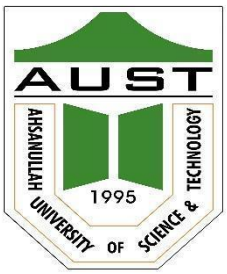

Ahsanullah University of Science & Technology

Department of Computer Science & Engineering

Course No : **CSE2214**
Course Title : **Assembly Language Programming Sessional**
Assignment No : **07**
Date of Performance : **28.01.2021**
Date of Submission : **04.02.2021**

Submitted To : **Ms. Tahsin Aziz & Ms. Moumita Choudhury**

Submitted By-
Group : **A_2**
Name : **Minhajul Islam Jim**
Id : **17.01.04.001**
Section : **A**

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

Ans:

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

MSG1 DB "ENTER A CHARACTER. $"
MSG2 DB 0DH,0AH, "ASCII CODE of the character
                  in binary is. $"
MSG3 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
MOV CX, 8
MOV AH, 2
OUTPUT:

SHL BL, 1
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
INT 21H
MAIN ENDP

END MAIN.
```

Q→2 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 character.

```
.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
  MOV COUNT, 48

START_2:
MOV AH, 1
INT 21H
CMP AL, ODH
JNE SKIP
CMP COUNT, 48
JBE START_1
JMP END

 SKIP
 CMP AL, "A"
 JB DECIMAL
 CMP AL, "F"
 JA START_1
 ADD AL, 09H
 JMP OK

DECIMAL:
CMP AL, 30H
JB START_1

CMP AL, 39H
JA START_1
```

```
OK:
INC COUNT
AND AL,0FH
MOV CL,4
SHL AL,CL
MOV CX,4
LOOP_1
  SHL AL,1
  RCL BX,1
  LOOP LOOP_1

  CMP COUNT,34H
  JE END
  JMP START_2
  END:
  MOV AH,9
  LEA DX,MSG2
  INT 21H
  MOV CX,16
  MOV AH,2
  LOOP_2
  SHL BX,1
  JC ONE
  MOV DL,30H
  JMP DISPLAY
```

ONE.

```
MOV DL, 31H
DISPLAY:
  INT 21H
LOOP LOOP_2
  MOV AH, 4CH
  INT 21H
MAIN ENDP
END MAIN.
```

Q-3 Write a program that prompts the user to enter two unsigned hex numbers, 0 to FFFF$_h$. and prints their sum in hex on the next line If the user enters an illigal 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.

```
.MODEL SMALL
.STACK 100H
.DATA
    MSG1 DB 0DH,0AH,'ENTER the first hex number (0000-FFFF):$'
    MSG2 DB 0DH,0AH,'ENTER the second hex number (0000-FFFF):$'
    MSG3 DB 0DH,0AH,'The sum of given hex number in hex form is:$'
    MSG4 DB 0DH,0AH,'Illegal character. Please try again.$'
    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
MOV  COUNT, 30H

START_3:
 MOV  AH, 1
  INT  21H
  CMP AL, 0DH
  JNE  SKIP_1
  CMP  COUNT, 30H
  JBE START_1
  JMP  END_1
SKIP_1
   CMP  AL, "A"
   JB   DECIMAL_1
   CMP  AL, "F"
   JA   START_1
   ADD  AL, 09H
   JMP   OK_1
  DECIMAL_1:
   CMP  AL, 30H
   JB START_1
   OK_1
   INC  COUNT
   AND  AL, 0FH
```

```
MOV CL, 4
SAL AL, CL
MOV CX, 4
LOOP_1:
  SHL AL, 1
  RCL BX, 1
  LOOP LOOP_1
  CMP COUNT, 34H
  JE END_1
  JMP START_3
END_1:
  MOV VALUE, BX
  MOV AH, 9
  LEA DX, MSG2
  INT 21
  XOR BX, BX
  MOV COUNT, 30H
START_4
MOV AH, 1
INT 21H
CMP AL, 0DH
JNE SKIP_2
```

```
CMP  COUNT, 30H
JBE  START_1
JMP  END_2

SKIP_2:
  CMP AL, "A"
  JB  DECIMAL_2
  CMP AL, "F"
  JA  JUMP
  ADD AL, 09H
  JMP OK_2
  DECIMAL_2
  CMP AL, 30H
  JB  JUMP
  CMP AL, 39H
  JA  JUMP
  JMP OK_2

  JUMP:
     JMP START_1

  OK_2
  INC COUNT
  AND AL, 0FH
  MOV CL, 4
  SHL AL, CL
```

```
MOV CX, 4.
LOOP_2:
SHL AL, 1
RCL BX, 1
LOOP LOOP_2
CMP COUNT, 34H
JE END_2
JMP START_4.
END_2:
MOV AH, 9
LEA DX, MSG3
INT 21H
ADD BX, VALUE
JNC SKIP
MOV AH, 2
MOV DL, 31H
INT 21H
SKIP:
MOV COUNT, 30H
LOOP3:
XOR DL, DL
MOV CX, 4
```

```
LOOP_4:
  SHL BX, 1
  RCL DL, 1
  LOOP LOOP_4
  MOV AH, 2
  CMP DL, 9
  JBE NUMERICAL_DIGIT
  SUB DL, 9
  OR DL, 40H
  JMP DISPLAY
  NUMERICAL_DIGIT
  OR DL, 30H
  DISPLAY:
  INT 21H
  INC COUNT
  CMP COUNT, 34H
  JNE LOOP_3
END:
  MOV AH, 4CH
  INT 21H
  MAIN ENDP
END MAIN.
```

Q4/ Write a program that prompts the user to enter a string of decimal digits, ending with a carriage return, and prints their sum in hex on the next line. If the user enters an illig illegal character, he or she should be prompted to begin again.

⇒

```
.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
XOR CX,CX
START_2:
MOV AH,1
INT 21H
INT 21H
INC CX
CMP AL,0DH
JNE SKIP
CMP CX,1
JB START_1
JMP END_INPUT
SKIP:
  CMP AL,30H
  JB START_1
  CMP AL,39H
  JA START_1
  AND AL,0FH
  XOR AH,A*H
```

```
ADD  BX, AX
JMP  START_2
END_INPUT
    MOV  AH, 9
    LEA  DX, MSG2
    INT  21H
    MOV  CH, 4
    MOV  AH, 2
    LOOP_1
    XOR  DX, DX
    LOOP_2
      SHL  BX, 1
      RCL  DL, 1
      INC  DH
      CMP  DH, 4
    JNE  LOOP_2
    CMP  DL, 9
    JBE  NUMERICAL_DIGIT
    SUB  DL, 9
    OR   DL, 40H
    JMP  DISPLAY
    NUMERICAL_DIGIT
     OR  DL, 30H
```

~~DISPLY~~

```
DISPLAY:
   INT 21H
   LOOP LOOP_1

MOV AH, 4CH
   INT 21H
   MAIN ENDP

END MAIN
```