```
;*****************************************************************
;
;       MOONLANDER FOR THE GT40 TERMINAL, FIRST VERSION.
;       LEM.090,        FEBRUARY 25, 1973       5:55 PM
;       WRITTEN BY JACK BURNESS.
;               FOR FANTASTIC PROGRAMS AT REASONABLE PRICES,
;               OR EXPERT CONSULTING DONE IN EVERY PHASE OF SYSTEM
;               DESIGN AND IMPLEMENTATION, WRITE
;                       JACK BURNESS
;                       P.O. BOX 411
;                       MAYNARD, MASS.
;                                       01754
;*****************************************************************
;
;       MODIFICATIONS TO GET THIS HUMMER RUNNING UNDER RT-11
;
;                       BY  AL KOSSOW
;                       22-JANUARY-80
;
;*****************************************************************
        IN1=R0                          ;INPUT REGISTER FOR SUBROUTINE CALLING.
        IN2=R1                          ;SECOND VALUE.
        RET1=R2                         ;HIGH ORDER PORTION OF SUBROUTINE RETURN.
        RET2=R3                         ;LOW ORDER PORTION.
        TEMP=R4                         ;SCRATCH REGISTER.
        TEMP2=R5                        ;SECOND SCRATCH.
;
;       MAJOR DEFINITIONS FOR THE SYSTEM.
;
        CLKFRQ=60.                      ;FREQUENCY OF THE RUN TIME CLOCK.
        FIVESECONDS=300.                ;DEFINE FIVE SECONDS WORTH OF TIME.
        CLKFDG=1500.                    ;CLKFRQ*25.
        CLKFG2=3000.                    ;CLKFRQ*50.
        CLKFG3=600.                     ;CLKFRQ*10.
        LKS=177546                      ;CLOCK ADDRESS.
        G=32.                           ;ACTUALLY 32.174 IN FUTURE CALCULATIONS.
        G1=16087.                       ;G*500
        G2=2670.                        ;MOON GRAVITY*500. (.166 EARTH'S).
        STATUS=177776                   ;WHERE STATUS IS LOCATED.
        DISTOP=173400                   ;THE DISPLAY STOP INSTRUCTION.
        SETSVM=106120                   ;INSTRUCTION TO SET SMALL VECTOR MODE.
        SETPNT=114000                   ;BASIC SET POINT INSTRUCTION.
        DISJMP=160000                   ;BASIC DISPLAY JUMP INSTRUCTION.
        DPC=172000                      ;DISPLAY PROGRAM COUNTER.
        DSR=172002                      ;DISPLAY STATUS REGISTER.
        XSR=172004                      ;X STATUS REGISTER.
        YSR=172006                      ;Y STATUS REGISTER.
        INT=40000                       ;INTENSITY BIT.
        LEFT=20000                      ;BIT TO SET FOR LEFT X IN LONG V
        DOWN=20000                      ;BIT TO SET FOR DOWN Y IN LONG VECTOR.
        OTHER=100                       ;BIT TO SET LEFT OR DOWN FOR SHORT MODE.
        INTTWO=200                      ;INTENSIFY BIT SHIFTED OVER.
        MINTRS=10.                      ;10% IS MIN. ROCKET WILL RUN AT.
        MAXTRS=10500.                   ;MAXIMUM THRUST OF ENGINE.
        EMPTY=14300.                    ;EMPTY WEIGHT OF SHIP (18200 DESCENT FUEL).
        FUELS=30000.                    ;INITIAL START QUANTITY OF FUEL.
        DX1=0.                          ;X POSITION FOR DATA MESSAGE.
        DX2=250.
        DX3=500.
        DX4=750.
        DY1=730.                        ;Y POSITION FOR DATA MESSAGE.
        DY2=730.
        DY3=730.
        DY4=730.
```

```
        ITEMX1=900.                     ;X POSITIONS FOR CHOICE OF DISPLAY.
        ITEMX2=ITEMX1
        ITEMX3=ITEMX1
        ITEMX4=ITEMX1
        ITEMX5=ITEMX1
        ITEMX6=ITEMX1
        ITEMX7=ITEMX1
        ITEMX8=ITEMX1
        ITEMX9=ITEMX1
        ITEMXT=ITEMX1
        ITEMXE=ITEMX1
        ITEMXS=ITEMX1
        ITEMDY=22.                      ;SET UP DELTA Y VALUES NOW.
        ITEMYS=5.                       ;DEFINE BOTTOM Y NOW.
        ITEMYE=ITEMYS+ITEMDY            ;DEFINE ALL THE OTHER Y'S AS DELTA'S.
        ITEMYT=ITEMYE+ITEMDY
        ITEMY9=ITEMYT+ITEMDY
        ITEMY8=ITEMY9+ITEMDY
        ITEMY7=ITEMY8+ITEMDY
        ITEMY6=ITEMY7+ITEMDY
        ITEMY5=ITEMY6+ITEMDY
        ITEMY4=ITEMY5+ITEMDY
        ITEMY3=ITEMY4+ITEMDY
        ITEMY2=ITEMY3+ITEMDY
        ITEMY1=ITEMY2+ITEMDY
        SLFTAX=945.                     ;COORDINATES FOR THE TURNING ARROWS.
        SLFTAY=375.
        BLFTAX=945.
        BLFTAY=330.
        SRGTAX=955.
        SRGTAY=375.
        BRGTAX=955.
        BRGTAY=330.
        BARLX=947.                      ;LEFT EDGE OF THROTTLE BAR.
        BARBY=450.                      ;BOTTOM OF THE THROTTLE BAR.
        BARTY=700.                      ;TOP OF THE BAR.
        BARSIZ=250.                     ;LENGTH OF THE BAR.
        BARFDG=BARBY+25.                ;ARITHEMENTIC FUDGE FACTOR.
        BARMXR=940.                     ;WHERE UNDERLINING IS TO START.
        BARMXL=43.+LEFT                 ;LENGTH OF UNDERLINE.
        BAREST=55.+LEFT                 ;WHERE TO POSITION FOR THE LEADING BLANKS.
        BARADD=13.                      ;AMOUNT TO ADD TO Y FOR UNDERLINE.
        .PAGE
;
;
;       MISC INFORMATION.
;       ---- ------------
;               ALL VELOCITIES ARE SIGNED AND STORED IN TENTHS OF FEET PER SECOND.
;               ALL ACCELERATIONS ARE 500 TIMES ACTUAL.
;               THE HORIZONTAL AND VERTICAL DISTANCE IS IN FEET (SIGNED).
;               THE FUEL IS IN TENTHS OF POUNDS.
;               WEIGHT OF THE CRAFT IS IN POUNDS, AS IS THRUST.
;               ORIENTATION OF THE SHIP IS STRICTLY COMPASS (I.E.
;               ZERO DEGREES IS AT THE TOP, 90 TO THE RIGHT, ETC.
;                       M A I N   D A T A   A R E A .
;
;               THE WORDS THAT FOLLOW ARE USED TO SAVE ALL THE STUFF
;               IN LOW CORE THAT MUST BE ALTERED TO RUN.
;
SYSTK:  .WORD                           ;SAVE THE SYSTEM STACK PTR
KBVEC:  .WORD
KBPSW:  .WORD
PFVEC:  .WORD
PFPSW:  .WORD
KWVEC:  .WORD
```

```
KWPSW:  .WORD
VTVEC:  .WORD
VTPSW:  .WORD
LPVEC:  .WORD
LPPSW:  .WORD
DPVEC:  .WORD
DPPSW:  .WORD
TURN:   .WORD   0                       ;PRESENT RATE OF TURN AS SELECTED BY USER.
ANGLE:  .WORD   0                       ;ANGLE CURRENTLY BEING USED FOR TRIG CALCULATIONS.
ANGLER: .WORD   0                       ;REMAINDER OF ANGLE FOR TURNING.
SINANG: .WORD   0                       ;SINE OF THAT ANGLE
COSANG: .WORD   0                       ;COSINE OF THAT ANGLE.
ACCEL:  .WORD   0                       ;CURRENT ACCEL OF SHIP CAUSED BY ROCKET.
HORACC: .WORD   0                       ;CURRENT HORIZONTAL ACCELERATION.
VERACC: .WORD   0                       ;CURRENT VERTICAL ACCELERATION (INCLUDING G).
HORVEL: .WORD   0                       ;CURRENT HORIZONTAL VELOCITY>
HORREM: .WORD   0                       ;REMAINDER OF HORIZONTAL VELOCITY AFTER DIVIDING.
VERVEL: .WORD   0                       ;CURRENT VERTICAL VELOCTIY.
VERREM: .WORD   0                       ;REMAINDER OF VERTICAL VELOCITY AFTER DIVIDING.
HORDIS: .WORD   0                       ;CURRENT HORIZONTAL DISTANCE.
VERDIS: .WORD   0                       ;CURRENT VERTICAL DISTANCE.
PERCNT: .WORD   0                       ;% OF THRUST FROM LIGHT BAR.
OLDPER: .WORD   0                       ;PREVIOUS % TO SAVE USELESS DISPLAY CALCULATION.
PERTRS: .WORD   0                       ;% THRUST THE USER IS REQUESTING.
THRUST: .WORD   0                       ;ACTUAL THRUST DELIVERED.
FUEL:   .WORD   0                       ;TENTHS OF # OF LBS (EARTH) OF FUEL REMIANING.
WEIGHT: .WORD   0                       ;CURRENT EARTH WEIGHT OF SHIP.
TICKS:  .WORD   0                       ;# OF TICKS. MOVED OVER TO CLOCK.
CLOCK:  .WORD   0                       ;# OF CLOCK TICKS SINCE LAST CALCULATION.
TIME:   .WORD   0                       ;ACCUMULATION OF "CLOCK". I.E. TOD.
BIGXCT: .WORD   0                       ;PRESENT POSITION OF SHIP (X RASTOR).
LEFTED: .WORD   0                       ;LEFT EDGE OF SCREEN FOR CLOSEUP WORK (DELTA X).
LEFEET: .WORD   0                       ;DISTANCE FROM LANDING THAT LEFTEDGE IS DURING CLOSEUP.
INDEXL: .WORD   0                       ;INDEX FOR FEATURE JUST TO LEFT OF SHIP.
AVERT:  .WORD   0                       ;AVERAGE RASTOR HEIGHT OF TERAIN.
AVERY:  .WORD   0                       ;AVERAGE Y HEIGHT OF THE TERRAIN.
RADARY: .WORD   0                       ;AVERAGE RADAR HEIGHT ABOVE THE TERRAIN.
LASTX:  .WORD   0                       ;USED TO KEEP TRACK OF WHERE DRAW WAS WORKING.
LASTY:  .WORD   0                       ;SAME FOR Y.
LOWY:   .WORD   0                       ;LOWER LIMIT TO SHOW INVISIBLE VECTOR.
LOWEST: .WORD   0                       ;LOWEST Y COORDINATE OF PRESENT SCREEN FIGURE.
SAVEY:  .WORD   0                       ;LOWEST Y OF LAST DRAWN SHIP.
FRAND:  .WORD   0                       ;RANDOMIZER WORD FOR ROCKET FLICKER.
FSHIFT: .WORD   0                       ;X SHIFTER FOR FLAME.
FINT:   .WORD   0                       ;INTENSITY OF FLAME.
FLINE:  .WORD   0                       ;LINE TYPE FOR FLAME.
SHIPTP: .WORD   0                       ;FLIP-FLOP TO DETERMINE WHICH BUFFER
                                        ;THE USER IS DISPLAYING SHIP FROM.
DNUM:   .WORD   0                       ;CURRENT PICTURE WE ARE DISPALYING.
DSTACK: .WORD   STACKD                  ;CURRENT DISPLAY STACK POINTER.
LPFLG1: .WORD   0                       ;CURRENT ADDRESS OF VALUE TO BE DISPLAYED.
OLDHIT: .WORD   0                       ;PREVIOUS LIGHT PEN ADDRESS COUNTER TO
HITCNT: .WORD   0                       ;AVOID SPURIOUS HITS.
MOON:   .WORD   .                       ;CURRENT STATUS OF PICTURE OF MOON.
DRAWCT: .WORD   0                       ;COUNTER FOR DRAWING THE MOON.
DRAWTY: .WORD   0                       ;LOCAL COUNTER FOR CHANGING THE INTENSITY OF
DRAWTZ: .WORD   0                       ;THE SURFACE OF THE MOON.
DINT:   .WORD   0
DTYPE:  .WORD   0
DFUDGE: .WORD   0                       ;ADDITIONAL MOON FUDGING FOR CLOSEUP.
XTYPE:  .WORD   0                       ;THIS IS THE RANDOMIZING WORD FOR
RADIUS: .WORD   0                       ;THE EXPLOSION, AND THE BLAST RADIUS.
DUSTX:  .WORD   .                       ;A RANDOM WORD FOR GENERATING DUST.
        .=.+40                          ; -- RESERVE 40 OCTAL WORDS FOR STACK
STACKD:                                 ;MAKE SURE THAT THERE IS ENOUGH SPACE FOR IT.
```

```
        .PAGE
;                               C  O  D  E
;                               -  -  -  -
;                           START OF THE SYSTEM.
START:  RESET                           ; DO A FEW HEAVY RESETS
        RESET
;
;               SAVE SYSTEM STACK POINTER
;
        MOV     SP,SYSTK
;
;               SAVE THE OLD VECTORS SO WE CAN GET BACK TO RT-11
;
        MOV     60,KBVEC
        MOV     62,KBPSW
        MOV     24,PFVEC
        MOV     26,PFPSW
        MOV     100,KWVEC
        MOV     102,KWPSW
        MOV     320,VTVEC
        MOV     322,VTPSW
        MOV     324,LPVEC
        MOV     326,LPPSW
        MOV     330,DPVEC
        MOV     332,DPPSW
;
;               INSTALL THE NEW VECTORS
;
        MOV     #POWERF,24              ; PWR FAIL VECTOR
        MOV     #340,26
        MOV     #KBDINT,60              ; KEYBOARD
        MOV     #340,62
        MOV     #TIMER,100              ; LINE CLOCK
        MOV     #300,102
        MOV     #DISPLY,320             ; VT-11
        MOV     #200,322
        MOV     #LIGHT,324
        MOV     #200,326
        MOV     #DBUSE,330
        MOV     #340,332
;
;               DISPLAY THE STARTING MESSAGE (ONCE)
;
        MOV     #STACK,SP
        MOV     #STARTM,DPC
        MOV     #100,177560             ; ENABLE KBD INTERRUPTS
        MOV     #100,LKS                ; ENABLE LINE CLOCK
        JSR     PC,DELAY                ; WAIT 10 SECS AND GET STARTED
        .WORD   10.
;
;               ONCE STARTED, ALL RESTARTS OCCUR HERE
;
RESTAR: MOV     #MOONST,IN1             ;NOW SET THE MOON AREA TO ALL DISTOP'S
        MOV     #MOONEN,IN2             ;FOLLOWED BY A ZERO. GET LOC TO GET COUNT.
        SUB     IN1,IN2                 ;AND GET # OF BYTES.
        ASR     IN2                     ;GET NUMBER OF WORDS TO CLEAR.
        ASR     IN2                     ;GET NUMBER OF DOUBLE WORDS TO CLEAR.
RESTL1: MOV     #DISTOP,(IN1)+          ;PUT IN THE DISPLAY STOP INSTRUCTIONS NOW.
        CLR     (IN1)+
        DEC     IN2
        BGT     RESTL1                  ;AND LOOP TILL DONE.
        CLR     STATUS                  ;CLEAR CPU STUFF.
        MOV     #STACK,SP               ;SET UP THE STACK POINTER NOW.
        MOV     #INIT,IN1               ;INITIALIZE THE SYSTEM NOW.
STARTL: MOV     (IN1)+,IN2              ;PICK UP NEXT COMMAND.
```

```
        BEQ     IDLE                    ;ZERO. ALL DONE.
        MOV     (IN1)+,(IN2)            ;DO COMMAND NOW.
        BR      STARTL                  ;AND LOOP FOREVER.
;
;               THIS IS THE MAIN WAIT POINT. IT IS
;               ACTIVATED IN THE NORMAL RUNNING MODE.
;               SUBROUTINE "EIDLE" CAN BE CALLED WHEN
;               IT IS TIME TO BLAST OFF FROM THE MOON
;               AFTER GETTING A HAMBURGER.
;
IDLE:   JSR     PC,EIDLE                ;WAIT FOR AN EVENT.
        BR      IDLE                    ;AND LOOP FOREVER.
EIDLE:  WAIT                            ;WAIT FOR AN INTERUPT TO OCCUR.
1$:     CMP     TICKS,#5                ;WAIT FOR CLOCK ON THE 40 AND 45
        BLT     1$
        MOV     TICKS,IN1               ;PICK UP THE NUMBER OF CLOCK TICKS NOW.
        BEQ     EXIDLE                  ;IF NONE, JUST EXIT.
        CLR     TICKS                   ;ELSE RESET THE LITTLE TICK COUNTER.
        MOV     IN1,CLOCK               ;AND THEN PUT THE # OF TICKS AWAY FOR CALCULATIONS.
        JSR     PC,SHOWSP               ;DO CALCULATIONS.
EXIDLE: RTS     PC                      ;AND THEN RETURN TO CALLER.
        .PAGE
;
;               THE INITIALIZATION CRAP.
;
INIT:   .WORD   TURN,-1.
        .WORD   ANGLE,-70.
        .WORD   ANGLER,0
        .WORD   HORREM,0
        .WORD   VERREM,0
        .WORD   HORVEL,10000.
        .WORD   VERVEL,-5000.
        .WORD   HORDIS,-22000.
        .WORD   VERDIS,23000.
        .WORD   PERCNT,75.
        .WORD   LPBARY,BARFDG+150.
        .WORD   OLDPER,-1.
        .WORD   FUEL,FUELS
        .WORD   TIME,0
        .WORD   CLOCK,0
        .WORD   TICKS,0
        .WORD   LOWY,0
        .WORD   DNUM,0
        .WORD   DSTACK,STACKD
        .WORD   STACKD,DTOP
        .WORD   LPFLG1,0
        .WORD   MOON,.
        .WORD   DPC,ITEME1
        .WORD   LKS,100
        .WORD   FSUBC,0
        .WORD   SHIPDP,0
        .WORD   MOONGO,0
        .WORD   SYSMES,0
        .WORD   DUSTON,0
        .WORD   DIALTB,ITEME1
        .WORD   DIALTB+2,ITEME3
        .WORD   DIALTB+4,ITEME8
        .WORD   DIALTB+6,ITEME9
        .WORD   ITEME1+2,117560
        .WORD   ITEME2+2,117560
        .WORD   ITEME3+2,117560
        .WORD   ITEME4+2,117560
        .WORD   ITEME5+2,117560
        .WORD   ITEME6+2,117560
        .WORD   ITEME7+2,117560
```

```
        .WORD   ITEME8+2,117560
        .WORD   ITEME9+2,117560
        .WORD   ITEMET+2,117560
        .WORD   ITEMEE+2,117560
        .WORD   ITEMES+2,117560
        .WORD   OLDHIT,.
        .WORD   HITCNT,10.
        .WORD   LPSW,DISTOP
        .WORD   0
        .PAGE
;
;               GO HERE IF THERE IS ANY ACTION FROM THE KEYBOARD
;
KBDINT: RESET
        MOV     KBVEC,60                ; RESTORE ALL THE VECTORS
        MOV     KBPSW,62
        MOV     PFVEC,24
        MOV     PFPSW,26
        MOV     KWVEC,100
        MOV     KWPSW,102
        MOV     VTVEC,320
        MOV     VTPSW,322
        MOV     LPVEC,324
        MOV     LPPSW,326
        MOV     DPVEC,330
        MOV     DPPSW,332
        MOV     #100,LKS
        MOV     #100,177560
        MOV     SYSTK,SP                ; RESTORE SYSTEM STACK
        MOV     KBVEC,-(SP)
        RTI
        .PAGE
;
;                       POWER FAIL ROUTINES HERE.
;
POWERF: MOV     #STACKD,DSTACK          ;SET UP ALL THE STACKS NOW.
        MOV     #DTOP,STACKD
        CLR     DNUM
        MOV     #ITEME1,DPC             ;THE DISPLAY IS GOING NOW.
        MOV     #100,LKS                ;SO IS THE CLOCK.
        MOV     #STACK,SP               ;NOW SO IS THE PROGRAM.
        CLR     STATUS
        JMP     IDLE                    ;AND REENTER.
;
;                       DISPLAY BUS ERROR ROUTINES.
;
DBUSE:  CLR     DNUM                    ;RECYCLE DISPLAY TO THE TOP OF THE LIST.
        MOV     IN1,-(SP)               ;MAKE IT LOOK LIKE A NORMAL INTERUPT.
        MOV     IN2,-(SP)
        MOV     #DITEM1,IN2
        JMP     DTOPOK                  ;AND REENTER IT NOW.
        .PAGE
;
;       THIS ROUTINE WILL DRAW THE SHIP RELATIVE TO IT'S CENTER.
;       THE ANGLES HAVE ALREADY BEEN GOTTEN FROM SOMEWHERE
;       ELSE AND HAVE BEEN LEFT IN COSANG AND SINANG. IN ADDITION,
;       IN1 POINTS TO THE COMMANDS ON HOW TO DRAW THE SHIP, AND IN2
;       POINTS TO WHERE TO LEAVE THEM.
;       COMMAND LIST OF THE THE FORM:
;               "ADDRESS OF ROUTINE"
;               "EXTRA DATA, IF NEEDED".
;       IF IT'S A DATA POINT TO BE ROTATED, THEN THE NEXT WORD
;       HAS THE X AND Y VALUES AS SEPARATE BYTES. THE X VALUE
;       IS THE BYTE ON THE RIGHT....... REMEMBER THAT.
;
```

```
DRAW:   MOV     IN1,TEMP                ;MOVE OVER THE INPUT LIST POINTER.
        MOV     IN2,TEMP2               ;AND THE OUTPUT LIST POINTER.
        CLR     LASTX                   ;SET TO CENTER OF OBJECT.
        CLR     LASTY
        MOV     #77777,LOWEST           ;SET UP LOWEST POINT WE HAVE DRAWN SO FAR.
DRAWLP: MOV     (TEMP)+,PC              ;GO TO FIRST INSTRUCTION.
DRAWIS: CLR     -(SP)                   ;DRAW INVISIBLE SHORT.
        JSR     PC,DRAWRT               ;ROTATE NEXT SET OF POINTS.
        BR      DRAWSC                  ;AND THEN INSERT THEM INTO LIST.
DRAWVS: MOV     #40000,-(SP)            ;DRAW VISIBLE SHORT.
        JSR     PC,DRAWRT               ;ROTATE RETURNS X IN RET1, Y IN RET2.
        CMP     LASTY,LOWY              ;SEE IF THIS LINE SHOULD BE DRAWN.
        BGE     DRAWV1                  ;YEP.
        CLR     (SP)                    ;NOPE. CLEAR OUT DISPLAY BIT.
        BR      DRAWSC
DRAWV1: CMP     RET2,LOWY               ;CHECK OTHER END NOW.
        BGE     DRAWSC                  ;IT'S ALSO OK.
        CLR     (SP)                    ;ELSE CLEAR THE DISPLAY BIT.
DRAWSC: MOV     RET1,IN1                ;MOVE OVER X.
        SUB     LASTX,RET1              ;FIGURE OUT DELTA X.
        BPL     DRAWS1                  ;IF POSITIVE, OK. ELSE
        NEG     RET1                    ;REVERSE SIGN AND SET MINUS BITS.
        BIS     #20000,(SP)             ;IN THE NEW SHORT VECTOR COMMAND.
DRAWS1: BIC     #-100,RET1              ;REMOVE STRAY BITS (IF ANY).
        SWAB    RET1                    ;GET INTO CORRECT POSITION.
        ROR     RET1
        MOV     IN1,LASTX               ;AND FINALLY UPDATE THE NEW X POSITION.
        MOV     RET2,IN2                ;AND NOW DO SAME THING WITH THE Y.
        SUB     LASTY,RET2
        BPL     DRAWS2
        NEG     RET2
        BIS     #100,(SP)
DRAWS2: BIC     #-100,RET2
        MOV     IN2,LASTY               ;UPDATE THE NEW Y NOW.
        CMP     IN2,LOWEST              ;SEE IF PRESENT Y IS LOWEST POINT ON SCREEN.
        BGE     DRAWNL                  ;NOT LOWEST.
        MOV     IN2,LOWEST              ;IF LOWEST, REMEMBER IT.
DRAWNL: BIS     RET1,RET2               ;NOW CREATE THE NEW INSTRUCTION.
        BIS     (SP)+,RET2
        MOV     RET2,(TEMP2)+           ;AND STORE IT AWAY.
        BR      DRAWLP                  ;AND FINALLY LOOP AROUND AGAIN.
DRAWIN: MOV     (TEMP)+,(TEMP2)+        ;THIS MOVES OVER ONE WORD.
        BR      DRAWLP
DRAWDN: MOV     #DISTOP,(TEMP2)+        ;THIS TERMINATES THE PICTURE
        CLR     (TEMP2)+                ;AND WILL CAUSE THE SUBROUTINE TO EXIT.
        RTS     PC
        .PAGE
;
;       THIS ROTATES THE X AND Y BYTE. (TEMP) POINTS TO THE X.
;
DRAWRT: MOVB    (TEMP)+,IN1             ;PICK UP THE X.
        MOV     SINANG,IN2
        JSR     PC,TRGMUL               ;MULTIPLY THEM.
        MOV     RET1,-(SP)              ;AND SAVE X*SINANG.
        MOVB    (TEMP),IN1              ;PICK UP THE Y.
        MOV     COSANG,IN2
        JSR     PC,TRGMUL               ;MULTIPLY THEM OUT.
        SUB     RET1,(SP)               ;X*SINANG-Y*COSANG (WE'LL REVERSE SIGNS LATER.
        MOVB    -1(TEMP),IN1            ;PICK UP THE X AGAIN.
        MOV     COSANG,IN2
        JSR     PC,TRGMUL               ;X*COSANG
        MOV     RET1,-(SP)
        MOVB    (TEMP)+,IN1             ;Y ONCE AGAIN.
        MOV     SINANG,IN2
        JSR     PC,TRGMUL               ;Y*SINE
```

```
        ADD     (SP)+,RET1              ;X*COS+Y*SIN
        MOV     (SP)+,RET2              ;X*SIN-Y*COS
        NEG     RET2                    ;Y*COS-X*SIN
        RTS     PC                      ;AND EXIT NOW.
        .PAGE
;
;       THIS ROUTINE FIGURES OUT THE ACCELERATION OF THE SHIP AND
;       THE AMOUNT OF FUEL REMAINING.
;
ROCKET: TST     FUEL                    ;SEE HOW MUCH FUEL WE HAVE ON BOARD NOW.
        BEQ     ROCKOFF                 ;NONE. TURN OFF THE ENGINE.
        MOV     PERTRS,IN1              ;PICK UP THE PERCENT THROTTLE NOW.
        MOV     #MAXTRS,IN2             ;GET THE MAXIMUM THRUST NOW.
        JSR     PC,MULTWO               ;GET PER CENT * MAXTHRUST.
        MOV     #100.,IN1               ;GET %*MAXIMUM THRUST /100.
        JSR     PC,DIVTWO
        MOV     RET2,THRUST             ;SAVE IT AWAY NOW.
        MOV     RET2,IN2
        MOV     CLOCK,IN1               ;PICK UP NUMBER OF CLOCK TICKS.
        JSR     PC,MULTWO
        MOV     #CLKFDG,IN1             ;PICK UP FREQUENCY*25.
        JSR     PC,DIVTWO               ;APPROX. 250LBS FUEL PER LB THRUST.
        SUB     FUEL,RET2               ;NOW SUBTRACT OFF REMAINING FUEL.
        BMI     FUELOK                  ;IF NEGATIVE, THEN THE FUEL IS OK.
        CLR     RET2                    ;ELSE MAKE SURE TOTAL IS ZERO.
        MOV     #DISTOP,LPSW            ;STOP THE LOW FUEL MESSAGE NOW.
        BR      FUELKO                  ;HE'S GOT NONE LEFT.
FUELOK: NEG     RET2                    ;GET FUEL LEFT
        TST     LPSW                    ;SEE IF FUEL IS ALREADY LOW.
        BEQ     FUELKO                  ;IT IS.
        CMP     RET2,#2000.             ;SEE IF UNDER 200 POUNDS.
        BGE     FUELKO                  ;NO. STILL OK.
        CLR     LPSW                    ;ELSE CLEAR THE SWITCH NOW.
        BIS     #0,DSR                  ;AND RING THE BELL <ONCE ONLY>
FUELKO: MOV     RET2,FUEL               ;AND SAVE IT AWAY.
        CLR     RET1                    ;NOW DIVIDE BY 10 TO CONVERT
        MOV     #10.,IN1                ;TENTHS OF POUNDS TO POUNDS.
        JSR     PC,DIVTWO               ;DO DIVISION NOW.
        ADD     #EMPTY,RET2             ;ADD IN EMPTY WEIGHT OF SHIP.
        MOV     RET2,WEIGHT             ;AND SAVE AWAY NOW.
        MOV     THRUST,IN1              ;FIGURE OUT ACCELERATION NOW.
        MOV     #G1,IN2
        JSR     PC,MULTWO
        MOV     TEMP,IN1                ;NOW DIVIDE BY WEIGHT.
        JSR     PC,DIVTWO
        MOV     RET2,ACCEL              ;AND SAVE THIS AWAY. A=TRS*G/W
        MOV     RET2,IN1
        MOV     SINANG,IN2              ;GET HORIZONTAL  ACCELERATION.
        JSR     PC,TRGMUL
        MOV     RET1,HORACC
        MOV     ACCEL,IN1               ;AND GET THE VERTICAL ACCELERATION.
        MOV     COSANG,IN2
        JSR     PC,TRGMUL
        SUB     #G2,RET1                ;DON'T FORGET ABOUT THE LUNAR GRAVITY.
        MOV     RET1,VERACC
        RTS     PC                      ;AND RETURN NOW.
ROCKOF: MOV     #DISTOP,LPSW            ;TURN OFF LOW FUEL MESSAGE WHEN OUT.
        CLR     THRUST                  ;IF ROCKET IS OFF, TURN OFF THRUST.
        MOV     #EMPTY,WEIGHT           ;UPDATE WEIGHT OF THE ROCKET NOW.
        ADD     FUEL,WEIGHT
        CLR     ACCEL                   ;CLEAR THE ROCKET'S ACCELERATION'S NOW.
        CLR     HORACC
        MOV     #-G2,VERACC             ;SET ROCKET TO FREE FALL.
        RTS     PC                      ;AND RETURN NOW.
        .PAGE
```

```
;
;       THIS ROUTINE WILL DRAW THE THRUST OF THE ROCKET, BASED ON
;       THE AMOUNT OF FUEL THAT IS BEING CONSUMED.
;
FLAME:  MOV     IN1,IN2                 ;MOVE OVER WHERE TO LEAVE FLAME.
        MOV     PERTRS,TEMP             ;PICK UP THE PERCENTAGE THRUST NOW.
        ASR     TEMP                    ;DIVDE BY EIGHT FOR THE TABLE LOOKUP.
        ASR     TEMP
        ASR     TEMP
        MOVB    YTHRST(TEMP),RET1       ;PICK UP THE Y LENGTH.
        INC     FRAND                   ;INCREMENT TO NEXT FLAME DIS@PLACEMENT.
        MOV     FRAND,RET2
        BIC     #-40,RET2               ;JUST THIRTY TWO VALUES POSSIBLE.
        MOVB    YUPDWN(RET2),RET2       ;PICK UP FLAME DISPLACEMENT.
        ADD     RET2,RET1               ;ADD THEM TOGETHER.
        ADD     RET1,FSHIFT             ;FIX UP HORIZONTAL FLICKER NOW.
        MOV     FSHIFT,RET2
        BIC     #-4,RET2                ;JUST 0 TO 3.
        MOV     #FLEN,TEMP              ;PICK UP #OF POINTS TO DO.
        MOV     #FLAMXS,TEMP2           ;PICK UP POINTER TO WHERE TO INSERT.
        ADD     #FLAMBT,RET2            ;LET RET2 POINT TO THE BYTE X TABLE.
FLAMLP: MOVB    (RET2)+,(TEMP2)+        ;INSERT X VALUE NOW.
        MOVB    RET1,(TEMP2)+           ;INSERT THE Y VALUE.
        ADD     #6,TEMP2                ;UPDATE PAST COMMANDS.
        DEC     TEMP                    ;AND LOOP TILL DONE.
        BGT     FLAMLP
        INC     FLINE                   ;INCREMENT THE LINE TYPE NOW.
        BIC     #-4,FLINE
        ADD     #600,FINT               ;AND THE INTENSITY NOW.
        BIC     #176177,FINT            ;AND MAKE JUST SEVEN BITS.
        MOV     #106124,FLAMEX          ;SET UP PROTOTYPE COMMAND.
        BIS     FLINE,FLAMEX            ;AND MOVE OVER THE DATA.
        BIS     FINT,FLAMEX
        MOV     #FLAMDO,IN1             ;FINALLY SET UP THE POINTER.
        JMP     DRAW                    ;AND DRAW THE FIGURE, AND RETURN
        .PAGE
;
;       THIS ROUTINE GETS THE SINE AND COSINE FROM THE TABLES.
;
TRIG:   MOV     ANGLE,IN1               ;PICK UP THE ANGLE NOW.
        BPL     TRIG2                   ;POSITIVE. SEE IF >180.
TRIG1:  CMP     IN1,#-180.              ;NEGATIVE. SEE IF BELOW -180.
        BGT     TRIG3                   ;>-180. IT'S OK.
        ADD     #360.,IN1               ;TOO SMALL. FUDGE IT AND LOOP.
        BPL     TRIG3                   ;IF STILL NEGATIVE. ELSE IT'S OK.
        BR      TRIG1
TRIG2:  CMP     IN1,#180.               ;POSITIVE. SEE IF IT'S >180.
        BLE     TRIG3                   ;NOPE. IT'S OK.
        SUB     #360.,IN1               ;DECREMENT BY ONE REVOLUTION
        BMI     TRIG2                   ;AND LOOP IF IT'S STILL TO BIG.
