```
.ORIG x3000


MSG_INSTRUCTIONS .STRINGZ "ENTER THE GRADES (0-100):"
MSG_MIN .STRINGZ "\nMin Grade: "
MSG_MAX .STRINGZ "\nMax Grade: "
MSG_AVG .STRINGZ "\nAverage: "
NEWLINE .STRINGZ "\n"
MSG_A .STRINGZ "A\n"
MSG_B .STRINGZ "B\n"
MSG_C .STRINGZ "C\n"
MSG_D .STRINGZ "D\n"
MSG_F .STRINGZ "F\n"
MSG_INVALID .STRINGZ "Invalid grade\n"

LETT_G .FILL #0041 ; 'A'
       .FILL #0042 ; 'B'
       .FILL #0043 ; 'C'
       .FILL #0044 ; 'D'
       .FILL #0046 ; 'F'

ASCII_1 .FILL x0031
ASCII_OFFSET  .FILL x0030
SPACE         .FILL x0020

SAVE_R0 .BLKW #1 ;
SAVE_R1 .BLKW #1
SAVE_R2 .BLKW #1
SAVE_R7 .BLKW #1

SCORES .BLKW #5 ;

MIN_G .FILL #0
MAX_G .FILL #0
AVG_G .FILL #0

LEA R0, MSG_INSTRUCTIONS  ;
PUTS                      ;


   JSR R_SCO
   JSR C_MIN
   JSR C_MAX
   JSR C_AVG;    JSR CVT_LETT
```

```
    JSR D_RES ;

    HALT

R_SCO
    ST R1, SAVE_R1
    ST R7, SAVE_R7

    LEA R0, SCORES
    AND R1, R1, #0
    ADD R1, R1, #5

    R_LOOP
    GETC
    OUT

    ADD R2, R0, R1
    ADD R2, R2, #-1

    STR R0, R2, #0

    ADD R1, R1, #-1
    BRp R_LOOP

    LD R1, SAVE_R1
    LD R7, SAVE_R7
    RET



C_MIN
    ST R0, SAVE_R0
    ST R1, SAVE_R1
    ST R2, SAVE_R2

    LEA R1, SCORES
    LDR R0, R1, #0
    ADD R2, R1, #5

    MIN_LOOP
    ADD R1, R1, #1
    LDR R3, R1, #0

    NOT R4, R0
```

```
    ADD R4, R4, #1
    ADD R4, R3, R4

    BRzp END_MIN_LOOP
    ADD R0, R3, #0

    END_MIN_LOOP
    ADD R2, R2, #-1
    BRp MIN_LOOP

    LD R0, SAVE_R0
    LD R1, SAVE_R1
    LD R2, SAVE_R2
    RET


C_MAX
    ST R0, SAVE_R0
    ST R1, SAVE_R1
    ST R2, SAVE_R2

    LEA R1, SCORES
    LDR R0, R1, #0
    ADD R2, R1, #5

    MAX_LOOP
    ADD R1, R1, #1
    LDR R3, R1, #0

    NOT R4, R3
    ADD R4, R4, #1
    ADD R4, R0, R4

    BRzp END_MAX_LOOP
    ADD R0, R3, #0

    END_MAX_LOOP
    ADD R2, R2, #-1
    BRp MAX_LOOP

    LD R0, SAVE_R0
    LD R1, SAVE_R1
    LD R2, SAVE_R2
    RET
```

```
C_AVG
    ST R0, SAVE_R0
    ST R1, SAVE_R1
    ST R2, SAVE_R2

    AND R0, R0, #0
    LEA R1, SCORES
    ADD R2, R1, #5

    AVG_LOOP
    LDR R3, R1, #0
    ADD R0, R0, R3
    ADD R1, R1, #1
    ADD R2, R2, #-1
    BRp AVG_LOOP

    ADD R0, R0, R0
    ADD R1, R0, #0
    ADD R0, R0, #-10
    BRn DIV_LOOP
    ADD R0, R0, #0

    DIV_LOOP
    ADD R1, R1, #10
    ADD R0, R0, #1
    BRn DIV_LOOP

    LD R0, SAVE_R0
    LD R1, SAVE_R1
