dac cos
          xor sin
          sma
          jmp csx-1
          lac (377777
          lio sin
          spi
          cma
          jmp csx

          lac cos
csx,      jmp .

si2,      cma
          add (62210
          sma
          jmp si3
          add (62210
          spa
          jmp .+3
          sub (62210
          jmp si3

          sub (62210
          jmp si1
```

/integer square root
 /input in ac, binary point to right of bit 17, jda sqt
 /answer in ac with binary point between bits 8 and 9
 /largest input number = 177777

```

sqt,      0
          dap sqx
          law i 23
          dac sq1
          dzm sq2
          lio sqt
          dzm sqt

sq3,      isp sq1
          jmp .+3
          lac sq2
sqx,      jmp .

          lac sq2
          sal 1s
          dac sq2
          lac sqt
          rcl 2s
          sza i
          jmp sq3
          dac sqt
          lac sq2
          sal 1s
          add (1
          sub sqt
          sma+sza-skp
          jmp sq3
          spa
          cma
          dac sqt
          idx sq2
          jmp sq3

sq1,      0
sq2,      0
  
```

```
/outline compiler
/ac=where to compile to, call jda oc
                               /ot=address of outline table
```

```
define
    plinst A
    lac A
    dac i oc
    idx oc
    terminate
```

```
define
    comtab A, B
    plinst A
    jsp ocs
    lac B
    jmp oce
    terminate
```

```
ocs,    dap ocz                /puts in swap
        dio i oc
        idx oc
        dio i oc
        idx oc
ocz,    jmp .
```

```
oc,      0                    /outline compiler proper
```

```
        dap ocx
        lac i ocx
        dap ocg
        plinst (stf 5
        dap ocm
        idx ocx
ock,    plinst (lac  $\overline{s}x1$ 
        plinst (lio  $\overline{s}y1$ 
        clf 6
```

```
ocj,    setup  $\overline{occ},6$ 
ocg,    lio .                  /outline table
och,    cla
```

```
        rcl  $\overline{3}s$ 
        dio  $\overline{oci}$ 
        lio (rcl  $\overline{9}s$ 
        dispatch
        opr
```

```
        jmp oc1
oco,    jmp oc2
ocq,    jmp oc3
ocp,    jmp oc4
ocr,    jmp oc5
        jmp oc6
```

```

plinst (szf 5      /7 code
add (4
dap ocn
plinst ocn
plinst (dac  $\overline{sx1}$ 
plinst (dio  $\overline{sy1}$ 
plinst (jmp sq6
plinst (clf 5
plinst (lac  $\overline{scm}$ 
plinst (cma
plinst (dac  $\overline{scm}$ 
plinst (lac  $\overline{ssm}$ 
plinst (cma
plinst (dac  $\overline{ssm}$ 
plinst (lac  $\overline{csm}$ 
plinst (lio  $\overline{ssd}$ 
plinst (dac  $\overline{ssd}$ 
plinst (dio  $\overline{csm}$ 

plinst (lac  $\overline{ssc}$ 
plinst (lio  $\overline{csn}$ 
plinst (dac  $\overline{csn}$ 
plinst (dio  $\overline{ssc}$ 
plinst ocm
ocx,    jmp .

ocm,    jmp .
ocn,    jmp .

oc1,    plinst (add  $\overline{ssn}$ 
        jsp ocs
        lac (sub  $\overline{scn}$ 
oce,    dac i oc
        idx oc
        jsp ocs
        plinst (ioh
        lac (dpy-4000
ocd,    dac i oc
        idx oc
        lio  $\overline{oci}$ 
        count  $\overline{occ}$ , och
        idx ocg
        jmp ocj

