1) Program to find the highest in a Series of numbers

```
DATA  SEGMENT

      MSGH1 DB    0DH, 0AH, "BHARAT ACHARYA EDUCATION $"
      MSGH2 DB    0DH, 0AH, "8086 PRACTICALS $"
      MSGH3 DB    0DH, 0AH, "------------------------------------ $"
      MSGH4 DB    0DH, 0AH, "TO FIND THE HIGHEST IN A GIVEN SERIES $"
      MSGH5 DB    0DH, 0AH, " $"

      MSG1  DB    0DH, 0AH, "PLEASE ENTER THE NUMBERS... $"
      MSG2  DB    ": $"
      MSG3  DB    0DH, 0AH, 0DH, 0AH, "HIGHEST: $"

      ARRN  DB    06 DUP (00H)
      LEN   DB    06
      HNUM  DB    00H

DATA  ENDS

CODE  SEGMENT
      ASSUME CS:CODE, DS:DATA
START:
      MOV AX, DATA      ; INITIALISE DS
      MOV DS, AX

      MOV AH, 09H       ; DISPLAY HEADERS
      LEA DX, MSGH1
      INT 21H
      LEA DX, MSGH2
      INT 21H
      LEA DX, MSGH3
      INT 21H
      LEA DX, MSGH4
      INT 21H
      LEA DX, MSGH5
      INT 21H

      MOV AH, 09H       ; ASK FOR THE NUMBERS
      LEA DX, MSG1
      INT 21H

      LEA SI, ARRN      ; GET THE NUMBERS
      MOV CL, LEN
      MOV BL, 01H
BK1:  MOV AH, 09H
      LEA DX, MSGH5
      INT 21H
      MOV AH, 02H
      MOV DL, BL
      ADD DL, 30H
      INT 21H
      INC BL
      MOV AH, 09H
      LEA DX, MSG2
      INT 21H
      CALL BAGET8
      MOV [SI], AL
      INC SI
      DEC CL
      JNZ BK1

CALC: LEA SI, ARRN      ; CALCULATE HIGHEST
      MOV CL, LEN
      MOV AL, HNUM

BCK2: CMP AL, [SI]
      JNC SKP2
      MOV AL, [SI]
SKP2: INC SI
      DEC CL
      JNZ BCK2
      MOV HNUM, AL

SHOW: MOV AH, 09        ; DISPLAY RESULT
      LEA DX, MSG3
      INT 21H
      LEA SI, HNUM
      CALL BAPUT8

EXIT: MOV AH, 02H
      MOV DL, 0DH
      INT 21H
      MOV DL, 0AH
      INT 21H
      MOV AH, 01H
      INT 21H
      MOV AH, 4CH       ; END THE PROGRAM... GO BACK TO DOS
      INT 21H


PROC  BAGET8            ; GETS AN 8 BIT NUMBER FROM THE SCREEN

      PUSH CX

      MOV AH, 01H
      INT 21H
      SUB AL, 30H
      CMP AL, 09H
      JLE G1
      SUB AL, 07H
G1:   MOV CL, 04H
      ROL AL, CL
      MOV CH, AL

      MOV AH, 01H
      INT 21H
      SUB AL, 30H
      CMP AL, 09H
      JLE G2
      SUB AL, 07H
G2:   ADD AL, CH          ; RETURNS THE NUMBER IN AL
      POP CX

      RET
ENDP  BAGET8


PROC  BAPUT8              ; DISPLAYS 8 BIT NUMBER ON THE SCREEN

      PUSH CX
      MOV AL, [SI]
      AND AL, 0F0H
      MOV CL, 04H
      ROL AL, CL
      ADD AL, 30H
      CMP AL, 39H
      JLE P1
      ADD AL, 07H
P1:   MOV AH, 02H
      MOV DL, AL
      INT 21H

      MOV AL, [SI]
      AND AL, 0FH
      ADD AL, 30H
      CMP AL, 39H
      JLE P2
      ADD AL, 07H
P2:   MOV AH, 02H
      MOV DL, AL
      INT 21H

      POP CX

      RET
ENDP  BAPUT8

CODE  ENDS
END   START
```

Notes from Bharat Acharya

1) *Copy and paste the above program in EMU8086 simulator as a new file*
2) *Click on "save", to save the file*
3) *Click on "emulate", to build the object code*
4) *Click on "run", to execute the program*
5) *Click on "vars" to observe the result in your variables*
6) *Run it several times and test different input numbers to verify the result*
7) *Erase the program and try to code it by yourself. Feel free to change variable names and even play with different registers. This is your first Assembly program… Own it!*

