.ORIG x3000

; Number of test scores

NUM\_SCORES .FILL x0005 ; Number of test scores (5)

; Initialize each score to 0 (5 scores)

SCORE\_1 .FILL x0000 ; Score 1

SCORE\_2 .FILL x0000 ; Score 2

SCORE\_3 .FILL x0000 ; Score 3

SCORE\_4 .FILL x0000 ; Score 4

SCORE\_5 .FILL x0000 ; Score 5

; Initialize variables for maximum, average, and sum of scores

MAX\_SCORE .FILL x0000 ; Maximum score

AVG\_SCORE .FILL x0000 ; Average score

SUM\_SCORES .FILL x0000 ; Sum of the scores

; Prompt for Input and Output Messages

PROMPT .STRINGZ "Enter test score (0-100) ODD ONlY: "

PROMPT1 .STRINGZ "Enter ten's digit: \n"

PROMPT2 .STRINGZ "Enter one's digit: \n"

ASCII\_C .FILL #-48

OUTPUT\_MIN .STRINGZ "Minimum Score: "

OUTPUT\_MAX .STRINGZ "Maximum Score: "

OUTPUT\_AVG .STRINGZ "Average Score: "

LETTER\_GRADE .STRINGZ "Letter Grade: "

NEWLINE .FILL x0A

SPACE .FILL x20

; Main Program

LEA R0, PROMPT ; Load address of prompt

PUTS ; Print prompt

JSR GETSCORES

; Input Loop

LD R2, NUM\_SCORES ; Load number of scores

LEA R1, SCORE\_1 ; Load address of first score

INPUT\_LOOP

JSR READ\_NUMBER ; Read a number from user

STR R0, R1, #0 ; Store number in current score location

ADD R1, R1, #1 ; Move to next score location

ADD R2, R2, #-1 ; Decrease counter

BRp INPUT\_LOOP ; Repeat for alll scores

GETSCORES

GETSCORES

;'TENS DIGIT'

JSR CLEARREG ; CLEARS ALL REGISTERS

LD R6, ASCII\_C ; LOAD ASCII\_C (#-48) INTO R6

LEA R0, PROMPT1 ;

PUTS

AND R0, R0, #0 ; CLEAR R0

IN ; GET 'TENS' DIGIT ; ECHO

ADD R1, R0, R6 ; ADD INPUT + R6 TO GET DECIMAL VALUE

OUT

;'ONES' DIGIT

AND R0, R0, #0 ; CLEAR R0

LEA R0, PROMPT2

PUTS

AND R0, R0, #0

IN ; GET 'ONES' DIGIT

OUT ; ECHO

ADD R2, R0, #0 ; MOVE 'ONES' DIGIT TO R2

ADD R2, R0, R6

ADD R3, R2, R1 ; ADD 'TENS' VALUE AND 'ONES VALUE

OUT ; ECHO

HALT

CLEARREG

AND R1, R1, #0

AND R2, R2, #0

AND R3, R3, #0

AND R4, R4, #0

AND R5, R5, #0

AND R6, R6, #0

RET

;Maximum Score

LEA R1, SCORE\_1 ; Load address of first score

LDR R0, R1, 0 ; first score

LD R2, NUM\_SCORES

ADD R2, R2, #-1

FIND\_MAX\_LOOP

ADD R1, R1, #1 ; move to next score

LDR R3, R1, 0 ; next score

LD R0, MAX\_SCORE ; Load current maximum

ADD R4, R0, R3

BRp SKIP\_MAX ; If R0 >= R3, skip

ST R3, MAX\_SCORE ; Update max

SKIP\_MAX

ADD R2, R2, #-1 ; Decrease counter

BRp FIND\_MAX\_LOOP ; Repeat until all scores are checked

; Compute Sum

LEA R1, SCORE\_1 ; Address of first score

LD R2, NUM\_SCORES ; Load number of scores

AND R6, R6, #0 ; Initialize sum to 0

SUM\_LOOP

LDR R3, R1, 0

ADD R6, R6, R3 ; Add to sum

ADD R1, R1, #1

ADD R2, R2, #-1 ; Decrease counter

BRp SUM\_LOOP

ST R6, SUM\_SCORES ; Store the sum

;Average

LD R0, NUM\_SCORES

LD R1, SUM\_SCORES

JSR DIVIDE ; Divide sum by number of scores (assuming a custom division subroutine)

ST R0, AVG\_SCORE ; Store average

; Display Results

LEA R0, OUTPUT\_MIN

PUTS

LEA R0, OUTPUT\_MAX

PUTS

LD R0, MAX\_SCORE

JSR PRINT\_NUMBER

LD R0, NEWLINE

OUT

LEA R0, OUTPUT\_AVG

PUTS

LD R0, AVG\_SCORE

JSR PRINT\_NUMBER

LD R0, NEWLINE

OUT

; Display for Letter Grade

LEA R0, LETTER\_GRADE

PUTS

; Adjust Average

ADD R4, AVG\_SCORE, #60 ; Adjusted average

; Check ranges

CMP R4, #LOW\_PASS\_SCORE ; Compare with passing score (60)

BLT FAIL\_GRADE ; Branch if less than passing score

CMP R4, #HIGH\_PASS\_SCORE ; Compare with high score (90)

BLT ABOVE\_AVG\_GRADE ; Branch if less than high score

LEA R0, ASCII\_A ; Load ASCII code for 'A'

BR DONE\_GRADE ; Branch to display 'A' grade

;Above avg grade

LEA R0, ASCII\_B ; Load ASCII code for 'B'

BR DONE\_GRADE ; Branch to display 'B' grade

;Fail grade

LEA R0, ASCII\_F ; Load ASCII code for 'F'

BR DONE\_GRADE ; Branch to display 'F' grade

DONE\_GRADE

OUT

LD R0, NEWLINE

OUT

HALT

; Subroutine to Read a Number

READ\_NUMBER

; Read a character

GETC

; Convert ASCII to integer

OUT

AND R0, R0, #15

RET

; Subroutine to Print a Number

PRINT\_NUMBER

; Convert integer to ASCII

OUT

RET

LOW\_PASS\_SCORE .FILL #60 ; Minimum passing score D

HIGH\_PASS\_SCORE .FILL #90 ; Minimum score for an A' grade

; ASCII Values for Letter Grades

ASCII\_A .FILL x0041 ; ASCII code for 'A'

ASCII\_B .FILL x0042 ; ASCII code for 'B'

ASCII\_C .FILL x0043 ; ASCII code for 'C'

ASCII\_D .FILL x0044 ; ASCII code for 'D'

ASCII\_E .FILL x0045 ; ASCII code for 'E'

ASCII\_F .FILL x0046 ; ASCII code for 'F'

.END