Ahsanullah University of Science & Technology

Department of Computer Science & Engineering

Course No : CSE2214

Course Title : Assembly Language Programming Sessional
Assignment No : 10

Date of Performance : 07-03-2021

Date of Submission : 14-03-2021

Submitted To : Ms. Tahsin Aziz & Mr. A.K.M. Amanat Ullah

Submitted By-

Group : B_2

Name : S. M Tasnimul Hasan

Id : 180204142

Section : B

<u>Question No</u>: 01

<u>Question</u>: Suppose the class records are stored as follows:

CLASS
DB ‘MARY ALLEN’, 67, 45, 98, 33
DB ‘SCOTT BAYLIS’, 70, 56, 87, 44
DB ‘GEORGE FRANK', 82, 72, 89, 40
DB ‘SAM WONG', 78, 76, 92, 60

Each name occupies 12 bytes. Write a program to print the name of each student and his or her average (truncated to an integer) for the four exams.

<u>Answer</u>:

```
.MODEL SMALL
.STACK 100H
.DATA
   PROMPT_1   DB  'The Class Marks are as follows : ',0DH,0AH,'$'
   PROMPT_2   DB  0DH,0AH,'The Average Marks of Students are as
follows : ',0DH,0AH,'$'

   AVERAGE  DW  4 DUP(0)
    CLASS   DB  'MARY ALLEN  ',67,45,98,33
          DB  'SCOTT BAYLIS',70,56,87,44
          DB  'GEORGE FRANK',82,72,89,40
          DB  'SAM WONG    ',78,76,92,60

.CODE
  MAIN PROC
    MOV AX, @DATA               ; initialize DS
    MOV DS, AX

    LEA DX, PROMPT_1            ; load and print the string PROMPT_1
    MOV AH, 9
    INT 21H

    LEA SI, CLASS               ; set SI=offset address of variable CLASS
    MOV BH, 4                   ; set BH=4
    MOV BL, 16                   ; set BL=16

    CALL PRINT_2D_ARRAY          ; call the procedure
PRINT_2D_ARRAY

    LEA DI, AVERAGE              ; set DI=offset address of variable
AVERAGE
    LEA SI, CLASS                ; set SI=offset address of variable CLASS
    ADD SI, 12                   ; set SI=SI+12
    MOV CX, 4                    ; set CX=4

    @COMPUTE_AVERAGE:            ; loop label
      XOR AX, AX                 ; clear AX
      MOV DX, 4                  ; set DX=4

      @SUM:                      ; loop label
        XOR BH, BH               ; clear BH
        MOV BL, [SI]             ; set BL=[SI]

        ADD AX, BX               ; set AX=AX+BX

        INC SI                   ; set SI=SI+1
        DEC DX                   ; set DX=DX-1
      JNZ @SUM                   ; jump to label @SUM if DX!=0

      MOV BX, 4                  ; set BX=4
      DIV BX                     ; set AX=DX:AX/BX , DX=DX:AX%BX

      MOV [DI], AX               ; set [DI]=AX
      ADD DI, 2                  ; set DI=DI+2
      ADD SI, 12                 ; set SI=SI+12
    LOOP @COMPUTE_AVERAGE        ; jump to label
@COMPUTE_AVERAGE while CX!=0

    LEA DX, PROMPT_2             ; load and print the string PROMPT_2
    MOV AH, 9
    INT 21H

    LEA SI, AVERAGE              ; set SI=offset address of variable
AVERAGE
    LEA DI, CLASS                ; set DI=offset address of variable CLASS
    MOV CX, 4                    ; set CX=4

    @PRINT_RESULT:               ; loop label
      MOV BX, 12                 ; set BX=12
      MOV AH, 2                  ; set output function

      @NAME:                     ; loop label
        MOV DL, [DI]             ; set DL=[DI]
        INT 21H                  ; print a character

        INC DI                   ; set DI=DI+1
        DEC BX                   ; set BX=BX-1
      JNZ @NAME                  ; jump to label @NAME if BX!=0

      MOV DL, 20H                ; set DL=20H
      INT 21H                    ; print a character

      MOV DL, ":"                ; set DL=":"
      INT 21H                    ; print a character

      MOV DL, 20H                ; set DL=20H
      INT 21H                    ; print a character

      XOR AH, AH                 ; clear AH
      MOV AL, [SI]               ; set AL=[SI]

      CALL OUTDEC                ; call the procedure OUTDEC

      MOV AH, 2                  ; set output function
```

```
       MOV DL, 0DH               ; carriage return
       INT 21H

       MOV DL, 0AH               ; line feed
       INT 21H

       ADD SI, 2              ; set SI=SI+2
       ADD DI, 4              ; set DI=DI+4
     LOOP @PRINT_RESULT          ; jump to label @PRINT_RESULT
while CX!=0

     MOV AH, 4CH                 ; return control to DOS
     INT 21H
   MAIN ENDP
 ;Procedure Definitions PRINT_2D_ARRAY
  PRINT_2D_ARRAY PROC
    ; this procedure will print the given 2D array
    ; input : SI=offset address of the 2D array
    ;       : BH=number of rows
    ;       : BL=number of columns
    ; output : none

   PUSH AX                    ; push BX onto the STACK
   PUSH CX                    ; push CX onto the STACK
   PUSH DX                    ; push DX onto the STACK
   PUSH SI                   ; push SI onto the STACK

   MOV CX, BX                   ; set CX=BX

   @OUTER_LOOP:                   ; loop label
     MOV CL, BL                ; set CL=BL
     MOV AH, 2                 ; set output function

     @PRINT_NAME:               ; loop label
       MOV DL, [SI]            ; set DL=[SI]
   INT 21H             ; print a character

   INC SI             ; set SI=SI+1
   DEC CL              ; set CL=CL-1

   CMP CL, 4            ; compare CL with 4
  JG @PRINT_NAME            ; jump to label @PRINT_NAME if CL>4

  MOV DL, 20H            ; set DL=20H
  INT 21H              ; print a character

  @INNER_LOOP:            ; loop label
   MOV AH, 2            ; set output function
   MOV DL, 20H           ; set DL=20H
   INT 21H             ; print a character

   XOR AH, AH
   MOV AL, [SI]          ; set AX=[SI]

   CALL OUTDEC           ; call the procedure OUTDEC

   INC SI             ; set SI=SI+1
   DEC CL              ; set CL=CL-1
  JNZ @INNER_LOOP           ; jump to label @INNER_LOOP if CL!=0

  MOV AH, 2             ; set output function
  MOV DL, 0DH            ; set DL=0DH
  INT 21H              ; print a character

  MOV DL, 0AH            ; set DL=0AH
  INT 21H              ; print a character

  DEC CH               ; set CH=CH-1
 JNZ @OUTER_LOOP            ; jump to label @OUTER_LOOP if
CX!=0
```

```
  POP SI                    ; pop a value from STACK into SI
  POP DX                      ; pop a value from STACK into DX
  POP CX                      ; pop a value from STACK into CX
  POP AX                      ; pop a value from STACK into AX

  RET
PRINT_2D_ARRAY ENDP

;OUTDEC Procedure
OUTDEC PROC
  ; this procedure will display a decimal number
  ; input : AX
  ; output : none

  PUSH BX                      ; push BX onto the STACK
  PUSH CX                      ; push CX onto the STACK
  PUSH DX                      ; push DX onto the STACK

  XOR CX, CX                    ; clear CX
  MOV BX, 10                    ; set BX=10

  @OUTPUT:                      ; loop label
    XOR DX, DX                  ; clear DX
    DIV BX                    ; divide AX by BX
    PUSH DX                     ; push DX onto the STACK
    INC CX                    ; increment CX
    OR AX, AX                   ; take OR of Ax with AX
  JNE @OUTPUT                    ; jump to label @OUTPUT if ZF=0

  MOV AH, 2                     ; set output function

  @DISPLAY:                     ; loop label
    POP DX                    ; pop a value from STACK to DX
    OR DL, 30H                  ; convert decimal to ascii code
    INT 21H                    ; print a character
   LOOP @DISPLAY                  ; jump to label @DISPLAY if CX!=0

   POP DX                      ; pop a value from STACK into DX
   POP CX                      ; pop a value from STACK into CX
   POP BX                      ; pop a value from STACK into BX

   RET                        ; return control to the calling procedure
 OUTDEC ENDP
END MAIN
```

