```
       ORG 0000
       MOVE total,B
back   SUB #$1,B  *set total to zero
       CMP #$0,B
       BNE back
       MOVE B,total



wait2  MOVE $E1,A *CHECK IF KEYBOARD READY
       CMP #$0,A  *check to see if its ready
       BEQ wait2 *go back if not ready
       MOVE $E0,A *READ FROM KEYBOARD
       CMP #^49,A *check for a 1
       BNE check1
       JSR sub
       JMP wait3

check1 CMP #^48,A *checks for a 0 *not equal to 0 go back
       BNE wait2

wait3  MOVE $E3,B *check if console output is ready
       CMP #$0,B
       BEQ wait3
       MOVE A,$E2 *WRITE TO CONSOLE
wait4  MOVE $E3,B
       CMP #$0,B
       BEQ  wait4
       MOVE #$2E,A
       MOVE A,$E2





wait5  MOVE $E1,A *CHECK IF KEYBOARD READY
       CMP #$0,A  *check to see if its ready
       BEQ wait5 *go back if not ready
       MOVE $E0,A *READ FROM KEYBOARD
       CMP #^49,A *check for a 1
       BNE check2
       JSR sub1
       JMP wait6 *FORWARDS

check2 CMP #^48,A *checks for a 0 *not equal to 0 go back
       BNE wait5 *GO BACK

wait6  MOVE $E3,B *check if console output is ready
       CMP #$0,B
       BEQ wait6 *GO BACK
       MOVE A,$E2 *WRITE TO CONSOLE
wait7  MOVE $E3,B
       CMP #$0,B
       BEQ  wait7 *GO BACK
       MOVE #$2E,A
       MOVE A,$E2



wait8  MOVE $E1,A *CHECK IF KEYBOARD READY
       CMP #$0,A  *check to see if its ready
       BEQ wait8 *go back if not ready

       MOVE $E0,A *READ FROM KEYBOARD
       CMP #^49,A *check for a 1
       BNE check3
       JSR sub2
       JMP wait9 *FORWARDS

check3 CMP #^48,A *checks for a 0 *not equal to 0 go back
       BNE wait8 *GO BACK

wait9  MOVE $E3,B *check if console output is ready
       CMP #$0,B
       BEQ wait9 *GO BACK
       MOVE A,$E2 *WRITE TO CONSOLE
wait10 MOVE $E3,B
       CMP #$0,B
       BEQ  wait10 *GO BACK
       MOVE #$2E,A
       MOVE A,$E2


wait11 MOVE $E1,A *CHECK IF KEYBOARD READY
       CMP #$0,A  *check to see if its ready
       BEQ wait11 *go back if not ready
       MOVE $E0,A *READ FROM KEYBOARD
       CMP #^49,A *check for a 1
       BNE check4
       JSR sub3 *jumps to subprogram
       JMP wait12 *FORWARDS

check4 CMP #^48,A *checks for a 0 *not equal to 0 go back
       BNE wait11 *GO BACK

wait12 MOVE $E3,B *check if console output is ready
       CMP #$0,B
       BEQ wait12 *GO BACK
       MOVE A,$E2 *WRITE TO CONSOLE
wait13 MOVE $E3,B
       CMP #$0,B
       BEQ  wait13 *GO BACK
       MOVE #$2E,A *.
       MOVE #$3D,A *=
       MOVE A,$E2 *DISPLAY .=
       MOVE A,inputs

       MOVE total,A
       CMP #$9,A *IF NUM HIGHER THAN 9 FIX IT
       BMI ten

back1  MOVE $E3,B  *display code final
       CMP #$0,B
       BEQ  back1
       MOVE A,$E2
       HALT


ten JSR adding1  *SUB FOR MAKING 1 FIRST DIGIT
    MOVE total,A
    SUB #$0a,A *SINGLE DIGIT LEFT
back2 MOVE $E3,B
    CMP #$0,B
    BEQ  back2
    MOVE A,$E2 *display code final*
    HALT



adding1 MOVE A,total
back3  MOVE $E3,B
       CMP #$0,B
       BEQ  back3
       MOVE A,$E2  *display 1 for first digit*
       RTS







sub CMP #^49,A
    BEQ do
    BNE Fix *double check that 1 has been pressed to prevnt code from adding 8 without 1 being pressed
do  MOVE A,inputs *clears A of stuff
    ADD #$7,A * ADD 8 TO A
    MOVE A,total
    MOVE inputs,A
Fix RTS

sub1 CMP #^49,A
    BEQ do1
    BNE Fix1 *double check that 1 has been pressed to prevnt code from adding 8 without 1 being pressed
do1  MOVE A,inputs *clears A of stuff
    ADD #$3,A * ADD 4 TO A
       MOVE A,total
    MOVE inputs,A
Fix1 RTS

sub2 CMP #^49,A
    BEQ do2
    BNE Fix2 *double check that 1 has been pressed to prevnt code from adding 8 without 1 being pressed
do2  MOVE A,inputs *clears A of stuff
    ADD #$1,A
    MOVE A,total
    MOVE inputs,A
Fix2 RTS

sub3 CMP #^49,A
     BEQ do3
     BNE Fix3 *double check that 1 has been pressed to prevnt code from adding 8 without 1 being pressed
do3  MOVE A,inputs *clears A of stuff
     MOVE A,total
     MOVE inputs,A
Fix3 RTS








inputs DC.W $1
total DC.W $0 *total in memory location 0
```