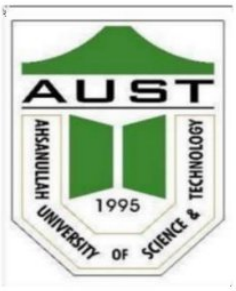

Ahsanullah University of Science & Technology

Department of Computer Science & Engineering

Course No : CSE2214

Course Title : Assembly Language Programming Sessional
Assignment No : 07

Date of Performance : 31-01-2021

Date of Submission : 28-02-2021

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

Question No : 01

Question : Write a program that prompts the user to enter a character, and in subsequent lines prints its ASCII code in binary and the number of 1 bit in its ASCII code.

Answer :

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

MSG1 DB "Enter a character : $"
MSG 2 DB 0DH, 0AH, "ASCII code of the character in binary is : $"
MSG 3 DB 0DH, 0AH, "Number of 1 bit in ASCII code is : $"

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

    MOV AH, 9
    LEA DX, MSG1
    INT 21H
```

```
MOV AH, 1
INT 21H

MOV BL, AL

MOV AH, 9
LEA DX, MSG2
INT 21H

XOR BH, BH      ; BH is counter for 1
MOV CX, 8       ; CX Loop counter
MOV AH, 2       ; single key input

OUTPUT:

SHL BL, 1       ; shift BL towards left by 1 position

JNC ZERO
INC BH
MOV DL, 31H
JMP DISPLAY

ZERO:
MOV DL, 30H

DISPLAY:

INT 21H
LOOP OUTPUT
```

```
    MOV AH, 9
    LEA DX, MSG3
    INT 21H

    OR BH, 30H

    MOV AH, 2
    MOV DL, BH
    INT 21H

    MOV AH, 4CH    ; Return 0
    INT 21H
    MAIN ENDP

END MAIN
```

Question No : 02

Question : Write a program that prompts the user to type a hex number of four hex digits or less, and outputs it in binary on the next line. If the user enters an illegal character, he or she should be prompted to begin again. Accept only uppercase letters. Your program may ignore any input beyond four characters.

Answer :

```
.MODEL SMALL
.STACK 100H
.DATA
    MSG1  DB  'Enter the hexadecimal number (max 4-digit) : $'
    MSG2  DB  0DH, 0AH, 'The equivalent 16 bit binary number is : $'
    MSG3  DB  0DH, 0AH, 'Illegal hex number. Please try again : $'

    COUNT DB  ?
```

```
.CODE

MAIN PROC

    MOV AX, @DATA

    MOV DS, AX

    MOV AH, 9

    LEA DX, MSG1

    INT 21H

    JMP START

    START_1:

        MOV AH, 9

        LEA DX, MSG3

        INT 21H

    START:

        XOR BX, BX      ; clear BX

        MOV COUNT, 48 ;  initialize loop counter

    START_2:

        MOV AH, 1

        INT 21H
```

```
CMP AL, 0DH     ; compare AL with CR
JNE SKIP        ; jump to label SKIP if AL != CR

CMP COUNT, 48   ; compare COUNT with 0
JBE START_1     ; jump to label START_1 if COUNT <= 0
JMP END         ; jump to label END

SKIP:
    CMP AL, "A"     ; compare AL with "A"
    JB DECIMAL      ; jump to label DECIMAL if AL<A

    CMP AL, "F"     ; compare AL with "F"
    JA START_1      ; jump to label START_1 if AL>F
    ADD AL, 09H     ; add 9 to AL
    JMP OK          ; jump to label OK

DECIMAL:
    CMP AL, 30H     ; compare AL with 0
    JB START_1      ; jump to label START_1 if AL<0
```

```
CMP AL, 39H      ; compare AL with 9
JA START_1       ; jump to label START_1 if AL > 9

OK:
INC COUNT        ; increment the COUNT variable
AND AL, 0FH      ; convert the ascii into binary code
MOV CL, 4        ; set CL = 4
SHL AL, CL       ; shift AL towards left by 4 positions
MOV CX, 4        ; set CX = 4

LOOP_1:
SHL AL, 1        ; shift AL towards left by 1 position
RCL BX, 1        ; rotate BX towards left by 1 position
                   through carry
LOOP LOOP_1      ; jump to label LOOP_1 if CX != 0
CMP COUNT, 34H   ; compare COUNT with 4
JE END           ; jump to label END if COUNT = 4
JMP START_2      ; jump to label START_2
```

```
END :

    MOV AH, 9
    LEA DX, MSG2
    INT 21H

    MOV CX, 16      ; set CX = 16
    MOV AH, 2       ; single key output

LOOP_2 :