oc2,    comtab (add  $\overline{scm}$ , (add  $\overline{ssm}$ 
oc3,    comtab (add  $\overline{ssc}$ , (sub  $\overline{csm}$ 
oc4,    comtab (sub  $\overline{scm}$ , (sub  $\overline{ssm}$ 
oc5,    comtab (add  $\overline{csn}$ , (sub  $\overline{ssd}$ 
oc6,    szf 6
        jmp oc9
        stf 6
        plinst (dac  $\overline{ssa}$ 
        lac (dio  $\overline{ssi}$ 
        jmp ocd
oc9,    clf 6
        plinst (lac  $\overline{ssa}$ 
        lac (lio  $\overline{ssi}$ 
        jmp ocd

```

/display a star

define

starp
add bx
swap
add by
swap
ioh
dpy-4000
terminate

blp, dap blx /star

szs 60
jmp blx
random
rar 9s
and (400700
spa
xor (377777
dac bx
lac ran
ral 4s
and (400700
spa
xor (377777
dac by
jsp bpt
ioh

blx, jmp .

bpt, dap bpx
random
sar 9s
sar 6s
spa
cma
sal 3s
add (bds
dap bjm
cla cli clf 6-opr-opr
dpy-4000

bjm, jmp .

bds, repeat 10, starp
szf 6

bpx, jmp .
stf 6
cma
swap
cma
swap
jmp bjm

/background display • 3/13/62, prs.

```
define
dislis J, Q
    dap fx+R
    clf 5
    lac flo+R
    dap fpo+R
fs,    dap fin+R
        dap fyn+R
        idx fyn+R

fin,    lac      /lac X
        sub fpr   /right margin
        sma
        sub (20000
        add (2000
frr,    spq
fou,    jmp fuu+R
fie,    sub (1000
        sal 8s
fyn,    lio      /lio Y
        dpy-1
        stf 5
fid,    idx fyn+R
        sad (lio Q+2
        jmp flp+R
        sad fpo+R
        jmp fx+R
        dap fin+R
        idx fyn+R
        jmp fin+R

fuu,    szf 5
fx,     jmp      /return
        idx flo+R
        idx flo+R
        sas (Q+2
        jmp fid+R
        law J
        dac flo+R
        jmp fid+R

flp,    lac (lio J
        sad fpo+R
        jmp fx+R
        dap fin+R
        law J+1
        dap fyn+R
        jmp fin+R

fpo,    lio
flo,    J
        terminate
```

```

define
background    jsp bck
               termin

bck,          dap bcx
               szs 40
               jmp bcx
               jsp 1m
               jsp 2m
               idx bcc
               and (1
               sza
               jsp 3m
bc1,          law 3
               and bcc
               sza i
               jsp 4m
bc3,          isp bkc
               jmp bcy
               law i 10
               dac bkc
               law i 1
               add fpr
               spa
               add (20000
               dac fpr
bcy,          jsp 1m
bcx,          jmp

1m,          dislis 1j, 1q
2m,          dislis 2j, 2q
3m,          dislis 3j, 3q
4m,          dislis 4j, 4q

bcc,          0
bkc,          0
fpr,          10000

```

mul=mus
div=dis

start

/main control routine for spaceships

```

nob=30                      /total number of colliding objects
m10,      setup mtc, 5000          /delay for loop
          init m11, mtb /loc of calc routines
          add (nob
          dap mx1          / x
nx1=mtb nob
          add (nob
          dap my1          / y
ny1=nx1 nob
          add (nob
          dap na1          / count for length of explosion or torp
na1=ny1 nob
          add (nob
          dap mb1          / count of instructions taken by calc routine
nb1=na1 nob
          add (nob
          dac mdx          / dx
ndx=nb1 nob
          add (nob
          dac mdy          / dy
ndy=ndx nob
          add (nob
          dap mom          /angular velocity
nom=ndy nob
          add (2
          dap mth          / angle
nth=nom 2
          add (2
          dac mfu          /fuel
nfu=nth 2
          add (2
          dac mtr          / no torps remaining
ntr=nfu 2
          add (2
          dap mot          / outline of spaceship
not=ntr 2
          add (2
          dap mco          / old control word
nco=not 2
          add (2
          dac mh1
nh1=nco 2
          add (2
          dac mh2
nh2=nh1 2
          add (2
          dac mh3
nh3=nh2 2
          add (2
          dac mh4
nh4=nh3 2
nnn=nh4 2

```