    LD R2, SAVE_R2
    RET


CVT_LETT
    ST R0, SAVE_R0
    ST R1, SAVE_R1
    ST R7, SAVE_R7

    LEA R1, SCORES
    ADD R2, R1, #5
    LEA R6, LETT_G
```

```
CVT_LOOP
   LDR R0, R1, #0
   ADD R0, R0, #-90
   BRzp GRADE_A
   ADD R0, R0, #10
   BRzp GRADE_B
   ADD R0, R0, #10
   BRzp GRADE_C
   ADD R0, R0, #10
   BRzp GRADE_D
   BRnzp GRADE_F

GRADE_A
   LDR R0, R6, #0
   BRnzp STORE_GRADE

GRADE_B
   LDR R0, R6, #1
   BRnzp STORE_GRADE

GRADE_C
   LDR R0, R6, #2
   BRnzp STORE_GRADE

GRADE_D
   LDR R0, R6, #3
   BRnzp STORE_GRADE

GRADE_F
   LDR R0, R6, #4
   BRnzp STORE_GRADE

ST_G
   STR R0, R1, #0
   ADD R1, R1, #1
   ADD R2, R2, #-1
   BRp CVT_LOOP

   LD R0, SAVE_R0
   LD R1, SAVE_R1
   LD R7, SAVE_R7
   RET
```

```
D_RES

    ST R0, SAVE_R0
    ST R1, SAVE_R1
    ST R2, SAVE_R2
    ST R7, SAVE_R7


    LEA R0, MSG_MIN
    PUTS


    LD R0, MIN_G
    LD R1, SAVE_R1
    JSR PG_ASCII
      LEA R1, LETT_G
    ADD R1, R1, R0
    LDR R0, R1, #0
    OUT



    LEA R0, MSG_MAX
    PUTS

    LD R0, MAX_G
    JSR CG_ASCII
    LEA R1, LETT_G
    ADD R1, R1, R0
    LDR R0, R1, #0
    OUT


    LEA R0, NEWLINE
    PUTS


    LEA R0, MSG_AVG
    PUTS


    LD R0, AVG_G
    JSR PG_ASCII
```

```
    LEA R1, LETT_G
    ADD R1, R1, R0
    LDR R0, R1, #0
    OUT


    LEA R0, NEWLINE
    PUTS

    LD R0, SAVE_R0
    LD R1, SAVE_R1
    LD R2, SAVE_R2
    LD R7, SAVE_R7
    RET ;

PG_ASCII
    ADD R1, R0, #0
    ADD R0, R0, #-10


    LD R0, ASCII_1 ;
    OUT

    ADD R0, R1, #-16
    ADD R0, R0, #-16
    ADD R0, R0, #-16

    OUT
    RET


.END
—----------------------------------------------------------------
VERSION 2
—----------------------------------------------------------------
.ORIG x3000

MSG_INSTRUCTIONS .STRINGZ "ENTER THE GRADES (0-100):"
MSG_MIN .STRINGZ "\nMin Grade: "
MSG_MAX .STRINGZ "\nMax Grade: "
MSG_AVG .STRINGZ "\nAverage: "
NEWLINE .STRINGZ "\n"
MSG_A .STRINGZ "A\n"
MSG_B .STRINGZ "B\n"
```

```
MSG_C .STRINGZ "C\n"
MSG_D .STRINGZ "D\n"
MSG_F .STRINGZ "F\n"
MSG_INVALID .STRINGZ "Invalid grade\n"

LETT_G .FILL #0041
       .FILL #0042
       .FILL #0043
       .FILL #0044
       .FILL #0046

ASCII_1 .FILL x0031
ASCII_OFFSET  .FILL x0030
SPACE         .FILL x0020

SAVE_R0 .BLKW #1
SAVE_R1 .BLKW #1
SAVE_R2 .BLKW #1
SAVE_R7 .BLKW #1

SCORES .BLKW #5

MIN_G .FILL #0
MAX_G .FILL #0