TRIG3:  MOV     IN1,ANGLE               ;AND NOW SAVE IT AWAY AGAIN.
        BPL     TRIG4                   ;CONVERT TO A GOOD COMPASS NOTATION.
        ADD     #360.,IN1               ;ELSE WRAP AROUND ONCE.
TRIG4:  ASL     IN1                     ;SHIFT IT LEFT BECAUSE IT'S NECESSARY.
        MOV     SINTAB(IN1),SINANG      ;NOW GET THE NECESSARY CONSTANTS.
        CMP     IN1,#540.               ;SEE IF GREATER THEN 269 DEGREES.
        BLT     TRIG5                   ;NO. IT'S ALRIGHT.
        SUB     #720.,IN1               ;TOO BIG. WRAP IT AROUND.
TRIG5:  MOV     COSTAB(IN1),COSANG      ;NOW MOVE OVER THE COSINE.
        RTS     PC                      ;AND RETURN WHEN DONE.
        .PAGE
;
;       THIS ROUTINE FIGURES OUT WHERE THE SHIP IS IN RELATIONSHIP TO THE MOON.
;
WHERE:  MOV     HORACC,IN2              ;PICK UP THE X ACCEL.
```

```
        BPL     WH1                     ;IF POSITIVE, FINE.
        NEG     IN2                     ;ELSE NEGATE IT FOR A SECOND.
WH1:    MOV     CLOCK,IN1               ;PICK UP ELAPSED TIME NOW.
        JSR     PC,MULTWO               ;MULTIPLY THEM TOGETHER.
        MOV     #CLKFG2,IN1             ;PICK UP TIMER FUDGE.
        JSR     PC,DIVTWO               ;AND DIVIDE IT OUT NOW.
        TST     HORACC                  ;GET ORIGINAL SIGN OF X.
        BPL     WH2
        NEG     RET2
WH2:    MOV     RET2,RET1               ;MOVE OVER A*T
        ASR     RET2                    ;GET .5*A*T
        ADD     HORVEL,RET2             ;GET V0+.5*A*T
        ADD     RET1,HORVEL             ;UPDATE THE HORIZONTAL VELOCITY NOW.
        MOV     RET2,-(SP)              ;SAVE SIGN OF VELOCITY FOR TEST LATER.
        MOV     RET2,IN2                ;VEL IN FPS*10
        BPL     WH3
        NEG     IN2
WH3:    MOV     CLOCK,IN1               ;NOW GET THE DISTANCE COVERED.
        JSR     PC,MULTWO
        MOV     #CLKFG3,IN1
        ADD     HORREM,RET2             ;BRING BACK THE REMAINDER FROM PREVIOUS OPERATIONS.
        ADC     RET1                    ;AND DON'T FORGET THE CARRY.
        JSR     PC,DIVTWO               ;ACTUAL DISTANCE COVERED.
        TST     (SP)+                   ;RECALL OLD SIGN OF VELOCITY.
        BPL     WH4
        NEG     RET2
WH4:    ADD     RET2,HORDIS             ;UPDATE THE DISTANCE NOW.
        MOV     RET1,HORREM             ;AND SAVE THE REMAINDER NOW.
        MOV     VERACC,IN2              ;NOW DO THE UP AND DOWN PART.
        BPL     WH5                     ;SAME SORT OF CRAP FOR THE Y.
        NEG     IN2
WH5:    MOV     CLOCK,IN1
        JSR     PC,MULTWO
        MOV     #CLKFG2,IN1
        JSR     PC,DIVTWO
        TST     VERACC
        BPL     WH6
        NEG     RET2
WH6:    MOV     RET2,RET1
        ASR     RET2
        ADD     VERVEL,RET2
        ADD     RET1,VERVEL
        MOV     RET2,-(SP)              ;SAVE SIGN FOR LATER USE.
        MOV     RET2,IN2
        BPL     WH7
        NEG     IN2
WH7:    MOV     CLOCK,IN1
        JSR     PC,MULTWO
        MOV     #CLKFG3,IN1
        ADD     VERREM,RET2             ;ADD IN PREVIOUS REMAINDER NOW.
        ADC     RET1                    ;AND FIDDLE IT ACCROSS.
        JSR     PC,DIVTWO
        TST     (SP)+
        BPL     WH8
        NEG     RET2
WH8:    ADD     RET2,VERDIS
        MOV     RET1,VERREM             ;AND SAVE THE REMAINDER NOW.
        RTS     PC
        .PAGE
;
;               THIS SECTION WILL FIGURE OUT THE POSITION OF THE
;       LUNAR MODULE AND WHICH VEIW TO USE.
;
SHOWSP: MOV     #-64.,LOWY              ;SET UP ABORT LIMIT OUT OF RANGE.
        MOV     TURN,TEMP2              ;PICK UP THE RATE OF TURN.
```

```
         CLR     TURN                    ;AND TURN IT OFF NOW.
         MOV     CLOCK,IN1               ;PICK UP THE NUMBER OF CLOCK TICKS.
         MOV     TEMP2,IN2               ;AND MOVE OVER THE RATE OF TURN.
         BPL     SHOWP1                  ;IF POSITIVE, THEN OK,
         NEG     IN2                     ;ELSE NEGATE IT.
SHOWP1:  JSR     PC,MULTWO               ;MULTIPLY THEM TOGETHER.
         MOV     #CLKFRQ,IN1             ;AND THEN GET ACTUAL # OF DEGREES OF TURN.
         ADD     ANGLER,RET2             ;DON'T FORGET TO ADD IN THE REMAINDER NOW.
         ADC     RET1                    ;AND THE CARRY, ALSO.
         JSR     PC,DIVTWO
         TST     TEMP2                   ;AND FINALLY SET TO THE CORRECT SIGN.
         BPL     SHOWP2
         NEG     RET2
SHOWP2:  ADD     RET2,ANGLE              ;UPDATE THE ANGLE NOW.
         MOV     RET1,ANGLER             ;SAVE THE REMAINDER NOW.
         JSR     PC,TRIG                 ;AND GET NEW SINES AND COSINES.
         ADD     #2,SHIPTP               ;GET WHICH SHIP'S
         MOV     SHIPTP,IN2              ;BUFFER TO USE.
         BIC     #-3,IN2                 ;SO WE HAVE NO PROBLEMS.
         MOV     SHIPLC(IN2),IN2         ;AND SET UP CALL.
         MOV     #DESIGN,IN1             ;SET UP SHIP'S DESIGN.
         MOV     IN2,-(SP)               ;REMEMBER THE BUFFER FOR LATER USE.
         JSR     PC,DRAW                 ;AND DRAW IT NOW.
         MOV     (SP)+,SHIPDP            ;AND FINALLY MOVE OVER BUFFER POINTER TO CALL.
         MOV     LOWEST,SAVEY            ;AND FINALLY SAVE THE LOWEST Y DRAWN.
SHOWNT:  MOV     PERCNT,PERTRS           ;PICK UP THE PRESENT FUEL SETTING.
         JSR     PC,ROCKET               ;FIGURE OUT ROCKET COMPONENTS.
         JSR     PC,WHERE                ;AND NOW FIGURE OUT WHERE WE ARE ON MOON.
         MOV     HORDIS,TEMP             ;PICK UP X IN FEET.
         ADD     #22400.,TEMP            ;FUDGE UP DOWNRANGE CALCULATION.
         ASR     TEMP                    ;32 FEET PER RASTOR ON BIG SCALE.
         ASR     TEMP
         ASR     TEMP
         ASR     TEMP
         ASR     TEMP
         MOV     TEMP,BIGXCT             ;AND SAVE AWAY FOR LATER USE.
         MOV     TEMP,TEMP2
         ASL     TEMP                    ;NOW GET THE CURRENT RADAR HEIGHT.
         ADD     #TERAIN,TEMP
         MOV     (TEMP)+,IN1             ;LEFT TERAIN.
         ADD     (TEMP),IN1              ;RIGHT TERAIN.
         ASR     IN1                     ;AVERAGE IT OUT.
         MOV     IN1,AVERY               ;AND SAVE IT AWAY.
         SUB     VERDIS,IN1              ;AND NOW GET THE RADAR HEIGHT.
         NEG     IN1
         MOV     IN1,RADARY              ;RADAR HEIGHT IS NOW CALCULATED.
         TST     THRUST                  ;NOW SEE ABOUT THE ROCKET FLAME.
         BLE     NOFLAM                  ;NO THRUST=NO FLAME.
         MOV     #FLAMIN,IN1             ;THIS IS WHERE TO LEAVE THE FLAME.
         JSR     PC,FLAME                ;GO DRAW THE FLAME NOW.
         MOV     #FLAMIN,FSUBC           ;AND TURN ON THE FLAME AGAIN.
         BR      YSFLAM                  ;AND LEAVE FLAME ON.
NOFLAM:  CLR     FSUBC                   ;IF NO THRUST, TURN OF ROCKET.
YSFLAM:  MOV     BIGXCT,TEMP             ;PICK UP X POSITION NOW.
         BMI     OFFLFT                  ;WE'RE OFF TO THE LEFT.
         CMP     TEMP,#10.
         BLE     OFFLFT                  ;WE ARE STILL OFF TO THE LEFT.
         CMP     TEMP,#890.
         BGE     OFFRGT                  ;SEE IF WE ARE TOO FAR TO THE RIGHT.
         MOV     VERDIS,TEMP             ;WE SEEM TO BE OK. HOW ABOUT HEIGHT.
         BMI     CLSEUP                  ;IF MINUS, GO INTO CLOSEUP MODE.
         CMP     TEMP,#25000.            ;IF OFF TOP, GIVE MESSAGE.
         BGE     OFFTOP
         CMP     TEMP,#450.              ;SEE IF WE SHOULD MAGNIFY.
         BLE     CLSEUP                  ;YES. MAGNIFY.
```

```
         MOV     BIGXCT,SHOWX            ;ELSE ESTABLISH THE Y COORDINATE.
         ASR     TEMP                    ;NOW SCALE DOWN THE VERTICAL DISTANCE.
         ASR     TEMP
         ASR     TEMP
         ASR     TEMP
         ASR     TEMP                    ;32 FEET PER RASTOR AT THIS RANGE.
         ADD     #43.,TEMP               ;UPDATE
         MOV     TEMP,SHOWY
         TST     MOON                    ;IS MOON DRAWN.
         BEQ     SHWMN1                  ;YES.
         JSR     PC,DRAWM1               ;NO.
         CLR     MOON                    ;RESET FLAG.
 SHWMN1: JSR     PC,DIAL                 ;DISPLAY THE DIALS NOW.
         MOV     RADARY,TEMP
         SUB     #16.,TEMP               ;FIX FOR CENTER OF THE SHIP.
         BPL     SHWOUT                  ;IT'S OK.
         MOV     #640.,IN1               ;NOT OK. WE'VE CRASHED. BIG DISTORTION.
         JSR     PC,ALTER                ;ALTER MOON NOW.
         MOV     SP,MOON                 ;INDICATE THAT WE'VE ALTERED THE MOON.
         JSR     PC,DRAWM1               ;NOW REDRAW IT.
         CLR     MOON                    ;AND REMEMBER THAT WE'VE REDRAWN IT.
         JSR     PC,EXPLOD               ;GIVE AN EXPLOSION.
 SHWOUT: RTS     PC                      ;AND RETURN NOW.
         .PAGE
 ;
 ;               OFF SCREEN ROUTINES GO HERE......
 ;
 OFFLFT: MOV     #13.,IN1                ;PICK UP A NEW X NOW.
         MOV     #LFTMSG,IN2             ;AND PICK UP A MESSAGE NOW.
         BR      OFFCOM                  ;AND DO THE COMMON ROUTINES NOW.
 OFFRGT: MOV     #887.,IN1               ;A RIGHT X VALUE.
         MOV     #RGTMSG,IN2
         BR      OFFCOM                  ;SAME CRAP.
 OFFTOP: MOV     TEMP,IN1                ;PRESENT X VALUE.
         MOV     #TOPMSG,IN2             ;TOP MESSAGE.
 OFFCOM: ASL     IN1                     ;GET A NEW HORDISTANCE.
         ASL     IN1
         ASL     IN1
         ASL     IN1
         ASL     IN1
         SUB     #22400.,IN1
         MOV     IN1,HORDIS              ;SET DISTANCE.
         MOV     IN2,SYSMES              ;SET MESSAGE NOW.
         CLR     FUEL                    ;TURN OFF FUEL NOW.
         CLR     HORVEL                  ;CLEAR SIDEWAYS MOTION.
         MOV     VERDIS,RET2             ;PICK UP VERTICAL DISTANCE NOW.
         ASR     RET2                    ;DIVIDE BY 4 TO GET A REASONALBE SPEED.
         ASR     RET2
         BMI     OFFCMD                  ;IF GOING DOWN, WE ARE OK.
         NEG     RET2                    ;ELSE MAKE US GO DOWN.
 OFFCMD: MOV     RET2,VERVEL             ;AND SAVE IT AWAY NOW.
         JMP     SHOWSP                  ;AND RELOOP AND REDISPLAY SHIP NOW.
         .PAGE
 ;
 ;               THIS ROUTINE WILL DRAW THE CLOSEUP PICTURES OF
 ;               THE MOON, WHEN THEY BECOME NECESSARY. RETURN GOES
 ;               BACK TO IDLE VIA THE OLD PC ON THE STACK
 ;               <OR IT SHOULD, ANYWAY.>.
 ;
 CLSEUP: CMP     MOON,(SP)               ;SEE IF WE HAVE THE MOON IN SIGHT.
         BEQ     CLSEC1                  ;YES.
         MOV     BIGXCT,IN1              ;NO. GET CURRENT RASTOR POSITION.
         SUB     #9.,IN1                 ;NOW SET A GOOD LEFT EDGE.
 CLSEFG: MOV     IN1,LEFTEDGE            ;LEFT EDGE ESTABLISHED NOW.
         ASL     IN1                     ;NOW CONVERT EDGE TO FEET.
```

```
        ASL     IN1
        ASL     IN1
        ASL     IN1
        ASL     IN1
        SUB     #22400.,IN1
        MOV     IN1,LEFEET              ;AND SAVE THAT AWAY.
        JSR     PC,DRAWM2               ;DRAW A CLOSE-UP OF THE MOON NOW.
        MOV     (SP),MOON               ;SET FLAG TO INDICATE WE HAVE DONE SO.
        MOV     HORDIS,IN1              ;PICK UP THE HORIZONTAL POSITION NOW.
        SUB     LEFEET,IN1              ;GET FEET FROM LEFT EDGE OF SCREEN.
CLSEOK: MOV     IN1,RET2                ;NOW GET RASTOR POSITION (3/2)
        ASL     IN1
        ADD     IN1,RET2
        ASR     RET2                    ;(3/2) OF FEET=RASTOR<X>
        MOV     RET2,SHOWX              ;AND SAVE IT AWAY NOW.
        CLR     RET1                    ;NOW CONVERT THIS POSITION TO A GOOD.
        MOV     #48.,IN1                ;NOW CONVERT BACK TO A TERAIN INDEX.
        JSR     PC,DIVTWO               ;DO THE DIVISION NOW.
        ADD     LEFTEDGE,RET2           ;MAKE INTO A TERAIN AND FEATURE INDEX.
        MOV     RET2,INDEXL             ;SAVE AWAY INDEX TO LEFT POSTION OF SHIP.
        ASL     RET2                    ;MULTIPLY BY TWO FOR THE TERAIN.
        MOV     RET2,TEMP               ;SAVE IT FOR A SECOND.
        MOV     #48.,IN1                ;NOW FIGURE OUT HOW MUCH SHIP IS TO LEFT
        SUB     RET1,IN1                ;OR RIGHT OF CENTER OF TERAIN. USE DIVISION REMAINDER.
        MOV     RET1,TEMP2              ;SAVE THAT AWAY.
        MOV     TERAIN(TEMP),IN2        ;PICK UP TERAIN NOW.
        JSR     PC,SGNMUL               ;MULTIPLY IT OUT NOW.
        MOV     RET2,-(SP)              ;SAVE FOR A SECOND NOW.
        MOV     TEMP2,IN1               ;RECALL REMAINDER NOW. DO RIGHT EDGE OF "TERRAIN".
        MOV     TERAIN+2(TEMP),IN2      ;OTHER HEIGHT.
        JSR     PC,SGNMUL               ;MULTIPLY THEM OUT.
        CLR     RET1                    ;NOW SET UP DIVIDE. THIS WILL CONVERT
        MOV     #48.,IN1                ;TERAIN HEIGHT*48 TO DISPLAY FUDGED HEIGHT.
        ADD     (SP)+,RET2              ;OLD FUDGED WEIGHTED HEIGHT.
        BPL     CLSEF1                  ;AVERAGE IS POSITIVE.
        NEG     RET2                    ;AVERAGE IS NEGATIVE.
        JSR     PC,DIVTWO               ;DO THE DIVIDE NOW.
        NEG     RET2                    ;AND NEGATE THE ANSWER.
        BR      CLSEF2
CLSEF1: JSR     PC,DIVTWO
CLSEF2: MOV     RET2,TEMP               ;MOVE OVER HEIGHT FOR DFAKE TO USE.
        ASR     RET2                    ;NOW CONTINUING FUDING TERRAIN BY KNOWN STANDARDS.
        ASR     RET2
        MOV     RET2,AVERY              ;SAVE AWAY AVERAGE TERAIN HEIGHT.
        JSR     PC,DFAKE                ;GET ACTUAL RASTER HEIGHT ABOVE THE MOON.
        MOV     TEMP,AVERT              ;SAVE IT AWAY NOW.
        MOV     VERDIS,IN1              ;PICK UP THE DISTANCE NOW.
        MOV     IN1,RET2                ;PREPARE TO CONVERT TO RASTORS.
        ASL     RET2                    ; THE OLD 3/2 GAMBIT.
        ADD     IN1,RET2
        ASR     RET2
        ADD     #23.,RET2               ;TO CONVERT TO VIEWING RASTORS.
        MOV     RET2,SHOWY              ;SAVE IT AWAY NOW.
        ADD     #24.,SHOWY              ;EXCEPT FOR THE FUDGE.
        SUB     TEMP,RET2               ;GET # OF RASTORS TO GO.
        MOV     RET2,TEMP               ;MOVE BACK OVER.
        BPL     CLSEF3                  ;IF POSITIVE, ALL IS WELL.
        NEG     RET2                    ;ELSE NEGATE IT NOW.
CLSEF3: CLR     RET1                    ;PREPARE TO DO THE DIVIDE.
        MOV     #3.,IN1                 ;MAKE IT THREE HALFS.
        ASL     RET2
        JSR     PC,DIVTWO
        TST     TEMP                    ;GET BACK THE ORIGINAL SIGN NOW.
        BPL     CLSEF4
        NEG     RET2
```

```
CLSEF4: MOV     RET2,RADARY             ;AND THIS IS THE RADAR HEIGHT.
        JSR     PC,INTEL                ;FIGURE OUT IF ANYTHING GREAT IS TO HAPPEN.
        JSR     PC,DUST                 ;AND ALSO IF WE SHOULD SHOW DUST.
AHNONE: RTS     PC                      ;AND WHEN DONE, RETURN NOW.
CLSEC1: MOV     HORDIS,IN1              ;MOON ALREADY BEING DISPLAYED.
        SUB     LEFEET,IN1              ;SEE IF IT'S STILL ON THE SCREEN.
        CMP     IN1,#30.                ;TO CLOSE TO LEFT?
        BLE     CLOL                    ;YES. SHIFT TO RIGHT.
        CMP     IN1,#580.               ;TOO CLOSE TO LEFT
        BLT     CLSEOK                  ;NOPE. DISPLAY AS IS.
        MOV     BIGXCT,IN1              ;SHIFT EVERYTHING LEFT NOW.
        DEC     IN1
        BR      CLSEFG                  ;SHOULD BE POIFECT.
CLOL:   MOV     BIGXCT,IN1              ;SHIFT EVERYTHING RIGHT IF TOO CLOSE TO THE LEFT.
        SUB     #17.,IN1
        BR      CLSEFG                  ;FUDGE IT NOW.
        .PAGE
;
;               GENERAL INTELIGENCE SECTION...
;               THIS CODE GOES HERE SO THAT IT MAY BE
;               REFERENCED BY SIMPLE BRANCH INSTRUCTIONS
;               FROM THE MAIN SECTION "INTEL".
;
AHMAC:  TST     #MACON                  ;OVER MACDONALD'S. SEE IF DRAWN.
        .=.-2                           ;DEFINE MACON AS BEING HERE.
MACON:  .WORD   0                       ;AND INITIALIZE IT TO ZERO.
        BEQ     AHNONE                  ;NO, NOT DRAWN. EXIT NOW.
        CMP     TEMP2,#30.              ;YES. SEE HOW CLOSE WE ARE.
        BGT     AHNONE                  ;NOT TOO CLOSE. ALL IS WELL.
        CLR     #MACTHR                 ;TOO CLOSE. WE'VE CRASHED INTO IT.
        .=.-2                           ;AND LIKEWISE DEFINE THE MAC DESTROYED
MACTHR: .WORD   MACTHR                  ;FLAG AS STILL HAVING MACDONALDS.
        MOV     #MACDED,SYSMES          ;TELL HIM WE'VE CRASHED INTO IT.
        JMP     QUICK                   ;AND KILL HIM OFF.
AHROCK: CMP     TEMP2,#15.              ;ARE WE TOO LOW OVER THE ROCK?
        BGT     AHNONE                  ;NOPE.
        CMP     VERVEL,#-600.           ;GREATER THAN CRASH SPEED?
        BLE     GODEAD                  ;YES. HE'S DEAD.
        MOV     #ROCKMS,SYSMES          ;START DISPLAYING THE ROCK MESSAGE NOW.
        CLR     SHIPDP                  ;STOP DISPLAYING THE SHIP NOW.
        JSR     PC,EXPLOD               ;EXPLOD NOW
AHSP1:  CMP     TEMP2,#26.              ;TOO HIGH?
        BGT     AHNONE                  ;IT'S ALRIGHT.
        CMP     TEMP,#-600.             ;TOO FAST.
        BLE     GODEAD
        MOV     #OLDMS,SYSMES           ;DISPLAY THE OLD SHIP MESSAGE.
        MOV     #3,-(SP)                ;SET UP A LEFT SHIP
        TST     HORVEL                  ;GET HORIZONTAL VELOCITY.
        BMI     AHSP1M                  ;DIRECTION OK.
        INC     (SP)                    ;MAKE IT RIGHT TILTING.
AHSP1M: MOV     IN1,-(SP)               ;PUSH THE INDEX ONTO THE STACK.
        JSR     PC,PUTFET               ;PLACE IN FEATURE TABLE NOW.
        SUB     #16.,SHOWY              ;BRING DOWN THE OLD SHIP.
        JSR     PC,DRAWM2               ;REDRAW THE MOON NOW.
        JSR     PC,EXPLOD               ;EXPLOD THE SHIP NOW.
AHFLG:  CMP     TEMP2,#26.              ;ARE WE TOO HIGH
        BGT     AHNONE                  ;YEP.
        TST     THRUST                  ;ENGINE ON?
        BEQ     AHNONE                  ;NOPE. EXIT.
        CLR     -(SP)                   ;CLEAR THE FLAG NOW.
        MOV     IN1,-(SP)
        JSR     PC,PUTFET               ;CLEAR INDICATOR FLAG NOW.
        MOV     #FLAGMS,SYSMES          ;GIVE THE FLAG MESSAGE NOW.
        JSR     PC,DRAWM2               ;REDRAW THE MOON NOW.
        BR      AHNONE                  ;AND CONTINUE CHECKING CRAP.
```

```
AHOLDS: CMP     TEMP2,#17.              ;SEE IF ON CRASHED SHIP.
        BGT     AHNONE                  ;NOPE.
        CMP     TEMP,#-600.             ;SEE IF TOO FAST.
GODEAD: BLE     DEAD                    ;ELSE DEAD.
        MOV     #OLDTLT,SYSMES          ;GIVE MESSAGE.
        JSR     PC,EXPLOD               ;AND EXPLOD NOW.
        .PAGE
;
;               ROUTINE WHICH FIGURES OUT WHAT TO DO NOW.
;
INTEL:  JSR     PC,DIAL                 ;DISPLAY USER'S DATA NOW.
        MOV     RADARY,TEMP2            ;PICK UP ALTITUDE NOW.
        BMI     INTELM                  ;IT'S NEGATIVE. FIGURE OUT SOMETHING SPECIAL.
        CMP     TEMP2,#3.               ;ARE WE CLOSE.
        BLE     VERYLOW                 ;WE CERTAINLY ARE.
        MOV     VERVEL,TEMP             ;NOT TOO LOW. GET VELOCITY NOW.
        CMP     TEMP,#-600.             ;SEE IF <=60 FPS
        BLT     AHAH                    ;YES. IS HE SCREWED UP.
        CMP     TEMP,#-300.             ;HOW ABOUT 30 FPS
        BLT     AHAH2                   ;HE'S NOT TOO SCREWED UP <YET!>
        CMP     TEMP,#-150.             ;HOW ABOUT FIFTEEN FPS.
        BLT     AHAH3                   ;GIVE ANOTHER MESSAGE.
        CMP     SYSMES,#N2FAST          ;WAS A NOT TOO FAST MESSAGE THE
        BNE     AHAHC                   ;ONE ISSUED. NO.
        CLR     SYSMES                  ;YES. DELETE IT NOW.
AHAHC:  MOV     INDEXL,IN1              ;PICK UP FEATURE INDEX NOW.
        MOV     IN1,-(SP)               ;PUSH DESIRED ITEM ONTO STACK.
        JSR     PC,GETFET               ;GET THE FEATURE NOW.
        MOV     (SP)+,IN2               ;AND POP THE ITEM OFF THE STACK.
        ASL     IN2                     ;MULTIPLY IT BY TWO.
        JMP     @AHTAB(IN2)             ;WE'VE GOT IT NOW.
AHAH:   MOV     #VFAST,SYSMES           ;GIVE A VERY FAST MESSAGE NOW.
        BR      AHAHC                   ;AND DO SOMETHING ELSE.
AHAH2:  MOV     #FAST,SYSMES            ;GIVE A FAST MESSAGE.
        BR      AHAHC
AHAH3:  MOV     #N2FAST,SYSMES          ;GIVE A "TAKE IT EASY MESSAGE".
        BR      AHAHC
INTELM: CMP     TEMP2,#-10.
        BLE     DEAD                    ;TOO FAR DOWN. HE'S DEAD.
VERYLO: CLR     PERCNT                  ;TURN OFF FUEL NOW.
        MOV     #BARBY,LPBARY           ;AND INIDCATE IT TO HIM.
        CLR     FSUBC                   ;TERMINATE FLAMES.
        CLR     DUSTON                  ;AND THE DUST.
        TST     TEMP2                   ;GET ALTITUDE NOW.
        BEQ     VD                      ;WE'RE DOWN.
        BPL     AHAHC                   ;IF POSITIVE, LOOK FOR FEATURES.
VD:     MOV     VERVEL,TEMP             ;PICK UP THE VERTICAL VELOCITY NOW.
        CMP     TEMP,#-600.             ;SEE IF GREATER THAN 60 FPS.
        BLE     DEAD                    ;YES. SPLAT
        CMP     TEMP,#-300.             ;HOW ABOUT 30FPS
        BLE     CRIPPLED                ;YES. HE'S CRIPLLED.
        CMP     TEMP,#-150.             ;HOW ABOUT 15FPS
        BLE     ROUGH                   ;IT'S A ROUGH ONE.
        CMP     TEMP,#-80.              ;HOW ABOUT 8FPS.
        BLE     GOOD                    ;IT'S A GOOD ONE.
        MOV     #GREATM,IN1             ;ISSUE GREAT MESSAGE
        BR      MAYBE                   ;AND NOW CHECK HORIZONTAL VELOCITY.
GOOD:   MOV     #GOODM,IN1              ;A GOOD LANDING.
        BR      MAYBE
ROUGH:  MOV     #ROUGHM,IN1             ;A ROUGH LANDING.
        BR      MAYBE
CRIPPL: MOV     #CRIPM,IN1              ;CRIPPLED. GIVE THAT MESSAGE.
        BR      MAYBE
DEAD:   MOV     #DEADM,SYSMES           ;ISSUE THE DEAD MESSAGE NOW.
QUICK:  MOV     #32.,IN1                ;ALTER THE SURFACE NOW.
```

```
         JSR     PC,ALTER
         JSR     PC,DRAWM2               ;REDRAW THE MOON NOW.
         CLR     SHIPDP                  ;STOP DISPLAYING THE SHIP.
         JSR     PC,EXPLOD               ;EXPLOD NOW.
MAYBE:   CLR     FSUBC                   ;TURN OFF FLAME IMAGE NOW.
         MOV     IN1,SYSMES              ;DISPLAY THE CALLER'S MESSAGE NOW.
         CMP     HORVEL,#100.            ;SEE IF TOO FAST SIDEWAYS.
         BGT     MAYBEB                  ;TOO FAST.
         CMP     HORVEL,#-100.           ;HOW ABOUT LEFT.
         BLT     MAYBEB                  ;TOO FAST.
         CMP     ANGLE,#-15.             ;SEE IF THE ANGLE IS OK.
         BLT     MAYBEC                  ;TOO FAR LEFT.
         CMP     ANGLE,#15.              ;RIGHT OK?
         BGT     MAYBEC                  ;YES. TIP HIM OVER.
         MOV     INDEXL,IN2              ;NOW FIGURE OUT IF TERAIN IS TOO ROUGH.
         ASL     IN2
         MOV     TERAIN+2(IN2),IN1       ;RIGHT TERAIN.
         SUB     TERAIN(IN2),IN1         ;LEFT TERAIN.
         MOV     IN1,RET1                ;MOVE OVER FOR ABSOLUTE VALUE.
         BPL     MAYBEP                  ;TAKE ABSOLUTE VALUE NOW.
         NEG     RET1
MAYBEP:  CMP     RET1,#48.               ;SEE IF GREATER THAN A 48 FOOT DROP.
         BGE     MAYBED                  ;YES. A BADDY.
                                         ;PLANT THE FLAG NOW!!!!!!!
                                         ; <OR GET A HAMBURGER>.
         JSR     PC,PALSY                ;WITH THE SPASTIC MAN.
MAYBEB:  MOV     HORVEL,IN1              ;PICK UP DIRECTION TO TIP SHIP.
         MOV     #SIDEM,IN2              ;PICK UP POINTER TO THE MESSAGE.
         BR      TILT                    ;AND TILT THE SHIP NOW.
MAYBEC:  MOV     ANGLE,IN1               ;GET THE DIRECTION TO TILT THE SHIP.
         MOV     #ANGLEM,IN2             ;AND PICK UP THE MESSAGE NOW.
         BR      TILT                    ;AND TILT IT NOW.
MAYBED:  MOV     #BUMPYM,IN2             ;PICK UP MESSAGE NOW. IN1=CORRECT SIGN.
TILT:    MOV     IN2,FSUBC               ;DISPLAY MESSAGE FROM FLAME POINT.
         MOV     INDEXL,IN2              ;PICK UP TERAIN INDEX AGAIN.
         MOV     #3,-(SP)                ;INDICATE IT'S TIPPED TO THE LEFT.
         TST     IN1                     ;PICK UP DIRECTION AGAIN.
         BMI     TILT1                   ;NO, MAKE IT TO THE RIGHT.
         INC     (SP)
TILT1:   MOV     IN2,-(SP)               ;AND NOW UPDATE THE FEATURE TABLE.
         JSR     PC,PUTFET
         ASL     IN2                     ;NOW GET TERAIN CHARACTERISTICS.
         MOV     TERAIN+2(IN2),RET1      ;RIGHT TERAIN.
         SUB     TERAIN(IN2),RET1        ;LEFT TERAIN.
         MOV     RET1,RET2               ;NOW WE'LL CONVERT FEET DIFFERENCE TO AN ANGLE.
         ASL     RET2                    ;MULTIPLY IT BY 3
         ADD     RET1,RET2               ;OK
         ASR     RET2                    ;NOW DIVIDE BY FOUR.
         ASR     RET2
         SUB     RET2,RET1               ;SUBTRACT FROM THE ORIGINAL.
         MOV     RET1,RET2               ;NOW HALF ANSWER AGAIN.
         ASR     RET2
         ADD     RET2,RET1               ;AND ADD IN NEW ANSWER. RESULT
                                         ;IS NUMBER OF DEGREES TO TILT.
         BPL     TILT2                   ;NOW SEE IF RESULT IS GREATER THAN
         CMP     RET1,#-45.              ;THE ABS(45).
         BGE     TILT3                   ;IF OK, THEN LEAVE ALONE.
         MOV     #-45.,RET1              ;ELSE SET IT TO -45.
         BR      TILT3
TILT2:   CMP     RET1,#45.               ;SEE IF IT'S GREATER THAN 45 DEGREES.
         BLE     TILT3                   ;NO, IT'S OK.
         MOV     #45.,RET1               ;ELSE FIX IT UP NOW.
TILT3:   MOV     #90.,RET2               ;PICK UP FALLEN SHIP ANGLE.
         TST     IN1                     ;GET WHICH WAY TO FALL.
         BPL     TILT4                   ;IT'S OK.
```

```
        NEG     RET2                    ;NEGATE IT NOW.
TILT4:  ADD     RET2,RET1               ;GET THE NEW SHIP ANGLE NOW.
        MOV     RET1,ANGLE              ;UPDATE SHIP ANGLE.
        MOV     SHIPTP,IN2              ;NOW TOGGLE SHIP SWITCH.
        ADD     #2,IN2
        BIC     #-3,IN2
        MOV     IN2,SHIPTP
        MOV     SHIPLC(IN2),IN2         ;PICK UP BUFER POINTER NOW.
        MOV     IN2,-(SP)               ;AND SAVE IT AWAY FOR LATER USE.
        JSR     PC,TRIG                 ;GET THE SINE AND COSINE NOW.
        MOV     #DESIGN,IN1             ;HOW TO DRAW SHIP.
        JSR     PC,DRAW                 ;DRAW IT NOW.
        MOV     (SP)+,SHIPDP            ;SHOW NEW SHIP NOW.
        SUB     #7.,SHOWY               ;BUT BRING IT DOWN A LITTLE BIT.
        JSR     PC,DELAY                ;DELAY FOR A WHILE
        .WORD   10.