```

law ss1
xor mtb
sza
jmp mdn
law ss2
xor mtb 1
sza
jmp mdn
law 1      / test if both ships out of torps
add ntr
spa
jmp md1
law 1
add ntr 1
spa 1
jmp mdn
md1,      xct tlf      / restart delay is 2X torpedo life
          sal 1s
          dac ntd
          jmp ml1

mdn,      count ntd,ml1
          stf 1
          stf 2
          law ss1
          xor mtb
          sza
          clf 1
          sza i
          idx 1sc
          law ss2
          xor mtb 1
          sza
          clf 2
          sza i
          idx 2sc
          clf 2
          jmp a

```

```

a1,    law mg2      / test word control
      dac cwg
      jmp a

a40,   law cwr      / here from start at 4
      dac cwg
      jmp a6

1sc,   0            /scores
2sc,   0

a,      dzn 8j1
      dzn 8j2
      lac gct
      sna
      jmp a5
      count gct, a5
      lac 1sc
      sas 2sc
      jmp a4
      law i 1
      dac gct
a5,     lat
      and (40
      sza i
      jmp a2
a4,     jmp fi1
      lat
      and (40
      sza
      jmp a2
      dzn 1sc
      dzn 2sc
a6,     lat
      rar 6s
      and (37
      sza
      cma
      dac gct
a2,     clear ntb, nnn-1      / clear out all tables
      law ss1
      dac ntb
      law ss2
      dac ntb 1
      lac (200000
      dac nx1
      dac ny1
      cma
      dac nx1 1
      dac ny1 1
      lac (144420
      dac nth

```

```

law nnn          / start of outline program
dac not
lio ddd
spi 1
jmp a3
jda oc          / compile outline
ot1
a3, dac not 1
dap fl1
jda oc
ot2
dap fl2
xct tno
dac ntr
dac ntr 1
lac foo
dac nfu
dac nfu+1
law 2000
dac nb1
dac nb1 1
xct mhs
dac nh2
dac nh2 1
jmp m10

```

/ control word get routines

```
mg1,    dap mg3
        iot 11
        jsp 8a
8j1,    0
        dac 11
        iot 111
        jsp 8a
8j2,    0
        lio 11
        ril 4s
        rcr 4s
mg3,    jmp .
mg2,    dap mg4
        lat
        swap
mg4,    jmp .
idl,    idx mth
        idx mfu
        idx mtr
        idx mco
        idx mot
        idx mom
        idx mh1
        idx mh2
        idx mh3
        idx mh4
ids,    idx mx1
        idx my1
        idx ma1
        idx mb1
        idx mdy
        idx mdx
ml1,    lac .           / 1st control word
        sza i           / zero if not active
        jmp mq1         / not active
        dap . 1
        jmp .
mb1,    lac .           / alter count of number of instructions
        add mtc
        dac mtc
mq1,    idx ml1
        sad (lac atb 1
        jmp idl
        sas (lac atb nob
        jmp ids
        background
        jsp blp         / display massive star
        count mtc, .    / use up rest of time of main loop
        jmp mlc         / repeat whole works
```

```

8a,    dap 8ay
      dap 8ax
      idx 8ax
      clf 7
      spi
      stf 5
      szf 1 5
      str 6
      ril 1
      spi
      jmp 8ao

```

```

8ay,   lac
      rir 8s
      spi
      cla
      rir 1s
      spi
      law 2
      dac 1 8ay

```

```

8ad,   rir 8s
      spi
      jmp 8aq
      rir 5s
      spi
      lor (14
      rir 1s
      spi
      lor (1
      jmp 8aa

```

```

8ao,   cla
      rir 9s
      spi
      law 2
      jmp 8ad

```

```

8aq,   rir 5s
      spi
      lor (1
      rir 1s
      spi
      lor (14

```

```

8as,   rir 1s
      spi 5
      lor (4
      spi 6
      lor (10
      rir 1s
      spi 5
      lor (10
      spi 6
      lor (4

```