Question No: 02

Question: Write a program that uses XLAT to (a) read a line of text, and (b) print it on the next line with all small letters converted to capitals. The input line may contain any characters - small letters, capital letters, digits, characters, punctuation, and so on.

Answer:

```
.MODEL SMALL
.STACK 100H
.DATA

MSG_1 DB 10,13,'ENTER ANY STRING: $'
MSG_2 DB 10,13,'THE ENTERED STRING: $'
MSG_3 DB 10,13,'CONVERTED STRING: $'

P LABEL BYTE
M DB 0FFH
L DB ?
Q DB 0FFH DUP('$')
DATA ENDS

DISPLAY MACRO MSG
   MOV AH,9
```

```
    LEA DX,MSG
    INT 21H

ENDM

.CODE

MAIN PROC
    MOV AX,DATA
    MOV DS,AX

    DISPLAY MSG_1

    LEA DX,P
    MOV AH,0AH
    INT 21H

    DISPLAY MSG_2

    DISPLAY Q

    DISPLAY MSG_3

    LEA SI,Q

    MOV CL,L
    MOV CH,0


 CHECK:
    CMP [SI],41H
    JB DONE

    CMP [SI],61H
    JB DONE
```

```
   CMP [SI],7BH
   JG DONE

   UPR:
   SUB [SI],20H
   JMP DONE

   DONE:
   INC SI
   LOOP CHECK

   DISPLAY Q

  MOV AH,4CH
  INT 21H

  MAIN ENDP
END MAIN
```