```
.MODEL SMALL
.STACK 100H
.DATA
    MSGA DB 'Enter the Length: $'
    MSGB DB 'Sum of the Fibonacci numbers  $'
    MSGC DB 'is $'
    COMA DB ', $'
    TEMP DW ?
    A DW ?
    B DW ?
    LEN DW ?
    SUM DW ?

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

    LEA DX,MSGA                 ;Print 'Enter the Length: $'
    MOV AH,9
    INT 21H

GET_INPUT:
    MOV AH,1
    MOV BX,0
    INT 21H

    CMP AL,0DH
    JE FIBONACCI_SERIES         ;If Enter

    INER_LOOP_1:
        MOV AH,0                ;Use full 16 bits of AX
        SUB AX,48

        MOV TEMP,AX
        MOV AX,10
        MUL BX                  ;AX = AX*BX
        MOV BX,AX
        ADD BX,TEMP

    MOV AH,1                    ;Input new digit
    INT 21H
    CMP AL,0DH
    JNE INER_LOOP_1             ;If Enter

    LEA DX,MSGB                 ;Print 'Fibonacci series: $'
    MOV AH,9
    INT 21H

FIBONACCI_SERIES:
    MOV LEN,BX
    CMP LEN,0
    JE LINK_UP

    MOV A,0
    MOV B,1

    MOV DL,0                    ;Print fibonacci first number
    ADD DL,48
    MOV AH,2
    INT 21H

    MOV DX,0
    ADD SUM,DX                  ;Sum first number

    LEA DX,COMA                     ;Print Space
    MOV AH,9
    INT 21H

    DEC LEN
    CMP LEN,0
    JE LINK_UP

    MOV DL,1                    ;Print fibonacci second number
    ADD DL,48
    MOV AH,2
    INT 21H

    MOV DX,1
    ADD SUM,DX                  ;Sum second number

    LEA DX,COMA                     ;Print Space
    MOV AH,9
    INT 21H

    DEC LEN
    JMP NEXT_NUMBER

LINK_UP:
    JMP GET_SUM

NEXT_NUMBER:
    CMP LEN,0
    JE LINK_UP

    MOV AX,A
    MOV BX,B
    ADD AX,BX
    MOV DX,AX                       ;Fibonacci number is in DX
    MOV AX,BX
    MOV BX,DX

    MOV A,AX
    MOV B,BX

    ADD SUM,DX

    MOV CX,0
    MOV AX,DX                       ;Prepear for multi digit printing
    MOV BX,10

    STOR_MUL_DIGIT:
        MOV DX,0
        DIV BX
        PUSH DX                     ;Stor each digit in stack
        INC CX
        CMP AX,0
        JNE STOR_MUL_DIGIT

    PRINT_FROM_STACK:
        MOV AH,2
        POP DX                      ;print each digit from stack
        ADD DL,48
        INT 21H
        LOOP PRINT_FROM_STACK

    DEC LEN
    LEA DX,COMA                     ;Print Space
    MOV AH,9
    INT 21H
    JMP NEXT_NUMBER
```

```
GET_SUM:
    LEA DX,MSGC                     ;Print is
    MOV AH,9
    INT 21H

    MOV CX,0
    MOV AX,SUM                      ;Prepear for sum printing
    MOV BX,10

    STOR_SUM:
        MOV DX,0
        DIV BX
        PUSH DX                     ;Stor each digit in stack
        INC CX
        CMP AX,0
        JNE STOR_SUM

    PRINT_SUM:
        MOV AH,2
        POP DX                      ;print each digit from stack
        ADD DL,48
        INT 21H
        LOOP PRINT_SUM

    MOV AH,4CH
    INT 21H

MAIN ENDP
    END MAIN
```