```

8ax,   jmp

```

```

col,      dap cox
          law 1
          add ml1
          sad (lac mtb nob
          jmp cox-1
          dap ml2
          law 1
          add mx1
          dap mx2
          law 1
          add my1
          dap my2
          law 1
          add ma1
          dap ma2
          law 1
          add mb1
          dap mb2
ml2,      lac .
          spq      / 2nd control word
          jmp mq2   / can it collide?
mx1,      lac .
          sub .     / no
mx2,      sub .     / calc if collision
          spa      / delta x
          cma      / take abs val
          dac mt1
          sub me1   / < EPSILON ?
          sma
          jmp mq2   / no
my1,      lac .
my2,      sub .
          spa
          cma
          sub me1   / < epsilon ?
          sma
          jmp mq2   / no
          add mt1
          sub me2
          spa
          jmp cox
mq2,      idx mx2   / end of comparison loop
          idx my2
          idx ma2
          idx mb2
          index ml2, (lac mtb nob, ml2
          idx cox
cox,      jmp .

```

/routine to set explosion

```
law 20
dac i mb1
dac i mb2
sex,   lac (mex 400000      /EXPLODE
      dac i ml1          / replace calc routine with explosion
      dac i ml2
      lac i mb1          / duration of explosion
mb2,   add .
      cma
      sar 8s
      add (1
ma1,   dac .
ma2,   dac .
```

/ misc calculation routines

/ explosion

```
mex,   lac i mdx
      sar 3s
      add i mx1
      dac i mx1
      lac i ndy
      sar 3s
      add i my1
      dac i my1
      law mst
      dap msh
      lac i mb1          / time involved
      cma cli-opr
      sar 3s
      dac mxc
      sub (140
      sma
      idx msh
mz1,   lac ran
      and (777
      lor (scl
      dac ml1
      random
      scr 9s
      sir 9s
msh,   xot .
ml1,   hlt
      add i my1
      swap
      add i mx1
      dpy-i 300
      count mxc, mz1
      count i ma1, mb1
      dzm i ml1
      jmp mb1

mst,   scr 1s
      scr 3s
```

/ torpedo calc routine

```
ter,      jsp col
          jmp sex-3
          count i ma1, tc1
          lac (mex 400000
          dac i ml1
          law i 2
          dac i ma1
          law 20
          dac i mb1
          jmp mb1

tc1,      lac i mx1
          sar 9s
          xct the
          add i mdy
          dac i mdy
          sar 3s
          add i my1
          dac i my1
          sar 9s
          xct the
          add i mdx
          dac i mdx
          sar 3s
          add i mx1
          dac i mx1
          dispt i, i my1, 1
          jmp mb1
```

/ hyperspace routines

/ this routine handles a non-colliding ship invisibly
/ in hyperspace

```
hp1,      count i ma1, mb1
          law hp3          / next step
          dac i ml1
          law 7
          dac i mb1
          random
          scr 9s
          sir 9s
          xct hr1
          add i mx1
          dac i mx1
          swap
          add i my1
          dac i my1
          dzm i mdx
          dzm i mdy
          xct hd2
          dac i ma1
          jmp mb1
```

/ this routine handles a ship breaking out of
/ hyperspace.

```
hp3,    jsp col
        jmp sex
        count i ma1, hp6
        law 2000
        dac i mb1
        lac i mh4
        add hur
        dac i mh4
        random
        ior (400000
        add i mh4
        sma
        jmp po1
        lac i mh1
        dac i ml1
        lac ran
        scr 9s
        sir 9s
        xct hr2
        dac i mdy
        dio i mdx
        setup hpt, 3
        lac ran
        dac i mth
hp4,    lac i mth
        sma
        sub (311040
        spa
        add (311040
        dac i mth
        count hpt, hp4
        count i mh2, hp7
        dzm i mh2
hp7,    xct hd3
        dac i mh3
hp6,    lac i mx1
        dispt i, i my1, 2
        jmp mb1
```