        .PAGE
;
;               THIS ROUTINE WILL KICK UP DUST ON THE MOON.
;
DUST:   CMP     RADARY,#150.            ;SEE IF WE ARE CLOSE TO THE MOON NOW.
        BGE     NODUST                  ;DON'T GENERATE ANY DUST NOW.
        MOV     #EXLIST,TEMP2           ;PICK UP ADDRESS OF WHERE TO LEAVE DUST.
        MOV     PERTRS,TEMP             ;GET THE TRUST NOW.
        CMP     TEMP,#63.               ;SEE IF MORE THAN 63%
        BLE     DUSTB1                  ;NO. OK.
        MOV     #63.,TEMP               ;IF MORE, SET TO 63% FOR SCALING.
DUSTB1: ROL     TEMP                    ;BECAUSE WE WILL USE MAGNITUDE TO
        ROL     TEMP                    ;CONTROL INTENSITY.
        ROL     TEMP                    ;MOVE OVER TO INTENSITY SPOT.
        ROL     TEMP
        BIC     #176177,TEMP            ;CLEAR OUT ANY STRAY BITS NOW.
        BIS     #116120,TEMP            ;NOW SET THE POSITIONING MASK.
        MOV     TEMP,(TEMP2)+           ;PLACE IN THE LIST NOW.
        CMP     ANGLE,#45.              ;SEE IF THE ANGLE IS TOO GREAT NOW.
        BGT     NODUST                  ;YEP.
        CMP     ANGLE,#-45.
        BLT     NODUST                  ;SAME HERE.
        MOV     SINANG,IN2              ;NOW GET THE PLACE TO PUT THE DUST
        BPL     DUSTP1                  ;BY FIGURING OUT THE TANGENT OF
        NEG     IN2                     ;OF THE ANGLE. MAKE POSITIVE
DUSTP1: MOV     SHOWY,IN1               ;SO DIVIDE WON'T BLOW. GET RASTOR Y NOW.
        SUB     AVERT,IN1               ;SUBTRACT OFF THE TERAIN HEIGHT NOW.
        MOV     IN1,TEMP                ;SAVE FOR A SECOND.
        JSR     PC,MULTWO               ;GET PRODUCT.
        MOV     COSANG,IN1              ;GET COSINE NOW.
        JSR     PC,DIVTWO               ;RET2=TAN(ANGLE)*DELTAY
        ADD     RET2,TEMP               ;TEMP=DELTAX+DELTAY=TOTAL FLAME DISTANCE.
        TST     SINANG                  ;FIGURE OUT WHICH WAY DUST IS TO GO.
        BMI     DUSTP2                  ;CORRECT DIRECTION NOW.
        NEG     RET2                    ;ELSE NEGATE IT NOW.
DUSTP2: ADD     SHOWX,RET2              ;CENTER OF DUST NOW.
        MOV     RET2,(TEMP2)+           ;INSERT INTO THE BUFFER.
        MOV     AVERT,(TEMP2)+          ;INSERT THE Y VALUE NOW.
        MOV     #130000,(TEMP2)+        ;INSERT RELATIVE POINT COMMAND INTO THE BUFFER.
        SUB     #150.,TEMP              ;GET APPROXIMATE DISTANCE TO GROUND.
        BPL     NODUST                  ;MORE THAN 50 FEET AWAY. PRODUCE NO DUST.
        NEG     TEMP                    ;MAKE THE DISTANCE POSITIVE AGAIN.
        MOV     TEMP,IN1                ;NOW MULTIPLY IT BY THE % THROTTLE.
        MOV     PERTRS,IN2
        JSR     PC,MULTWO               ;NOW HAVE ANUMBER BETWEEN 0 AND 5000
        ASR     RET2                    ;NOW BRING IT DOWN TO A CIVILIZED NUMBER.
        ASR     RET2                    ;BY DIVIDING BY 16.
        ASR     RET2
        ASR     RET2
```

```
         BEQ     NODUST                  ;IF ZERO, GENERATE NO DUST.
         MOV     RET2,-(SP)              ;SAVE THE COUNTER ON THE STACK.
         MOV     #STACK-200.-EXLIST-12.,RET1     ;PICK UP SIZE OF DUST AREA.
         ASR     RET1                    ;GET NUMBER OF DOUBLE WORD ENTRIES.
         ASR     RET1
         CMP     RET1,(SP)               ;SEE IF CALCULATED NUMBER IS OK.
         BHIS    DUSTWF                  ;IF GREATER OR EQUAL, ALL IS WELL.
         MOV     RET1,(SP)               ;ELSE SET A FINITE LIMIT NOW.
 DUSTWF: MOV     DUSTX,RET1              ;PICK UP A RANDOM DUST GENERATING WORD.
         MOV     #-100,RET2              ;AND LEAVE A GOOD MASK IN RET2.
 DUSTL:  ADD     TIME,RET1               ;GENERATE THE RANDOM WORD NOW.
         INC     RET1
         BIC     RET2,RET1               ;NOW LEAVE JUST LOW SIX BITS.
         MOVB    YTHRST(RET1),IN1        ;PICK UP AN X NOW.
         ADD     VERACC,RET1             ;NOW A LITTLE MORE RANDOMIZING.
         BIC     RET2,RET1               ;INDEX SET TO PICK UP Y.
         BIC     RET2,IN1                ;MAKE X JUST SIX BITS.
         SWAB    IN1                     ;X OVER TO LEFT.
         ROR     IN1                     ;AND INTO POSITION NOW.
         COM     TEMP                    ;COMPLEMENT X DIRECTION NOW.
         BIC     #-20001,TEMP            ;LEAVE JUST THE SIGN BIT.
         BIS     #40000,TEMP             ;SET THE INTENSITY BIT NOW.
         BIS     TEMP,IN1                ;AND SET THEM IN THE X NOW.
         MOVB    YTHRST(RET1),IN2        ;PICK UP THE Y NOW.
         BIC     RET2,IN2                ;Y IS ALWAYS POSITIVE.
         BIS     IN2,IN1                 ;MAKE THE COMMAND NOW.
         MOV     IN1,(TEMP2)+            ;AND STICK IT AWAY.
         ADD     #20100,IN1              ;FLIP X AND Y SIGNS.
         BIC     #140000,IN1             ;TURN OFF DISPLAY BIT.
         MOV     IN1,(TEMP2)+            ;AND SAVE AWAY NOW.
         DEC     (SP)                    ;DECREMENT THE COUNTER NOW.
         BGT     DUSTL                   ;LOOP AROUND UNTIL DONE.
         MOV     #DISTOP,(TEMP2)+        ;TERMINATE THE LIST NOW.
         CLR     (TEMP2)                 ;WITH A DISPLAY STOP INSTRUCTION
         MOV     RET1,DUSTX              ;SAVE DUSTX NOW.
         TST     (SP)+                   ;INCREMENT THE COUNTER NOW.
         MOV     #EXLIST,DUSTON          ;TURN ON THE DUST NOW.
         RTS     PC                      ;AND RETURN TO THE CALLER.
 NODUST: CLR     DUSTON                  ;TURN OFF THE DUST DISPLAY NOW.
         RTS     PC                      ;AND RETURN.

         .PAGE
 ;
 ;               THIS ROUTINE WILL PLANT AN AMERICAN FLAG ON THE MOON.
 ;                               <WITH A M*A*N, NO LESS>
 ;                               OR EVEN MORE SUPER, IT WILL BUY
 ;                               SOME HAMBURGERS.
 ;
 PALSY:  JSR     PC,DELAY                ;WAIT FOUR SECONDS BEFORE
         .WORD   -4.                     ;DOING ANYTHING GREAT.
         MOV     SHOWX,MANX              ;POSITION MAN CORRECTLY.
         MOV     SHOWY,MANY
         MOV     #MAN,DUSTON             ;SHOW THE MAN NOW.
 PALSY1: BIT     #177,TIME               ;WAIT FOR CLOCK TO LINE UP NOW.
         BNE     PALSY1                  ;WHICH SHOULD TAKE A RANDOM AMOUNT OF TIME.
         TST     MACON                   ;SEE IF THERE'S A MACDONALD'S PRESENT.
         BEQ     PALNOR                  ;NONE. NORMAL PLANT.
 ;                               MOVE THE MAN TO THE MACDONALD'S NOW!
         MOV     MACX,RET2               ;PICK UP THE MACDONALD'S X
         SUB     SHOWX,RET2              ;FIGURE OUT WHICH WAY TO GO.
         MOV     MACY,RET1               ;PICK UP THE HEIGHT OF THE MACDONALD'S.
         SUB     MANY,RET1               ;GET THE HEIGHT OF THE MAN NOW.
         ADD     #3,RET1                 ;LIFT HIM UP OFF THE FLOOR.
         MOV     RET1,-(SP)              ;AND PUSH DIFFERENCE ONTO STACK.
         BEQ     PALMX1                  ;IF NO X, THEN ONLY MOVE Y.
```

```
         JSR     PC,MOVMAN               ;MOVE THE MAN NOW.
PALMX1:  MOV     MACX,RET2               ;NOW FIGURE OUT HOW FAR TO MOVE THE MAN.
         SUB     MANX,RET2
         ADD     #25.,RET2               ;BRING HIM INTO THE PLACE.
         MOV     RET2,-(SP)              ;REMEMBER WHERE WE WERE.
         CLR     RET1                    ;NO Y MOVE.
         JSR     PC,MOVMAN               ;MOVE THE MAN NOW.
         MOV     #ORDER,SYSMES           ;ORDER THE FOOD NOW.
         JSR     PC,DELAY                ;WAIT FOR AWHILE.
         .WORD   -8.
         CLR     SYSMES                  ;REMOVE THE MESSAGE.
         JSR     PC,DELAY                ;WAIT SOME MORE NOW.
         .WORD   -2.                     ;AND THEN FINALLY START HIM BACK.
         NEG     (SP)                    ;GET THE NEGATIVE DISTANCE.
         MOV     (SP),RET2
         CLR     RET1                    ;NO Y.
         JSR     PC,MOVMAN
         MOV     (SP)+,RET2              ;AND THEN GO UP INTO THE SHIP.
         MOV     (SP)+,RET1              ;BY RECALLING THE VALUES.
         NEG     RET1                    ;INVERT SIGN, AND DON'T
         BEQ     PALMX2                  ;FORGET POSSIBILITY OF SHIP LEVEL.
         JSR     PC,MOVMAN               ;UP HE GOES NOW.
PALMX2:  JSR     PC,DELAY                ;HOLD HIM IN SHIP A FEW SECONDS.
         .WORD   -2.
         ADD     #4.,VERDIS              ;THEN LIFT HIM UP, AND TAKE HIM OFF.
         ADD     #2000.,FUEL             ;GIVE HIM A LITTLE BIT MORE FUEL.
         CLR     VERVEL                  ;AND NO UPWARD VELOCITY.
         CLR     TICKS                   ;RESET NUMBER OF TICKS THAT HAVE ELAPSED.
         MOV     #STACK,SP               ;AND SET SP TO TOP OF CORE <AGAIN>.
PALOFF:  CLR     SYSMES                  ;LOOP WITHOUT GIVING ANY MESSAGES.
         CLR     ANGLE                   ;STRAIGHT UP.
         CLR     HORVEL                  ;WITH NO SIDEWAYS MOTION.
         MOV     #30.,PERCNT             ;WITH ENOUGH THRUST TO RISE UP.
         JSR     PC,EIDLE                ;AND DO EVERYTHING RIGHT.
         TST     MOON                    ;AND WAIT TILL MOON GETS BIG.
         BNE     PALOFF                  ;AND THEN MAYBE WE'LL FALL THROUGH.
         CLR     SHIPDP                  ;WHEN MOON IS BIG, REMOVE
         CLR     FSUBC                   ;SHIP AND FLAME.
         JSR     PC,DELAY                ;AND THEN JUST WAIT FOR A FEW SECONDS.
         .WORD   3.
         .PAGE
;
;               THIS ROUTINE PLANTS AN AMERICAN FLAG ON THE MOON.
;
PALNOR:  MOV     #1,-(SP)                ;INDICATE TO MEMORY THAT SHIP HAS LANDED.
         MOV     INDEXL,-(SP)
         JSR     PC,PUTFET               ;AWAY GO THE FEATURES NOW.
         MOV     #-24.,RET1              ;PREPARE TO MOVE THE MAN OUT NOW.
         MOV     #48.,RET2
         MOV     TIME,TEMP2              ;FUDGE WHICH WAY TO MOV THE MAN.
         ROR     TEMP2                   ;BY DIDDLING LOW BIT OF TIME.
         BCC     PALN1                   ;IF LOW BIT OFF, GO RIGHT.
         NEG     RET2                    ;IF ON, GO LEFT.
PALN1:   MOV     RET2,-(SP)              ;REMEMBER IT ANYWAY.
         JSR     PC,MOVMAN               ;MOVE THE MAN NOW.
         MOV     (SP)+,RET2              ;AND THEN THE FINAL DISTANCE.
         CLR     RET1                    ;WITH NO Y.
         JSR     PC,MOVMAN
         MOV     MANX,FLAGX              ;DISPLAY THE FLAG NOW.
         MOV     MANY,FLAGY
         ADD     #20.,FLAGX              ;OFFSET IT A LITTLE BIT.
         MOV     #FLAGL,FSUBC            ;PLACE FLAG IN THE FLAME LIST.
         MOV     INDEXL,IN1              ;PICK UP THE INDEX NOW.
         JSR     PC,PALSYI               ;UPDATE IT ALSO.
         JSR     PC,PALSYI               ;SO WE KNOW WHERE TO REMEMBER FLAG.
```

```
        MOV     #2,-(SP)                ;SAVE IT AWAY NOW.
        MOV     IN1,-(SP)               ;IN THE FEATURE TABLE
        JSR     PC,PUTFET
        MOV     #MANMSG,SYSMES          ;DISPLAY THE MESSAGE NOW.
        JSR     PC,DELAY                ;WAIT TEN MORE SECONDS.
        .WORD   10.
        .PAGE
;
;               THIS ROUTINE MOVES THE MAN ALONG THE SURFACE
;               OF THE MOON. RET1=DELTA Y; RET2=DELTA X.
;               IF Y=0, THEN ONLY X IS USED. IF Y<>0, THEN ONLY
;               THE SIGN OF THE X IS USED.
;
MOVMAN: MOV     #INC,TEMP2              ;GET "INC IN1" TO INDICATE X TO RIGHT.
        TST     RET2                    ;GET X DIRECTION NOW.
        BPL     MOVM1                   ;TO THE RIGHT.
        MOV     #DEC,TEMP2              ;ELSE MAKE THE X GO THE OTHER WAY.
        NEG     RET2                    ;AND SET THE COUNTER TO GO DOWN.
MOVM1:  MOV     TEMP2,PALSYI            ;SAVE AWAY THE GENERATED INSTRUCTION NOW.
        CLR     TEMP2                   ;CLEAR THE DELTA Y ADD NOW.
        TST     RET1                    ;NOW GET THE AMOUNT OF THE DELTA "Y".
        BEQ     MOVJX                   ;NO Y. JUST X MOVE.
        INC     TEMP2                   ;SOME Y. SET TO UP.
        MOV     RET1,RET2               ;AND USE THE Y AS THE COUNTER NOW.
        BPL     MOVJX                   ;IF UP, ALL IS WELL,
        NEG     RET2                    ;ELSE NEGATE COUNT AND THE
        NEG     TEMP2                   ;AND THE DELTA Y ADD.
                                        ;RET2=+COUNT TO DO, TEMP2=DELTA Y.
MOVJX:  MOV     MANX,IN1                ;PICK UP THE PRESENT POSITION NOW.
MOVLUP: JSR     PC,PALSYI               ;MOVE OVER THE X.
        MOV     IN1,MANX                ;AND PLACE IN MEMORY.
        ADD     TEMP2,MANY              ;UPDATE THE Y ALSO.
        JSR     PC,PALSYW               ;WAIT FOR 8 CLOCK TICKS TO GO BY.
        DEC     RET2                    ;ELSE DECREMENT THE COUNTER.
        BGT     MOVLUP                  ;AND LOOP TILL DONE.
        RTS     PC                      ;AND THEN RETURN.
PALSYI: .WORD   0                       ;EITHER AN INC OR DEC 'IN1'
        RTS     PC                      ;AND THEN RETURN WITH IN1 UPDATED.
PALSYW: BIT     #7,TIME                 ;WAIT FOR NONE MULTIPLE OF 8.
        BEQ     PALSYW
        JSR     PC,DIAL                 ;AND DISPLAY THE USER'S INFORMATION NOW.
PALSYV: BIT     #7,TIME                 ;AND THEN WAIT FOR 8 MORE <OR LESS>
        BNE     PALSYV
        JSR     PC,DIAL                 ;WHILE DOING THIS, DISPLAY THE USER'S INFO.
        RTS     PC                      ;AND THEN RETURN.
        .PAGE
;
;       THESE ROUTINE WILL INSERT OR CHANGE A LUNAR
;       FEATURE OR WILL RETRIEVE IT'S VALUE.
;
PUTFET: MOV     TEMP,-(SP)              ;SAVE TEMPORARY NOW.
        MOV     4(SP),TEMP              ;PICK UP AN INDEX NOW.
        ASR     TEMP                    ;MAKE IT A BYTE INDEX.
        BCC     FETRGT                  ;IF EVEN, IT'S THE RIGHT BYTE.
        ASL     6(SP)                   ;LEFT BYTE. SHIFT OVER THE VALUE.
        ASL     6(SP)
        ASL     6(SP)
        ASL     6(SP)
        BICB    #360,FEATUR(TEMP)       ;CLEAR OUT THE BYTE NOW.
        BR      PUTCOM                  ;INSERT IT NOW.
FETRGT: BICB    #17,FEATUR(TEMP)        ;CLEAR OUT THE RIGHT BYTE NOW.
PUTCOM: BISB    6(SP),FEATUR(TEMP)      ;INSERT THE BYTE NOW.
        MOV     (SP)+,TEMP              ;RESTORE THE REGISTER NOW.
        MOV     (SP),4(SP)
        ADD     #4,SP                   ;UPDATE THE STACK AFTER MOVING OVER THE PC.
```

```
         RTS     PC                      ;RETURN NOW.
 GETFET: MOV     TEMP,-(SP)              ;SAVE A SCRATCH REGISTER NOW.
         MOV     4(SP),TEMP              ;PICK UP THE INDEX NOW.
         ASR     TEMP                    ;BYTE ADDRESS IT.
         MOVB    FEATUR(TEMP),TEMP       ;PICK UP THE FEATUR NOW.
         BCC     GETDUN                  ;IF RIGHT BYTE, NO SHIFTING IS NECESSARY.
         ASR     TEMP                    ;ELSE SHIFT IT OVER.
         ASR     TEMP
         ASR     TEMP
         ASR     TEMP
 GETDUN: BIC     #177360,TEMP            ;JUST RETURN 4 BITS NOW.
         MOV     TEMP,4(SP)
         MOV     (SP)+,TEMP              ;MOVE OVER ANSWER AND RESTORE SCRATCH
         RTS     PC                      ;REGISTER AND EXIT.
         .PAGE
 ;
 ;               THIS ROUTINE WILL "DRAW" THE MOON FROM THE TABLE OF Y VALUES.
 ;
 DRAWM1: JSR     PC,DRAWMC               ;SET UP COMMON CRAP.
         MOV     TERAIN,TEMP             ;PICK UP FIRST Y POS.
         ASR     TEMP                    ;DIVIDE BY 32
         ASR     TEMP
         ASR     TEMP
         ASR     TEMP
         ASR     TEMP
         ADD     #23.,TEMP
         MOV     TEMP,(TEMP2)+
         MOV     TEMP,LASTY              ;SAVE Y FOR TOP OF SCREEN CHECK.
         MOV     #SETSVM,(TEMP2)+        ;AFTER STORING Y POS, SET GRAPH PLOT MODE.
         MOV     #TERAIN,IN1             ;POINTER TO THE TERAIN.
 DRAW1L: ADD     #8.,IN1                 ;POSITION TO THE NEXT Y.
         MOV     (IN1),TEMP
         ASR     TEMP
         ASR     TEMP
         ASR     TEMP
         ASR     TEMP
         ASR     TEMP
         ADD     #23.,TEMP               ;SCALE IT AROUND NOW.
         JSR     PC,DRAWIC               ;INSERT ONE Y CO-ORD
         BR      DRAW1L                  ;AND LOOP TILL DONE.
         MOV     #DISTOP,(TEMP2)+        ;AND FINALLY TERMINATE THE LIST
         CLR     (TEMP2)
         MOV     #MOONST,MOONGO          ;START DISPLAYING THE MOON NOW.
         JSR     PC,DRAWRS               ;RESTORE REGISTERS
         RTS     PC                      ;AND EXIT NOW.
         .PAGE
 ;
 ;               THIS ROUTINE WILL ITIALIZE CERTAIN
 ;               REGISTERS, SAVE REGISTERS AND DO OTHER CRAP NECESSARY
 ;               FOR THE MOON DRAWING ROUTINES TO WORK.
 ;
 DRAWMC: MOV     RET1,-(SP)
         MOV     RET2,-(SP)
         MOV     IN1,-(SP)
         MOV     IN2,-(SP)
         MOV     TEMP,-(SP)
         MOV     #225.,DRAWCT            ;NUMBER OF VECTORS TO DRAW.
         CLR     MOONGO                  ;TURN OFF DRAWING OF MOON.
         CLR     MACON                   ;SET FLAG TO "NO MACDONALD'S ON SCREEN".
         MOV     #MOONST,TEMP2           ;SET UP THE POINTER NOW.
         MOV     #114124,(TEMP2)+        ;PLOT POINT MODE.
         MOV     #170240,(TEMP2)+        ;SET STATUS MODE.
         CLR     (TEMP2)+                ;SET X POSITION TO ZERO NOW.
         JMP     @10.(SP)                ;AND RETURN NOW.
 DRAWRS: MOV     (SP),12.(SP)            ;MOVE OVER RETURN ADDRESS.
```

```
        MOV     (SP)+,TEMP              ;THROW AWAY TOP ITEM.
        MOV     (SP)+,TEMP              ;AND RESTORE THE REGISTERS.
        MOV     (SP)+,IN2
        MOV     (SP)+,IN1
        MOV     (SP)+,RET2
        MOV     (SP)+,RET1
        RTS     PC                      ;AND RETURN NOW.
        .PAGE
;
;               THIS ROUTINE WILL INSERT ONE "Y" POINT INTO THE
;               MOON LIST. IT WILL RANDOMIZE THE LINE TYPE AND INTENSITY
;
DRAWIC: MOV     #1000,-(SP)             ;MAKE THE VECTOR INVISIBLE FOR NOW.
        CMP     TEMP,#1024.             ;SEE IF IT'S TOO BIG.
        BLO     DRAWI1                  ;NOT TOO BIG.
        MOV     #1023.,TEMP             ;TOO BIG, SET TO TOP, AND TURN OFF
        CMP     TEMP,LASTY              ;IS IT STILL ON TOP?
        BEQ     DRAWI3                  ;YES. DON'T DRAW THE MOUNTAIN.
DRAWI1: TST     TEMP                    ;SEE IF VALUE IS LESS THAN ZERO.
        BPL     DRAWI2                  ;YES.
        CLR     TEMP                    ;NO IT ISN'T.
        TST     LASTY                   ;WAS LAST RESULT 0?
        BEQ     DRAWI3                  ;YES. DO NOT PUT ON VISIBLE BIT.
DRAWI2: BIS     #40000,(SP)             ;TURN ON VISIBLE BIT.
DRAWI3: DEC     DRAWTY                  ;SEE WHAT TYPE OF INTENSITY
        BPL     DRWIN                   ;AND LINE TYPE IS REQUIRED, IF ANY.
        INC     DRAWTZ                  ;COUNTER OVERFLOWED. CHANGE TYPE.
        BIC     #-4,DRAWTZ
        INC     DRAWTZ
        MOV     DRAWTZ,DRAWTY           ;RESET BIGGY AND LITTLY.
        ADD     #1200,DINT              ;RANDOMIZE INTENSITY AND TYPE.
        BIC     #176177,DINT
        INC     DTYPE
        BIC     #-4,DTYPE
        MOV     DINT,(TEMP2)            ;CREATE NEW PLOT POINT INSTRUCTION.
        BIS     DTYPE,(TEMP2)
        BIS     #106004,(TEMP2)+        ;GRAPH Y WITH BLANK INTENSITY'S AND TYPES.
DRWIN:  SUB     LASTY,TEMP              ;GET THE DELTA Y.
        BPL     DRWINP                  ;FIX UP DIRECTION OF VECTOR.
        NEG     TEMP
        BIC     #-100,TEMP              ;REMOVE CRAP NOW.
        SUB     TEMP,LASTY              ;UPDATE PREVIOUS Y NOW.
        BIS     #100,TEMP
        BR      DRWINQ
DRWINP: BIC     #-100,TEMP              ;CLEAN IT UP.
        ADD     TEMP,LASTY              ;AND UPGRADE PREVIOUS Y NOW.
DRWINQ: BIS     (SP)+,TEMP              ;MAKE IT A GOOD INSTRUCTION NOW.
        MOV     TEMP,(TEMP2)+           ;AND SAVE IT AWAY NOW.
        DEC     DRAWCT                  ;DECREMENT MASTER COUNTER.
        BGT     DRAWRR                  ;NORMAL RETURN.
        ADD     #2,(SP)                 ;END RETURN.
DRAWRR: RTS     PC                      ;AND EXIT NOW.
        .PAGE
;
;               ALTER WILL ALTER THE FACE OF THE MOON. IN1
;               CONTAINS A NUMBER TO BE USED IN ALTERING IT.
;
ALTER:  MOV     BIGXCT,IN2              ;PICK UP PRESENT INDEX.
        MOV     IN2,TEMP                ;WORK BOTH FORWARD AND BACKWARD.
        ASL     IN2
        ADD     #TERAIN,IN2
        MOV     #3,-(SP)                ;INDICATE A CRASHED SHIP.
        MOV     (IN2)+,TEMP2            ;PICK UP Y.
        MOV     IN2,RET2
        SUB     (IN2),TEMP2             ;FIGURE OUT WHICH WAY TO TILT SHIP.
```

```
         BMI     ALERTP                  ;LEAVE TILTED TO LEFT.
         INC     (SP)
 ALERTP: MOV     TEMP,-(SP)              ;PUSH THE INDEX ONTO THE STACK.
         JSR     PC,PUTFET               ;AND PUT AWAY THE FEATURE.
 ALERTL: SUB     IN1,(IN2)+              ;NOW ALTER BOTH WAYS.
         SUB     IN1,-(RET2)
         ASR     IN1                     ;DIVIDE BY TWO.
         NEG     IN1                     ;AND NEGATE IT.
         BNE     ALERTL                  ;AND LOOP TILL DONE.
         RTS     PC                      ;ELSE RETURN NOW.
         .PAGE
 ;
 ;               THIS SECTION WILL CAUSE THE SKIP TO
 ;               EXPLODE ON THE SCREEN BY SHOOTING OUT A SERIES OF
 ;               "DOTS" FOR A GIVEN PERIOD OF TIME.
 ;
 EXPLOD: CLR     RADIUS                  ;SET THE RADIUS TO ZERO.
         CLR     FSUBC                   ;TURN OFF THE ROCKET NOW.
         CLR     DUSTON                  ;TURN OFF THE DUST, IF ANY.
         BIS     #0,DSR                  ;"RING THE BELL".
 EXPLD1: MOV     #EXLIST,TEMP2           ;GET POINTER TO WHERE TO SHOW EXPLOSION.
         MOV     XTYPE,TEMP              ;PICK UP RANDOMIZING WORD NOW.
         INC     TEMP                    ;AND JUST PLAY WITH YOURSELF FOR A WHILE.
         SWAB    TEMP
         ROR     TEMP
         ADD     (TEMP2),TEMP            ;ADD IN OLD COMMAND.
         BIC     #176177,TEMP            ;CLEAR OUT CRAP NOW.
         BIS     #116120,TEMP            ;MAKE IT A GOOD COMMAND NOW.
         MOV     TEMP,(TEMP2)+           ;AND INSERT IT INTO THE BUFFER.
         JSR     PC,EXGEN                ;GENERATE SOME SMOKE NOW.
         BIS     #0,DSR                  ;GIVE A GOOD BEEP NOW.
         SUB     #10.,RADIUS             ;NOW PRODUCE SOME MORE DOTS.
         JSR     PC,EXGEN
         MOV     #DISTOP,(TEMP2)+        ;TERMINATE THE PICTURE NOW.
         CLR     (TEMP2)
         MOV     #EXLIST,FSUBC           ;MAKE SURE WE'RE SHOWING IT.
         BIS     #0,DSR                  ;GIVE A RANDOM BEEP, AT TIMES.
         ADD     #33.,RADIUS             ;FINALLY INCREMENT RADIUS AND CHECK
         CMP     RADIUS,#300             ;DONE?
         BLE     EXPLD1                  ;NOPE.
         JSR     PC,DELAY                ;DELAY NOW BEFORE RESTARTING.
         .WORD   5.                      ;FIVE SECONDS.
 EXGEN:  MOV     #-30.,ANGLE             ;INITIALIZE THE ANGLE NOW.
         MOV     #241.,-(SP)             ;AND SET COUNTER TO DO -30 TO 210 DEGREES.
 EXGENL: JSR     PC,TRIG                 ;FIGURE OUT VARIOUS ANGLES NOW.
         MOV     -6(SP),IN1              ;PICK UP RANDOM INTERUPT NUMBER NOW.
         ASR     IN1                     ;JUST IN CASE IT WAS EVEN.
         INC     IN1                     ;INCREMENT IT NOW.
         ADD     TIME,IN1                ;ADD IN THE TIME ALSO.
         ADD     XTYPE,IN1               ;AND ALSO THE PREVIOUS NUMBER.
         MOV     IN1,XTYPE               ;AND SAVE IT AWAY NOW.
         BIC     #-40,IN1                ;MAKE IT BETWEEN 0 AND 31
         MOVB    YUPDWN(IN1),TEMP        ;SO THAT WE CAN USE THE FLAME TABLE.
         ADD     RADIUS,TEMP             ;GET A NEW RADIUS NOW.
         BMI     EXGENP                  ;IF MINUS, FORGET ABOUT IT.
         MOV     TEMP,IN1                ;ELSE GET THE SINES AND COSINES.
         MOV     COSANG,IN2              ;SO WE CAN GET THE X AND THE
         JSR     PC,TRGMUL               ;Y POINTS FOR THE EXPLOSION.
         ADD     SHOWX,RET1              ;WE'VE GOT OUR X NOW.
         BMI     EXGENP                  ;MINUS IS A BADDY.
         BIS     #INT,RET1               ;DON'T FORGET TO TURN ON INTENSIFY BIT.
         MOV     RET1,(TEMP2)+           ;AND PLACE IT AWAY NOW.
         MOV     TEMP,IN1                ;NOW GET THE Y.
         MOV     SINANG,IN2
         JSR     PC,TRGMUL
```

```
         ADD     SHOWY,RET1
         BMI     EXGENO                  ;IF THIS IS BAD, WE MUST BACK UP.
         MOV     RET1,(TEMP2)+
 EXGEND: INC     ANGLE                   ;NOW GET THE NEXT ANGLE.
         DEC     (SP)                    ;AND WAIT TILL DONE.
         BGT     EXGENL                  ;AND LOOP TILL DONE.
         TST     (SP)+                   ;AND THEN EXIT.
         RTS     PC                      ;THIS INSTRUCTION HELPS IF ONE LEAVES IT IN.
 EXGENO: CLR     -(TEMP2)                ;IF X IS ALREADY THERE, TAKE IT BACK.
 EXGENP: CLR     (TEMP2)+                ;INSERT DUMMY'S SO DISPLAY WON'T SCREW UP.
         CLR     (TEMP2)+
         BR      EXGEND                  ;AND SEE IF ALL DONE.
         .PAGE
 ;
 ;               NOTE, ANY ADDITIONAL CRAP YOU WISH TO PUT
 ;               IN MAY GO HERE.
 ;
 DRAWM2: JSR     PC,DRAWMC               ;THIS ROUTINE WILL THE CLOSE-UP
         CLR     DFUDGE                  ;VIEW OF THE MOON.
         MOV     LEFTEDGE,IN1            ;PICK UP LEFT SIDE OF THE SCREEN.
         ASL     IN1
         ADD     #TERAIN,IN1             ;GET POINTER TO LEFT Y.
         MOV     (IN1),TEMP              ;GET LEFT Y.
         JSR     PC,DFAKE                ;MAKE INTO GOOD RASTOR.
         TST     TEMP                    ;SEE IF ON THE SCREEN NOW.
         BPL     DRW2L5                  ;IF POSITIVE OK,
         CLR     TEMP                    ;ELSE ZERO IT OUT.
         BR      DRW2M
 DRW2L5: CMP     TEMP,#1024.             ;SEE IF TOO BIG.
         BLO     DRW2M                   ;NOPE. OK
         MOV     #1023.,TEMP             ;ELSE FIX IT UP.
 DRW2M:  MOV     TEMP,(TEMP2)+           ;INSERT STARTING Y INTO BUFFER.
         MOV     TEMP,LASTY              ;ESTABLISH LAST Y POSITION NOW.
         MOV     #SETSVM,(TEMP2)+        ;INSERT THE GRAPH PLOT MODE COMMAND <Y>.
 DRW2L:  MOV     TEMP,-(SP)              ;NOW GET Y(NEW) AND DELTA Y.
         MOV     (IN1)+,TEMP
         JSR     PC,DFAKE
         MOV     TEMP,IN2
         MOV     (SP)+,TEMP
         CLR     RET1                    ;CLEAR AWAY LEFT MOST PART FOR THE DIVIDE.
         MOV     IN1,-(SP)               ;SAVE IN1 NOW.
         MOV     #12.,IN1                ;PICK UP A TWELVE FOR THE DIVIDE.
         SUB     TEMP,IN2                ;GET DELTA Y NOW.
         BPL     DRAW2G
         SUB     #6.,IN2
         NEG     IN2                     ;MAKE IN2 POSITIVE NOW.
         MOV     IN2,RET2                ;MOVE OVER TO DIVISOR PART.
         JSR     PC,DIVTWO               ;DIVIDE BY TWO NOW.