2) To find highest and lowest in a series of numbers

```
DATA  SEGMENT

      MSGH1 DB    0DH, 0AH, "BHARAT ACHARYA EDUCATION $"
      MSGH2 DB    0DH, 0AH, "8086 PRACTICALS $"
      MSGH3 DB    0DH, 0AH, "------------------------------ $"
      MSGH4 DB    0DH, 0AH, "TO FIND THE HIGHEST AND LOWEST $"
      MSGH5 DB    0DH, 0AH, " $"

      MSG1  DB    0DH, 0AH, "PLEASE ENTER THE NUMBERS... $"
      MSG2  DB    ": $"
      MSG3  DB    0DH, 0AH, 0DH, 0AH, "HIGHEST: $"
      MSG4  DB    0DH, 0AH, "LOWEST: $"

      ARRN  DB    06 DUP (00H)
      LEN   DB    06
      HNUM  DB    00H
      LNUM  DB    0FFH

DATA  ENDS

CODE  SEGMENT
      ASSUME CS:CODE, DS:DATA
START:
      MOV AX, DATA      ; INITIALISE DS
      MOV DS, AX

      MOV AH, 09H       ; DISPLAY HEADERS
      LEA DX, MSGH1
      INT 21H
      LEA DX, MSGH2
      INT 21H
      LEA DX, MSGH3
      INT 21H
      LEA DX, MSGH4
      INT 21H
      LEA DX, MSGH5
      INT 21H

      MOV AH, 09H       ; ASK FOR THE NUMBERS
      LEA DX, MSG1
      INT 21H

      LEA SI, ARRN      ; GET THE NUMBERS
      MOV CL, LEN
      MOV BL, 01H
BK1:  MOV AH, 09H
      LEA DX, MSGH5
      INT 21H
      MOV AH, 02H
      MOV DL, BL
      ADD DL, 30H
      INT 21H
      INC BL
      MOV AH, 09H
      LEA DX, MSG2
      INT 21H
      CALL BAGET8
      MOV [SI], AL
      INC SI
      DEC CL
      JNZ BK1

CALC: LEA SI, ARRN      ; CALCULATE HIGHEST AND LOWEST
      MOV CL, LEN
      MOV AL, HNUM
      MOV AH, LNUM

BCK2: CMP AL, [SI]
      JNC SKP2
      MOV AL, [SI]
SKP2: CMP AH, [SI]
      JC SKP3
      MOV AH, [SI]
SKP3: INC SI
      DEC CL
      JNZ BCK2
      MOV HNUM, AL
      MOV LNUM, AH

SHOW: MOV AH, 09        ; DISPLAY RESULT - HIGHEST
      LEA DX, MSG3
      INT 21H
      LEA SI, HNUM
      CALL BAPUT8
      MOV AH, 09        ; DISPLAY RESULT - LOWEST
      LEA DX, MSG4
      INT 21H
      LEA SI, LNUM
      CALL BAPUT8

EXIT: MOV AH, 02H
      MOV DL, 0DH
      INT 21H
      MOV DL, 0AH
      INT 21H
      MOV AH, 01H
      INT 21H
      MOV AH, 4CH       ; END THE PROGRAM... GO BACK TO DOS
      INT 21H


PROC  BAGET8            ; GETS AN 8 BIT NUMBER FROM THE SCREEN

      PUSH CX

      MOV AH, 01H
      INT 21H
      SUB AL, 30H
      CMP AL, 09H
```

```
        JLE G1
        SUB AL, 07H
G1:     MOV CL, 04H
        ROL AL, CL
        MOV CH, AL

        MOV AH, 01H
        INT 21H
        SUB AL, 30H
        CMP AL, 09H
        JLE G2
        SUB AL, 07H
G2:     ADD AL, CH              ; RETURNS THE NUMBER IN AL
        POP CX

        RET
ENDP    BAGET8


PROC    BAPUT8                  ; DISPLAYS 8 BIT NUMBER ON THE SCREEN

        PUSH CX
        MOV AL, [SI]
        AND AL, 0F0H
        MOV CL, 04H
        ROL AL, CL
        ADD AL, 30H
        CMP AL, 39H
        JLE P1
        ADD AL, 07H
P1:     MOV AH, 02H
        MOV DL, AL
        INT 21H

        MOV AL, [SI]
        AND AL, 0FH
        ADD AL, 30H
        CMP AL, 39H
        JLE P2
        ADD AL, 07H
P2:     MOV AH, 02H
        MOV DL, AL
        INT 21H

        POP CX

        RET
ENDP    BAPUT8
CODE    ENDS
END     START
```