```

/ spaceship calc
ss1,    jsp i cwg
        dio scw
        jmp sr0

ss2,    jsp i cwg
        rir 4s
        dio scw

sr0,
sc1,    clf 6
        jsp col
        jmp sex
        lio scw
        clf 6 cla-opr /update angle
        spi
        add maa
        ril 1s
        spi
        sub maa
mom,    add .
        dac i mom
        szs 10
        jmp . 3
        dzm i mom
        ral 7s
        ril 1s
        spi
        stf 6
        lio i mfu
        spi i
        clf 6

mth,    add .
        sma
        sub (311040
        spa
        add (311040
        dac i mth
        jda sin
        dac sn
        dzm bx
        dzm by
        szs 60
        jmp bsg
        lac i mx1
        dac t1
        mul t1
        scr 1s
        dac acx
        cla
        scr 2s
        dio fox
        lac i my1
        dac t1
        mul t1
        scr 1s
        dac acy

```

```

cla
scr 2s
swap
add  $\overline{10x}$ 
swap
scr 2s
add  $\overline{acx}$ 
add  $\overline{acy}$ 
sub str
sma 1 sza-skip
jmp pof
add str
varsft
dac  $\overline{t1}$ 
jda sct
mul  $\overline{t1}$ 
undosft
scr 9s
scr 6s
szs 1 20      / switch 2 for light star
scr 2s
sza
jmp bsg
scr 1s
dio  $\overline{t1}$ 
integrate mx1,  $\overline{bx}$ 
integrate my1,  $\overline{by}$ 
bsg, lac 1 mth
jda cos
dac  $\overline{cs}$ 
sar 9s
xct sac
szf 1 6
cla
add  $\overline{by}$ 
diff_mdy, my1, (sar 3s
lac  $\overline{sn}$ 
sar 9s
xct sac
cma
szf 1 6
cla
add  $\overline{bx}$ 
diff_mdx, mx1, (sar 3s
scale  $\overline{sn}$ , 5s,  $\overline{ssn}$ 
scale  $\overline{cs}$ , 5s,  $\overline{scn}$ 
lac 1 mx1
sub  $\overline{ssn}$ 
dac  $\overline{sx1}$ 
sub  $\overline{ssn}$ 
dac str

```

```

lac i my1
add scn
dac sy1
add scn
dac sty
scale sn, 9s, ssn
scale cs, 9s, scn
dac scm
lac ssn
dac ssm
add scn
dac ssc
dac ssd
lac ssn
sub scn
dac csn
cma
dac csm
cla cli-opr
dpy-4000
mot,sp5,      jmp i .
sq6,      loh
           lio scw
           ril 2s
           spi i          / not blasting
           jmp sq9        / no tail
           ranct sar 9s, sar 4s, src
           scale sn, 8s, ssn
           scale cs, 8s, scn
sq7,      count i mfu, st2
           dzm i mfu
           jmp sq9
st2,      yincr sx1, sy1, sub
           dispt i, sy1
           count src,sq7
sq9,      count i ma1, sr5          / check if torp tube reloaded
           dzm i ma1          / prevent count around
mco,      lac .          / previous control word
           cma
           szs i 30
           clc
           and scw          / present control word
           ral 3s          / torpedo bit to bit 0
           sma
           jmp sr5          / no launch
           count i mtr, st1          / check if torpedos exhausted
           dzm i mtr          / prevent count around
           jmp sr5
st1,      init sr1, mtb nob-1          /search for unused object
sr1,      lac .
           sza i          / 0 if unused
           jmp sr2
           law i 1
           add sr1
           dap sr1
           sas (lac mtb-1
           jmp sr1
           hlt          / no space for new objects
           jmp sr5

```

```

sr2,    lac (tcr          / set up torpedo calc
        dac i sr1
        law nob
        add sr1
        dap ss3
        lio stx
ss3,    dio .
        add (nob
        dap ss4
        lio sty
ss4,    dio .
        add (nob
        dap sr6
        add (nob
        dap sr7
        add (nob
        dap sr3
        add (nob
        dap sr4
        lac sn
        xct tvl
        cma
        add i mdx
sr3,    dac .
        lac cs
        xct tvl
        add i mdy
sr4,    dac .
        xct rlt
        dac i ma1          / permit torp tubes to cool
        xct tlf          / life of torpedo
sr6,    dac .
        lac (lac mth nob-1
        sub sr1
        sal 3s
        add (30
sr7,    dap .          / length of torp calc.
sr5,    lac scw
        dac i mco
        count i mh3, mb1
        dzm i mh3
        lac i mh2
        sza i
        jmp mb1
        lac scw
        and (600000
        xor (600000
        sza
        jmp mb1
        lac i ml1
        dac i mh1
        lac (hp1 400000
        dac i ml1
        xct hd1
        dac i ma1
        law 2
        dac i mb1
        jmp mb1

```

/ here to handle spaceships dragged into star

/ spaceship in star