         NEG     RET2                    ;NEGATE THE ANSWER NOW.
         BR      DRAW2H
 DRAW2G: ADD     #6.,IN2
         MOV     IN2,RET2                ;MOVE IT OVER NOW.
         JSR     PC,DIVTWO               ;DO THE DIVISION NOW.
 DRAW2H: MOV     RET2,IN2                ;MOVE OVER THE DELTA NOW.
         MOV     IN1,RET1                ;ESTABLISH THE COUNT NOW.
         MOV     (SP)+,IN1               ;RESTORE THE REGISTER NOW.
 DRAW22: INC     @#DFUDGE                ;---<<INSTRUCTION CHANGED>>---
         CMP     DFUDGE,#3.
         BLT     DRAW2V
         MOV     (PC)+,@(PC)+
         DEC     @(PC)+
         .WORD   DRAW22
         BR      DRAW2W
 DRAW2V: CMP     DFUDGE,#-3.
         BGT     DRAW2W
```

```
         MOV     (PC)+,@(PC)+
         INC     @(PC)+
         .WORD   DRAW22
 DRAW2W: ADD     DFUDGE,TEMP             ;PERTIBATE SURFACE NOW.
         ADD     IN2,TEMP                ;UPDATE TEMP NOW.
         MOV     TEMP,-(SP)              ;SAVE Y FOR A SECOND.
         JSR     PC,DRAWIC               ;INSERT
         BR      DRAW2C                  ;STILL GOING.
         BR      DRAW2X                  ;GONE.
 DRAW2C: MOV     (SP)+,TEMP              ;RECALL THE Y NOW.
         DEC     RET1                    ;SEE IF DONE WITH THE EIGHT.
         BGT     DRAW22                  ;NO.
         BR      DRW2L                   ;YES.
 DRAW2X: MOV     LEFTEDGE,RET1           ;PICK UP LEFTEDGE AGAIN.
         MOV     #19.,(SP)               ;PREPARE TO EXAMINE THE TERAIN NOW.
         MOV     #24.,IN1
         MOV     RET1,RET2               ;SET UP TWO POINTERS NOW.
         ASL     RET2
         ADD     #TERAIN,RET2
 DRAW2Y: MOV     RET1,-(SP)              ;PUSH FEATURE INDEX ONTO THE STACK.
         INC     RET1                    ;INCREMENT INDEX POINTER TO NEXT FEATURE.
         JSR     PC,GETFET               ;GET THE FEATURE NOW.
         MOV     (SP)+,TEMP              ;AND PUT IT IN A USEFUL REGISTER.
         ASL     TEMP                    ;MAKE IT INTO A GOOD INDEX.
         JSR     PC,@DRAWTB(TEMP)
         TST     (RET2)+                 ;BYPASS
         ADD     #48.,IN1
         DEC     (SP)
         BGT     DRAW2Y                  ;KEEP LOOKING NOW.
         TST     (SP)+                   ;ELSE UPDATE.
         MOV     #DISTOP,(TEMP2)+        ;TERMINATE THE DISPLAY NOW.
         CLR     (TEMP2)
         JSR     PC,DRAWRS               ;RESTORE THE REGISTERS.
         MOV     #MOONST,MOONGO          ;START DISPLAYING THE MOON NOW.
         RTS     PC                      ;AND RETURN NOW.
         .PAGE
 ;
 ;               JUMP TABLE FOR THE VARIOUS KINDS OF
 ;               CRAP ONE IS LIKELY TO FIND ON THE MOON.
 ;
 AHTAB:  .WORD   AHNONE                  ;HOW TO PROCESS THIS CRAP.
         .WORD   AHSP1
         .WORD   AHFLG
         .WORD   AHOLDS
         .WORD   AHOLDS
         .WORD   AHROCK
         .WORD   AHMAC
         .WORD   AHMAC
 DRAWTB: .WORD   DRAWRR                  ;0=NOTHING=RETURN.
         .WORD   OLDSHP                  ;1=OLD SPACE SHIP
         .WORD   FLAG                    ;2=PLANTED FLAG.
         .WORD   LEFTSP                  ;3=CRASHED SHIP ON LEFT SIDE.
         .WORD   RIGHTS                  ;4=CRASHED SHIP ON RIGHT SIDE.
         .WORD   ROCK                    ;5=A ROCK.
         .WORD   AHNONE                  ;6=LEFT OR RIGHT OF MAC'S. RETURN.
         .WORD   MACDON                  ;7=CENTER OF MAC'S. THIS COVERS 6.
 DFAKE:  MOV     TEMP,-(SP)              ;THIS ROUTINE FIGURES OUT
         ASL     TEMP                    ;Y*3/2/4 OR ((3*Y)/2)/4
         ADD     (SP)+,TEMP
         ASR     TEMP
         ASR     TEMP                    ;ALMOST THERE.
         ASR     TEMP
         ADD     #23.,TEMP               ;AND THEN IT CORRECTS THE BASE
         RTS     PC                      ;SCREEN Y.
         .PAGE
```

```
;
;               THESE ROUTINES WILL DRAW OLD AND CRASHED SHIPS
;               ONE THE SURFACE OF THE MOON.
;
OLDSHP: JSR     PC,DEADSP
        .WORD   0                       ;UPRIGHT
        .WORD   23.                     ;DISTANCE ABOVE SURFACE.
        .WORD   -24.                    ;LOWEST POINT TO DRAW.
LEFTSP: JSR     PC,DEADSP               ;LEFT TILTED SHIP.
        .WORD   -90.
        .WORD   16.
        .WORD   -18.
RIGHTS: JSR     PC,DEADSP
        .WORD   90.
        .WORD   16.
        .WORD   -18.
DEADSP: CMP     TEMP2,#MOONEN-102.      ;SEE IF ENOUGH ROOM FOR THE SHIP.
        BHI     DEADBY                  ;NOPE. EXIT.
        MOV     IN2,-(SP)
        MOV     RET1,-(SP)
        MOV     RET2,-(SP)
        MOV     TEMP,-(SP)
        MOV     IN1,-(SP)
        MOV     10.(SP),IN1             ;PICK UP OLD PC.
        MOV     #SETPNT,(TEMP2)+        ;SET POINT MODE.
        MOV     (SP),(TEMP2)+           ;MOV OVER THE X NOW.
        MOV     ANGLE,-(SP)             ;PUSH DOWN THE OLD ANGLE.
        MOV     (RET2),TEMP             ;PICK UP A Y.
        JSR     PC,DFAKE
        MOV     TEMP,-(SP)
        MOV     2(RET2),TEMP            ;NEXT Y.
        JSR     PC,DFAKE
        ADD     (SP)+,TEMP
        ASR     TEMP                    ;AVERAGE OF THE Y'S, WHICH IS CENTER.
        MOV     (IN1)+,ANGLE            ;MOVE OVER THE CALLER'S ANGLE NOW.
        ADD     (IN1)+,TEMP             ;AND CENTER THE SHIP.
        MOV     (IN1),LOWY              ;AND SET THE LOW Y ALL AT THE SAME TIME.
        MOV     TEMP,(TEMP2)+           ;INSERT THE Y NOW.
        CMP     (RET2),2(RET2)          ;SEE WHICH Y IS BIGGER.
        BEQ     DEADOK
        BHI     DEADPL
        ADD     #-22.,ANGLE
        BR      DEADOK
DEADPL: ADD     #22.,ANGLE
DEADOK: JSR     PC,TRIG                 ;FIGURE OUT SINES AND COSINES.
        MOV     TEMP2,IN2
        MOV     #DESIGN,IN1             ;SET UP DRAW CALL.
        JSR     PC,DRAW                 ;DRAW THE SHIP NOW.
        CLR     -(TEMP2)                ;REMOVE THE DISPLAY STOP INSTRUCTION.
        CLR     -(TEMP2)
        MOV     (SP)+,ANGLE             ;RESTORE THE ANGLE.
        JSR     PC,TRIG                 ;AND THE VALUES.
        MOV     (SP)+,IN1               ;POP THE VALUES NOW.
        MOV     (SP)+,TEMP
        MOV     (SP)+,RET2
        MOV     (SP)+,RET1
        MOV     (SP)+,IN2
        TST     (SP)+                   ;IGNORE ORIGINAL CALL.
DEADBY: RTS     PC                      ;AND RETURN NOW.
        .PAGE
;
;               THIS ROUTINE WILL PLANT AN AMERICAN FLAG ON THE MOON.
;
FLAG:   CMP     TEMP2,#MOONEN-FLAGEN+FLAGL-6    ;SEE IF FLAG CAN FIT.
        BHI     ROCKRT                  ;NOPE. NO ROOM. DO NOT INSERT IT
```

```
         MOV     IN1,FLAGX
         MOV     (RET2),TEMP
         JSR     PC,DFAKE                ;GET THE Y'S
         MOV     TEMP,IN2
         MOV     2(RET2),TEMP
         JSR     PC,DFAKE
         ADD     IN2,TEMP
         ASR     TEMP
         MOV     TEMP,FLAGY
         MOV     #FLAGL,IN2              ;NOW FAKE OUT A ROUTINE.
         BR      ROCKLP                  ;AND NOW ENTER ROCK PUTTER.
         .PAGE
 ;
 ;               THIS ROUTINE WILL DRAW A ROCK ON THE MOON.
 ;
 ROCK:   CMP     TEMP2,#MOONEN-ROCKEN+ROCKL-6    ;SEE IF THE ROCK CAN FIT.
         BHI     ROCKRT                  ;CAN'T FIT NOW.
         MOV     IN1,ROCKX               ;SAVE THE ROCK'S X AND Y POINT.
         MOV     (RET2),TEMP             ;AFTER FUDGING Y.
         JSR     PC,DFAKE
         MOV     TEMP,IN2
         MOV     2(RET2),TEMP
         JSR     PC,DFAKE
         ADD     IN2,TEMP
         ASR     TEMP
         MOV     TEMP,ROCKY              ;SAVE IT AWAY NOW.
         MOV     #ROCKL,IN2              ;PICK UP POINTER TO PROTOTYPE ROCK.
 ROCKLP: MOV     (IN2)+,TEMP
         CMP     TEMP,#DISTOP            ;DONE?
         BEQ     ROCKRT                  ;YES.
         MOV     TEMP,(TEMP2)+           ;NO. PUT AWAY NOW.
         BR      ROCKLP
 ROCKRT: RTS     PC                      ;RETURN NOW.
 MACDON: TST     MACTHR                  ;IS MACDONALD'S STILL IN EXISTANCE.
         BEQ     MACEX                   ;NO. EXIT.
         CMP     IN1,#25.                ;IT IS. ARE WE CENTERED ENOUGH
         BLE     MACEX                   ;TO DRAW IT. NO. TO FAR LEFT.
         CMP     IN1,#880.
         BGE     MACEX                   ;NO. TO FAR TO THE RIGHT.
         MOV     2(RET2),TEMP            ;PICK UP RIGHT Y NOW.
         CMP     (RET2),TEMP             ;AND SEE IF IT'S SMALLER THAN LEFT Y.
         BGE     MACB1                   ;IT IS.
         MOV     (RET2),TEMP             ;IT ISN'T. PICK SMALLEST Y NOW.
 MACB1:  JSR     PC,DFAKE                ;GET SCREEN COORDINATES NOW.
         MOV     TEMP,MACY               ;SAVE AWAY THE Y
         MOV     IN1,MACX                ;AND THE X ALSO.
         MOV     SP,MACON                ;AND SET THE MAC IS DRAWN FLAG.
         MOV     #DISTOP,(TEMP2)+        ;AND PLACE CALL TO IT IN THE BUFFER.
         MOV     #MACS,(TEMP2)+
 MACEX:  RTS     PC                      ;AND RETURN NOW. SIMPLE, ISN'T IT.
         .PAGE
 ;
 ;       THESE ARE SOME MISC MULTIPLY ROUTINES (AND DIVIDE) WHICH
 ;       ARE NECESSARY FOR THE SMOOTH OPERATION OF THE SYSTEM.
 ;
 MULTWO: CLR     RET2                    ;CLEAR LOW ORDER RETURN VALUE.
         CMP     IN2,IN1                 ;SEE WHICH ONE IS SMALLER.
         BHIS    MULT1L                  ;IN1 IS SMALLER (AS EXPECTED).
         MOV     IN2,RET1                ;PLACE IN2 IN RET1, AND
         BEQ     MULTDN                  ;IF ZERO, EXIT. ELSE THEN PLACE
         MOV     IN1,IN2                 ;IN1 IN IN2. <INVERT>.
         CLC                             ;CLEAR THE CARRY BEFORE WE BLOW IT!
         BR      MULTCM                  ;AND START MULTIPLY GOING.
 MULT1L: MOV     IN1,RET1                ;MOVE OVER IN1, BECAUSE IT IS SMALLEST.
         BEQ     MULTDN                  ;IF IT'S ZERO, THEN WE ARE DONE.
```

```
MULTCM: ROL     RET1                    ;ROTATE MULTIPLIER ONE PLACE.
        BCS     ADD16                   ;AND START MULTIPLY WHEN SET.
        ROL     RET1                    ;ROTATE MULTIPLIER ONE PLACE.
        BCS     ADD15                   ;AND START MULTIPLY WHEN SET.
        ROL     RET1                    ;ROTATE MULTIPLIER ONE PLACE.
        BCS     ADD14                   ;AND START MULTIPLY WHEN SET.
        ROL     RET1                    ;ROTATE MULTIPLIER ONE PLACE.
        BCS     ADD13                   ;AND START MULTIPLY WHEN SET.
        ROL     RET1                    ;ROTATE MULTIPLIER ONE PLACE.
        BCS     ADD12                   ;AND START MULTIPLY WHEN SET.
        ROL     RET1                    ;ROTATE MULTIPLIER ONE PLACE.
        BCS     ADD11                   ;AND START MULTIPLY WHEN SET.
        ROL     RET1                    ;ROTATE MULTIPLIER ONE PLACE.
        BCS     ADD10                   ;AND START MULTIPLY WHEN SET.
        ROL     RET1                    ;ROTATE MULTIPLIER ONE PLACE.
        BCS     ADD9                    ;AND START MULTIPLY WHEN SET.
        ROL     RET1                    ;ROTATE MULTIPLIER ONE PLACE.
        BCS     ADD8                    ;AND START MULTIPLY WHEN SET.
        ROL     RET1                    ;ROTATE MULTIPLIER ONE PLACE.
        BCS     ADD7                    ;AND START MULTIPLY WHEN SET.
        ROL     RET1                    ;ROTATE MULTIPLIER ONE PLACE.
        BCS     ADD6                    ;AND START MULTIPLY WHEN SET.
        ROL     RET1                    ;ROTATE MULTIPLIER ONE PLACE.
        BCS     ADD5                    ;AND START MULTIPLY WHEN SET.
        ROL     RET1                    ;ROTATE MULTIPLIER ONE PLACE.
        BCS     ADD4                    ;AND START MULTIPLY WHEN SET.
        ROL     RET1                    ;ROTATE MULTIPLIER ONE PLACE.
        BCS     ADD3                    ;AND START MULTIPLY WHEN SET.
        ROL     RET1                    ;ROTATE MULTIPLIER ONE PLACE.
        BCS     ADD2                    ;AND START MULTIPLY WHEN SET.
        CLR     RET1                    ;RET1 MUST BE 100000, BECAUSE
        MOV     IN2,RET2                ;IF WE GOT HERE, THEN IT MUST
MULTDN: RTS     PC                      ;BE A MULTIPLY BY 1.
ADD16:  MOV     IN2,RET2                ;IF A BIGGY, JUST MOVE OVER CRAP.
        ASL     RET2                    ;OK, NOW LET'S SHIFT AND START GOING.
        ROL     RET1                    ;DO NEXT BIT NOW. OK TO MULT?
        BCC     NADD15                  ;NO. BYPASS THE ADD NOW.
ADD15:  ADD     IN2,RET2                ;BIT ON. ADD IN MULTIPLICAND.
        ADC     RET1                    ;DON'T FORGET ABOUT ADDING IN CARRY.
NADD15: ASL     RET2                    ;ALL DONE WITH PREV. OPER.
        ROL     RET1                    ;DO NEXT BIT NOW. OK TO MULT?
        BCC     NADD14                  ;NO. BYPASS THE ADD NOW.
ADD14:  ADD     IN2,RET2                ;BIT ON. ADD IN MULTIPLICAND.
        ADC     RET1                    ;DON'T FORGET ABOUT ADDING IN CARRY.
NADD14: ASL     RET2                    ;ALL DONE WITH PREV. OPER.
        ROL     RET1                    ;DO NEXT BIT NOW. OK TO MULT?
        BCC     NADD13                  ;NO. BYPASS THE ADD NOW.
ADD13:  ADD     IN2,RET2                ;BIT ON. ADD IN MULTIPLICAND.
        ADC     RET1                    ;DON'T FORGET ABOUT ADDING IN CARRY.
NADD13: ASL     RET2                    ;ALL DONE WITH PREV. OPER.
        ROL     RET1                    ;DO NEXT BIT NOW. OK TO MULT?
        BCC     NADD12                  ;NO. BYPASS THE ADD NOW.
ADD12:  ADD     IN2,RET2                ;BIT ON. ADD IN MULTIPLICAND.
        ADC     RET1                    ;DON'T FORGET ABOUT ADDING IN CARRY.
NADD12: ASL     RET2                    ;ALL DONE WITH PREV. OPER.
        ROL     RET1                    ;DO NEXT BIT NOW. OK TO MULT?
        BCC     NADD11                  ;NO. BYPASS THE ADD NOW.
ADD11:  ADD     IN2,RET2                ;BIT ON. ADD IN MULTIPLICAND.
        ADC     RET1                    ;DON'T FORGET ABOUT ADDING IN CARRY.
NADD11: ASL     RET2                    ;ALL DONE WITH PREV. OPER.
        ROL     RET1                    ;DO NEXT BIT NOW. OK TO MULT?
        BCC     NADD10                  ;NO. BYPASS THE ADD NOW.
ADD10:  ADD     IN2,RET2                ;BIT ON. ADD IN MULTIPLICAND.
        ADC     RET1                    ;DON'T FORGET ABOUT ADDING IN CARRY.
NADD10: ASL     RET2                    ;ALL DONE WITH PREV. OPER.
```

```
         ROL     RET1                    ;DO NEXT BIT NOW. OK TO MULT?
         BCC     NADD9                   ;NO. BYPASS THE ADD NOW.
 ADD9:   ADD     IN2,RET2                ;BIT ON. ADD IN MULTIPLICAND.
         ADC     RET1                    ;DON'T FORGET ABOUT ADDING IN CARRY.
 NADD9:  ASL     RET2                    ;ALL DONE WITH PREV. OPER.
         ROL     RET1                    ;DO NEXT BIT NOW. OK TO MULT?
         BCC     NADD8                   ;NO. BYPASS THE ADD NOW.
 ADD8:   ADD     IN2,RET2                ;BIT ON. ADD IN MULTIPLICAND.
         ADC     RET1                    ;DON'T FORGET ABOUT ADDING IN CARRY.
 NADD8:  ASL     RET2                    ;ALL DONE WITH PREV. OPER.
         ROL     RET1                    ;DO NEXT BIT NOW. OK TO MULT?
         BCC     NADD7                   ;NO. BYPASS THE ADD NOW.
 ADD7:   ADD     IN2,RET2                ;BIT ON. ADD IN MULTIPLICAND.
         ADC     RET1                    ;DON'T FORGET ABOUT ADDING IN CARRY.
 NADD7:  ASL     RET2                    ;ALL DONE WITH PREV. OPER.
         ROL     RET1                    ;DO NEXT BIT NOW. OK TO MULT?
         BCC     NADD6                   ;NO. BYPASS THE ADD NOW.
 ADD6:   ADD     IN2,RET2                ;BIT ON. ADD IN MULTIPLICAND.
         ADC     RET1                    ;DON'T FORGET ABOUT ADDING IN CARRY.
 NADD6:  ASL     RET2                    ;ALL DONE WITH PREV. OPER.
         ROL     RET1                    ;DO NEXT BIT NOW. OK TO MULT?
         BCC     NADD5                   ;NO. BYPASS THE ADD NOW.
 ADD5:   ADD     IN2,RET2                ;BIT ON. ADD IN MULTIPLICAND.
         ADC     RET1                    ;DON'T FORGET ABOUT ADDING IN CARRY.
 NADD5:  ASL     RET2                    ;ALL DONE WITH PREV. OPER.
         ROL     RET1                    ;DO NEXT BIT NOW. OK TO MULT?
         BCC     NADD4                   ;NO. BYPASS THE ADD NOW.
 ADD4:   ADD     IN2,RET2                ;BIT ON. ADD IN MULTIPLICAND.
         ADC     RET1                    ;DON'T FORGET ABOUT ADDING IN CARRY.
 NADD4:  ASL     RET2                    ;ALL DONE WITH PREV. OPER.
         ROL     RET1                    ;DO NEXT BIT NOW. OK TO MULT?
         BCC     NADD3                   ;NO. BYPASS THE ADD NOW.
 ADD3:   ADD     IN2,RET2                ;BIT ON. ADD IN MULTIPLICAND.
         ADC     RET1                    ;DON'T FORGET ABOUT ADDING IN CARRY.
 NADD3:  ASL     RET2                    ;ALL DONE WITH PREV. OPER.
         ROL     RET1                    ;DO NEXT BIT NOW. OK TO MULT?
         BCC     NADD2                   ;NO. BYPASS THE ADD NOW.
 ADD2:   ADD     IN2,RET2                ;BIT ON. ADD IN MULTIPLICAND.
         ADC     RET1                    ;DON'T FORGET ABOUT ADDING IN CARRY.
 NADD2:  ASL     RET2                    ;ALL DONE WITH PREV. OPER.
         ROL     RET1                    ;DO NEXT BIT NOW. OK TO MULT?
         BCC     NADD1                   ;NO. BYPASS THE ADD NOW.
         ADD     IN2,RET2                ;BIT ON. ADD IN MULTIPLICAND.
         ADC     RET1                    ;DON'T FORGET ABOUT ADDING IN CARRY.
 NADD1:  RTS     PC                      ;AND RETURN WHEN DONE.
         .PAGE
 ;
 ;               THIS ROUTINE WILL MULTIPLY TWO SIGNED NUMBERS.
 ;
 SGNMUL: TST     IN1                     ;GET SIGN OF THE FIRST ARGUMENT.
         BPL     SGNMP1                  ;IT'S OK.
         NEG     IN1                     ;NOT OK.
         TST     IN2                     ;HOW ABOUT SECOND.
         BPL     SGNMP2                  ;IT'S POSITIVE.
         NEG     IN2                     ;BOTH NEGATIVE=POSITIVE.
 SGNMP3: JSR     PC,MULTWO               ;MULTIPLY THEM OUT.
         RTS     PC                      ;AND RETURN NOW.
 SGNMP1: TST     IN2                     ;TEST SIGN OF THE SECOND ONE.
         BPL     SGNMP3                  ;IT'S ALSO POSITIVE. DO MULTIPLY.
         NEG     IN2                     ;ELSE NEGATE IT, ANDINVERT ANSWER.
 SGNMP2: JSR     PC,MULTWO               ;MULTIPLY THEM OUT.
         NEG     RET2                    ;AND DO A DOUBLE PRECISION NEGATE.
         ADC     RET1
         NEG     RET1
         RTS     PC                      ;AND THEN RETURN. ALL IS WELL.
```

```
TRGMUL: JSR     PC,SGNMUL
        ASL     RET2                    ;FUDGE THE RETURN ANSWER NOW.
        ROL     RET1
        ASL     RET2
        ROL     RET1
        RTS     PC
        .PAGE
;
;               THIS ROUTINE WILL DIVIDE RET1-RET2 BY IN1 AND LEAVE THE
;               ANSWER IN RET2. DOES IT BY 16 SUBTRACTIONS. CRUDE BUT FAST.
;
DIVTWO: ASL     RET2                    ;WE FORBID LEFT-MOST BIT TO BE ON.
        ROL     RET1
        SUB     IN1,RET1                ;SEE IF IT FITS.
        BPL     DVOKA                   ;IT FITS. GENERATE A "1".
DVBADA: ASL     RET2                    ;NOT YET. PUT IN A ZERO.
        ROL     RET1                    ;AND PICK UP THE NEXT BIT.
        ADD     IN1,RET1                ;2*(Y-X)+X=2*Y-X.
        BPL     DVOKB                   ;IT FITS. GENERATE A "1".
DVBADB: ASL     RET2                    ;NOT YET. PUT IN A ZERO.
        ROL     RET1                    ;AND PICK UP THE NEXT BIT.
        ADD     IN1,RET1                ;2*(Y-X)+X=2*Y-X.
        BPL     DVOKC                   ;IT FITS. GENERATE A "1".
DVBADC: ASL     RET2                    ;NOT YET. PUT IN A ZERO.
        ROL     RET1                    ;AND PICK UP THE NEXT BIT.
        ADD     IN1,RET1                ;2*(Y-X)+X=2*Y-X.
        BPL     DVOKD                   ;IT FITS. GENERATE A "1".
DVBADD: ASL     RET2                    ;NOT YET. PUT IN A ZERO.
        ROL     RET1                    ;AND PICK UP THE NEXT BIT.
        ADD     IN1,RET1                ;2*(Y-X)+X=2*Y-X.
        BPL     DVOKE                   ;IT FITS. GENERATE A "1".
DVBADE: ASL     RET2                    ;NOT YET. PUT IN A ZERO.
        ROL     RET1                    ;AND PICK UP THE NEXT BIT.
        ADD     IN1,RET1                ;2*(Y-X)+X=2*Y-X.
        BPL     DVOKF                   ;IT FITS. GENERATE A "1".
DVBADF: ASL     RET2                    ;NOT YET. PUT IN A ZERO.
        ROL     RET1                    ;AND PICK UP THE NEXT BIT.
        ADD     IN1,RET1                ;2*(Y-X)+X=2*Y-X.
        BPL     DVOKG                   ;IT FITS. GENERATE A "1".
DVBADG: ASL     RET2                    ;NOT YET. PUT IN A ZERO.
        ROL     RET1                    ;AND PICK UP THE NEXT BIT.
        ADD     IN1,RET1                ;2*(Y-X)+X=2*Y-X.
        BPL     DVOKH                   ;IT FITS. GENERATE A "1".
DVBADH: ASL     RET2                    ;NOT YET. PUT IN A ZERO.
        ROL     RET1                    ;AND PICK UP THE NEXT BIT.
        ADD     IN1,RET1                ;2*(Y-X)+X=2*Y-X.
        BPL     DVOKI                   ;IT FITS. GENERATE A "1".
DVBADI: ASL     RET2                    ;NOT YET. PUT IN A ZERO.
        ROL     RET1                    ;AND PICK UP THE NEXT BIT.
        ADD     IN1,RET1                ;2*(Y-X)+X=2*Y-X.
        BPL     DVOKJ                   ;IT FITS. GENERATE A "1".
DVBADJ: ASL     RET2                    ;NOT YET. PUT IN A ZERO.
        ROL     RET1                    ;AND PICK UP THE NEXT BIT.
        ADD     IN1,RET1                ;2*(Y-X)+X=2*Y-X.
        BPL     DVOKK                   ;IT FITS. GENERATE A "1".
DVBADK: ASL     RET2                    ;NOT YET. PUT IN A ZERO.
        ROL     RET1                    ;AND PICK UP THE NEXT BIT.
        ADD     IN1,RET1                ;2*(Y-X)+X=2*Y-X.
        BPL     DVOKL                   ;IT FITS. GENERATE A "1".
DVBADL: ASL     RET2                    ;NOT YET. PUT IN A ZERO.
        ROL     RET1                    ;AND PICK UP THE NEXT BIT.
        ADD     IN1,RET1                ;2*(Y-X)+X=2*Y-X.
        BPL     DVOKM                   ;IT FITS. GENERATE A "1".
DVBADM: ASL     RET2                    ;NOT YET. PUT IN A ZERO.
        ROL     RET1                    ;AND PICK UP THE NEXT BIT.
```

```
         ADD     IN1,RET1                ;2*(Y-X)+X=2*Y-X.
         BPL     DVOKN                   ;IT FITS. GENERATE A "1".
 DVBADN: ASL     RET2                    ;NOT YET. PUT IN A ZERO.
         ROL     RET1                    ;AND PICK UP THE NEXT BIT.
         ADD     IN1,RET1                ;2*(Y-X)+X=2*Y-X.
         BPL     DVOKO                   ;IT FITS. GENERATE A "1".
 DVBADO: ASL     RET2                    ;NOT YET. PUT IN A ZERO.
         ROL     RET1                    ;AND PICK UP THE NEXT BIT.
         ADD     IN1,RET1                ;2*(Y-X)+X=2*Y-X.
         BPL     DVOKP                   ;IT FITS. GENERATE A "1".
 DVBADP: ADD     IN1,RET1                ;ON LAST FAIL, ADD BACK IN TO GET A
         RTS     PC                      ;CORRECT REMAINDER, AND THEN EXIT.
 DVOKA:  INC     RET2                    ;SUBTRACTION IS GOOD. INSERT A "1".
         ASL     RET2                    ;NOW CHECK THE NEXT ONE.
         ROL     RET1                    ;TO SEE IF IT GOES.
         SUB     IN1,RET1
         BMI     DVBADB                  ;GOES NOT GO. INSERT A ZERO AND START ADDING.
 DVOKB:  INC     RET2                    ;SUBTRACTION IS GOOD. INSERT A "1".
         ASL     RET2                    ;NOW CHECK THE NEXT ONE.
         ROL     RET1                    ;TO SEE IF IT GOES.
         SUB     IN1,RET1
         BMI     DVBADC                  ;GOES NOT GO. INSERT A ZERO AND START ADDING.
 DVOKC:  INC     RET2                    ;SUBTRACTION IS GOOD. INSERT A "1".
         ASL     RET2                    ;NOW CHECK THE NEXT ONE.
         ROL     RET1                    ;TO SEE IF IT GOES.
         SUB     IN1,RET1
         BMI     DVBADD                  ;GOES NOT GO. INSERT A ZERO AND START ADDING.
 DVOKD:  INC     RET2                    ;SUBTRACTION IS GOOD. INSERT A "1".
         ASL     RET2                    ;NOW CHECK THE NEXT ONE.
         ROL     RET1                    ;TO SEE IF IT GOES.
         SUB     IN1,RET1
         BMI     DVBADE                  ;GOES NOT GO. INSERT A ZERO AND START ADDING.
 DVOKE:  INC     RET2                    ;SUBTRACTION IS GOOD. INSERT A "1".
         ASL     RET2                    ;NOW CHECK THE NEXT ONE.
         ROL     RET1                    ;TO SEE IF IT GOES.
         SUB     IN1,RET1
         BMI     DVBADF                  ;GOES NOT GO. INSERT A ZERO AND START ADDING.
 DVOKF:  INC     RET2                    ;SUBTRACTION IS GOOD. INSERT A "1".
         ASL     RET2                    ;NOW CHECK THE NEXT ONE.
         ROL     RET1                    ;TO SEE IF IT GOES.
         SUB     IN1,RET1
         BMI     DVBADG                  ;GOES NOT GO. INSERT A ZERO AND START ADDING.
 DVOKG:  INC     RET2                    ;SUBTRACTION IS GOOD. INSERT A "1".
         ASL     RET2                    ;NOW CHECK THE NEXT ONE.
         ROL     RET1                    ;TO SEE IF IT GOES.
         SUB     IN1,RET1
         BMI     DVBADH                  ;GOES NOT GO. INSERT A ZERO AND START ADDING.
 DVOKH:  INC     RET2                    ;SUBTRACTION IS GOOD. INSERT A "1".
         ASL     RET2                    ;NOW CHECK THE NEXT ONE.
         ROL     RET1                    ;TO SEE IF IT GOES.
         SUB     IN1,RET1
         BMI     DVBADI                  ;GOES NOT GO. INSERT A ZERO AND START ADDING.
 DVOKI:  INC     RET2                    ;SUBTRACTION IS GOOD. INSERT A "1".
         ASL     RET2                    ;NOW CHECK THE NEXT ONE.
         ROL     RET1                    ;TO SEE IF IT GOES.
         SUB     IN1,RET1
         BMI     DVBADJ                  ;GOES NOT GO. INSERT A ZERO AND START ADDING.
 DVOKJ:  INC     RET2                    ;SUBTRACTION IS GOOD. INSERT A "1".
         ASL     RET2                    ;NOW CHECK THE NEXT ONE.
         ROL     RET1                    ;TO SEE IF IT GOES.
         SUB     IN1,RET1
         BMI     DVBADK                  ;GOES NOT GO. INSERT A ZERO AND START ADDING.
 DVOKK:  INC     RET2                    ;SUBTRACTION IS GOOD. INSERT A "1".
         ASL     RET2                    ;NOW CHECK THE NEXT ONE.
         ROL     RET1                    ;TO SEE IF IT GOES.
```

```
         SUB     IN1,RET1
         BMI     DVBADL                  ;GOES NOT GO. INSERT A ZERO AND START ADDING.
 DVOKL:  INC     RET2                    ;SUBTRACTION IS GOOD. INSERT A "1".
         ASL     RET2                    ;NOW CHECK THE NEXT ONE.
         ROL     RET1                    ;TO SEE IF IT GOES.
         SUB     IN1,RET1
         BMI     DVBADM                  ;GOES NOT GO. INSERT A ZERO AND START ADDING.
 DVOKM:  INC     RET2                    ;SUBTRACTION IS GOOD. INSERT A "1".
         ASL     RET2                    ;NOW CHECK THE NEXT ONE.
         ROL     RET1                    ;TO SEE IF IT GOES.
         SUB     IN1,RET1
         BMI     DVBADN                  ;GOES NOT GO. INSERT A ZERO AND START ADDING.
 DVOKN:  INC     RET2                    ;SUBTRACTION IS GOOD. INSERT A "1".
         ASL     RET2                    ;NOW CHECK THE NEXT ONE.
         ROL     RET1                    ;TO SEE IF IT GOES.
         SUB     IN1,RET1
         BMI     DVBADO                  ;GOES NOT GO. INSERT A ZERO AND START ADDING.
 DVOKO:  INC     RET2                    ;SUBTRACTION IS GOOD. INSERT A "1".