    SHL BX, 1       ; shift BX towards left by 1 position
    JC ONE          ; jump to label ONE if CF = 1
    MOV DL, 30H     ; set DL = 0
    JMP DISPLAY     ; jump to label DISPLAY

ONE :

    MOV DL, 31H     ; set DL = 1

DISPLAY :

    INT 21H         ; display a character
    LOOP LOOP_2     ; jump to label LOOP_2 if CX != 0

    MOV AH, 4CH     ; Return 0
    INT 21H
    MAIN ENDP
END MAIN
```

Question No : 03

Question : Write a program that prompts the users to enter two unsigned hex numbers, 0 to FFFFh, and prints their sum in hex on the next line. If the user enters an illegal character, he or she should be prompted to begin again. Your program should be able to handle the possibility of unsigned overflow. Each input ends with a carriage return.

Answer :

```
.MODEL SMALL
.STACK 100H
.DATA
    MSG1  DB  0DH,0AH,'Enter the first Hex Numbers (0000-FFFF): $'
    MSG2  DB  0DH,0AH,'Enter the second Hex Number (0000-FFFF): $'
    MSG3  DB  0DH,0AH,'The sum of given Hex Numbers in Hex form : $'
    MSG4  DB  0DH,0AH,'Illegal character. Please try again. $'
    COUNT DB  ?
    VALUE DW  ?
```

```
.CODE
MAIN PROC
    MOV AX, @DATA
    MOV DS, AX

    JMP START_2

    START_1:
        MOV AH, 9
        LEA DX, MSG4
        INT 21H

    START_2:
        MOV AH, 9
        LEA DX, MSG1
        INT 21H

        XOR BX, BX      ; clear BX
        MOV COUNT, 30H  ; initialize loop counter

    START_3:
        MOV AH, 1
        INT 21H

        CMP AL, 0DH     ; compare AL with CR
        JNE SKIP_1      ; jump to label SKIP_1 if AL != CR
```

```
CMP  COUNT, 30H      ; compare COUNT with 0
JBE  START_1         ; jump to label if COUNT <= 0
JMP  END_1           ; jump to label END

SKIP_1 :
    CMP  AL, "A"     ; compare AL with "A"
    JB   DECIMAL_1   ; jump to label DECIMAL_1 if AL<A

    CMP  AL, "F"     ; compare AL with "F"
    JA   START_1     ; jump to label START_1 if AL>F
    ADD  AL, 09H     ; add 9 to AL
    JMP  OK_1        ; jump to label OK_1

DECIMAL_1 :
    CMP  AL, 30H     ; compare AL with 0
    JB   START_1     ; jump to label if AL<0

    CMP  AL, 39H     ; compare AL with 9
    JA   START_1     ; jump to label if AL>9
```

```
OK_1 :

    INC  COUNT      ; increment the count variable

    AND  AL, 0FH    ; convert the ascii into binary code

    MOV  CL, 4      ; set CL=4

    SHL  AL, CL     ; shift AL towards left by 4 positions

    MOV  CX, 4      ; set CX=4

LOOP_1 :

    SHL  AL, 1
    RCL  BX, 1      ; rotate BX towards left by 1 position through carry
    LOOP LOOP_1

    CMP  COUNT, 34H ; compare count with 4

    JE   END_1      ; jump to label if COUNT=4

    JMP  START_3

END_1 :

    MOV  VALUE, BX

    MOV  AH, 9
    LEA  DX, MSG2
    INT  21H
```

```
XOR BX, BX      ; clear BX
MOV COUNT, 30H  ; initialize loop counter

START_4 :

    MOV AH, 1
    INT 21H

    CMP AL, 0DH     ; compare AL with CR
    JNE SKIP_2      ; jump to label SKIP_2 if AL != CR

    CMP COUNT, 30H  ; compare COUNT with 0
    JBE START_1     ; jumb to label if COUNT <= 0
    JMP END_2       ; jump to label END_2

SKIP_2 :

    CMP AL, "A"
    JB DECIMAL_2    ; jump to label if AL<A

    CMP AL, "F"
    JA JUMP         ; jump to label if AL>F
    ADD AL, 09H     ; add 9 to AL
    JMP OK_2        ; jump to label OK_2
```

```
DECIMAL_2 :
    CMP  AL, 30H    ; compare AL with 0
    JB   JUMP       ; jump to label if AL<0

    CMP  AL, 39H    ; compare AL with 9
    JA   JUMP       ; jump to label if AL>9

    JMP  OK_2

JUMP :
    JMP  START_1

OK_2 :
    INC  COUNT      ; increment the COUNT variable
    AND  AL, 0FH    ; convert the ascii into binary code
    MOV  CL, 4      ; set CL = 4
    SHL  AL, CL     ; shift AL towards left by 4 positions