```
pof,      dzm i mdx
           dzm i mdy
           szs 50
           jmp po1
           lac (377777
           dac i mx1
           dac i my1
           lac i mb1
           dac ssn
           count ssn, .
           jmp mb1
```

```
po1,      lac (mex 400000          / now go bang
           dac i m11
           law i 10
           dac i ma1
           jmp mb1
```

/ outlines of spaceships

```
ot1,      111131
           111111
           111111
           111163
           311111
           146111
           111114
           700000
```

. 5/

```
ot2,      013113
           113111
           116313
           131111
           161151
           111633
           365114
           700000
```

. 5/

variables
constants

.-64/

p, . 100/ / space for patches

/display score routine

```
fss,      0          /set size of spaceship
          dap fs1
          lac fss
          dac scm
          dac ssc
          dac ssd
          dac scn
          dac csm
          cma
          dac csn
          dzm ssn
          dzm ssm
fs1,      jmp .
fi1,      law .      /set return of compiled outline
          sub c21
          dac t6
fi2,      law .
          sub c21
          dac t7
fis,      lac c23
          dac t4
          szf 3
          lio 2sc     /get score
          szf 1 3
          lio 1sc
          scl 1s
          cla
          div c12
          hlt
          sza
          jmp fx1
flr,      dio t3
          law 400
          jda fss
          law fys
          dap frt
flt,      idx t3
          cma
          dac t3
          law fus
          dap i t6
          dap i t7
fus,      lac c20
          szf 3
          cma
          dac sy1
          lac t4
          add c30
          dac t4
          dac sx1
```

```

fds,      szf 3      /display spaceship
          law not
          szf 1 3
          law not 1
          dap fug
          idx t5
          ral 9s
          cli
          dpy-4000+700
          isp t3
fug,      jmp 1 .
frt,      jmp .

fys,      law 4000
          jda fss
          law fub
          dap i t6
          dap i t7
          lac c26
          dac sx1
          lac c20
          szf i 3
          cma
          add c30
          dac sy1
          law i 2
          dac t3
          jmp fds
fub,      szf 3
          jmp . 3
          stf 3
          jmp . 2
          clf 3
          iot 11
          dio  $\overline{t1}$ 
          iot 111
          dio t2
          law 1
          and  $\overline{t1}$ 
          and  $\overline{t2}$ 
          sza
          jmp fik
          law 2
          and  $\overline{t1}$ 
          and  $\overline{t2}$ 
          sza i
          jmp fis
          law a4+1
          dap fiu
          jmp fwt
fik,      law 4
          dap fiu
fwt,      add .
          dac  $\overline{t1}$ 
          isp  $\overline{t1}$ 
          jmp .-1
fiu,      jmp .

```

```

fx1,    dio t1
        dac t3
        law fx2
        dap frt
        law 1100
        jda fss
        jmp flt
fx2,    lio t1
        jmp fkr