         ASL     RET2                    ;NOW CHECK THE NEXT ONE.
         ROL     RET1                    ;TO SEE IF IT GOES.
         SUB     IN1,RET1
         BMI     DVBADP                  ;GOES NOT GO. INSERT A ZERO AND START ADDING.
 DVOKP:  INC     RET2                    ;LAST SUBTRACTION SUCCESSFULL.
 DVOKX:  RTS     PC                      ;INSERT A "1" AND RETURN. ALL IS WELL.
         .PAGE
 ;
 ;               THIS IS THE CLOCK INTERUPT ROUTINE.
 ;
 TIMER:  BIS     #100,LKS                ;JUST MAKE SURE CLOCK IS STILL GOING.
         INC     TICKS                   ;UPDATE CLOCKS.
         INC     TIME                    ;INCREMENT BOTH CLOCKS.
         BIT     #17,TIME                        ;SEE IF IT'S TIME TO CLEAR ARROWS.
         BNE     TIMERT                  ;NOPE. EXIT.
         CLR     DLIST1                  ;YES. CLEAR THEM.
         CLR     DLIST2
 TIMERT: RTI                             ;AND RETURN NOW.
         .PAGE
 ;
 ;                                       THIS IS THE DELAY ROUTINE.
 ;                       IT WILL WAIT "N" SECONDS BEFORE RESTARTING THE SYSTEM.
 ;
 DELAY:  MOV     #DISTOP,LPSW            ;IF DELAYED CALLED, TURN OFF LOW FUEL MESSAGE.
         MOV     @0(SP),IN1              ;PICK UP THE TIME.
         MOV     TIME,IN2                ;GET CURRENT TIME.
         MOV     IN1,RET1                ;MOVE OVER TIME REQUESTED.
         BPL     DELAY1                  ;IF POSITIVE, IT'S ALRIGHT.
         NEG     IN1                     ;ELSE NEGATE IT NOW.
 DELAY1: ADD     #CLKFRQ,IN2             ;CALCULATE FUTURE TIME
         DEC     IN1                     ;WAIT TILL MULTIPLICATION IS DONE.
         BGT     DELAY1
 DELAY2: WAIT                            ;WAIT FOR AN INTERUPT TO OCCUR.
         JSR     PC,DIAL                 ;WHILE WAITING, DISPLAY THE USER'S DATA.
         CMP     IN2,TIME                ;CHECK THE TIMES NOW.
         BHI     DELAY2
         ADD     #2,(SP)                 ;UPDATE THE RETURN PC NOW.
         TST     RET1                    ;TEST SIGN OF THE WAIT.
         BMI     DVOKX                   ;IF NEGATIVE, THEN RETURN.
         JMP     RESTART                 ;ELSE RESTART THE SYSTEM.
         .PAGE
 ;
 ;       THIS IS THE START OF THE DISPLAY SECTION. PLEASE BE MERICFUL.
 ;
 DISPLY: MOV     IN1,-(SP)               ;COMES HERE ON AN INTERUPT.
         MOV     IN2,-(SP)
         MOV     DSTACK,IN1              ;PICK UP OLD STACK.
```

```
         MOV     @DPC,IN2                ;FIND OUT WHY WE STOPPED.
         BEQ     DSUBRT                  ;DISPLAY SUBROUTINE RETURN.
         MOV     DPC,-(IN1)              ;SUBROUTINE CALL. PUSH RETURN DPC ON STACK.
         ADD     #2,(IN1)                ;AND FUDGE PAST POINTER.
         MOV     #GOTSUB,-(IN1)          ;AND PUSH RETURN ADDRESS ONTO STACK.
 DISRT:  CMP     (IN2),#DISTOP           ;SEE IF WHERE HE WANTS TO START IS
         BNE     DISRST                  ;ANOTHER SUBROUTINE CALL.
         TST     2(IN2)                  ;IF IT IS, THEN FIGURE OUT WHETHER
         BEQ     DSUBRT                  ;TO RETURN OR GO DOWN ONE LEVEL.
         MOV     IN2,-(IN1)              ;PUSH NEXT DPC ONTO STACK.
         ADD     #4,(IN1)                ;AND FUDGE IT UP A BIT.
         MOV     #GOTSUB,-(IN1)          ;PUT ADDRESS ONTO STACK.
         MOV     2(IN2),IN2              ;UPDATE NEW PC NOW.
         BR      DISRT                   ;AND LOOP AGAIN.
 DISRST: MOV     IN1,DSTACK              ;SAVE THE STACK NOW.
         MOV     IN2,DPC                 ;START THE DISPLAY GOING NOW.
         MOV     (SP)+,IN2               ;RESTORE SAVED REGISTERS.
         MOV     (SP)+,IN1
         RTI                             ;AND RETURN NOW.
 DSUBRT: MOV     (IN1)+,PC               ;GO TO SUBROUTINE NOW.
 GOTSUB: MOV     (IN1)+,IN2              ;PICK UP OLD DPC
         BR      DISRT                   ;AND RESTART AND CONTINUE.
 DTOP:   INC     DNUM                    ;THIS ROUTINE GOES DOWN THE MASTER LIST.
 DTOP2:  MOV     DNUM,IN2
         ASL     IN2                     ;MAKE INTO A GOOD INDEX.
         MOV     DLIST(IN2),IN2
         BNE     DTOPOK                  ;IT'S OK TO START NOW.
         CLR     DNUM                    ;AT BOTTOM. LOOP.
         BR      DTOP2
 DTOPOK: MOV     #DTOP,STACKD
         MOV     #STACKD,IN1
         BR      DISRT
         .PAGE
 ;
 ;               LIGHT PEN HIT ROUTINE.
 ;
 LIGHT:  MOV     IN1,-(SP)
         MOV     IN2,-(SP)               ;SAVE REGISTERS NOW.
         MOV     DNUM,IN2                ;PICK UP THE INDEXED NUMBER NOW.
         CMP     IN2,OLDHIT              ;SEE IF INDEX IS SAME AS PREVIOUS.
         BEQ     LIGHTG                  ;YES. MAYBE WE CAN PROCESS IT.
         MOV     #15.,HITCNT             ;NOT SAME. SET UP HIT COUNTER
         MOV     IN2,OLDHIT              ;AND THE OLD REGISTER  NOW.
         BR      LPRESM                  ;AND EXIT NOW.
 LIGHTG: DEC     HITCNT                  ;HAVE WE BEEN HIT ENOUGH?
         BPL     LPRESM                  ;NO. GET OUT NOW.
         ASL     IN2                     ;ELSE SHIFT THE INDEX OVER
         JMP     @LPTAB(IN2)             ;AND GO PROCESS INTERUPT.
         .PAGE
 ;
 ;               THIS ROUTINE IS CALLED WHENEVER THE LIGHT PEN HITS
 ;               THE THROTTLE BAR.
 ;
 LPBARH: MOV     YSR,IN2                 ;GET THE Y COORDINATE.
         BIC     #-2000,IN2              ;MAKE IT JUST 10 BITS.
         ADD     #BARADD,IN2             ;ADD FUDGE FACTOR NOW.
         MOV     LPBARY,IN1              ;PICK UP OLD Y NOW.
         ASL     IN1                     ;AND MULTIPLY IT BY 8.
         ASL     IN1
         ASL     IN1
         SUB     LPBARY,IN1              ;WE REALLY ONLY WANTED 7.
         ADD     IN1,IN2                 ;NOW MAKE IT 7*OLD+NEW Y.
         ASR     IN2
         ASR     IN2
         ASR     IN2                     ;NOW AVERAGE IT OUT.
```

```
         MOV     IN2,LPBARY              ;SAVE THE Y VALUE NOW.
         SUB     #BARFDG,IN2             ;SUBTRACT BASE Y NOW.
         ASR     IN2                     ;DIVIDE BY TWO.
         CMP     IN2,#MINTRS             ;COMPARE AGAINST MINIMUM THRUST.
         BPL     LPBARP                  ;GREATER OR EQUAL. IT'S OK.
         MOV     #MINTRS,IN2             ;ELSE SET TO THE MINIMUM THRUST.
 LPBARP: CMP     #100.,IN2               ;LIKEWISE IF GREATER THEN 100, MAKE
         BGE     LPBARQ                  ;IT ONLY 100.
         MOV     #100.,IN2
 LPBARQ: MOV     IN2,PERCNT              ;SAVE AWAY THE VALUE NOW.
 LPRESM: BIS     #1,DPC                  ;RESUME THE DISPLAY NOW.
         MOV     (SP)+,IN2               ;RESTORE THE REGISTERS NOW.
         MOV     (SP)+,IN1               ;AND THEN EXIT.
         RTI
 LPIT1:  MOV     DLIST(IN2),IN2          ;GET POINTER TO USER'S DISPAY ITEM.
         MOV     LPFLG1,IN1              ;AND PICK UP PREVIOUS ITEM NOW.
         BEQ     LPIT1P                  ;IF ZERO, NO PREVIOUS BLINKING.
         BIC     #10,2(IN1)              ;ELSE CLEAR OUT THE BLINK BITS.
 LPIT1P: MOV     IN2,LPFLG1              ;SAVE IT AWAY FOR DISPLAY TRANSFER.
         BIS     #30,2(IN2)              ;SET IT BLINKING NOW.
         BR      LPRESM                  ;AND RESUME THE DISPLAY NOW.
 LPIT2:  MOV     LPFLG1,IN1              ;GET HIS POINTER.
         BEQ     LPRESM                  ;NONE SPECIFIED.
         CLR     LPFLG1                  ;CLEAR POINTER NOW.
         MOV     IN1,DIALTB-LPBASE+DLIST(IN2)    ;CHANGE OUR POINTER NOW.
         BIC     #10,2(IN1)              ;AND CLEAR THE BLINK FLAG NOW.
         BR      LPRESM                  ;AND RESUME THE DISPLAY NOW.
 ARROW:  MOV     DLIST(IN2),IN2          ;ON ARROW HIT, COME HERE. PICK UP ADDRESS.
         MOV     -2(IN2),TURN            ;AND MOVE OVER RATE OF TURN.
         MOV     IN2,DLIST1              ;AND ALSO MAKE IT BRIGHTER.
         MOV     IN2,DLIST2
         BR      LPRESM                  ;AND RESUME NOW.
         .PAGE
 ;
 ;               DISPLAY POINTERS.
 ;
 LPBASE:
 DLIST:  .WORD   DITEM1
         .WORD   DITEM2
         .WORD   DITEM3
         .WORD   DITEM4
         .WORD   ITEME1
         .WORD   ITEME2
         .WORD   ITEME3
         .WORD   ITEME4
         .WORD   ITEME5
         .WORD   ITEME6
         .WORD   ITEME7
         .WORD   ITEME8
         .WORD   ITEME9
         .WORD   ITEMET
         .WORD   ITEMEE
         .WORD   ITEMES
         .WORD   DRWSHP
         .WORD   ONFIRE
         .WORD   INFO
         .WORD   DRWDST
         .WORD   DRWLUN
         .WORD   SLEFTA
         .WORD   SRGTA
         .WORD   BLEFTA
         .WORD   BRGTA
         .WORD   LPBAR
 DLIST1: .WORD   0                       ;EXTRA BRIGHTNESS WORD
 DLIST2: .WORD   0                       ;IF DISPLAYING THE ARROWS.
```

```
        .WORD   0
        .PAGE
;
;               ;LIGHT PEN VECTORS.
;
LPTAB:  .WORD   LPIT2,LPIT2,LPIT2,LPIT2
        .WORD   LPIT1,LPIT1,LPIT1,LPIT1
        .WORD   LPIT1,LPIT1,LPIT1,LPIT1
        .WORD   LPIT1,LPIT1,LPIT1,LPIT1
        .WORD   DSUBRT,DSUBRT,DSUBRT,DSUBRT,DSUBRT
        .WORD   ARROW,ARROW,ARROW,ARROW
        .WORD   LPBARH
        .WORD   ARROW,ARROW             ;TERMINATING LIGHT PEN HITS.
        .PAGE
;
;               THIS ROUTINE WILL CALCULATE THE VALUES THAT THE
;               USER WISHES TO HAVE DISPLAYED.
;
DIAL:   MOV     IN1,-(SP)               ;SAVE ALL THE IMPORTANT REGISTERS NOW.
        MOV     IN2,-(SP)
        MOV     RET1,-(SP)
        MOV     RET2,-(SP)
        MOV     TEMP,-(SP)
        MOV     TEMP2,-(SP)
        CLR     TEMP                    ;NOW SET POINTER TO FIRST ITEM.
DIALL:  MOV     DIALTB(TEMP),TEMP2      ;PICK UP POINTER TO AN ITEM ENTRY.
        BEQ     DIALD1                  ;IF ZERO, THEN END OF LIST.
        MOV     DIALTC(TEMP),IN2        ;PICK UP POINTER TO WHERE TO
        MOV     TEMP2,RET2              ;DISPLAY CALCULATED VALUES.
        ADD     #10.,RET2               ;CALCULATE "ITEMFX" ENTRY ADDRESS.
        MOV     RET2,18.(IN2)           ;AND INSERT IT IN THE DISJMP OF
        MOV     @-4(TEMP2),RET2         ;THE DITEMX TABLE. PICK UP USERS NUMBER NOW.
        MOV     -2(TEMP2),IN1           ;PICK UP THE DIVISOR <IF ANY>.
        BEQ     DIALND                  ;NONE. DO NOT DIVIDE. DISPLAY AS IS.
        TST     RET2                    ;NOW MAKE ANSWER POSITIVE.
        BPL     DIALLP
        NEG     RET2
DIALLP: CLR     RET1                    ;GET RID OF THE HIGH PART.
        JSR     PC,DIVTWO               ;DO THE DIVISION NOW.
        TST     @-4(TEMP2)              ;NOW RESTORE THE SIGN.
        BPL     DIALND
        NEG     RET2
DIALND: MOV     DIALTC(TEMP),TEMP2      ;PICK UP DITEMX POINTER AGAIN.
        MOV     RET2,IN1                ;MOVE OVER NUMBER NOW.
        MOV     #10.,IN2                ;NOW CREATE ADDRESS OF WHERE TO LEAVE ANSWER
        ADD     TEMP2,IN2               ;AFTER CONVERTING TO ASCII STRING.
        JSR     PC,SASCII               ;DO THE CONVERSION NOW.
        TST     (TEMP)+                 ;FINALLY INCREMENT POINTER TO NEXT ITEM.
        BR      DIALL                   ;AND GO BACK UP TO TOP AND GET NEXT VALUE.
DIALD1: CMP     PERCNT,OLDPER           ;CONVERT PERCENTAGE NOW. ANY CHANGE?
        BEQ     DIALRT                  ;NO. JUST EXIT.
        MOV     #LPBARC,IN2             ;YES. GET ADDRESS OF WHERE TO PLACE STRING.
        MOV     PERCNT,IN1              ;GET THE NUMBER NOW.
        MOV     IN1,OLDPER              ;UPDATE OLD PERCENTAGE NOW.
        JSR     PC,SASCII               ;AND CONVERT IT NOW.
DIALRT: MOV     (SP)+,TEMP2             ;RESTORE ALL THE ACTIVE REGISTERS NOW.
        MOV     (SP)+,TEMP
        MOV     (SP)+,RET2
        MOV     (SP)+,RET1
        MOV     (SP)+,IN2
        MOV     (SP)+,IN1
        RTS     PC                      ;AND RETURN NOW.
        .PAGE
;
;               THIS ROUTINE CONVERT THE BINARY NUMBER (UNSIGNED)
```

```
;               IN THE REGISTER "IN1" INTO AN ASCII STRING SIX
;               CHARACTERS LONG AND LEAVES THE ANSWER TO WHERE "IN2"
;               POINTS TO. THE ANSWER IS PUT RIGHT JUSTIFIED WITH
;               LEADING BLANKS.
;
ASCII:  MOVB    #' ,(IN2)+              ;FIRST CHARACTER WILL BE BLANK ALWAYS.
        CMP     IN1,#10000.             ;IS IT GREATER THEN 9999.?
        BHIS    ASCIIG                  ;YES. DO SOMETHING SPECIAL.
        MOVB    #' ,(IN2)+              ;NO. JUST INSERTA BLANK.
        CLR     -(SP)                   ;AND REMEMBER TO DELETE LEADING ZEROS.
ASCIIL: CMP     IN1,#100.               ;IS IT GREATER THAN 99?
        BLT     ASCIID                  ;NO! NO DIVIDE IS NECESSARY.
        MOV     IN1,RET2                ;GREATER THAN 99. DIVIDE.
        CLR     RET1                    ;CLEAR HIGH ORDER WORD OF AC-MQ.
        MOV     #100.,IN1               ;DIVIDE BY 100.
        JSR     PC,DIVTWO               ;DIVIDE NOW.
        ASL     RET2                    ;NOW LOOK UP ANSWER IN TABLE.
        ADD     #TENTAB,RET2            ;TWO BYTES WHICH COMPOSE NEXT TWO DIGITS.
ASCIIB: JSR     PC,ASCDO                ;INSERT FIRST.
        JSR     PC,ASCDO                ;INSERT SECOND.
        MOV     RET1,RET2               ;GET THE REMAINDER NOW.
        ASL     RET2
        ADD     #TENTAB,RET2            ;AND MAKE IT A GOOD ADDRESS.
        JSR     PC,ASCDO                ;AND DO THE NEXT TWO DIGITS.
        MOV     PC,(SP)                 ;BUT MAKE SURE WE GET A GOOD ZERO.
        JSR     PC,ASCDO
        ADD     #2,SP                   ;BUMP THE STACK NOW.
        RTS     PC                      ;AND RETURN.
ASCIIG: MOV     PC,-(SP)                ;UPDATE LEADING ZERO FLAG TO NO DELETE.
        MOVB    #'0,(IN2)+              ;INSERT IT INTO THE BUFFER NOW.
ASCIIM: CMP     IN1,#10000.             ;SEE IF IT'S STILL ABOVE 9999.
        BLO     ASCIIL                  ;NO. EXIT
        INCB    -1(IN2)                 ;YES CHANGE DIGIT AND UPDATE.
        SUB     #10000.,IN1
        BR      ASCIIM                  ;AND LOOP TILL DONE.
ASCIID: MOV     IN1,RET1                ;SMALL NUMBER. SET UP REMAINDER.
        MOV     #TENTAB,RET2            ;AND ADDRESS FOR THE FIRST 0.
        BR      ASCIIB                  ;AND INSERT THEM NOW.
ASCDO:  TST     2(SP)                   ;INSERT ANYTHING?
        BNE     ASCDBP                  ;YES.
        CMPB    (RET2),#'0              ;NO. IS THIS CHARACTER A ZERO?
        BNE     ASCDO1                  ;NO. FIRST GOOD CHARACTER.
        MOVB    #' ,(IN2)+              ;STILL ZERO. INSERT A BLANK.
        INC     RET2                    ;AND BOP INPUT POINTER.
        RTS     PC                      ;AND RETURN NOW.
ASCDO1: MOV     SP,2(SP)                ;GOOD CHARACTER. CLEAR ZERO FLAG.
ASCDBP: MOVB    (RET2)+,(IN2)+          ;AND MOVE OVER THE CHARACTER NOW.
        RTS     PC                      ;AND THEN RETURN.
        .PAGE
;
;               THIS ROUTINE WILL CONVERT A SIGNED NUMBER TO AN ASCII
;               STRING ACCORDING TO THE RULES USED IN "ASCII".
;
SASCII: TST     IN1                     ;GET THE SIGN OF THE INPUT.
        BPL     ASCII                   ;IF POSITIVE, NOTHING SPECIAL.
        NEG     IN1                     ;ELSE NEGATE IT.
        JSR     PC,ASCII                ;AND CONVERT IT.
        MOV     IN2,IN1                 ;MOV BACK LAST POINTER.
SASCIL: CMPB    #' ,-(IN1)              ;AND LOOK FOR THE FIRST BACKWARD BLANK.
        BNE     SASCIL
        MOVB    #'-,(IN1)               ;WHEN FOUND, CHANGE IT.
        RTS     PC                      ;AND RETURN NOW.
        .PAGE
;
;               THESE LISTS ARE USED IN PRINTING OUT THE USER
```

```
;               REQUESTED VALUES.
;
DIALTB: .WORD   ITEME1
        .WORD   ITEME2
        .WORD   ITEME3
        .WORD   ITEME4
        .WORD   0                       ;0=END OF LIST.
DIALTC: .WORD   DITEM1                  ;WHERE OUR TABLES ARE.
        .WORD   DITEM2
        .WORD   DITEM3
        .WORD   DITEM4
;
;               VALUE DISPLAY ITEMS.
;
DITEM1: .WORD   170240                  ;ENABLE LIGHT PEN STUFF.
        .WORD   117560                  ;AND MOVE THE POINTER AROUND.
        .WORD   DX1
        .WORD   DY1
        .WORD   100000                  ;ENTER CHARACTER MODE NOW.
        .ASCII  '      '                ;SIX CHARACTERS OF ZERO,
        .WORD   DISJMP                  ;AND NOW CALL THE NEXT SUBROUTINE.
        .WORD   ITEMF1
DITEM2: .WORD   170240
        .WORD   117560
        .WORD   DX2
        .WORD   DY2
        .WORD   100000
        .ASCII  '      '
        .WORD   DISJMP
        .WORD   ITEMF2
DITEM3: .WORD   170240
        .WORD   117560
        .WORD   DX3
        .WORD   DY3
        .WORD   100000
        .ASCII  '      '
        .WORD   DISJMP
        .WORD   ITEMF3
DITEM4: .WORD   170240
        .WORD   117560
        .WORD   DX4
        .WORD   DY4
        .WORD   100000
        .ASCII  '      '
        .WORD   DISJMP
        .WORD   ITEMF4
;
;               THESE TABLES CONTAIN THE ACTUAL MESSAGES THAT THE
;               USER HAS A CHOICE OF SEEING.
;
ITEM1:  .WORD   RADARY
        .WORD   0                       ;WHERE ITEM IS AND WHAT TO DIVIDE IT BY.
ITEME1: .WORD   170260                  ;ACTUALLY DISPLAY PORTION HERE.
        .WORD   117560
        .WORD   ITEMX1
        .WORD   ITEMY1
        .WORD   100000
ITEMF1: .ASCII  ' HEIGHT '
        .WORD   DISTOP
        .WORD   0
ITEM2:  .WORD   VERDIS
        .WORD   0
ITEME2: .WORD   170260
        .WORD   117560
        .WORD   ITEMX2
```

```
         .WORD   ITEMY2
         .WORD   100000
 ITEMF2: .ASCII  ' ALTITUDE '
         .WORD   DISTOP
         .WORD   0
 ITEM3:  .WORD   HORDIS
         .WORD   0
 ITEME3: .WORD   170260
         .WORD   117560
         .WORD   ITEMX3
         .WORD   ITEMY3
         .WORD   100000
 ITEMF3: .ASCII  ' DISTANCE '
         .WORD   DISTOP
         .WORD   0
 ITEM4:  .WORD   FUEL
         .WORD   10.
 ITEME4: .WORD   170260
         .WORD   117560
         .WORD   ITEMX4
         .WORD   ITEMY4
         .WORD   100000
 ITEMF4: .ASCII  ' FUEL LEFT'
         .WORD   DISTOP
         .WORD   0
 ITEM5:  .WORD   WEIGHT
         .WORD   0
 ITEME5: .WORD   170260
         .WORD   117560
         .WORD   ITEMX5
         .WORD   ITEMY5
         .WORD   100000
 ITEMF5: .ASCII  ' WEIGHT '
         .WORD   DISTOP
         .WORD   0
 ITEM6:  .WORD   THRUST
         .WORD   0
 ITEME6: .WORD   170260
         .WORD   117560
         .WORD   ITEMX6
         .WORD   ITEMY6
         .WORD   100000
 ITEMF6: .ASCII  ' THRUST '
         .WORD   DISTOP
         .WORD   0
 ITEM7:  .WORD   ANGLE
         .WORD   0
 ITEME7: .WORD   170260
         .WORD   117560
         .WORD   ITEMX7
         .WORD   ITEMY7
         .WORD   100000
 ITEMF7: .ASCII  ' ANGLE'
         .WORD   DISTOP
         .WORD   0
 ITEM8:  .WORD   VERVEL
         .WORD   10.
 ITEME8: .WORD   170260
         .WORD   117560
         .WORD   ITEMX8
         .WORD   ITEMY8
         .WORD   100000
 ITEMF8: .ASCII  ' VER VEL'
         .WORD   DISTOP
         .WORD   0
```

```
ITEM9:  .WORD   HORVEL
        .WORD   10.
ITEME9: .WORD   170260
        .WORD   117560
        .WORD   ITEMX9
        .WORD   ITEMY9
        .WORD   100000
ITEMF9: .ASCII  ' HOR VEL'
        .WORD   DISTOP
        .WORD   0
ITEM10: .WORD   VERACC
        .WORD   500.
ITEMET: .WORD   170260
        .WORD   117560
        .WORD   ITEMXT
        .WORD   ITEMYT
        .WORD   100000
ITEMFT: .ASCII  ' VER ACC'
        .WORD   DISTOP
        .WORD   0
ITEM11: .WORD   HORACC
        .WORD   500.
ITEMEE: .WORD   170260
        .WORD   117560
        .WORD   ITEMXE
        .WORD   ITEMYE
        .WORD   100000
ITEMFE: .ASCII  ' HOR ACC'
        .WORD   DISTOP
        .WORD   0
ITEM12: .WORD   TIME
        .WORD   60.
ITEMES: .WORD   170260
        .WORD   117560
        .WORD   ITEMXS
        .WORD   ITEMYS
        .WORD   100000
ITEMFS: .ASCII  ' SECONDS'
        .WORD   DISTOP
        .WORD   0
        .PAGE
;
;               THIS IS WHERE THE MOON GOES, WHEN IT IS CORRECTLY
;               DRAWN. IT IS PLACED HERE SO NO CODE GOES OVER 15000-20000
;               AND SO THAT NO CODE GOES OVER 35000-37776 EITHER. THIS AREA
;               IS INITIALIZED TO ALL "DISTOP,0" BY THE RESTART ROUTINE.
;
MOONST:                                 ;DEFINE IT AS STARTING HERE.
        .=.+2002.                       ;AND BEING 1000 DECIMAL WORDS LONG.
MOONEN:                                 ;NEXT WORD AFTER MOON TO DETERMINE
                                        ;HOW FAR TO INITIALIZE THIS SECTION.
;
;               MESSAGES FOR GOING OFF SCREEN <LEFT,RIGHT AND TOP.>.
;
LFTMSG: .WORD   117520
        .WORD   30.
        .WORD   600.
        .WORD   100000
        .ASCII  'BOY, ARE YOU'
        .BYTE   0
        .=.-1
        .EVEN
        .WORD   103730
        .ASCII  ' INEPT'
        .BYTE   0
```

```
         .=.-1
         .EVEN
         .WORD   DISTOP
         .WORD   0
 RGTMSG: .WORD   117520
         .WORD   525.
         .WORD   600.
         .WORD   100000
         .ASCII  '  YOU HAVE JUST CRASHED'
         .BYTE   0
         .=.-1
         .EVEN
         .WORD   117520
         .WORD   525.
         .WORD   570.
         .WORD   100000
         .ASCII  'INTO THE EDGE OF THE MOON'
         .BYTE   0
         .=.-1
         .EVEN
         .WORD   DISTOP
         .WORD   0
 TOPMSG: .WORD   117520
         .WORD   50.
         .WORD   650.
         .WORD   100000
         .ASCII  'SORRY, BUT WHEN YOU LOSE TV COVERAGE, YOU ALSO LOSE YOUR FUEL'
         .BYTE   0
         .=.-1
         .EVEN
         .WORD   DISTOP
         .WORD   0
         .PAGE
 ;
 ;               SPEED WARNING MASSAGES...
 ;
 VFAST:  .WORD   117530
         .WORD   100.
         .WORD   2.
         .WORD   100000
         .ASCII  "TOO FAST. YOU'RE GOING TO CRASH"
         .BYTE   0
         .=.-1
         .EVEN
         .WORD   DISTOP
         .WORD   0
 FAST:   .WORD   117520
         .WORD   100.
         .WORD   700.
         .WORD   100000
         .ASCII  'BETTER START SLOWING IT UP PRETTY SOON'
         .BYTE   0
         .=.-1
         .EVEN
         .WORD   DISTOP
         .WORD   0
 N2FAST: .WORD   117520
         .WORD   100.
         .WORD   2.
         .WORD   100000
         .ASCII  'TAKE IT NICE AND EASY. A PERFECT LANDING IS UNDER 8 FPS'
         .BYTE   0
         .=.-1
         .EVEN
         .WORD   DISTOP
```

```
        .WORD   0
GREATM: .WORD   117520
        .WORD   100.
        .WORD   600.
        .WORD   100000
        .ASCII  'FANTASTIC, A PERFECT LANDING'
        .BYTE   0
        .=.-1
        .EVEN
        .WORD   DISTOP
        .WORD   0
GOODM:  .WORD   117520
        .WORD   100.
        .WORD   600.
        .WORD   100000
        .ASCII  'CONGRATULATIONS ON A GOOD LANDING'
        .BYTE   0
        .=.-1
        .EVEN
        .WORD   DISTOP
        .WORD   0
ROUGHM: .WORD   117520
        .WORD   100.
        .WORD   600.
        .WORD   100000
        .ASCII  'THE LANDING WAS A LITTLE FAST'
        .BYTE   0
        .=.-1
        .EVEN
        .WORD   DISTOP
        .WORD   0
CRIPM:  .WORD   117520
        .WORD   100.
        .WORD   600.
        .WORD   100000
        .ASCII  'THE LANDING WAS TOO FAST AND DAMAGE WAS DONE TO THE SHIP'
        .BYTE   0
        .=.-1
        .EVEN
        .WORD   DISTOP
        .WORD   0
DEADM:  .WORD   117530
        .WORD   100.
        .WORD   550.
        .WORD   100000
        .ASCII  'WELL, YOU CERTAINLY BLEW THAT ONE. THERE WERE NO SURVIRORS'
        .BYTE   0
        .=.-1
        .EVEN
        .WORD   DISTOP
        .WORD   0
ANGLEM: .WORD   117520
        .WORD   100.
        .WORD   570.
        .WORD   100000
        .ASCII  'BUT THE ANGLE WAS TOO GREAT AND THE SHIP TIPPED OVER'
        .BYTE   0
        .=.-1
        .EVEN
ANGLEJ: .WORD   117520
        .WORD   100.
        .WORD   540.
        .WORD   100000
        .ASCII  'SORRY, BUT THERE WERE '
        .BYTE   0
```

```
        .=.-1
        .EVEN
        .WORD   103630
        .ASCII  'NO'
        .WORD   103520
        .ASCII  ' SURVIVORS'
        .BYTE   0
        .=.-1
        .EVEN
        .WORD   DISTOP
        .WORD   0
SIDEM:  .WORD   117520
        .WORD   100.
        .WORD   570.
        .WORD   100000
        .ASCII  'BUT THE HORIZONTAL VELOCITY WAS TOO GREAT, AND YOU CRASHED ANYWAY'
        .BYTE   0
        .=.-1
        .EVEN
        .WORD   DISJMP
        .WORD   ANGLEJ                  ;DO A DISPLAY JUMP INTO ANGLEM.
BUMPYM: .WORD   117520
        .WORD   100.
        .WORD   570.
        .WORD   100000
        .ASCII  'BUT THE TERRAIN IS TOO ROUGH, AND YOU TIPPED OVER'
        .BYTE   0
        .=.-1
        .EVEN
        .WORD   DISJMP
        .WORD   ANGLEJ
ROCKMS: .WORD   117520
        .WORD   100.
        .WORD   570.
        .WORD   100000
        .ASCII  'YOU JUST CRASHED INTO THAT ROCK'
        .BYTE   0
        .=.-1
        .EVEN
        .WORD   DISJMP
        .WORD   ANGLEJ
OLDMS:  .WORD   117520
        .WORD   100.
        .WORD   570.
        .WORD   100000
        .ASCII  'YOU JUST CRASHED ON TOP OF AN OLD LUNAR MODULE'
        .BYTE   0
        .=.-1
        .EVEN
        .WORD   DISJMP
        .WORD   ANGLEJ
FLAGMS: .WORD   117520
        .WORD   50.
        .WORD   3.
        .WORD   100000
        .ASCII  'YOU HAVE JUST VAPORIZED A PREVIOUSLY PLANTED AMERICAN FLAG'
        .BYTE   0
        .=.-1
        .EVEN
        .WORD   DISTOP
        .WORD   0
OLDTLT: .WORD   117520
        .WORD   100.
        .WORD   570.
        .WORD   100000
```

```
        .ASCII  'NICE WORK. YOU JUST CRASHED INTO A PREVIOUSLY CRASHED SHIP'
        .BYTE   0
        .=.-1
        .EVEN
        .WORD   DISJMP
        .WORD   ANGLEJ
MACDED: .WORD   117520
        .WORD   10.
        .WORD   570.
        .WORD   100000
        .ASCII  'W'
        .BYTE   145,154,154
        .ASCII  ', '
        .BYTE   171,157,165
        .ASCII  "'"
        .BYTE   166,145,40,152,165,163,164,40,144,145,163
        .BYTE   164,162,157,171,145,144,40,164,150,145,40
        .BYTE   157,156,154,171
        .ASCII  ' M'
        .BYTE   141,143
        .ASCII  'D'
        .BYTE   157,156,141,154,144
        .ASCII  "'"
        .BYTE   163
        .BYTE   0
        .=.-1
        .EVEN
        .WORD   117520
        .WORD   10.
        .WORD   540.
        .WORD   100000
        .BYTE   157,156,40,164,150,145,40,155,157,157,156
        .ASCII  '. W'
        .BYTE   150,141,164,40,141
        .ASCII  ' CLOD.'
        .BYTE   0
        .=.-1
        .EVEN
        .WORD   DISTOP
        .WORD   0
ORDER:  .WORD   117520
        .WORD   150.
        .WORD   2.
        .WORD   100000
        .WORD   170260
        .ASCII  'TWO CHEESEBURGERS AND A BIG MAC TO GO.'
        .BYTE   0
        .=.-1
        .EVEN
        .WORD   DISTOP
        .WORD   0
MANMSG: .WORD   117520
        .WORD   50.
        .WORD   2.
