Ahsanullah University of Science & Technology

Department of Computer Science and Engineering

Course No : CSE 2214

Course Title : Assembly Language Programming Sessional

Assignment no : 07

Date of Performance : 19.08.20

Date of Submission : 25.08.20

Submitted To : Ms.Tahsin Aziz & Md.Siam Ansary

Submitted By:

Name : Mahin opu

ID : 17.02.04.006

Year : 2nd

Semester : 2nd

Group : A1

Section : A

Question 01 :

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 bits in its ASCII code.

Answer:

```
.MODEL SMALL
 .STACK 100H

 .DATA
   PROMPT_1  DB  'Enter the character : $'
   PROMPT_2  DB  0DH,0AH,'The ASCII code of the given number in binary form is : $\'
   PROMPT_3  DB  0DH,0AH,'The number of 1 bits in ASCII code are : $\'

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

     LEA DX, PROMPT_1
     MOV AH, 9
     INT 21H

     MOV AH, 1
     INT 21H
```

```
XOR BX, BX
MOV BL, AL

LEA DX, PROMPT_2
MOV AH, 9
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

    LEA DX, PROMPT_3
    MOV AH, 9
    INT 21H

    OR BH, 30H

    MOV AH, 2
    MOV DL, BH
    INT 21H

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

Question 02:

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
  PROMPT_1  DB  'Enter the hexadecimal number ( max 4-digit ) : $'
  PROMPT_2  DB  0DH,0AH,'The equivalent 16-bit binary number is : $'
  ILLEGAL   DB  0DH,0AH,'Illegal hex number. Try again : $'

  COUNT     DB  ?

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

    LEA DX, PROMPT_1
    MOV AH,9
    INT 21H

    JMP @START

    @START_1:
      LEA DX, ILLEGAL
      MOV AH, 9
      INT 21H
```

```
@START:
  XOR BX, BX
  MOV COUNT, 30H

@START_2:
  MOV AH, 1
  INT 21H

  CMP AL, 0DH
  JNE @SKIP

  CMP COUNT, 30H
  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:

LEA DX, PROMPT_2
MOV AH, 9
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
```

Question 03 :

Write a program that prompts the user 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:

```
.STACK 100H

 .DATA
  PROMPT_1  DB  0DH,0AH,'Enter the first Hex number ( 0000 - FFFF ) : $'
  PROMPT_2  DB  0DH,0AH,'Enter the second Hex number ( 0000 - FFFF ) : $'
  PROMPT_3  DB  0DH,0AH,'The SUM of given Hex numbers in Hex form is : $'
```

```
  ILLEGAL   DB  0DH,0AH,'Illegal character. Try again.$'

  COUNT     DB  ?
  VALUE     DW  ?

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

    JMP @START_2

    @START_1:
      LEA DX, ILLEGAL
      MOV AH, 9
      INT 21H

    @START_2:
      LEA DX, PROMPT_1
      MOV AH, 9
      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

 CMP AL, 39H
 JA @START_1

@OK_1:

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_1
```

```
JMP @START_3

@END_1:

MOV VALUE, BX

LEA DX, PROMPT_2
MOV AH, 9
INT 21H

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:

LEA DX, PROMPT_3
MOV AH, 9
INT 21H
```

```
ADD BX, VALUE
JNC @SKIP
MOV AH, 2
MOV DL, 31H
INT 21H

@SKIP:

MOV COUNT, 30H

@LOOP_3:
  XOR DL, DL
  MOV CX, 4

  @LOOP_4:
    SHL BX, 1
    RCL DL, 1

  LOOP @LOOP_4

  MOV AH, 2

  CMP DL, 9
  JBE @NUMERIC_DIGIT
```

```
    SUB DL, 9
    OR DL, 40H
    JMP @DISPLAY

    @NUMERIC_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
```

Question 04:

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 illegal character, he or she should be prompted to begin again.

Answer:

```
.MODEL SMALL
 .STACK 100H

 .DATA
  PROMPT_1  DB  'Enter a decimal digit string : $'
  PROMPT_2  DB  0DH,0AH,'The sum of the decimal digit string in Hex is : $'
  ILLEGAL   DB  0DH,0AH,'Illegal character. Try again : $'

  TEMP      DW  ?
  VALUE     DW  ?
  v         dw  ?

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

```
LEA DX, PROMPT_1
MOV AH, 9
INT 21H

JMP @START_2

@START_1:
  LEA DX, ILLEGAL
  MOV AH, 9
  INT 21H

XOR BX, BX
XOR CX, CX

@START_2:
  MOV AH, 1
  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, AH
    ADD BX, AX

    JMP @START_2

@END_INPUT:

LEA DX, PROMPT_2
MOV AH, 9
INT 21H

MOV CX, 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 @NUMERIC_DIGIT
 SUB DL, 9
 OR DL, 40H
 JMP @DISPLAY

 @NUMERIC_DIGIT:
  OR DL, 30H

 @DISPLAY:
 INT 21H
LOOP @LOOP_1
```

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