Lab: 9
Solving Miscellaneous Problems with Assembly Language Programming

Miscellaneous Problems
In most of the cases we have to take input from the keyboard and display the result in the screen. As you know that if you take input from the user it is in ASCII format and to display a character it should be in ASCII format again. For the characters it is ok, but what happens when you have to process numbers. In this case you have to convert the ASCII character in to numeric form when taking input and before displaying you have to convert the number into ASCII character form.

To convert a character that is read by `INT 21H` function 01 we process as follows.
```
MOV AH,01
INT 21H
SUB AH,30H ; Converts the ASCII character to binary number
```
When taking multiple digit number you have to make a loop to read characters (actually numbers) convert the characters to numbers as above and store the final number in some location in data segment.
For multiple digit number we process as follows.
```
        .DATA
NUM   DB 0         ;Space for three digit number
COUNT DB 0         ;Counter to count the no of digits entered
NEWLN DB 0DH,0AH,'$'
        .CODE
MAIN PROC FAR
        MOV AX,@DATA
        MOV DS,AX

        MOV CX,03        ;For three digit decimal number
        MOV SI,OFFSET NUM

        ;Read characters for a number and add with prev num
  L1:   MOV AH,01
        INT 21H
        CMP AL,0DH       ;Stop if the user presses return
        JE LP
        SUB AL,30H       ;Convert the digit to ASCII
        PUSH AX              ;Save the digit for further purpose
        MOV AL,10        ;Multiply NUM by ten and store it back
        MUL NUM          ;As we are using byte operation the result
        MOV NUM,AL       ;will not be greater than byte
        POP AX               ;recover back the number
        ADD NUM,AL       ;Add it with the previous result
        INC COUNT        ;Count the number of digits entered
  LP:   LOOPNE L1
```

Without converting the digits entered at the same instant, the read digit are first stored in data segment then processed later. (See Abel's book page no 250). Read a string store in the memory and then convert to binary it later.
For the displaying purpose also we proceed in the same way. In this case we do the reverse process; convert the binary number to ASCII decimal format.
To display a number in memory at NUM into decimal format we process as follows
```
        ;Back to the ASCII format for display
        MOV CX,10
        MOV AL,NUM
        MOV AH,0
        MOV BX,0

        ;Find each decimal digit of the number and store in stack
  L4:   MOV DX,0
        DIV CX    ;Divide by 10
        ADD DX,30H       ;Convert the digit to characters
        PUSH DX          ;Store the decimal digit in the stack
```

```
        INC BX
        CMP AX,0 ;Stop if the number is <= 0
        JA L4

        ;Get characters from stack and display as decimal number
        MOV AH,02
        MOV CX,BX
  DISP: POP DX
        INT 21H          ;Display the character
        LOOP DISP
```

The processing can be done directly in the data segment. (See Abel's book page no 251). Convert the binary number into ASCII decimal format and display the final ASCII string that represents the decimal number.

To convert the binary number into ASCII Hexadecimal format divide the number by 16 instead of 10 and add 30H to the remainder numbers from 0 to 9 and add 41H to the remainder numbers from 10 to 15.

Assignments:
1. Write a program to find the sum of numbers from 1 to *n*. Read n from the user and display the sum as the decimal format. (also try to display the sum in Hexadecimal format)
2. Write a program to find the sum of the following series up to the terms specified by the user and display the result in decimal format. (also try to display the sum in Hex format)

    2 x 4 + 3 x 6 + … to n terms
3. Write a program that takes a string and count the number of words in the string. Display the count in decimal format
4. Write a program that takes a string and count the no of vowels in the string. Display the count in decimal format.
5. Write a program to add ten 16-bit numbers stored in memory and store and display the result in decimal format.