        .WORD   100000
        .WORD   170260
        .ASCII  "THAT'S ONE SMALL STEP FOR A MAN, ONE GIANT LEAP FOR MANKIND."
        .BYTE   0
        .=.-1
        .EVEN
        .WORD   DISTOP
        .WORD   0
        .PAGE
;
;               THIS LITTLE SECTION CONTAINS THE CODE
```

```
;               FOR DRAWING THE CONTROLLING ARROWS
;               FOR HANDLING THE DEGREES OF ROTATION.
;               SMALL LEFT ARROW.
;
        .WORD   -15.
SLEFTA: .WORD   114020
        .WORD   170240
        .WORD   SLFTAX
        .WORD   SLFTAY
        .WORD   113144
        .WORD   20.+LEFT+INT
        .WORD   0.
LEFTC:  .WORD   12.+INT
        .WORD   8.
        .WORD   12.+LEFT
        .WORD   8.+DOWN
        .WORD   12.+INT
        .WORD   8.+DOWN
        .WORD   DISTOP
        .WORD   0
        .WORD   -100.
BLEFTA: .WORD   114020
        .WORD   170240
        .WORD   BLFTAX
        .WORD   BLFTAY
        .WORD   113144
        .WORD   40.+LEFT+INT
        .WORD   0
        .WORD   DISJMP
        .WORD   LEFTC
        .WORD   15.
SRGTA:  .WORD   114020                  ;SMALL RIGHT ARROW.
        .WORD   170240
        .WORD   SRGTAX
        .WORD   SRGTAY
        .WORD   113144
        .WORD   20.+INT
        .WORD   0
RGTC:   .WORD   12.+INT+LEFT
        .WORD   8.
        .WORD   12.
        .WORD   8.+DOWN
        .WORD   12.+LEFT+INT
        .WORD   8.+DOWN
        .WORD   DISTOP
        .WORD   0
        .WORD   100.
BRGTA:  .WORD   114020
        .WORD   170240
        .WORD   BRGTAX
        .WORD   BRGTAY
        .WORD   113144
        .WORD   40.+INT
        .WORD   0
        .WORD   DISJMP
        .WORD   RGTC
        .PAGE
;
;               THIS IS THE LIGHT PEN BAR WHICH CONTROLS THE
;               THRUST OF THE ROCKET ENGINE.
;
LPBAR:  .WORD   116764
        .WORD   170200
        .WORD   BARLX+0
        .WORD   BARTY
```

```
        .WORD   110140
        .WORD   INT
        .WORD   BARSIZ+DOWN
        .WORD   3
        .WORD   BARSIZ
        .WORD   INT
        .WORD   BARSIZ+DOWN
        .WORD   3
        .WORD   BARSIZ
        .WORD   INT
        .WORD   BARSIZ+DOWN
        .WORD   114100
        .WORD   BARMXR                  ;NOW MOVE OVER TO THE BAR POSITION.
LPBARY: .WORD   0                       ;VERTICAL HEIGHT GOES HERE.
        .WORD   110000
        .WORD   BARMXL+INT              ;BE CAREFULL OF SIGN OF THIS ONE.
        .WORD   0
        .WORD   BAREST
        .WORD   0
        .WORD   100000                  ;ENTER CHARACTER MODE NOW.
LPBARC: .ASCII  '      % '              ;OVERLAYED WITH A NUMBER.
LPSW:   .WORD   DISTOP                  ;0 ON FUEL LOW--FALL TO NEXT MSG
        .WORD   0
        .WORD   117130
        .WORD   350.
        .WORD   700.
        .WORD   100000
        .ASCII  'FUEL LOW'
        .WORD   DISTOP
        .WORD   0
        .PAGE
;
;                       ;THIS IS THE BASIC ROCK COMMANDS.
;
ROCKL:  .WORD   116727
        .WORD   170240
ROCKX:  .WORD   0
ROCKY:  .WORD   0
        .WORD   104000
        .WORD   14.+OTHER*200+8.+OTHER
        .WORD   OTHER+6+INTTWO*200+8.
        .WORD   4+INTTWO*200+6
        .WORD   0+INTTWO*200+6
        .WORD   2+INTTWO*200+2
        .WORD   2+INTTWO*200+4
        .WORD   6+INTTWO*200+6
        .WORD   INTTWO*200+4
        .WORD   4+INTTWO*200+2
        .WORD   2+INTTWO*200+0
        .WORD   2+INTTWO*200+2+OTHER
        .WORD   6+INTTWO*200+2+OTHER
        .WORD   4+INTTWO*200+2+OTHER
        .WORD   2+INTTWO*200+6+OTHER
        .WORD   6+INTTWO*200+6+OTHER
        .WORD   INTTWO*200+4+OTHER
        .WORD   2+INTTWO*200+4+OTHER
        .WORD   2+OTHER+INTTWO*200+4+OTHER
        .WORD   2+OTHER+INTTWO*200+8.+OTHER
        .WORD   DISTOP
        .WORD   0
ROCKEN:                                 ;NEXT LOCATION AFTER ROCKS.
        .PAGE
;
;               THIS LIST EXPLAINS HOW TO DRAW THE MACDONALD'S.
;
```

```
MACS:   117724
        170240
MACX:   .WORD   0
MACY:   .WORD   0
        .WORD   104000
        .WORD   18.*200
        .WORD   107000
        .WORD   30.+INT
        .WORD   54.*200+INT
        .WORD   30.+OTHER+INT
        .WORD   54.+OTHER*200+INT
        .WORD   105324
        .WORD   12.*200
        .WORD   8.+INT
        .WORD   6.*200+INT
        .WORD   8.+OTHER+INT
        .WORD   19.+OTHER*200
        .WORD   110000
        .WORD   73.+LEFT
        .WORD   0.
        .WORD   DISTOP
        .WORD   ARCH
        .WORD   104000
        .WORD   22.*200
;
;               NOTICE HOW I FALL THROUGH TO DRAW THE
;               SECOND ARCH. CLEVER, CLEVER.
;
ARCH:   .WORD   107724
        .WORD   17.+OTHER*200+OTHER+3.
        .WORD   1*200+10.+INT
        .WORD   1*200+9.+INT
        .WORD   1*200+6+INT
        .WORD   1*200+5+INT
        .WORD   1*200+4+INT
        .WORD   1*200+2+INT
        .WORD   1*200+4+INT
        .WORD   2*200+5+INT
        .WORD   1*200+1+INT
        .WORD   2*200+4+INT
        .WORD   3*200+2+INT
        .WORD   1*200+1+INT
        .WORD   2*200+0+INT
        .WORD   1*200+1+INT+OTHER
        .WORD   3*200+2+INT+OTHER
        .WORD   2*200+4+INT+OTHER
        .WORD   1*200+1+INT+OTHER
        .WORD   2*200+5+INT+OTHER
        .WORD   1*200+4+INT+OTHER
        .WORD   1*200+2+INT+OTHER
        .WORD   1*200+4+INT+OTHER
        .WORD   1*200+5+INT+OTHER
        .WORD   1*200+6+INT+OTHER
        .WORD   1*200+9.+INT+OTHER
        .WORD   1*200+10.+INT+OTHER
        .WORD   17.+OTHER*200+3.
        .WORD   DISTOP
        .WORD   0
        .PAGE
;
;               THIS LIST EXPLAINS HOW TO DRAW A MAN.
;
MAN:    .WORD   116720                  ;DON'T MAKE HIM TOO BRIGHT.
MANX:   .WORD   0
MANY:   .WORD   0
```

```
        .WORD   104000                  ;ALL SHORT VECTORS.
        .WORD   400                     ;INVISIBLE 2 RIGHT.
        .WORD   INT+20000+404           ;LEFT TWO, UP FOUR.
        .WORD   INT+20000+1104          ;LEFT FOUR, DOWN 4.
        .WORD   1004                    ;INVISIBLE, UP 4, RIGHT 4
        .WORD   INT+5                   ;UP FIVE.
        .WORD   INT+200                 ;RIGHT ONE
        .WORD   INT+201                 ;RIGHT ONE, UP ONE.
        .WORD   INT+1                   ;UP ONE
        .WORD   INT+20000+201           ;LEFT ONE, UP ONE.
        .WORD   INT+20000+400           ;LEFT TWO.
        .WORD   INT+20000+301           ;LEFT ONE, DOWN ONE.
        .WORD   INT+101                 ;DOWN ONE.
        .WORD   INT+301                 ;DOWN ONE, RIGHT ONE.
        .WORD   INT+200                 ;RIGHT ONE.
        .WORD   101                     ;DOWN ONE.
        .WORD   INT+1203                ;RIGHT FIVE, UP THREE.
        .WORD   20000+1305              ;LEFT FIVE, DOWN FIVE.
        .WORD   INT+20000+1403          ;LEFT SIX, UP THREE.
        .WORD   DISTOP
        .WORD   0
        .PAGE
;
;               THIS LIST EXPLAINS HOW TO DRAW A FLAG.
;
FLAGL:  .WORD   115324
        .WORD   170240
FLAGX:  .WORD   0
FLAGY:  .WORD   0
        .WORD   104000
        .WORD   18.+INT
        .WORD   112727
        .WORD   16.+INT
        .WORD   0
        .WORD   INT
        .WORD   8.+DOWN
        .WORD   16.+INT+LEFT
        .WORD   0
        .WORD   112326
        .WORD   0
        .WORD   3
        .WORD   16.+INT
        .WORD   0
        .WORD   0
        .WORD   2
        .WORD   16.+INT+LEFT
        .WORD   0
        .WORD   DISTOP
        .WORD   0
FLAGEN:                                 ;NEXT LOCATION AFTER FLAG.
        .PAGE
;
;       COMMANDS FOR DISPLAYING THE LUNAR MODULE.
;
DESIGN: .WORD   DRAWIN,170200           ;LOAD STATUS.
        .WORD   DRAWIN,107124           ;AND SHORT VECTORS, INTENSITY 4.
        .WORD   DRAWIS                  ;DRAW BODY OF SHIP NOW.
        .BYTE   -6.,0.
        .WORD   DRAWVS
        .BYTE   -14.,8.
        .WORD   DRAWVS
        .BYTE   -14.,20.
        .WORD   DRAWVS
        .BYTE   -6.,29.
        .WORD   DRAWVS
```

```
        .BYTE   6.,29.
        .WORD   DRAWVS
        .BYTE   14.,20.
        .WORD   DRAWVS
        .BYTE   14.,8.
        .WORD   DRAWVS
        .BYTE   6.,0.
        .WORD   DRAWVS
        .BYTE   -6.,0.                  ;TOP OF SHIP DONE <OCTAGON>.
        .WORD   DRAWIS
        .BYTE   -17.,0.
        .WORD   DRAWVS
        .BYTE   -17.,-16.
        .WORD   DRAWVS
        .BYTE   17.,-16.
        .WORD   DRAWVS
        .BYTE   17.,0.
        .WORD   DRAWVS
        .BYTE   -17.,0.                 ;LOWER BODY OF SHIP DONE.
        .WORD   DRAWIN,107524           ;CHANGE TO LEVEL 3 INTENSITY.
        .WORD   DRAWVS                  ;DRAW LANDING LEGS NOW.
        .BYTE   -32.,-24.
        .WORD   DRAWIS                  ;POSITION OVER TO OTHER SIDE.
        .BYTE   17.,0.
        .WORD   DRAWVS
        .BYTE   32.,-24.
        .WORD   DRAWIN,106324           ;LEVEL ONE INTENSITY.
        .WORD   DRAWIS
        .BYTE   -17.,-14.
        .WORD   DRAWVS
        .BYTE   -28.,-18.
        .WORD   DRAWIS
        .BYTE   17.,-14.
        .WORD   DRAWVS
        .BYTE   28.,-18.                ;LOWER LANDING BRACE DONE.
        .WORD   DRAWIN,107124           ;INTENSITY 4 FOR LANDING PODS.
        .WORD   DRAWIS
        .BYTE   36.,-24.
        .WORD   DRAWVS
        .BYTE   28.,-24.
        .WORD   DRAWIS
        .BYTE   -28.,-24.
        .WORD   DRAWVS
        .BYTE   -36.,-24.               ;END OF LANDING PODS.
        .WORD   DRAWIS                  ;DRAW THE ENGINE NOW.
        .BYTE   -3.,-16.
        .WORD   DRAWVS
        .BYTE   -7.,-21.
        .WORD   DRAWVS
        .BYTE   7.,-21.
        .WORD   DRAWVS
        .BYTE   3.,-16.                 ;END OF THE ENGINE.
        .WORD   DRAWIS                  ;NOW BRING VECTOR BACK UP TO CENTER
        .BYTE   0,0                     ;OF THE LUNAR MODULE.
        .WORD   DRAWDN                  ;AND TERMINATE THE PICTURE.
        .PAGE
;
;       THESE TABLES CONTROL THE DRAWING OF THE ROCKET
;       ENGINE FLAME, AND ALL IT'S PURTIBATIONS.
;       Y THRUST TABLE ACCORDING TO PERCENTAGE THROTTLE.
;       13 SEGMENTS, EACH ONE APPROX 8% THROTTLE.
;
YTHRST: .BYTE   0.,-30.,-31.,-32.,-34.,-36.,-38.,-41.,-44.,-47.,-50.,-53.,-56.

;
```

```
;       Y UP DOWN IS A TABLE WHICH WILL SORT OF
;       RANDOMIZE THE HEIGHT OF THE FLAME AND SHOULD ENHANCE THE
;       FLICKERING APPEARANCE.
;
YUPDWN: .BYTE   0,1,3,6,4,3,1,-2,-6,-7,-5,-2,2,3,5,6,2,1
        .BYTE   -1,-4,-6,-5,-3,0,4,5,7,4,0,-1,-3,-1
;
;       "FLAME BOTTOM" CONTAINS THE TABLE OF THE X VALUES
;       FOR THE BOTTOM OF THE ROCKET FLAME. THEY ARE INDEXED SLIGHTLY
;       RANDOMLY AT TIMES.
;
FLAMBT: .BYTE   -20.,-16.,-13.,-10.,-7.,-4.,-2.
        .BYTE   0.,2.,4.,7.,10.,13.,16.,20.
;
;       THIS SECTION CONTAINS THE ACTUAL LIST OF COMMANDS TO
;       BE FOLLOWED WHEN DRAWING THE ROCKET FLAME.
;
        .EVEN                           ;REALIGN NOW IF NECESSARY.
FLAMDO: .WORD   DRAWIN,170200           ;LOAD STATUS PROPERLY.
        .WORD   DRAWIN                  ;INSERT COMMAND NOW.
FLAMEX: .WORD   0                       ;FLAME COMMAND GOES HERE.
        .WORD   DRAWIS                  ;MOVE POINTER OVER TO BOTTOM
        .BYTE   -6.,-21.                ;OFF THE ROCKET ENGINE.
        .WORD   DRAWVS                  ;NOW DRAW THE VECTORS
FLAMXS: .BYTE   0.,0.                   ;WHICH WE SHOULD SET UP HERE.
        .WORD   DRAWVS                  ;AND MOVE BACK UP AGAIN.
        .BYTE   -5.,-21.                ;BOTTOM OF THE ENGINE.
        .WORD   DRAWVS
        .BYTE   0.,0.                   ;OVERLAYED HERE ALSO.
        .WORD   DRAWVS
        .BYTE   -4.,-21.
        .WORD   DRAWVS
        .BYTE   0.,0.
        .WORD   DRAWVS
        .BYTE   -3.,-21.
        .WORD   DRAWVS
        .BYTE   0.,0.
        .WORD   DRAWVS
        .BYTE   -2.,-21.
        .WORD   DRAWVS
        .BYTE   0.,0.
        .WORD   DRAWVS
        .BYTE   -1.,-21.
        .WORD   DRAWVS
        .BYTE   0.,0.
        .WORD   DRAWVS
        .BYTE   0.,-21.
        .WORD   DRAWVS
        .BYTE   0.,0.
        .WORD   DRAWVS
        .BYTE   1.,-21.
        .WORD   DRAWVS
        .BYTE   0.,0.
        .WORD   DRAWVS
        .BYTE   2.,-21.
        .WORD   DRAWVS
        .BYTE   0.,0.
        .WORD   DRAWVS
        .BYTE   3.,-21.
        .WORD   DRAWVS
        .BYTE   0.,0.
        .WORD   DRAWVS
        .BYTE   4.,-21.
        .WORD   DRAWVS
        .BYTE   0.,0.
```

```
        .WORD   DRAWVS
        .BYTE   5.,-21.
        .WORD   DRAWVS
        .BYTE   0.,0.
        .WORD   DRAWVS
        .BYTE   6.,-21.
        .WORD   DRAWIS                  ;RETURN TO CENTER NOW.
        .BYTE   0.,0.
        .WORD   DRAWDN                  ;AND END THE COMMAND LIST.
        FLEN=12.                        ;NUMBER OF ITEMS TO INSERT IN THE TABLE.
        .PAGE
;
;               THESE ARE THE HIDEOUS TABLES THAT THE
;               SYSTEM NEEDS TO FUNCTION PROPERLY. THEY INCLUDE
;               TERRAIN TABLES (ALTITUDE), FEATURE TABLES (ROCKS,
;               OLD SHIPS, ETC.), CONVERSION TABLES, AND THE
;               SINE AND COSINE TABLES. THEY SHOULD BE BETWEEN
;               20000 AND 35000 OCTAL (LOCATION), OR YOU CAN BE SCREWED.
;               THIS IS THE CONVERSION TABLE FROM A BINARY
;               NUMBER LESS THAN 100 TO AN ASCII STRING. SEE THE
;               SUBROUTINE "ASCII".
;
TENTAB: .BYTE   '0,'0
        .BYTE   '0,'1
        .BYTE   '0,'2
        .BYTE   '0,'3
        .BYTE   '0,'4
        .BYTE   '0,'5
        .BYTE   '0,'6
        .BYTE   '0,'7
        .BYTE   '0,'8
        .BYTE   '0,'9
        .BYTE   '1,'0
        .BYTE   '1,'1
        .BYTE   '1,'2
        .BYTE   '1,'3
        .BYTE   '1,'4
        .BYTE   '1,'5
        .BYTE   '1,'6
        .BYTE   '1,'7
        .BYTE   '1,'8
        .BYTE   '1,'9
        .BYTE   '2,'0
        .BYTE   '2,'1
        .BYTE   '2,'2
        .BYTE   '2,'3
        .BYTE   '2,'4
        .BYTE   '2,'5
        .BYTE   '2,'6
        .BYTE   '2,'7
        .BYTE   '2,'8
        .BYTE   '2,'9
        .BYTE   '3,'0
        .BYTE   '3,'1
        .BYTE   '3,'2
        .BYTE   '3,'3
        .BYTE   '3,'4
        .BYTE   '3,'5
        .BYTE   '3,'6
        .BYTE   '3,'7
        .BYTE   '3,'8
        .BYTE   '3,'9
        .BYTE   '4,'0
        .BYTE   '4,'1
        .BYTE   '4,'2
```

```
        .BYTE   '4,'3
        .BYTE   '4,'4
        .BYTE   '4,'5
        .BYTE   '4,'6
        .BYTE   '4,'7
        .BYTE   '4,'8
        .BYTE   '4,'9
        .BYTE   '5,'0
        .BYTE   '5,'1
        .BYTE   '5,'2
        .BYTE   '5,'3
        .BYTE   '5,'4
        .BYTE   '5,'5
        .BYTE   '5,'6
        .BYTE   '5,'7
        .BYTE   '5,'8
        .BYTE   '5,'9
        .BYTE   '6,'0
        .BYTE   '6,'1
        .BYTE   '6,'2
        .BYTE   '6,'3
        .BYTE   '6,'4
        .BYTE   '6,'5
        .BYTE   '6,'6
        .BYTE   '6,'7
        .BYTE   '6,'8
        .BYTE   '6,'9
        .BYTE   '7,'0
        .BYTE   '7,'1
        .BYTE   '7,'2
        .BYTE   '7,'3
        .BYTE   '7,'4
        .BYTE   '7,'5
        .BYTE   '7,'6
        .BYTE   '7,'7
        .BYTE   '7,'8
        .BYTE   '7,'9
        .BYTE   '8,'0
        .BYTE   '8,'1
        .BYTE   '8,'2
        .BYTE   '8,'3
        .BYTE   '8,'4
        .BYTE   '8,'5
        .BYTE   '8,'6
        .BYTE   '8,'7
        .BYTE   '8,'8
        .BYTE   '8,'9
        .BYTE   '9,'0
        .BYTE   '9,'1
        .BYTE   '9,'2
        .BYTE   '9,'3
        .BYTE   '9,'4
        .BYTE   '9,'5
        .BYTE   '9,'6
        .BYTE   '9,'7
        .BYTE   '9,'8
        .BYTE   '9,'9
        .PAGE
;
;               THIS TABLE HOLDS THE TERAIN OF THE MOON IN
;               FEET ABOVE THE MEAN HEIGHT. IT GOES SOMEEWHAT FORWARD AND
;               BACKWARD BECAUSE THE PROGRAM MAY "SCAN" A LITTLE BIT.
;
        .WORD     718.                  ;X=-10
        .WORD     718.                  ;X= -9
```

```
         .WORD      718.                ;X= -8
         .WORD      750.                ;X= -7
         .WORD      750.                ;X= -6
         .WORD      750.                ;X= -5
         .WORD      782.                ;X= -4
         .WORD      782.                ;X= -3
         .WORD      782.                ;X= -2
         .WORD      814.                ;X= -1
TERAIN:  .WORD      780.                ;X=  0
         .WORD      750.                ;X=  1
         .WORD      703.                ;X=  2
         .WORD      650.                ;X=  3
         .WORD      620.                ;X=  4
         .WORD      575.                ;X=  5
         .WORD      500.                ;X=  6
         .WORD      420.                ;X=  7
         .WORD      400.                ;X=  8
         .WORD      350.                ;X=  9
         .WORD      270.                ;X= 10
         .WORD      246.                ;X= 11
         .WORD      200.                ;X= 12
         .WORD      180.                ;X= 13
         .WORD      107.                ;X= 14
         .WORD       24.                ;X= 15
         .WORD       54.                ;X= 16
         .WORD       53.                ;X= 17
         .WORD       51.                ;X= 18
         .WORD       82.                ;X= 19
         .WORD       80.                ;X= 20
         .WORD       78.                ;X= 21
         .WORD      109.                ;X= 22
         .WORD      107.                ;X= 23
         .WORD       74.                ;X= 24
         .WORD       72.                ;X= 25
         .WORD       70.                ;X= 26
         .WORD       69.                ;X= 27
         .WORD       99.                ;X= 28
         .WORD       98.                ;X= 29
         .WORD      128.                ;X= 30
         .WORD      126.                ;X= 31
         .WORD      125.                ;X= 32
         .WORD      123.                ;X= 33
         .WORD      122.                ;X= 34
         .WORD      152.                ;X= 35
         .WORD      182.                ;X= 36
         .WORD      174.                ;X= 37
         .WORD      166.                ;X= 38
         .WORD      158.                ;X= 39
         .WORD      181.                ;X= 40
         .WORD      173.                ;X= 41
         .WORD      165.                ;X= 42
         .WORD      156.                ;X= 43
         .WORD      148.                ;X= 44
         .WORD      140.                ;X= 45
         .WORD      131.                ;X= 46
         .WORD       91.                ;X= 47
         .WORD       83.                ;X= 48
         .WORD       75.                ;X= 49
         .WORD      182.                ;X= 50
         .WORD      161.                ;X= 51
         .WORD      236.                ;X= 52
         .WORD      120.                ;X= 53
         .WORD       67.                ;X= 54
         .WORD      174.                ;X= 55
         .WORD      153.                ;X= 56
```

```
        .WORD      197.                ;X= 57
        .WORD       80.                ;X= 58
        .WORD       59.                ;X= 59
        .WORD      866.                ;X= 60
        .WORD      123.                ;X= 61
        .WORD      201.                ;X= 62
        .WORD      250.                ;X= 63
        .WORD      275.                ;X= 64
        .WORD      300.                ;X= 65
        .WORD      375.                ;X= 66
        .WORD      410.                ;X= 67
        .WORD      579.                ;X= 68
        .WORD     926.                 ;X= 69
        .WORD     1832.                ;X= 70
        .WORD     1907.                ;X= 71
        .WORD     2013.                ;X= 72
        .WORD     2119.                ;X= 73
        .WORD     2257.                ;X= 74
        .WORD     2395.                ;X= 75
        .WORD     2533.                ;X= 76
        .WORD     2639.                ;X= 77
        .WORD     2745.                ;X= 78
        .WORD     2883.                ;X= 79
        .WORD     3021.                ;X= 80
        .WORD     3128.                ;X= 81
        .WORD     3226.                ;X= 82
        .WORD     3197.                ;X= 83
        .WORD     3231.                ;X= 84
        .WORD     3170.                ;X= 85
        .WORD     3140.                ;X= 86
        .WORD     3143.                ;X= 87
        .WORD     3145.                ;X= 88
        .WORD     3180.                ;X= 89
        .WORD     3246.                ;X= 90
        .WORD     3153.                ;X= 91
        .WORD     3252.                ;X= 92
        .WORD     3286.                ;X= 93
        .WORD     3353.                ;X= 94
        .WORD     3387.                ;X= 95
        .WORD     3390.                ;X= 96
        .WORD     3424.                ;X= 97
        .WORD     3523.                ;X= 98
        .WORD     3429.                ;X= 99
        .WORD     3440.                ;X=100
        .WORD     3450.                ;X=101
        .WORD     3429.                ;X=102
        .WORD     3407.                ;X=103
        .WORD     3386.                ;X=104
        .WORD     3396.                ;X=105
        .WORD     3375.                ;X=106
        .WORD     3321.                ;X=107
        .WORD     3300.                ;X=108
        .WORD     3246.                ;X=109
        .WORD     3257.                ;X=110
        .WORD     3203.                ;X=111
        .WORD     3182.                ;X=112
        .WORD     3128.                ;X=113
        .WORD     3107.                ;X=114
        .WORD     3085.                ;X=115
        .WORD     3064.                ;X=116
        .WORD     3074.                ;X=117
        .WORD     3085.                ;X=118
        .WORD     3095.                ;X=119
        .WORD     3195.                ;X=120
        .WORD     3167.                ;X=121
```

```
        .WORD     3139.                 ;X=122
        .WORD     3239.                 ;X=123
        .WORD     2954.                 ;X=124
        .WORD     2926.                 ;X=125
        .WORD     2834.                 ;X=126
        .WORD     2710.                 ;X=127
        .WORD     2746.                 ;X=128
        .WORD     2718.                 ;X=129
        .WORD     2753.                 ;X=130
        .WORD     2469.                 ;X=131
        .WORD     2569.                 ;X=132
        .WORD     2509.                 ;X=133
        .WORD     2513.                 ;X=134
        .WORD     2580.                 ;X=135
        .WORD     2584.                 ;X=136
        .WORD     2620.                 ;X=137
        .WORD     2720.                 ;X=138
        .WORD     2820.                 ;X=139
        .WORD     2919.                 ;X=140
        .WORD     2859.                 ;X=141
        .WORD     2703.                 ;X=142
        .WORD     2739.                 ;X=143
        .WORD     2967.                 ;X=144
        .WORD     2906.                 ;X=145
        .WORD     3134.                 ;X=146
        .WORD     3042.                 ;X=147
        .WORD     2918.                 ;X=148
        .WORD     2858.                 ;X=149
        .WORD     2759.                 ;X=150
        .WORD     2468.                 ;X=151
        .WORD     2498.                 ;X=152
        .WORD     2399.                 ;X=153
        .WORD     2396.                 ;X=154
        .WORD     2362.                 ;X=155
        .WORD     2071.                 ;X=156
        .WORD     1940.                 ;X=157
        .WORD     1970.                 ;X=158
        .WORD     2063.                 ;X=159
        .WORD     2028.                 ;X=160
        .WORD     1930.                 ;X=161
        .WORD     1799.                 ;X=162
        .WORD     1668.                 ;X=163
        .WORD     1762.                 ;X=164
        .WORD     1823.                 ;X=165
        .WORD     1660.                 ;X=166
        .WORD     1626.                 ;X=167
        .WORD     1527.                 ;X=168
        .WORD     1428.                 ;X=169
        .WORD     1650.                 ;X=170
        .WORD     1615.                 ;X=171
        .WORD     1676.                 ;X=172
        .WORD     1770.                 ;X=173
        .WORD     1703.                 ;X=174
        .WORD     1636.                 ;X=175
        .WORD     1666.                 ;X=176
        .WORD     1599.                 ;X=177
        .WORD     1436.                 ;X=178
        .WORD     1402.                 ;X=179
        .WORD     1431.                 ;X=180
        .WORD     1332.                 ;X=181
        .WORD     1426.                 ;X=182
        .WORD     1647.                 ;X=183
        .WORD     1516.                 ;X=184
        .WORD     1354.                 ;X=185
        .WORD     1319.                 ;X=186
```

```
        .WORD     1380.                 ;X=187
        .WORD     1090.                 ;X=188
        .WORD     1055.                 ;X=189
        .WORD     1148.                 ;X=190
        .WORD     1146.                 ;X=191
        .WORD     1207.                 ;X=192
        .WORD     1172.                 ;X=193
        .WORD     1202.                 ;X=194
        .WORD     1071.                 ;X=195
        .WORD      780.                 ;X=196
        .WORD      746.                 ;X=197
        .WORD      711.                 ;X=198
        .WORD      933.                 ;X=199
        .WORD     1092.                 ;X=200
        .WORD     1348.                 ;X=201
        .WORD     1539.                 ;X=202
        .WORD     1827.                 ;X=203
        .WORD     1858.                 ;X=204
        .WORD     2114.                 ;X=205
        .WORD     2337.                 ;X=206
        .WORD     2497.                 ;X=207
        .WORD     2624.                 ;X=208
        .WORD     2752.                 ;X=209
        .WORD     2847.                 ;X=210
        .WORD     3007.                 ;X=211
        .WORD     3166.                 ;X=212
        .WORD     3422.                 ;X=213
        .WORD     3581.                 ;X=214
        .WORD     3709.                 ;X=215
        .WORD     3964.                 ;X=216
        .WORD     4124.                 ;X=217
        .WORD     4315.                 ;X=218
        .WORD     4443.                 ;X=219
        .WORD     4449.                 ;X=220
        .WORD     4711.                 ;X=221
        .WORD     4845.                 ;X=222
        .WORD     4883.                 ;X=223
        .WORD     4985.                 ;X=224
        .WORD     5055.                 ;X=225
        .WORD     5061.                 ;X=226
        .WORD     5260.                 ;X=227
        .WORD     5362.                 ;X=228
        .WORD     5592.                 ;X=229
        .WORD     5726.                 ;X=230
        .WORD     5860.                 ;X=231
        .WORD     5994.                 ;X=232
        .WORD     6160.                 ;X=233
        .WORD     6259.                 ;X=234
        .WORD     6422.                 ;X=235
        .WORD     6584.                 ;X=236
        .WORD     6715.                 ;X=237
        .WORD     6877.                 ;X=238
        .WORD     7008.                 ;X=239
        .WORD     7138.                 ;X=240
        .WORD     7301.                 ;X=241
        .WORD     7432.                 ;X=242
        .WORD     7562.                 ;X=243
        .WORD     7693.                 ;X=244
        .WORD     7791.                 ;X=245
        .WORD     7954.                 ;X=246
        .WORD     8084.                 ;X=247
        .WORD     8215.                 ;X=248
        .WORD     8345.                 ;X=249
        .WORD     8400.                 ;X=250
        .WORD     8678.                 ;X=251
```

```
        .WORD     8765.                 ;X=252
        .WORD     8851.                 ;X=253
        .WORD     9033.                 ;X=254
        .WORD     9152.                 ;X=255
        .WORD     9366.                 ;X=256
        .WORD     9388.                 ;X=257
        .WORD     9539.                 ;X=258
        .WORD     9945.                 ;X=259
        .WORD    10191.                 ;X=260
        .WORD    10377.                 ;X=261
        .WORD    10659.                 ;X=262
        .WORD    10781.                 ;X=263
        .WORD    11063.                 ;X=264
        .WORD    11312.                 ;X=265
        .WORD    11370.                 ;X=266
        .WORD    11396.                 ;X=267
        .WORD    11646.                 ;X=268
        .WORD    11768.                 ;X=269
        .WORD    11985.                 ;X=270
        .WORD    12203.                 ;X=271
        .WORD    12357.                 ;X=272
        .WORD    12607.                 ;X=273
        .WORD    12857.                 ;X=274
        .WORD    12965.                 ;X=275
        .WORD    13073.                 ;X=276
        .WORD    13117.                 ;X=277
        .WORD    13193.                 ;X=278
        .WORD    13238.                 ;X=279
        .WORD    13282.                 ;X=280
        .WORD    13358.                 ;X=281
        .WORD    13434.                 ;X=282
        .WORD    13478.                 ;X=283
        .WORD    13555.                 ;X=284
        .WORD    13631.                 ;X=285
        .WORD    13707.                 ;X=286
        .WORD    13815.                 ;X=287
        .WORD    13923.                 ;X=288
        .WORD    13967.                 ;X=289
        .WORD    14029.                 ;X=290
        .WORD    13996.                 ;X=291
        .WORD    14121.                 ;X=292
        .WORD    14215.                 ;X=293
        .WORD    14308.                 ;X=294
        .WORD    14466.                 ;X=295
        .WORD    14591.                 ;X=296
        .WORD    14749.                 ;X=297
        .WORD    14874.                 ;X=298
        .WORD    15000.                 ;X=299
        .WORD    14754.                 ;X=300
        .WORD    14604.                 ;X=301
        .WORD    14326.                 ;X=302
        .WORD    14176.                 ;X=303
        .WORD    14090.                 ;X=304
        .WORD    13940.                 ;X=305
        .WORD    13662.                 ;X=306
        .WORD    13480.                 ;X=307
        .WORD    13394.                 ;X=308
        .WORD    13116.                 ;X=309
        .WORD    12548.                 ;X=310
        .WORD    11469.                 ;X=311
        .WORD    10582.                 ;X=312
        .WORD    10015.                 ;X=313
        .WORD     9160.                 ;X=314
        .WORD     8960.                 ;X=315
        .WORD     8696.                 ;X=316
```

```
        .WORD     8432.                 ;X=317
        .WORD     8200.                 ;X=318
        .WORD     7936.                 ;X=319
        .WORD     7770.                 ;X=320
        .WORD     7509.                 ;X=321
        .WORD     7375.                 ;X=322
        .WORD     7210.                 ;X=323
        .WORD     7076.                 ;X=324
        .WORD     6911.                 ;X=325
        .WORD     6745.                 ;X=326
        .WORD     6644.                 ;X=327
        .WORD     6478.                 ;X=328
        .WORD     6185.                 ;X=329
        .WORD     6019.                 ;X=330
        .WORD     5758.                 ;X=331
        .WORD     5656.                 ;X=332
        .WORD     5491.                 ;X=333
        .WORD     5261.                 ;X=334
        .WORD     5096.                 ;X=335
        .WORD     4867.                 ;X=336
        .WORD     4712.                 ;X=337
        .WORD     4398.                 ;X=338
        .WORD     4020.                 ;X=339
        .WORD     3834.                 ;X=340
        .WORD     3296.                 ;X=341
        .WORD     3014.                 ;X=342
        .WORD     2668.                 ;X=343
        .WORD     2642.                 ;X=344
        .WORD     2424.                 ;X=345
        .WORD     2398.                 ;X=346
        .WORD     2212.                 ;X=347
        .WORD     2186.                 ;X=348
        .WORD     1840.                 ;X=349
        .WORD     1828.                 ;X=350
        .WORD     2008.                 ;X=351
        .WORD     2156.                 ;X=352
        .WORD     2272.                 ;X=353
        .WORD     2356.                 ;X=354
        .WORD     2312.                 ;X=355
        .WORD     2268.                 ;X=356
        .WORD     2224.                 ;X=357
        .WORD     2276.                 ;X=358
        .WORD     2328.                 ;X=359
        .WORD     2413.                 ;X=360
        .WORD     2511.                 ;X=361
        .WORD     2673.                 ;X=362
        .WORD     2836.                 ;X=363
        .WORD     2998.                 ;X=364
        .WORD     3129.                 ;X=365
        .WORD     3227.                 ;X=366
        .WORD     3390.                 ;X=367
        .WORD     3488.                 ;X=368
        .WORD     3586.                 ;X=369
        .WORD     3717.                 ;X=370
        .WORD     3879.                 ;X=371
        .WORD     4010.                 ;X=372