AVG_G .FILL #0

LEA R0, MSG_INSTRUCTIONS
PUTS

JSR R_SCO
JSR C_MIN
JSR C_MAX
JSR C_AVG
JSR CVT_LETT
JSR D_RES

HALT

R_SCO
  ST R1, SAVE_R1
  ST R7, SAVE_R7

  LEA R0, SCORES
  AND R1, R1, #0
```

```
  ADD R1, R1, #5

R_LOOP
  GETC
  OUT

  LD R2, ASCII_OFFSET
  ADD R0, R0, #-10
  ADD R0, R0, R2

  STR R0, R1, #0
  ADD R1, R1, #1
  ADD R1, R1, #-1
  BRp R_LOOP

  LD R1, SAVE_R1
  LD R7, SAVE_R7
  RET


C_MIN
    ST R0, SAVE_R0
    ST R1, SAVE_R1
    ST R2, SAVE_R2

    LEA R1, SCORES
    LDR R0, R1, #0
    ADD R2, R1, #5

MIN_LOOP
    ADD R1, R1, #1
    LDR R3, R1, #0

    NOT R4, R0
    ADD R4, R4, #1
    ADD R4, R3, R4

    BRzp END_MIN_LOOP
    ADD R0, R3, #0

END_MIN_LOOP
    ADD R2, R2, #-1
    BRp MIN_LOOP
```

```
    LD R0, SAVE_R0
    LD R1, SAVE_R1
    LD R2, SAVE_R2
    RET

C_MAX
    ST R0, SAVE_R0
    ST R1, SAVE_R1
    ST R2, SAVE_R2

    LEA R1, SCORES
    LDR R0, R1, #0
    ADD R2, R1, #5

MAX_LOOP
    ADD R1, R1, #1
    LDR R3, R1, #0

    NOT R4, R3
    ADD R4, R4, #1
    ADD R4, R0, R4

    BRzp END_MAX_LOOP
    ADD R0, R3, #0

END_MAX_LOOP
    ADD R2, R2, #-1
    BRp MAX_LOOP

    LD R0, SAVE_R0
    LD R1, SAVE_R1
    LD R2, SAVE_R2
    RET

C_AVG
    ST R0, SAVE_R0
    ST R1, SAVE_R1
    ST R2, SAVE_R2

    AND R0, R0, #0
    LEA R1, SCORES
    ADD R2, R1, #5

AVG_LOOP
```

```
    LDR R3, R1, #0
    ADD R0, R0, R3
    ADD R1, R1, #1
    ADD R2, R2, #-1
    BRp AVG_LOOP

    ADD R0, R0, R0
    ADD R1, R0, #0
    ADD R0, R0, #-10
    BRn DIV_LOOP
    ADD R0, R0, #0

DIV_LOOP
    ADD R1, R1, #10
    ADD R0, R0, #1
    BRn DIV_LOOP

    LD R0, SAVE_R0
    LD R1, SAVE_R1
    LD R2, SAVE_R2
    RET

CVT_LETT
    ST R0, SAVE_R0
    ST R1, SAVE_R1
    ST R7, SAVE_R7

    LEA R1, SCORES
    ADD R2, R1, #5
    LEA R6, LETT_G

CVT_LOOP
    LDR R0, R1, #0
    ADD R0, R0, #-16
    ADD R0, R0, #-16
    ADD R0, R0, #-16
    ADD R0, R0, #-16
    ADD R0, R0, #-16
    ADD R0, R0, #-10

    BRzp GRADE_A

    ADD R0, R0, #10
    BRzp GRADE_B
```

```
    ADD R0, R0, #10
    BRzp GRADE_C

D_RES
    ST R0, SAVE_R0
    ST R1, SAVE_R1
    ST R2, SAVE_R2
    ST R7, SAVE_R7

    LEA R0, MSG_MIN
    PUTS

    LD R0, MIN_G
    LD R1, SAVE_R1
    JSR PG_ASCII

    LEA R1, LETT_G
    ADD R1, R1, R0
    LDR R0, R1, #0
    OUT

    LEA R0, MSG_MAX
    PUTS

    LD R0, MAX_G
    JSR PG_ASCII

    LEA R1, LETT_G