    MOV  CX, 4      ; set CX = 4

LOOP_2 :
    SHL  AL, 1      ; shift AL towards left by 1 position
    RCL  BX, 1      ; rotate BX towards left by 1 position
                    through carry.
```

```
LOOP  LOOP_2        ; jump to label LOOP_2 if CX != 0

CMP  COUNT, 34H     ; compare count with 4

JE  END_2           ; jump to label if COUNT = 4

JMP  START_4

END_2 :

    MOV  AH, 9
    LEA  DX, MSG3
    INT  21H

    ADD  BX, VALUE      ; add BX and VALUE

    JNC  SKIP           ; jump to label if CF = 1

    MOV AH, 2
    MOV  DL, 31H        ; set DL = 1
    INT  21H

SKIP :

    MOV  COUNT, 30H     ; set  COUNT = 0

LOOP_3 :

    XOR  DL, DL         ; clear DL
    MOV  CX, 4          ; set  CX = 4
```

```
LOOP_1 :
    SHL  BX, 1      ; shift BX towards left by 1 position
    RCL  DL, 1      ; rotate DL towards left by 1 position through carry
    LOOP LOOP_4     ; jump to label if CX != 0

    MOV  AH, 2
    CMP  DL, 9      ; compare DL with 9
    JBE  NUMERIC_DIGIT  ; jump to label if DL<=9
    SUB  DL, 9      ; convert it to number
    OR   DL, 40H    ; convert number to letter
    JMP  DISPLAY

NUMERIC_DIGIT :
    OR   DL, 30H    ; convert decimal to ascii code

DISPLAY :
    INT  21H
    INC  COUNT      ; increment COUNT variable
    CMP  COUNT, 34H ; compare COUNT with 4
    JNE  LOOP_3     ; jump to label if COUNT != 4
```

```
END :
    MOV AH, 1CH    ; Return 0
    INT 21H
    MAIN ENDP
END MAIN
```

Question No : 04

Question : Write a program that prompts the user to enter a string of decimal digits, ending with a carriage return, and print their sum in hex on the next line. If the user enters an illegal character, he or she should be prompted to begin again.

Answer :

```
. MODEL SMALL
. STACK 100H
. DATA
    MSG1 DB 'Enter a string of decimal digits : $'
    MSG2 DB 0DH, 0AH, 'The sum of the string of decimal digits
                       in Hex is : $'
```

```
    MSG3    DB    0DH, 0AH, 'Illegal character. Please try again: $'

    VALUE   DW    ?

.CODE

MAIN PROC

    MOV  AX, @DATA
    MOV  DS, AX

    MOV  AH, 9
    LEA  DX, MSG1
    INT  21H

    JMP  START_2

    START_1 :

        MOV  AH, 9
        LEA  DX, MSG3
        INT  21H

        XOR  BX, BX     ; clear BX
        XOR  CX, CX     ; clear CX

    START_2 :

        MOV  AH, 1
        INT  21H
```

```
INC   CX      ; increcement CX

CMP   AL, 0DH   ; comporce AL with CR

JNE   SKIP      ; jump to label if AL! = CR

CMP   CX, 1     ; compore cx with 1

JB    START_1   ; jump to label if CX<1

JMP   END_INPUT

SKIP :

   CMP   AL, 30H    ; compare AL with 0

   JB    START_1    ; jump to label AL<0

   CMP   AL, 39H    ; compare AL with 9

   JA    START_1    ; jump to label AL>9

   AND   AL, 0FH    ; convert the ascii to decimal code

   XOR   AH, AH     ; clear AH

   ADD   BX, AX

   JMP   START_2
```

```
END INPUT :

    MOV AH,9
    LEA DX, MSG2
    INT 21H

    MOV CX,4    ; initialize loop counter
    MOV AH,2    ; set output function

LOOP_1 :
    XOR DX, DX    ; clear DX

LOOP_2 :
    SHL BX,1    ; shift BX towards left by1position
    RCL DL,1    ; rotate DL towards left by 1 position
    INC DH      ; increment DH
    CMP DH,4    ; compare DH with 4
    JNE LOOP_2  ; jump to label if DH!=4

    CMP DL,9
    JBE NUMERIC_DIGIT ; jump to label if DL<=9
    SUB DL,9          ; convert it to numbers
    OR DL,40H         ; convert numbers to letters
    JMP DISPLAY
```

```
NUMERIC_DIGIT :

    OR  DL, 30H     ; convert decimal to ascii code

DISPLAY :

    INT 21H     ; print the character
    LOOP LOOP_1 ; jump to label if CX!=0

    MOV AH, 4CH     ; Return 0
    INT 21H
    MAIN ENDP

END MAIN
```