        .WORD     4108.                 ;X=373
        .WORD     4239.                 ;X=374
        .WORD     4369.                 ;X=375
        .WORD     4531.                 ;X=376
        .WORD     4694.                 ;X=377
        .WORD     4824.                 ;X=378
        .WORD     4987.                 ;X=379
        .WORD     5117.                 ;X=380
        .WORD     5216.                 ;X=381
```

```
        .WORD     5314.                 ;X=382
        .WORD     5445.                 ;X=383
        .WORD     5543.                 ;X=384
        .WORD     5705.                 ;X=385
        .WORD     5804.                 ;X=386
        .WORD     5966.                 ;X=387
        .WORD     6097.                 ;X=388
        .WORD     6195.                 ;X=389
        .WORD     6294.                 ;X=390
        .WORD     6392.                 ;X=391
        .WORD     6522.                 ;X=392
        .WORD     6685.                 ;X=393
        .WORD     6815.                 ;X=394
        .WORD     6914.                 ;X=395
        .WORD     7076.                 ;X=396
        .WORD     7239.                 ;X=397
        .WORD     7337.                 ;X=398
        .WORD     7436.                 ;X=399
        .WORD     7074.                 ;X=400
        .WORD     6809.                 ;X=401
        .WORD     6448.                 ;X=402
        .WORD     6118.                 ;X=403
        .WORD     5629.                 ;X=404
        .WORD     5268.                 ;X=405
        .WORD     5002.                 ;X=406
        .WORD     4769.                 ;X=407
        .WORD     4535.                 ;X=408
        .WORD     4270.                 ;X=409
        .WORD     3909.                 ;X=410
        .WORD     3515.                 ;X=411
        .WORD     3108.                 ;X=412
        .WORD     2796.                 ;X=413
        .WORD     2389.                 ;X=414
        .WORD     2077.                 ;X=415
        .WORD     1509.                 ;X=416
        .WORD     1326.                 ;X=417
        .WORD     1078.                 ;X=418
        .WORD      895.                 ;X=419
        .WORD      583.                 ;X=420
        .WORD      144.                 ;X=421
        .WORD       64.                 ;X=422
        .WORD      -15.                 ;X=423
        .WORD     -127.                 ;X=424
        .WORD     -114.                 ;X=425
        .WORD     -101.                 ;X=426
        .WORD      -56.                 ;X=427
        .WORD      -10.                 ;X=428
        .WORD        2.                 ;X=429
        .WORD      -15.                 ;X=430
        .WORD       29.                 ;X=431
        .WORD       74.                 ;X=432
        .WORD       56.                 ;X=433
        .WORD       69.                 ;X=434
        .WORD       82.                 ;X=435
        .WORD       95.                 ;X=436
        .WORD       88.                 ;X=437
        .WORD       49.                 ;X=438
        .WORD       73.                 ;X=439
        .WORD       34.                 ;X=440
        .WORD       59.                 ;X=441
        .WORD       51.                 ;X=442
        .WORD       44.                 ;X=443
        .WORD       68.                 ;X=444
        .WORD       29.                 ;X=445
        .WORD       22.                 ;X=446
```

```
        .WORD       14.                 ;X=447
        .WORD        7.                 ;X=448
        .WORD       32.                 ;X=449
        .WORD       85.                 ;X=450
        .WORD      171.                 ;X=451
        .WORD      256.                 ;X=452
        .WORD      310.                 ;X=453
        .WORD      460.                 ;X=454
        .WORD      555.                 ;X=455
        .WORD      618.                 ;X=456
        .WORD      681.                 ;X=457
        .WORD      776.                 ;X=458
        .WORD      839.                 ;X=459
        .WORD      870.                 ;X=460
        .WORD      933.                 ;X=461
        .WORD      996.                 ;X=462
        .WORD     1092.                 ;X=463
        .WORD     1187.                 ;X=464
        .WORD     1250.                 ;X=465
        .WORD     1345.                 ;X=466
        .WORD     1408.                 ;X=467
        .WORD     1471.                 ;X=468
        .WORD     1566.                 ;X=469
        .WORD     1597.                 ;X=470
        .WORD     1628.                 ;X=471
        .WORD     1692.                 ;X=472
        .WORD     1755.                 ;X=473
        .WORD     1818.                 ;X=474
        .WORD     1881.                 ;X=475
        .WORD     1944.                 ;X=476
        .WORD     2007.                 ;X=477
        .WORD     2070.                 ;X=478
        .WORD     2133.                 ;X=479
        .WORD     2196.                 ;X=480
        .WORD     2260.                 ;X=481
        .WORD     2323.                 ;X=482
        .WORD     2386.                 ;X=483
        .WORD     2481.                 ;X=484
        .WORD     2544.                 ;X=485
        .WORD     2607.                 ;X=486
        .WORD     2670.                 ;X=487
        .WORD     2733.                 ;X=488
        .WORD     2764.                 ;X=489
        .WORD     2828.                 ;X=490
        .WORD     2859.                 ;X=491
        .WORD     2922.                 ;X=492
        .WORD     2985.                 ;X=493
        .WORD     3080.                 ;X=494
        .WORD     3111.                 ;X=495
        .WORD     3174.                 ;X=496
        .WORD     3237.                 ;X=497
        .WORD     3300.                 ;X=498
        .WORD     3396.                 ;X=499
        .WORD     3356.                 ;X=500
        .WORD     3252.                 ;X=501
        .WORD     2957.                 ;X=502
        .WORD     3173.                 ;X=503
        .WORD     3390.                 ;X=504
        .WORD     3446.                 ;X=505
        .WORD     3406.                 ;X=506
        .WORD     3303.                 ;X=507
        .WORD     3231.                 ;X=508
        .WORD     3448.                 ;X=509
        .WORD     3142.                 ;X=510
        .WORD     3156.                 ;X=511
```

```
        .WORD     3107.                 ;X=512
        .WORD     3089.                 ;X=513
        .WORD     2944.                 ;X=514
        .WORD     2835.                 ;X=515
        .WORD     2726.                 ;X=516
        .WORD     2617.                 ;X=517
        .WORD     2476.                 ;X=518
        .WORD     2367.                 ;X=519
        .WORD     2373.                 ;X=520
        .WORD     2410.                 ;X=521
        .WORD     2511.                 ;X=522
        .WORD     2580.                 ;X=523
        .WORD     2618.                 ;X=524
        .WORD     2719.                 ;X=525
        .WORD     2724.                 ;X=526
        .WORD     2793.                 ;X=527
        .WORD     2798.                 ;X=528
        .WORD     2772.                 ;X=529
        .WORD     2740.                 ;X=530
        .WORD     2709.                 ;X=531
        .WORD     2645.                 ;X=532
        .WORD     2614.                 ;X=533
        .WORD     2582.                 ;X=534
        .WORD     2551.                 ;X=535
        .WORD     2519.                 ;X=536
        .WORD     2488.                 ;X=537
        .WORD     2456.                 ;X=538
        .WORD     2425.                 ;X=539
        .WORD     2393.                 ;X=540
        .WORD     2362.                 ;X=541
        .WORD     2362.                 ;X=542
        .WORD     2363.                 ;X=543
        .WORD     2363.                 ;X=544
        .WORD     2296.                 ;X=545
        .WORD     2228.                 ;X=546
        .WORD     2224.                 ;X=547
        .WORD     2092.                 ;X=548
        .WORD     2153.                 ;X=549
        .WORD     2032.                 ;X=550
        .WORD     2072.                 ;X=551
        .WORD     2144.                 ;X=552
        .WORD     2248.                 ;X=553
        .WORD     2128.                 ;X=554
        .WORD     1976.                 ;X=555
        .WORD     2016.                 ;X=556
        .WORD     1864.                 ;X=557
        .WORD     1936.                 ;X=558
        .WORD     1912.                 ;X=559
        .WORD     1856.                 ;X=560
        .WORD     1704.                 ;X=561
        .WORD     1616.                 ;X=562
        .WORD     1496.                 ;X=563
        .WORD     1344.                 ;X=564
        .WORD     1320.                 ;X=565
        .WORD     1232.                 ;X=566
        .WORD      952.                 ;X=567
        .WORD      992.                 ;X=568
        .WORD     1032.                 ;X=569
        .WORD     1136.                 ;X=570
        .WORD     1048.                 ;X=571
        .WORD     1056.                 ;X=572
        .WORD      968.                 ;X=573
        .WORD      688.                 ;X=574
        .WORD      792.                 ;X=575
        .WORD      768.                 ;X=576
```

```
        .WORD      744.                ;X=577
        .WORD      624.                ;X=578
        .WORD      343.                ;X=579
        .WORD      298.                ;X=580
        .WORD      444.                ;X=581
        .WORD      494.                ;X=582
        .WORD      289.                ;X=583
        .WORD      211.                ;X=584
        .WORD      165.                ;X=585
        .WORD      152.                ;X=586
        .WORD       74.                ;X=587
        .WORD      156.                ;X=588
        .WORD      174.                ;X=589
        .WORD      225.                ;X=590
        .WORD      147.                ;X=591
        .WORD      -58.                ;X=592
        .WORD      -39.                ;X=593
        .WORD      -85.                ;X=594
        .WORD     -163.                ;X=595
        .WORD      143.                ;X=596
        .WORD      449.                ;X=597
        .WORD      755.                ;X=598
        .WORD      934.                ;X=599
        .WORD      787.                ;X=600
        .WORD      673.                ;X=601
        .WORD      654.                ;X=602
        .WORD      636.                ;X=603
        .WORD      618.                ;X=604
        .WORD      471.                ;X=605
        .WORD      325.                ;X=606
        .WORD      403.                ;X=607
        .WORD      416.                ;X=608
        .WORD      270.                ;X=609
        .WORD      220.                ;X=610
        .WORD      105.                ;X=611
        .WORD       87.                ;X=612
        .WORD      101.                ;X=613
        .WORD      -13.                ;X=614
        .WORD      -63.                ;X=615
        .WORD       46.                ;X=616
        .WORD      155.                ;X=617
        .WORD      137.                ;X=618
        .WORD      183.                ;X=619
        .WORD      164.                ;X=620
        .WORD       50.                ;X=621
        .WORD       31.                ;X=622
        .WORD       13.                ;X=623
        .WORD      123.                ;X=624
        .WORD      104.                ;X=625
        .WORD      214.                ;X=626
        .WORD      228.                ;X=627
        .WORD      273.                ;X=628
        .WORD      287.                ;X=629
        .WORD      141.                ;X=630
        .WORD      250.                ;X=631
        .WORD      328.                ;X=632
        .WORD      278.                ;X=633
        .WORD      323.                ;X=634
        .WORD      401.                ;X=635
        .WORD      511.                ;X=636
        .WORD      364.                ;X=637
        .WORD      218.                ;X=638
        .WORD      200.                ;X=639
        .WORD      212.                ;X=640
        .WORD      160.                ;X=641
```

```
        .WORD     172.              ;X=642
        .WORD     152.              ;X=643
        .WORD     132.              ;X=644
        .WORD     112.              ;X=645
        .WORD      92.              ;X=646
        .WORD      72.              ;X=647
        .WORD      52.              ;X=648
        .WORD      63.              ;X=649
        .WORD      94.              ;X=650
        .WORD      93.              ;X=651
        .WORD      92.              ;X=652
        .WORD      90.              ;X=653
        .WORD      89.              ;X=654
        .WORD     120.              ;X=655
        .WORD     119.              ;X=656
        .WORD     117.              ;X=657
        .WORD     116.              ;X=658
        .WORD      83.              ;X=659
        .WORD      81.              ;X=660
        .WORD      80.              ;X=661
        .WORD      79.              ;X=662
        .WORD     110.              ;X=663
        .WORD     108.              ;X=664
        .WORD     139.              ;X=665
        .WORD     138.              ;X=666
        .WORD     136.              ;X=667
        .WORD     103.              ;X=668
        .WORD     134.              ;X=669
        .WORD     165.              ;X=670
        .WORD     131.              ;X=671
        .WORD     162.              ;X=672
        .WORD     193.              ;X=673
        .WORD     159.              ;X=674
        .WORD     185.              ;X=675
        .WORD     179.              ;X=676
        .WORD     140.              ;X=677
        .WORD     102.              ;X=678
        .WORD     128.              ;X=679
        .WORD     153.              ;X=680
        .WORD     147.              ;X=681
        .WORD     172.              ;X=682
        .WORD     134.              ;X=683
        .WORD     160.              ;X=684
        .WORD     153.              ;X=685
        .WORD     179.              ;X=686
        .WORD     204.              ;X=687
        .WORD     166.              ;X=688
        .WORD     160.              ;X=689
        .WORD     153.              ;X=690
        .WORD     179.              ;X=691
        .WORD     172.              ;X=692
        .WORD     166.              ;X=693
        .WORD     160.              ;X=694
        .WORD     153.              ;X=695
        .WORD     125.              ;X=696
        .WORD     110.              ;X=697
        .WORD     95.               ;X=698
        .WORD     80.               ;X=699
        .WORD     85.               ;X=700
        .WORD      82.              ;X=701
        .WORD      87.              ;X=702
        .WORD     84.               ;X=703
        .WORD     81.               ;X=704
        .WORD     86.               ;X=705
        .WORD     103.              ;X=706
```

```
        .WORD     125.              ;X=707
        .WORD     140.              ;X=708
        .WORD     152.              ;X=709
        .WORD     135.              ;X=710
        .WORD     130.              ;X=711
        .WORD     125.              ;X=712
        .WORD      88.              ;X=713
        .WORD      83.              ;X=714
        .WORD      46.              ;X=715
        .WORD      40.              ;X=716
        .WORD      35.              ;X=717
        .WORD      30.              ;X=718
        .WORD      25.              ;X=719
        .WORD      52.              ;X=720
        .WORD      47.              ;X=721
        .WORD      42.              ;X=722
        .WORD      37.              ;X=723
        .WORD      63.              ;X=724
        .WORD      61.              ;X=725
        .WORD      90.              ;X=726
        .WORD     120.              ;X=727
        .WORD     117.              ;X=728
        .WORD     115.              ;X=729
        .WORD     112.              ;X=730
        .WORD      78.              ;X=731
        .WORD      75.              ;X=732
        .WORD     104.              ;X=733
        .WORD      70.              ;X=734
        .WORD      67.              ;X=735
        .WORD      33.              ;X=736
        .WORD      62.              ;X=737
        .WORD      60.              ;X=738
        .WORD      57.              ;X=739
        .WORD      55.              ;X=740
        .WORD      52.              ;X=741
        .WORD      81.              ;X=742
        .WORD     111.              ;X=743
        .WORD     140.              ;X=744
        .WORD     138.              ;X=745
        .WORD     167.              ;X=746
        .WORD     165.              ;X=747
        .WORD     162.              ;X=748
        .WORD     192.              ;X=749
        .WORD     137.              ;X=750
        .WORD     179.              ;X=751
        .WORD     124.              ;X=752
        .WORD     230.              ;X=753
        .WORD     336.              ;X=754
        .WORD     377.              ;X=755
        .WORD     387.              ;X=756
        .WORD     301.              ;X=757
        .WORD     406.              ;X=758
        .WORD     288.              ;X=759
        .WORD     426.              ;X=760
        .WORD     467.              ;X=761
        .WORD     541.              ;X=762
        .WORD     423.              ;X=763
        .WORD     528.              ;X=764
        .WORD     506.              ;X=765
        .WORD     452.              ;X=766
        .WORD     365.              ;X=767
        .WORD     311.              ;X=768
        .WORD     193.              ;X=769
        .WORD     298.              ;X=770
        .WORD     436.              ;X=771
```

```
        .WORD     510.              ;X=772
        .WORD     487.              ;X=773
        .WORD     465.              ;X=774
        .WORD     347.              ;X=775
        .WORD     260.              ;X=776
        .WORD     334.              ;X=777
        .WORD     440.              ;X=778
        .WORD     417.              ;X=779
        .WORD     427.              ;X=780
        .WORD     341.              ;X=781
        .WORD     318.              ;X=782
        .WORD     232.              ;X=783
        .WORD     146.              ;X=784
        .WORD      27.              ;X=785
        .WORD      69.              ;X=786
        .WORD     175.              ;X=787
        .WORD     280.              ;X=788
        .WORD     258.              ;X=789
        .WORD     204.              ;X=790
        .WORD     277.              ;X=791
        .WORD     415.              ;X=792
        .WORD     425.              ;X=793
        .WORD     466.              ;X=794
        .WORD     412.              ;X=795
        .WORD     326.              ;X=796
        .WORD     463.              ;X=797
        .WORD     537.              ;X=798
        .WORD     547.              ;X=799
        .WORD     798.              ;X=800
        .WORD     665.              ;X=801
        .WORD     660.              ;X=802
        .WORD     592.              ;X=803
        .WORD     555.              ;X=804
        .WORD     646.              ;X=805
        .WORD     514.              ;X=806
        .WORD     765.              ;X=807
        .WORD     760.              ;X=808
        .WORD     755.              ;X=809
        .WORD     875.              ;X=810
        .WORD     994.              ;X=811
        .WORD    1081.              ;X=812
        .WORD    1136.              ;X=813
        .WORD    1223.              ;X=814
        .WORD    1311.              ;X=815
        .WORD    1398.              ;X=816
        .WORD    1485.              ;X=817
        .WORD    1604.              ;X=818
        .WORD    1691.              ;X=819
        .WORD    1747.              ;X=820
        .WORD    1834.              ;X=821
        .WORD    1921.              ;X=822
        .WORD    1976.              ;X=823
        .WORD    2095.              ;X=824
        .WORD    2183.              ;X=825
        .WORD    2238.              ;X=826
        .WORD    2357.              ;X=827
        .WORD    2412.              ;X=828
        .WORD    2468.              ;X=829
        .WORD    2422.              ;X=830
        .WORD    2440.              ;X=831
        .WORD    2523.              ;X=832
        .WORD    2381.              ;X=833
        .WORD    2336.              ;X=834
        .WORD    2380.              ;X=835
        .WORD    2392.              ;X=836
```

```
        .WORD   2404.           ;X=837
        .WORD   2449.           ;X=838
        .WORD   2493.           ;X=839
        .WORD   2505.           ;X=840
        .WORD   2517.           ;X=841
        .WORD   2562.           ;X=842
        .WORD   2574.           ;X=843
        .WORD   2586.           ;X=844
        .WORD   2598.           ;X=845
        .WORD   2643.           ;X=846
        .WORD   2687.           ;X=847
        .WORD   2731.           ;X=848
        .WORD   2775.           ;X=849
        .WORD   2849.           ;X=850
        .WORD   2890.           ;X=851
        .WORD   2932.           ;X=852
        .WORD   2781.           ;X=853
        .WORD   2759.           ;X=854
        .WORD   2832.           ;X=855
        .WORD   2810.           ;X=856
        .WORD   2787.           ;X=857
        .WORD   2509.           ;X=858
        .WORD   2742.           ;X=859
        .WORD   2752.           ;X=860
        .WORD   2825.           ;X=861
        .WORD   2899.           ;X=862
        .WORD   2844.           ;X=863
        .WORD   2822.           ;X=864
        .WORD   2543.           ;X=865
        .WORD   2777.           ;X=866
        .WORD   2882.           ;X=867
        .WORD   2892.           ;X=868
        .WORD   2773.           ;X=869
        .WORD   2879.           ;X=870
        .WORD   2792.           ;X=871
        .WORD   2770.           ;X=872
        .WORD   2747.           ;X=873
        .WORD   2629.           ;X=874
        .WORD   2670.           ;X=875
        .WORD   2391.           ;X=876
        .WORD   2497.           ;X=877
        .WORD   2570.           ;X=878
        .WORD   2804.           ;X=879
        .WORD   2781.           ;X=880
        .WORD   2823.           ;X=881
        .WORD   2544.           ;X=882
        .WORD   2490.           ;X=883
        .WORD   2435.           ;X=884
        .WORD   2477.           ;X=885
        .WORD   2518.           ;X=886
        .WORD   2368.           ;X=887
        .WORD   2313.           ;X=888
        .WORD   2419.           ;X=889
        .WORD   2524.           ;X=890
        .WORD   2502.           ;X=891
        .WORD   2479.           ;X=892
        .WORD   2553.           ;X=893
        .WORD   2562.           ;X=894
        .WORD   2540.           ;X=895
        .WORD   2549.           ;X=896
        .WORD   2591.           ;X=897
        .WORD   2664.           ;X=898
        .WORD   2386.           ;X=899
        .WORD   2455.           ;X=900
        .WORD   2428.           ;X=901
```

```
        .WORD     2145.                 ;X=902
        .WORD     2119.                 ;X=903
        .WORD     2156.                 ;X=904
        .WORD     2001.                 ;X=905
        .WORD     1974.                 ;X=906
        .WORD     2204.                 ;X=907
        .WORD     2145.                 ;X=908
        .WORD     2374.                 ;X=909
        .WORD     2476.                 ;X=910
        .PAGE
;
;               THIS TABLE HOLDS THE FEATURES OF THE MOON
;               THAT MAY CHANGE FROM TIME TO TIME.
;               0=NOTHING SPECIAL.
;               1=LANDED SHIP.
;               2=PLANTED AMERICAN FLAG.
;               3=DEAD SHIP TILTED LEFT.
;               4=DEAD SHIP TILTED RIGHT.
;               5=BOULDER. <ROCK TO YOU STUPID PEOPLES>
;               6=MACDONALD'S <LEFT OR RIGHT EDGES ONLY>
;               7=CENTER OF MACDONALD'S <WHERE TO DRAW CENTER OF IT>
;
        .BYTE   0*20+0                ;X= -9  X=-10
        .BYTE   5*20+0                ;X= -7  X= -8
        .BYTE   0*20+0                ;X= -5  X= -6
        .BYTE   0*20+0                ;X= -3  X= -4
        .BYTE   0*20+0                ;X= -1  X= -2
FEATUR: .BYTE   0*20+5                ;X=  1  X=  0
        .BYTE   0*20+0                ;X=  3  X=  2
        .BYTE   0*20+0                ;X=  5  X=  4
        .BYTE   0*20+0                ;X=  7  X=  6
        .BYTE   0*20+0                ;X=  9  X=  8
        .BYTE   0*20+0                ;X= 11  X= 10
        .BYTE   0*20+0                ;X= 13  X= 12
        .BYTE   0*20+0                ;X= 15  X= 14
        .BYTE   0*20+0                ;X= 17  X= 16
        .BYTE   0*20+0                ;X= 19  X= 18
        .BYTE   0*20+0                ;X= 21  X= 20
        .BYTE   0*20+0                ;X= 23  X= 22
        .BYTE   0*20+0                ;X= 25  X= 24
        .BYTE   0*20+5                ;X= 27  X= 26
        .BYTE   5*20+5                ;X= 29  X= 28
        .BYTE   0*20+0                ;X= 31  X= 30
        .BYTE   0*20+0                ;X= 33  X= 32
        .BYTE   0*20+0                ;X= 35  X= 34
        .BYTE   0*20+0                ;X= 37  X= 36
        .BYTE   0*20+0                ;X= 39  X= 38
        .BYTE   0*20+0                ;X= 41  X= 40
        .BYTE   0*20+0                ;X= 43  X= 42
        .BYTE   0*20+5                ;X= 45  X= 44
        .BYTE   0*20+0                ;X= 47  X= 46
        .BYTE   0*20+0                ;X= 49  X= 48
        .BYTE   5*20+5                ;X= 51  X= 50
        .BYTE   0*20+5                ;X= 53  X= 52
        .BYTE   5*20+5                ;X= 55  X= 54
        .BYTE   0*20+0                ;X= 57  X= 56
        .BYTE   0*20+0                ;X= 59  X= 58
        .BYTE   0*20+5                ;X= 61  X= 60
        .BYTE   0*20+5                ;X= 63  X= 62
        .BYTE   5*20+5                ;X= 65  X= 64
        .BYTE   5*20+0                ;X= 67  X= 66
        .BYTE   0*20+5                ;X= 69  X= 68
        .BYTE   5*20+5                ;X= 71  X= 70
        .BYTE   5*20+5                ;X= 73  X= 72
        .BYTE   5*20+0                ;X= 75  X= 74
```

```
        .BYTE   0*20+5                  ;X= 77  X= 76
        .BYTE   0*20+5                  ;X= 79  X= 78
        .BYTE   0*20+0                  ;X= 81  X= 80
        .BYTE   0*20+5                  ;X= 83  X= 82
        .BYTE   5*20+0                  ;X= 85  X= 84
        .BYTE   0*20+5                  ;X= 87  X= 86
        .BYTE   0*20+0                  ;X= 89  X= 88
        .BYTE   0*20+5                  ;X= 91  X= 90
        .BYTE   0*20+0                  ;X= 93  X= 92
        .BYTE   0*20+0                  ;X= 95  X= 94
        .BYTE   5*20+5                  ;X= 97  X= 96
        .BYTE   0*20+0                  ;X= 99  X= 98
        .BYTE   0*20+5                  ;X=101  X=100
        .BYTE   5*20+5                  ;X=103  X=102
        .BYTE   0*20+5                  ;X=105  X=104
        .BYTE   5*20+5                  ;X=107  X=106
        .BYTE   0*20+5                  ;X=109  X=108
        .BYTE   0*20+5                  ;X=111  X=110
        .BYTE   5*20+0                  ;X=113  X=112
        .BYTE   5*20+5                  ;X=115  X=114
        .BYTE   5*20+5                  ;X=117  X=116
        .BYTE   5*20+5                  ;X=119  X=118
        .BYTE   0*20+5                  ;X=121  X=120
        .BYTE   5*20+5                  ;X=123  X=122
        .BYTE   5*20+5                  ;X=125  X=124
        .BYTE   5*20+5                  ;X=127  X=126
        .BYTE   0*20+5                  ;X=129  X=128
        .BYTE   5*20+5                  ;X=131  X=130
        .BYTE   0*20+5                  ;X=133  X=132
        .BYTE   5*20+5                  ;X=135  X=134
        .BYTE   5*20+5                  ;X=137  X=136
        .BYTE   5*20+5                  ;X=139  X=138
        .BYTE   0*20+5                  ;X=141  X=140
        .BYTE   0*20+5                  ;X=143  X=142
        .BYTE   5*20+5                  ;X=145  X=144
        .BYTE   5*20+5                  ;X=147  X=146
        .BYTE   5*20+0                  ;X=149  X=148
        .BYTE   5*20+5                  ;X=151  X=150
        .BYTE   5*20+5                  ;X=153  X=152
        .BYTE   5*20+5                  ;X=155  X=154
        .BYTE   5*20+0                  ;X=157  X=156
        .BYTE   0*20+0                  ;X=159  X=158
        .BYTE   5*20+5                  ;X=161  X=160
        .BYTE   0*20+5                  ;X=163  X=162
        .BYTE   0*20+5                  ;X=165  X=164
        .BYTE   5*20+5                  ;X=167  X=166
        .BYTE   5*20+5                  ;X=169  X=168
        .BYTE   5*20+5                  ;X=171  X=170
        .BYTE   5*20+5                  ;X=173  X=172
        .BYTE   0*20+5                  ;X=175  X=174
        .BYTE   5*20+5                  ;X=177  X=176
        .BYTE   0*20+5                  ;X=179  X=178
        .BYTE   5*20+0                  ;X=181  X=180
        .BYTE   0*20+5                  ;X=183  X=182
        .BYTE   5*20+5                  ;X=185  X=184
        .BYTE   5*20+5                  ;X=187  X=186
        .BYTE   5*20+5                  ;X=189  X=188
        .BYTE   5*20+5                  ;X=191  X=190
        .BYTE   5*20+5                  ;X=193  X=192
        .BYTE   5*20+5                  ;X=195  X=194
        .BYTE   5*20+5                  ;X=197  X=196
        .BYTE   0*20+5                  ;X=199  X=198
        .BYTE   0*20+0                  ;X=201  X=200
        .BYTE   5*20+0                  ;X=203  X=202
        .BYTE   0*20+0                  ;X=205  X=204
```

```
        .BYTE   0*20+0                  ;X=207  X=206
        .BYTE   0*20+0                  ;X=209  X=208
        .BYTE   0*20+5                  ;X=211  X=210
        .BYTE   0*20+0                  ;X=213  X=212
        .BYTE   0*20+0                  ;X=215  X=214
        .BYTE   0*20+0                  ;X=217  X=216
        .BYTE   0*20+0                  ;X=219  X=218
        .BYTE   5*20+0                  ;X=221  X=220
        .BYTE   5*20+5                  ;X=223  X=222
        .BYTE   0*20+0                  ;X=225  X=224
        .BYTE   0*20+0                  ;X=227  X=226
        .BYTE   0*20+0                  ;X=229  X=228
        .BYTE   0*20+0                  ;X=231  X=230
        .BYTE   0*20+0                  ;X=233  X=232
        .BYTE   0*20+5                  ;X=235  X=234
        .BYTE   0*20+0                  ;X=237  X=236
        .BYTE   0*20+0                  ;X=239  X=238
        .BYTE   0*20+5                  ;X=241  X=240
        .BYTE   0*20+0                  ;X=243  X=242
        .BYTE   0*20+5                  ;X=245  X=244
        .BYTE   0*20+0                  ;X=247  X=246
        .BYTE   0*20+0                  ;X=249  X=248
        .BYTE   0*20+5                  ;X=251  X=250
        .BYTE   0*20+5                  ;X=253  X=252
        .BYTE   0*20+0                  ;X=255  X=254
        .BYTE   0*20+0                  ;X=257  X=256
        .BYTE   0*20+0                  ;X=259  X=258
        .BYTE   0*20+0                  ;X=261  X=260
        .BYTE   0*20+0                  ;X=263  X=262
        .BYTE   0*20+0                  ;X=265  X=264
        .BYTE   0*20+5                  ;X=267  X=266
        .BYTE   0*20+0                  ;X=269  X=268
        .BYTE   0*20+0                  ;X=271  X=270
        .BYTE   0*20+0                  ;X=273  X=272
        .BYTE   0*20+0                  ;X=275  X=274
        .BYTE   0*20+0                  ;X=277  X=276
        .BYTE   0*20+0                  ;X=279  X=278
        .BYTE   5*20+0                  ;X=281  X=280
        .BYTE   0*20+0                  ;X=283  X=282
        .BYTE   0*20+0                  ;X=285  X=284
        .BYTE   0*20+0                  ;X=287  X=286
        .BYTE   0*20+0                  ;X=289  X=288
        .BYTE   0*20+0                  ;X=291  X=290
        .BYTE   0*20+0                  ;X=293  X=292
        .BYTE   0*20+0                  ;X=295  X=294
        .BYTE   0*20+0                  ;X=297  X=296
        .BYTE   0*20+0                  ;X=299  X=298
        .BYTE   5*20+5                  ;X=301  X=300
        .BYTE   5*20+5                  ;X=303  X=302
        .BYTE   5*20+0                  ;X=305  X=304
        .BYTE   5*20+5                  ;X=307  X=306
        .BYTE   0*20+5                  ;X=309  X=308
        .BYTE   5*20+5                  ;X=311  X=310
        .BYTE   0*20+5                  ;X=313  X=312
        .BYTE   5*20+5                  ;X=315  X=314
        .BYTE   5*20+5                  ;X=317  X=316
        .BYTE   5*20+5                  ;X=319  X=318
        .BYTE   5*20+5                  ;X=321  X=320
        .BYTE   5*20+5                  ;X=323  X=322
        .BYTE   5*20+5                  ;X=325  X=324
        .BYTE   5*20+5                  ;X=327  X=326
        .BYTE   5*20+5                  ;X=329  X=328
        .BYTE   5*20+5                  ;X=331  X=330
        .BYTE   5*20+5                  ;X=333  X=332