    ADD R1, R1, R0
    LDR R0, R1, #0
    OUT

    LEA R0, NEWLINE
    PUTS

    LEA R0, MSG_AVG
    PUTS

    LD R0, AVG_G
    JSR PG_ASCII

    LEA R1, LETT_G
    ADD R1, R1, R0
```

```
    LDR R0, R1, #0
    OUT

    LEA R0, NEWLINE
    PUTS

    LD R0, SAVE_R0
    LD R1, SAVE_R1
    LD R2, SAVE_R2
    LD R7, SAVE_R7
    RET

; Letter grade conversion
GRADE_A .FILL #0041 ; ASCII code for 'A'
GRADE_B .FILL #0042 ; ASCII code for 'B'
GRADE_C .FILL #0043 ; ASCII code for 'C'
GRADE_D .FILL #0044 ; ASCII code for 'D'
GRADE_F .FILL #0046 ; ASCII code for 'F'

PG_ASCII
    ADD R1, R0, #0
    ADD R0, R0, #-10

    LD R0, ASCII_1
    OUT

    ADD R0, R1, #-16
    ADD R0, R0, #-16
    ADD R0, R0, #-16

    OUT
    RET

.END
—----------------------------------------------------------------------
VERSION 3
—----------------------------------------------------------------------
.ORIG x3000

MSG_INSTRUCTIONS .STRINGZ "ENTER THE GRADES (0-100):"
MSG_MIN .STRINGZ "\nMin Grade: "
MSG_MAX .STRINGZ "\nMax Grade: "
MSG_AVG .STRINGZ "\nAverage: "
NEWLINE .STRINGZ "\n"
```

```
MSG_A .STRINGZ "A\n"
MSG_B .STRINGZ "B\n"
MSG_C .STRINGZ "C\n"
MSG_D .STRINGZ "D\n"
MSG_F .STRINGZ "F\n"
MSG_INVALID .STRINGZ "Invalid grade\n"

LETT_G .FILL #0041
       .FILL #0042
       .FILL #0043
       .FILL #0044
       .FILL #0046

ASCII_1 .FILL x0031
ASCII_OFFSET  .FILL x0030
SPACE         .FILL x0020

SAVE_R0 .BLKW #1
SAVE_R1 .BLKW #1
SAVE_R2 .BLKW #1
SAVE_R7 .BLKW #1

SCORES .BLKW #5

MIN_G .FILL #0
MAX_G .FILL #0
AVG_G .FILL #0

LEA R0, MSG_INSTRUCTIONS
PUTS

JSR R_SCO
JSR C_MIN
JSR C_MAX
JSR C_AVG
JSR CVT_LETT
JSR D_RES

HALT

R_SCO
  ST R1, SAVE_R1
  ST R7, SAVE_R7
```

```
  LEA R0, SCORES
  AND R1, R1, #0
  ADD R1, R1, #5

R_LOOP
  GETC
  OUT

  LD R2, ASCII_OFFSET
  ADD R0, R0, #-10
  ADD R0, R0, R2

  STR R0, R1, #0
  ADD R1, R1, #1
  ADD R1, R1, #-1
  BRp R_LOOP

  LD R1, SAVE_R1
  LD R7, SAVE_R7
  RET


C_MIN
    ST R0, SAVE_R0
    ST R1, SAVE_R1
    ST R2, SAVE_R2

    LEA R1, SCORES
    LDR R0, R1, #0
    ADD R2, R1, #5

MIN_LOOP
    ADD R1, R1, #1
    LDR R3, R1, #0

    NOT R4, R0
    ADD R4, R4, #1
    ADD R4, R3, R4

    BRzp END_MIN_LOOP
    ADD R0, R3, #0

END_MIN_LOOP
    ADD R2, R2, #-1
```

```
    BRp MIN_LOOP

    ST R0, MIN_G

    LD R0, SAVE_R0
    LD R1, SAVE_R1
    LD R2, SAVE_R2
    RET

C_MAX