c12,    12
c20,    200000
c21,    21
c23,    -200000-30000
c26,    -260000
c30,    30000
t3,     0
t4,     0
t5,     0
t6,     0
t7,     0

```

mtb, / table of objects and their properties

start 4

spacewar 4.2 5/11/63 ddp : pt 1 - pass 1
spacewar 4.2 5/16/63 ddp : pt 2 - pass 1
stars by prs for s/w 2b - pass 1
spacewar 4.2 5/11/63 ddp : pt 1 - pass 2
spacewar 4.2 5/16/63 ddp : pt 2 - pass 2
stars by prs for s/w 2b - pass 2
spacewar 4.2 syms 5/2363 jem

Constants area, inclusive

from	to
2714	3042

j	6077
1m	676
1q	6117
1sc	1305
2j	6121
2m	753
2q	6141
2sc	1306
3j	6143
3m	1030
3q	6403
4j	6405
4m	1105
4q	7747
8a	1502
8aa	1552
8ad	1525
8ao	1537
8aq	1544
8ax	1564
8ay	1515
8j1	1422
8j2	1426
a	1307
a1	1277
a2	1344
a3	1375
a4	1327
a40	1302
a5	1323
a6	1336
acx	2705
acy	2707
bc1	657
bc3	663
bcc	1162
bck	646
bex	675
bey	674
bds	534
bjm	533
bkc	1163
blp	466
blx	513
bpt	514
bpx	635
bsg	2277
bx	2662
by	2663
c12	3354
c21	3356
c23	3357
c26	3360
c20	3355

c30	3361
col	1565
cos	100
cox	1645
cs	2710
csm	2654
csn	2657
csx	146
cwg	2676
cwr	40
ddd	20
fds	3253
f11	3206
f12	3211
fid	23
fie	16
fik	3334
fin	7
fis	3214
fiu	3342
fkr	3230
flo	54
flp	44
flt	3235
foo	12
fou	15
fpo	53
fpr	1164
frr	14
frr	3266
fs	4
fs1	3205
fss	3171
fub	3306
fug	3265
fus	3243
fuu	33
fwf	3336
fx	34
fx1	3343
fx2	3352
fyn	20
fys	3267
get	2677
grv	32
hd1	23
hd2	24
hd3	25
hp1	1773
hp3	2024
hp4	2062
hp6	2077
hp7	2075
hpt	2702

hr1	26
hr2	27
hur	30
idl	1440
ids	1452
iox	2706
ma1	1661
ma2	1662
maa	13
mb1	1465
mb2	1655
mco	2442
md1	1251
mdn	1255
mdx	2665
mdy	2666
me1	16
me2	17
mex	1663
mfu	2667
mg1	1417
mg2	1433
mg3	1432
mg4	1437
mh1	2671
mh2	2672
mh3	2673
mh4	2674
mhs	22
mi1	1720
ml1	1460
ml2	1607
ml0	1165
mom	2123
mot	2373
mq1	1470
mq2	1635
msh	1717
mst	1734
mt1	2700
mtb	3367
mtc	2664
mth	2137
mtr	2670
mx1	1612
mx2	1613
mxo	2701
my1	1622
my2	1623
mz1	1704
na1	3477
nb1	3527
nco	3651
ndx	3557

ndy	3607
nfu	3643
nh1	3653
nh2	3655
nh3	3657
nh4	3661
nnn	3663
nob	30
nom	3637
not	3647
ntd	2675
nth	3641
ntr	3645
nx1	3417
ny1	3447
oc	230
oc1	375
oc2	420
oc3	426
oc4	434
oc5	442
oc6	450
oc9	460
occ	2650
ocd	411
oce	402
ocg	252
och	253
oci	2651
ocj	250
ock	241
ocm	373
ocn	374
oco	264
ocp	266
ocq	265
ocr	267
ocs	222
ocx	372
ocz	227
ot1	2606
ot2	2623
p	3071
po1	2601
pof	2565
ran	31
rlt	10
sac	14
sbf	61
sc1	2111
scm	2652
scn	2641
scw	2703
sex	1651

sft	66
sl1	112
sl2	147
sl3	117
sin	106
sn	2704
sp5	2373
sq1	220
sq2	221
sq3	171
sq6	2374
sq7	2421
sq9	2437
sqt	162
sqx	174
sr1	2460
sr2	2472
sr3	2521
sr4	2525
sr5	2537
sr6	2531
sr7	2536
src	2713
sr0	2111
ss1	2103
ss2	2106
ss3	2500
ss4	2504
ssa	2660
ssc	2656
ssd	2655
ssi	2661
ssm	2653
ssn	2640
st1	2456
st2	2425
str	15
stx	2711
sty	2712
sx1	2646
sy1	2647
t1	2643
t2	2644
t3	3362
t4	3363
t5	3364
t6	3365
t7	3366
tc1	1751
ter	1736
the	21
tlr	11
tno	6
tv1	7

v2
xys
xyt

3
2642
2645