        .BYTE   5*20+5                  ;X=335  X=334
```

```
        .BYTE   5*20+5                  ;X=337  X=336
        .BYTE   5*20+5                  ;X=339  X=338
        .BYTE   5*20+5                  ;X=341  X=340
        .BYTE   5*20+5                  ;X=343  X=342
        .BYTE   5*20+5                  ;X=345  X=344
        .BYTE   5*20+5                  ;X=347  X=346
        .BYTE   5*20+5                  ;X=349  X=348
        .BYTE   0*20+0                  ;X=351  X=350
        .BYTE   0*20+0                  ;X=353  X=352
        .BYTE   0*20+0                  ;X=355  X=354
        .BYTE   5*20+0                  ;X=357  X=356
        .BYTE   5*20+5                  ;X=359  X=358
        .BYTE   5*20+5                  ;X=361  X=360
        .BYTE   0*20+0                  ;X=363  X=362
        .BYTE   0*20+0                  ;X=365  X=364
        .BYTE   0*20+0                  ;X=367  X=366
        .BYTE   0*20+0                  ;X=369  X=368
        .BYTE   0*20+0                  ;X=371  X=370
        .BYTE   0*20+0                  ;X=373  X=372
        .BYTE   0*20+0                  ;X=375  X=374
        .BYTE   0*20+0                  ;X=377  X=376
        .BYTE   0*20+5                  ;X=379  X=378
        .BYTE   0*20+0                  ;X=381  X=380
        .BYTE   0*20+5                  ;X=383  X=382
        .BYTE   0*20+0                  ;X=385  X=384
        .BYTE   0*20+5                  ;X=387  X=386
        .BYTE   0*20+0                  ;X=389  X=388
        .BYTE   0*20+0                  ;X=391  X=390
        .BYTE   0*20+0                  ;X=393  X=392
        .BYTE   0*20+0                  ;X=395  X=394
        .BYTE   0*20+0                  ;X=397  X=396
        .BYTE   0*20+0                  ;X=399  X=398
        .BYTE   5*20+5                  ;X=401  X=400
        .BYTE   5*20+5                  ;X=403  X=402
        .BYTE   5*20+5                  ;X=405  X=404
        .BYTE   5*20+5                  ;X=407  X=406
        .BYTE   5*20+5                  ;X=409  X=408
        .BYTE   5*20+5                  ;X=411  X=410
        .BYTE   5*20+5                  ;X=413  X=412
        .BYTE   5*20+5                  ;X=415  X=414
        .BYTE   5*20+0                  ;X=417  X=416
        .BYTE   5*20+5                  ;X=419  X=418
        .BYTE   5*20+5                  ;X=421  X=420
        .BYTE   0*20+5                  ;X=423  X=422
        .BYTE   0*20+5                  ;X=425  X=424
        .BYTE   0*20+0                  ;X=427  X=426
        .BYTE   0*20+0                  ;X=429  X=428
        .BYTE   0*20+5                  ;X=431  X=430
        .BYTE   0*20+0                  ;X=433  X=432
        .BYTE   5*20+0                  ;X=435  X=434
        .BYTE   5*20+0                  ;X=437  X=436
        .BYTE   0*20+0                  ;X=439  X=438
        .BYTE   0*20+0                  ;X=441  X=440
        .BYTE   0*20+0                  ;X=443  X=442
        .BYTE   0*20+0                  ;X=445  X=444
        .BYTE   0*20+0                  ;X=447  X=446
        .BYTE   0*20+0                  ;X=449  X=448
        .BYTE   5*20+0                  ;X=451  X=450
        .BYTE   5*20+0                  ;X=453  X=452
        .BYTE   0*20+5                  ;X=455  X=454
        .BYTE   0*20+0                  ;X=457  X=456
        .BYTE   5*20+0                  ;X=459  X=458
        .BYTE   0*20+5                  ;X=461  X=460
        .BYTE   0*20+5                  ;X=463  X=462
        .BYTE   5*20+5                  ;X=465  X=464
```

```
        .BYTE   0*20+5                  ;X=467  X=466
        .BYTE   0*20+0                  ;X=469  X=468
        .BYTE   0*20+5                  ;X=471  X=470
        .BYTE   0*20+5                  ;X=473  X=472
        .BYTE   0*20+5                  ;X=475  X=474
        .BYTE   0*20+0                  ;X=477  X=476
        .BYTE   0*20+5                  ;X=479  X=478
        .BYTE   0*20+0                  ;X=481  X=480
        .BYTE   0*20+0                  ;X=483  X=482
        .BYTE   0*20+0                  ;X=485  X=484
        .BYTE   0*20+0                  ;X=487  X=486
        .BYTE   0*20+0                  ;X=489  X=488
        .BYTE   0*20+0                  ;X=491  X=490
        .BYTE   5*20+0                  ;X=493  X=492
        .BYTE   0*20+0                  ;X=495  X=494
        .BYTE   0*20+0                  ;X=497  X=496
        .BYTE   0*20+0                  ;X=499  X=498
        .BYTE   5*20+5                  ;X=501  X=500
        .BYTE   5*20+5                  ;X=503  X=502
        .BYTE   5*20+5                  ;X=505  X=504
        .BYTE   5*20+5                  ;X=507  X=506
        .BYTE   5*20+5                  ;X=509  X=508
        .BYTE   5*20+5                  ;X=511  X=510
        .BYTE   5*20+5                  ;X=513  X=512
        .BYTE   5*20+0                  ;X=515  X=514
        .BYTE   5*20+0                  ;X=517  X=516
        .BYTE   5*20+5                  ;X=519  X=518
        .BYTE   5*20+5                  ;X=521  X=520
        .BYTE   5*20+5                  ;X=523  X=522
        .BYTE   5*20+5                  ;X=525  X=524
        .BYTE   5*20+0                  ;X=527  X=526
        .BYTE   5*20+5                  ;X=529  X=528
        .BYTE   5*20+5                  ;X=531  X=530
        .BYTE   5*20+5                  ;X=533  X=532
        .BYTE   0*20+5                  ;X=535  X=534
        .BYTE   5*20+5                  ;X=537  X=536
        .BYTE   5*20+0                  ;X=539  X=538
        .BYTE   5*20+0                  ;X=541  X=540
        .BYTE   5*20+5                  ;X=543  X=542
        .BYTE   5*20+5                  ;X=545  X=544
        .BYTE   5*20+5                  ;X=547  X=546
        .BYTE   5*20+5                  ;X=549  X=548
        .BYTE   5*20+5                  ;X=551  X=550
        .BYTE   5*20+5                  ;X=553  X=552
        .BYTE   5*20+0                  ;X=555  X=554
        .BYTE   5*20+5                  ;X=557  X=556
        .BYTE   5*20+5                  ;X=559  X=558
        .BYTE   5*20+5                  ;X=561  X=560
        .BYTE   5*20+5                  ;X=563  X=562
        .BYTE   5*20+5                  ;X=565  X=564
        .BYTE   5*20+5                  ;X=567  X=566
        .BYTE   5*20+5                  ;X=569  X=568
        .BYTE   5*20+5                  ;X=571  X=570
        .BYTE   5*20+0                  ;X=573  X=572
        .BYTE   5*20+5                  ;X=575  X=574
        .BYTE   0*20+5                  ;X=577  X=576
        .BYTE   0*20+5                  ;X=579  X=578
        .BYTE   0*20+5                  ;X=581  X=580
        .BYTE   5*20+0                  ;X=583  X=582
        .BYTE   5*20+0                  ;X=585  X=584
        .BYTE   5*20+5                  ;X=587  X=586
        .BYTE   5*20+5                  ;X=589  X=588
        .BYTE   5*20+5                  ;X=591  X=590
        .BYTE   5*20+5                  ;X=593  X=592
        .BYTE   5*20+5                  ;X=595  X=594
```

```
        .BYTE   5*20+5                  ;X=597  X=596
        .BYTE   5*20+5                  ;X=599  X=598
        .BYTE   0*20+0                  ;X=601  X=600
        .BYTE   0*20+0                  ;X=603  X=602
        .BYTE   0*20+0                  ;X=605  X=604
        .BYTE   0*20+0                  ;X=607  X=606
        .BYTE   0*20+0                  ;X=609  X=608
        .BYTE   0*20+0                  ;X=611  X=610
        .BYTE   0*20+0                  ;X=613  X=612
        .BYTE   0*20+0                  ;X=615  X=614
        .BYTE   5*20+0                  ;X=617  X=616
        .BYTE   5*20+0                  ;X=619  X=618
        .BYTE   0*20+0                  ;X=621  X=620
        .BYTE   0*20+0                  ;X=623  X=622
        .BYTE   0*20+0                  ;X=625  X=624
        .BYTE   0*20+0                  ;X=627  X=626
        .BYTE   0*20+0                  ;X=629  X=628
        .BYTE   0*20+0                  ;X=631  X=630
        .BYTE   5*20+0                  ;X=633  X=632
        .BYTE   0*20+0                  ;X=635  X=634
        .BYTE   0*20+0                  ;X=637  X=636
        .BYTE   0*20+0                  ;X=639  X=638
        .BYTE   0*20+0                  ;X=641  X=640
        .BYTE   0*20+0                  ;X=643  X=642
        .BYTE   0*20+5                  ;X=645  X=644
        .BYTE   0*20+0                  ;X=647  X=646
        .BYTE   0*20+0                  ;X=649  X=648
        .BYTE   0*20+5                  ;X=651  X=650
        .BYTE   0*20+0                  ;X=653  X=652
        .BYTE   0*20+0                  ;X=655  X=654
        .BYTE   0*20+0                  ;X=657  X=656
        .BYTE   0*20+0                  ;X=659  X=658
        .BYTE   0*20+0                  ;X=661  X=660
        .BYTE   0*20+0                  ;X=663  X=662
        .BYTE   0*20+0                  ;X=665  X=664
        .BYTE   0*20+0                  ;X=667  X=666
        .BYTE   5*20+0                  ;X=669  X=668
        .BYTE   5*20+5                  ;X=671  X=670
        .BYTE   0*20+0                  ;X=673  X=672
        .BYTE   0*20+0                  ;X=675  X=674
        .BYTE   0*20+0                  ;X=677  X=676
        .BYTE   0*20+0                  ;X=679  X=678
        .BYTE   0*20+0                  ;X=681  X=680
        .BYTE   0*20+0                  ;X=683  X=682
        .BYTE   0*20+0                  ;X=685  X=684
        .BYTE   0*20+0                  ;X=687  X=686
        .BYTE   5*20+5                  ;X=689  X=688
        .BYTE   0*20+5                  ;X=691  X=690
        .BYTE   0*20+0                  ;X=693  X=692
        .BYTE   0*20+0                  ;X=695  X=694
        .BYTE   0*20+0                  ;X=697  X=696
        .BYTE   0*20+0                  ;X=699  X=698
        .BYTE   0*20+0                  ;X=701  X=700
        .BYTE   7*20+6                  ;X=703  X=702
        .BYTE   0*20+6                  ;X=705  X=704
        .BYTE   5*20+0                  ;X=707  X=706
        .BYTE   5*20+5                  ;X=709  X=708
        .BYTE   5*20+0                  ;X=711  X=710
        .BYTE   0*20+0                  ;X=713  X=712
        .BYTE   5*20+0                  ;X=715  X=714
        .BYTE   5*20+0                  ;X=717  X=716
        .BYTE   0*20+0                  ;X=719  X=718
        .BYTE   5*20+5                  ;X=721  X=720
        .BYTE   5*20+5                  ;X=723  X=722
        .BYTE   0*20+0                  ;X=725  X=724
```

```
        .BYTE   0*20+5                  ;X=727  X=726
        .BYTE   0*20+0                  ;X=729  X=728
        .BYTE   5*20+0                  ;X=731  X=730
        .BYTE   0*20+5                  ;X=733  X=732
        .BYTE   5*20+5                  ;X=735  X=734
        .BYTE   5*20+0                  ;X=737  X=736
        .BYTE   0*20+0                  ;X=739  X=738
        .BYTE   5*20+0                  ;X=741  X=740
        .BYTE   5*20+0                  ;X=743  X=742
        .BYTE   0*20+5                  ;X=745  X=744
        .BYTE   0*20+0                  ;X=747  X=746
        .BYTE   0*20+0                  ;X=749  X=748
        .BYTE   5*20+0                  ;X=751  X=750
        .BYTE   0*20+5                  ;X=753  X=752
        .BYTE   0*20+0                  ;X=755  X=754
        .BYTE   5*20+0                  ;X=757  X=756
        .BYTE   5*20+0                  ;X=759  X=758
        .BYTE   0*20+0                  ;X=761  X=760
        .BYTE   0*20+0                  ;X=763  X=762
        .BYTE   5*20+0                  ;X=765  X=764
        .BYTE   0*20+0                  ;X=767  X=766
        .BYTE   5*20+5                  ;X=769  X=768
        .BYTE   5*20+0                  ;X=771  X=770
        .BYTE   0*20+0                  ;X=773  X=772
        .BYTE   5*20+0                  ;X=775  X=774
        .BYTE   0*20+0                  ;X=777  X=776
        .BYTE   5*20+0                  ;X=779  X=778
        .BYTE   0*20+0                  ;X=781  X=780
        .BYTE   0*20+0                  ;X=783  X=782
        .BYTE   0*20+5                  ;X=785  X=784
        .BYTE   0*20+0                  ;X=787  X=786
        .BYTE   0*20+0                  ;X=789  X=788
        .BYTE   0*20+0                  ;X=791  X=790
        .BYTE   0*20+0                  ;X=793  X=792
        .BYTE   0*20+0                  ;X=795  X=794
        .BYTE   0*20+0                  ;X=797  X=796
        .BYTE   0*20+0                  ;X=799  X=798
        .BYTE   0*20+0                  ;X=801  X=800
        .BYTE   0*20+0                  ;X=803  X=802
        .BYTE   0*20+0                  ;X=805  X=804
        .BYTE   0*20+5                  ;X=807  X=806
        .BYTE   0*20+0                  ;X=809  X=808
        .BYTE   0*20+0                  ;X=811  X=810
        .BYTE   0*20+0                  ;X=813  X=812
        .BYTE   0*20+0                  ;X=815  X=814
        .BYTE   0*20+0                  ;X=817  X=816
        .BYTE   0*20+0                  ;X=819  X=818
        .BYTE   0*20+0                  ;X=821  X=820
        .BYTE   0*20+0                  ;X=823  X=822
        .BYTE   0*20+0                  ;X=825  X=824
        .BYTE   5*20+0                  ;X=827  X=826
        .BYTE   0*20+5                  ;X=829  X=828
        .BYTE   0*20+0                  ;X=831  X=830
        .BYTE   0*20+0                  ;X=833  X=832
        .BYTE   0*20+0                  ;X=835  X=834
        .BYTE   5*20+5                  ;X=837  X=836
        .BYTE   0*20+0                  ;X=839  X=838
        .BYTE   5*20+0                  ;X=841  X=840
        .BYTE   0*20+5                  ;X=843  X=842
        .BYTE   0*20+0                  ;X=845  X=844
        .BYTE   0*20+0                  ;X=847  X=846
        .BYTE   5*20+0                  ;X=849  X=848
        .BYTE   5*20+5                  ;X=851  X=850
        .BYTE   5*20+5                  ;X=853  X=852
        .BYTE   5*20+0                  ;X=855  X=854
```

```
        .BYTE   5*20+5                  ;X=857  X=856
        .BYTE   5*20+5                  ;X=859  X=858
        .BYTE   5*20+5                  ;X=861  X=860
        .BYTE   5*20+5                  ;X=863  X=862
        .BYTE   5*20+5                  ;X=865  X=864
        .BYTE   5*20+5                  ;X=867  X=866
        .BYTE   5*20+0                  ;X=869  X=868
        .BYTE   5*20+0                  ;X=871  X=870
        .BYTE   5*20+5                  ;X=873  X=872
        .BYTE   0*20+5                  ;X=875  X=874
        .BYTE   5*20+5                  ;X=877  X=876
        .BYTE   5*20+5                  ;X=879  X=878
        .BYTE   5*20+5                  ;X=881  X=880
        .BYTE   5*20+5                  ;X=883  X=882
        .BYTE   0*20+0                  ;X=885  X=884
        .BYTE   5*20+5                  ;X=887  X=886
        .BYTE   5*20+5                  ;X=889  X=888
        .BYTE   5*20+5                  ;X=891  X=890
        .BYTE   5*20+5                  ;X=893  X=892
        .BYTE   5*20+5                  ;X=895  X=894
        .BYTE   5*20+5                  ;X=897  X=896
        .BYTE   5*20+5                  ;X=899  X=898
        .BYTE   5*20+5                  ;X=901  X=900
        .BYTE   5*20+5                  ;X=903  X=902
        .BYTE   5*20+5                  ;X=905  X=904
        .BYTE   5*20+5                  ;X=907  X=906
        .BYTE   5*20+5                  ;X=909  X=908
        .BYTE   0*20+5                  ;X=911  X=910
        .EVEN                           ;MAKE SURE TO BRING TO BRING BYTES BACK.
        .PAGE
;
;               SINE AND COSINE TABLES FROM 0 TO 359 DEGREES.
;
SINTAB: 000000                  ;    0 DEGREES.
        000436                  ;    1 DEGREES.
        001074                  ;    2 DEGREES.
        001531                  ;    3 DEGREES.
        002167                  ;    4 DEGREES.
        002624                  ;    5 DEGREES.
        003261                  ;    6 DEGREES.
        003715                  ;    7 DEGREES.
        004350                  ;    8 DEGREES.
        005003                  ;    9 DEGREES.
        005435                  ;   10 DEGREES.
        006066                  ;   11 DEGREES.
        006516                  ;   12 DEGREES.
        007146                  ;   13 DEGREES.
        007574                  ;   14 DEGREES.
        010220                  ;   15 DEGREES.
        010644                  ;   16 DEGREES.
        011266                  ;   17 DEGREES.
        011707                  ;   18 DEGREES.
        012326                  ;   19 DEGREES.
        012744                  ;   20 DEGREES.
        013360                  ;   21 DEGREES.
        013772                  ;   22 DEGREES.
        014402                  ;   23 DEGREES.
        015010                  ;   24 DEGREES.
        015414                  ;   25 DEGREES.
        016016                  ;   26 DEGREES.
        016416                  ;   27 DEGREES.
        017014                  ;   28 DEGREES.
        017407                  ;   29 DEGREES.
        020000                  ;   30 DEGREES.
        020366                  ;   31 DEGREES.
```

```
        020752                  ;   32 DEGREES.
        021333                  ;   33 DEGREES.
        021712                  ;   34 DEGREES.
        022265                  ;   35 DEGREES.
        022636                  ;   36 DEGREES.
        023204                  ;   37 DEGREES.
        023547                  ;   38 DEGREES.
        024107                  ;   39 DEGREES.
        024443                  ;   40 DEGREES.
        024775                  ;   41 DEGREES.
        025323                  ;   42 DEGREES.
        025646                  ;   43 DEGREES.
        026165                  ;   44 DEGREES.
        026501                  ;   45 DEGREES.
        027012                  ;   46 DEGREES.
        027316                  ;   47 DEGREES.
        027620                  ;   48 DEGREES.
        030115                  ;   49 DEGREES.
        030407                  ;   50 DEGREES.
        030675                  ;   51 DEGREES.
        031157                  ;   52 DEGREES.
        031435                  ;   53 DEGREES.
        031707                  ;   54 DEGREES.
        032155                  ;   55 DEGREES.
        032417                  ;   56 DEGREES.
        032655                  ;   57 DEGREES.
        033106                  ;   58 DEGREES.
        033334                  ;   59 DEGREES.
        033555                  ;   60 DEGREES.
        033772                  ;   61 DEGREES.
        034202                  ;   62 DEGREES.
        034406                  ;   63 DEGREES.
        034606                  ;   64 DEGREES.
        035001                  ;   65 DEGREES.
        035170                  ;   66 DEGREES.
        035352                  ;   67 DEGREES.
        035527                  ;   68 DEGREES.
        035700                  ;   69 DEGREES.
        036044                  ;   70 DEGREES.
        036203                  ;   71 DEGREES.
        036336                  ;   72 DEGREES.
        036464                  ;   73 DEGREES.
        036605                  ;   74 DEGREES.
        036722                  ;   75 DEGREES.
        037031                  ;   76 DEGREES.
        037134                  ;   77 DEGREES.
        037232                  ;   78 DEGREES.
        037323                  ;   79 DEGREES.
        037407                  ;   80 DEGREES.
        037466                  ;   81 DEGREES.
        037541                  ;   82 DEGREES.
        037606                  ;   83 DEGREES.
        037646                  ;   84 DEGREES.
        037702                  ;   85 DEGREES.
        037730                  ;   86 DEGREES.
        037752                  ;   87 DEGREES.
        037766                  ;   88 DEGREES.
        037776                  ;   89 DEGREES.
COSTAB: 040000                  ;   90 DEGREES.
        037776                  ;   91 DEGREES.
        037766                  ;   92 DEGREES.
        037752                  ;   93 DEGREES.
        037730                  ;   94 DEGREES.
        037702                  ;   95 DEGREES.
        037646                  ;   96 DEGREES.
```

```
037606                  ;   97 DEGREES.
037541                  ;   98 DEGREES.
037466                  ;   99 DEGREES.
037407                  ;  100 DEGREES.
037323                  ;  101 DEGREES.
037232                  ;  102 DEGREES.
037134                  ;  103 DEGREES.
037031                  ;  104 DEGREES.
036722                  ;  105 DEGREES.
036605                  ;  106 DEGREES.
036464                  ;  107 DEGREES.
036336                  ;  108 DEGREES.
036203                  ;  109 DEGREES.
036044                  ;  110 DEGREES.
035700                  ;  111 DEGREES.
035527                  ;  112 DEGREES.
035352                  ;  113 DEGREES.
035170                  ;  114 DEGREES.
035001                  ;  115 DEGREES.
034606                  ;  116 DEGREES.
034406                  ;  117 DEGREES.
034202                  ;  118 DEGREES.
033772                  ;  119 DEGREES.
033555                  ;  120 DEGREES.
033334                  ;  121 DEGREES.
033106                  ;  122 DEGREES.
032655                  ;  123 DEGREES.
032417                  ;  124 DEGREES.
032155                  ;  125 DEGREES.
031707                  ;  126 DEGREES.
031435                  ;  127 DEGREES.
031157                  ;  128 DEGREES.
030675                  ;  129 DEGREES.
030407                  ;  130 DEGREES.
030115                  ;  131 DEGREES.
027620                  ;  132 DEGREES.
027316                  ;  133 DEGREES.
027012                  ;  134 DEGREES.
026501                  ;  135 DEGREES.
026165                  ;  136 DEGREES.
025646                  ;  137 DEGREES.
025323                  ;  138 DEGREES.
024775                  ;  139 DEGREES.
024443                  ;  140 DEGREES.
024107                  ;  141 DEGREES.
023547                  ;  142 DEGREES.
023204                  ;  143 DEGREES.
022636                  ;  144 DEGREES.
022265                  ;  145 DEGREES.
021712                  ;  146 DEGREES.
021333                  ;  147 DEGREES.
020752                  ;  148 DEGREES.
020366                  ;  149 DEGREES.
020000                  ;  150 DEGREES.
017407                  ;  151 DEGREES.
017014                  ;  152 DEGREES.
016416                  ;  153 DEGREES.
016016                  ;  154 DEGREES.
015414                  ;  155 DEGREES.
015010                  ;  156 DEGREES.
014402                  ;  157 DEGREES.
013772                  ;  158 DEGREES.
013360                  ;  159 DEGREES.
012744                  ;  160 DEGREES.
012326                  ;  161 DEGREES.
```

```
011707                  ;  162 DEGREES.
011266                  ;  163 DEGREES.
010644                  ;  164 DEGREES.
010220                  ;  165 DEGREES.
007574                  ;  166 DEGREES.
007146                  ;  167 DEGREES.
006516                  ;  168 DEGREES.
006066                  ;  169 DEGREES.
005435                  ;  170 DEGREES.
005003                  ;  171 DEGREES.
004350                  ;  172 DEGREES.
003715                  ;  173 DEGREES.
003261                  ;  174 DEGREES.
002624                  ;  175 DEGREES.
002167                  ;  176 DEGREES.
001531                  ;  177 DEGREES.
001074                  ;  178 DEGREES.
000436                  ;  179 DEGREES.
000000                  ;  180 DEGREES.
177344                  ;  181 DEGREES.
176706                  ;  182 DEGREES.
176247                  ;  183 DEGREES.
175613                  ;  184 DEGREES.
175156                  ;  185 DEGREES.
174521                  ;  186 DEGREES.
174065                  ;  187 DEGREES.
173430                  ;  188 DEGREES.
172775                  ;  189 DEGREES.
172343                  ;  190 DEGREES.
171712                  ;  191 DEGREES.
171262                  ;  192 DEGREES.
170634                  ;  193 DEGREES.
170206                  ;  194 DEGREES.
167560                  ;  195 DEGREES.
167134                  ;  196 DEGREES.
166512                  ;  197 DEGREES.
166073                  ;  198 DEGREES.
165452                  ;  199 DEGREES.
165036                  ;  200 DEGREES.
164421                  ;  201 DEGREES.
164010                  ;  202 DEGREES.
163400                  ;  203 DEGREES.
162772                  ;  204 DEGREES.
162364                  ;  205 DEGREES.
161762                  ;  206 DEGREES.
161362                  ;  207 DEGREES.
160766                  ;  208 DEGREES.
160371                  ;  209 DEGREES.
160002                  ;  210 DEGREES.
157412                  ;  211 DEGREES.
157026                  ;  212 DEGREES.
156445                  ;  213 DEGREES.
156070                  ;  214 DEGREES.
155513                  ;  215 DEGREES.
155142                  ;  216 DEGREES.
154574                  ;  217 DEGREES.
154233                  ;  218 DEGREES.
153673                  ;  219 DEGREES.
153335                  ;  220 DEGREES.
153005                  ;  221 DEGREES.
152455                  ;  222 DEGREES.
152134                  ;  223 DEGREES.
151613                  ;  224 DEGREES.
151277                  ;  225 DEGREES.
150770                  ;  226 DEGREES.
```

```
150462                  ;  227 DEGREES.
150162                  ;  228 DEGREES.
147663                  ;  229 DEGREES.
147373                  ;  230 DEGREES.
147105                  ;  231 DEGREES.
146623                  ;  232 DEGREES.
146345                  ;  233 DEGREES.
146073                  ;  234 DEGREES.
145625                  ;  235 DEGREES.
145363                  ;  236 DEGREES.
145125                  ;  237 DEGREES.
144672                  ;  238 DEGREES.
144446                  ;  239 DEGREES.
144225                  ;  240 DEGREES.
144010                  ;  241 DEGREES.
143576                  ;  242 DEGREES.
143372                  ;  243 DEGREES.
143174                  ;  244 DEGREES.
143001                  ;  245 DEGREES.
142612                  ;  246 DEGREES.
142430                  ;  247 DEGREES.
142253                  ;  248 DEGREES.
142102                  ;  249 DEGREES.
141736                  ;  250 DEGREES.
141575                  ;  251 DEGREES.
141442                  ;  252 DEGREES.
141314                  ;  253 DEGREES.
141173                  ;  254 DEGREES.
141060                  ;  255 DEGREES.
140747                  ;  256 DEGREES.
140644                  ;  257 DEGREES.
140550                  ;  258 DEGREES.
140457                  ;  259 DEGREES.
140371                  ;  260 DEGREES.
140312                  ;  261 DEGREES.
140241                  ;  262 DEGREES.
140174                  ;  263 DEGREES.
140132                  ;  264 DEGREES.
140100                  ;  265 DEGREES.
140050                  ;  266 DEGREES.
140030                  ;  267 DEGREES.
140012                  ;  268 DEGREES.
140004                  ;  269 DEGREES.
140000                  ;  270 DEGREES.
140004                  ;  271 DEGREES.
140012                  ;  272 DEGREES.
140030                  ;  273 DEGREES.
140050                  ;  274 DEGREES.
140100                  ;  275 DEGREES.
140132                  ;  276 DEGREES.
140174                  ;  277 DEGREES.
140241                  ;  278 DEGREES.
140312                  ;  279 DEGREES.
140371                  ;  280 DEGREES.
140457                  ;  281 DEGREES.
140550                  ;  282 DEGREES.
140644                  ;  283 DEGREES.
140747                  ;  284 DEGREES.
141060                  ;  285 DEGREES.
141173                  ;  286 DEGREES.
141314                  ;  287 DEGREES.
141442                  ;  288 DEGREES.
141575                  ;  289 DEGREES.
141736                  ;  290 DEGREES.
142102                  ;  291 DEGREES.
```

```
142253                  ;  292 DEGREES.
142430                  ;  293 DEGREES.
142612                  ;  294 DEGREES.
143001                  ;  295 DEGREES.
143174                  ;  296 DEGREES.
143372                  ;  297 DEGREES.
143576                  ;  298 DEGREES.
144010                  ;  299 DEGREES.
144225                  ;  300 DEGREES.
144446                  ;  301 DEGREES.
144672                  ;  302 DEGREES.
145125                  ;  303 DEGREES.
145363                  ;  304 DEGREES.
145625                  ;  305 DEGREES.
146073                  ;  306 DEGREES.
146345                  ;  307 DEGREES.
146623                  ;  308 DEGREES.
147105                  ;  309 DEGREES.
147373                  ;  310 DEGREES.
147663                  ;  311 DEGREES.
150162                  ;  312 DEGREES.
150463                  ;  313 DEGREES.
150770                  ;  314 DEGREES.
151277                  ;  315 DEGREES.
151613                  ;  316 DEGREES.
152134                  ;  317 DEGREES.
152455                  ;  318 DEGREES.
153005                  ;  319 DEGREES.
153335                  ;  320 DEGREES.
153673                  ;  321 DEGREES.
154233                  ;  322 DEGREES.
154574                  ;  323 DEGREES.
155142                  ;  324 DEGREES.
155513                  ;  325 DEGREES.
156070                  ;  326 DEGREES.
156445                  ;  327 DEGREES.
157026                  ;  328 DEGREES.
157412                  ;  329 DEGREES.
160000                  ;  330 DEGREES.
160371                  ;  331 DEGREES.
160766                  ;  332 DEGREES.
161362                  ;  333 DEGREES.
161762                  ;  334 DEGREES.
162364                  ;  335 DEGREES.
162772                  ;  336 DEGREES.
163400                  ;  337 DEGREES.
164010                  ;  338 DEGREES.
164422                  ;  339 DEGREES.
165036                  ;  340 DEGREES.
165452                  ;  341 DEGREES.
166073                  ;  342 DEGREES.
166512                  ;  343 DEGREES.
167134                  ;  344 DEGREES.
167560                  ;  345 DEGREES.
170206                  ;  346 DEGREES.
170634                  ;  347 DEGREES.
171262                  ;  348 DEGREES.
171712                  ;  349 DEGREES.
172343                  ;  350 DEGREES.
172775                  ;  351 DEGREES.
173430                  ;  352 DEGREES.
174065                  ;  353 DEGREES.
174521                  ;  354 DEGREES.
175156                  ;  355 DEGREES.
175613                  ;  356 DEGREES.
```

```
        176247                  ;  357 DEGREES.
        176706                  ;  358 DEGREES.
        177344                  ;  359 DEGREES.
        .PAGE
;
;               THIS IS THE SHIP BUFFER FOR STORING THE SPACE SHIP.
;               THERE ARE TWO OF THEM TO AVOID PROBLEMS.
;
SHIPLC: .WORD   SHIPB1                  ;BUFFER POINTERS.
        .WORD   SHIPB2
DRWSHP: .WORD   114120                  ;BASIC SHIP START.
        .WORD   170200
SHOWX:  .WORD   0
SHOWY:  .WORD   0
        .WORD   DISTOP                  ;THE SUBROUTINE CALL NOW.
SHIPDP: .WORD   0
        .WORD   DISTOP
        .WORD   0
;
;               ROCKET FLAME CONTROL, AND BUFFER.
;
ONFIRE: .WORD   DISTOP
FSUBC:  .WORD   0
        .WORD   DISTOP
        .WORD   0
;
;               MESSAGE DISPLAY CONTROL.
;
INFO:   .WORD   DISTOP
SYSMES: .WORD   0
        .WORD   DISTOP
        .WORD   0
;
;               THIS SECTION CONTROLS THE DUST WHEN THE
;               ROCKET FLAME HITS THE SURFACE OF THE MOON.
;
DRWDST: .WORD   DISTOP
DUSTON: .WORD   0
        .WORD   DISTOP
        .WORD   0
;
;               THIS SECTION DRAWS AND STORES THE MOON, WHEN NECESSARY.
;
DRWLUN: .WORD   DISTOP
MOONGO: .WORD   0                       ;OVERLAYED BY POINTER
        .WORD   DISTOP
        .WORD   0
SHIPB1:
        .=.+84.
SHIPB2:
        .=.+84.
FLAMIN:
        .=.+66.
;
;               THIS SECTION IS WHERE THE EXPLOSION GOES (ALSO THE
;               DUST FROM THE ROCKET ENGINE).
;
EXLIST:
        .=.+2602.                       ;IT'S BIG (241*2*2+1) BUT NOT USED
                                        ;JUST TO GIVE US A SIZE FEELING.
        .=.+100.                        ;ALLOCATE 100 WORDS FOR THE STACK
STACK:  .WORD                           ;DO NOT PUT ANYTHING IN IT
                                        ;JUST DEFINE IT.
        .PAGE
;
```

```
;               THIS IS THE MESSAGE THAT IS DISPLAYED WHEN
;               STARTING UP THE FIRST TIME
;
STARTM: .WORD   116720
        .WORD   0.
        .WORD   650.
        .WORD   170240
        .WORD   100000
        .ASCII  '            R T - 1 1  L U N A R  L A N D E R'
        .BYTE   15,12
        .ASCII  '              ----------------------------'
        .BYTE   15,12,12,12
        .ASCII  '    ALL INTERNAL NAVIGATIONAL SYSTEMS HAVE DIED ON YOU'
        .BYTE   15,12
        .ASCII  /       IT LOOKS AS THOUGH YOU'RE GOING TO HAVE TO/
        .BYTE   15,12
        .ASCII  '             LAND THIS THING ON YOUR OWN'
        .BYTE   15,12,12
        .ASCII  '       ***** GOOD BYE, GOOD LUCK, AND SORRY  *****'
        .BYTE   0
        .=.-1
        .EVEN
        .WORD   DISJMP
        .WORD   STARTM
        .END    START                   ;RESTART ADDRESS.
```