    ST R0, SAVE_R0
    ST R1, SAVE_R1
    ST R2, SAVE_R2

    LEA R1, SCORES
    LDR R0, R1, #0
    ADD R2, R1, #5

MAX_LOOP
    ADD R1, R1, #1
    LDR R3, R1, #0

    NOT R4, R3
    ADD R4, R4, #1
    ADD R4, R0, R4

    BRzp END_MAX_LOOP
    ADD R0, R3, #0

END_MAX_LOOP
    ADD R2, R2, #-1
    BRp MAX_LOOP

    ST R0, MAX_G
    LD R0, SAVE_R0
    LD R1, SAVE_R1
    LD R2, SAVE_R2
    RET

C_AVG
    ST R0, SAVE_R0
    ST R1, SAVE_R1
    ST R2, SAVE_R2
```

```
    AND R0, R0, #0
    LEA R1, SCORES
    ADD R2, R1, #5

AVG_LOOP
    LDR R3, R1, #0
    ADD R0, R0, R3
    ADD R1, R1, #1
    ADD R2, R2, #-1
    BRp AVG_LOOP

    ADD R0, R0, R0
    ADD R1, R0, #0
    ADD R0, R0, #-10
    BRn DIV_LOOP
    ADD R0, R0, #0

DIV_LOOP
    ADD R1, R1, #10
    ADD R0, R0, #1
    BRn DIV_LOOP

    LD R0, SAVE_R0
    LD R1, SAVE_R1
    LD R2, SAVE_R2
    RET

CVT_LETT
    ST R0, SAVE_R0
    ST R1, SAVE_R1
    ST R7, SAVE_R7

    LEA R1, SCORES
    ADD R2, R1, #5
    LEA R6, LETT_G

CVT_LOOP
    LDR R0, R1, #0
    ADD R0, R0, #-16
    ADD R0, R0, #-16
    ADD R0, R0, #-16
    ADD R0, R0, #-16
    ADD R0, R0, #-16
    ADD R0, R0, #-10
```

```
    BRzp GRADE_A

    ADD R0, R0, #10
    BRzp GRADE_B

    ADD R0, R0, #10
    BRzp GRADE_C

D_RES
    ST R0, SAVE_R0
    ST R1, SAVE_R1
    ST R2, SAVE_R2
    ST R7, SAVE_R7

    LEA R0, MSG_MIN
    PUTS

    LD R0, MIN_G
    LD R1, SAVE_R1
    JSR PG_ASCII

    LEA R1, LETT_G
    ADD R1, R1, R0
    LDR R0, R1, #0
    OUT

    LEA R0, MSG_MAX
    PUTS

    LD R0, MAX_G
    JSR PG_ASCII

    LEA R1, LETT_G
    ADD R1, R1, R0
    LDR R0, R1, #0
    OUT

    LEA R0, NEWLINE
    PUTS

    LEA R0, MSG_AVG
    PUTS
```

```
    LD R0, AVG_G
    JSR PG_ASCII

    LEA R1, LETT_G
    ADD R1, R1, R0
    LDR R0, R1, #0
    OUT

    LEA R0, NEWLINE
    PUTS

    LD R0, SAVE_R0
    LD R1, SAVE_R1
    LD R2, SAVE_R2
    LD R7, SAVE_R7
    RET

; Letter grade conversion
GRADE_A .FILL #0041
GRADE_B .FILL #0042
GRADE_C .FILL #0043
GRADE_D .FILL #0044
GRADE_F .FILL #0046

PG_ASCII
    ADD R1, R0, #0
    ADD R0, R0, #-10

    LD R0, ASCII_1
    OUT

    ADD R0, R1, #-16
    ADD R0, R0, #-16
    ADD R0, R0, #-16

    OUT
    RET

.END
```

—----------

These are the versions before the final code, these versions were made before working “together”. These were the first attempts.

—----------