Experiment 10

Experiment no 10: Generate the first ‘n’ Fibonacci numbers using Procedure concept

Roll No.	17
Name	Manav Jawrani
Class	D10-A
Subject	Microprocessor Lab
LO Mapped	LO5: Write programs based on string and procedure for 8086 microprocessors.

Aim: Generate the first ‘n’ Fibonacci numbers using Procedure concept.

Introduction:

In mathematics, the Fibonacci numbers, commonly denoted F_n, form a sequence, the Fibonacci sequence, in which each number is the sum of the two preceding ones. The sequence commonly starts from 0 and 1, although some authors omit the initial terms and start the sequence from 1 and 1 or from 1 and 2. Starting from 0 and 1, the next few values in the sequence are:-

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, ...

Theory:

Algorithm:

Step 1 - Initialize the data section and create a variable ‘num’ to store the input from the user.
Step 2 - Initialize the counter and make decrement by 2.
Step 3 - Call the procedures to read and print the output.
Step 4 - Create the loop for the fibonacci condition and define/write the condition.
Step 5 - Execute the loop operations to generate the fibonacci series and print them.
Step 6 - Now, create the procedure to display 8 bit characters in decimal value in the ‘DL’ register.
Step 7 - Also, create a procedure to read 8 bit numbers and store them in the ‘AL’ register.
Step 8 - Lastly, create the procedure to print the output message.
Step 9 - Stop.

Code:

```
.model small
.data

msg_1 db 10,13,'Enter the Nth Number: $'
msg_2 db 10,13,'Fibonacci series is: 0 1 $'
num dw ?
a dw 0h
b dw 01h
```

```
.code
mov AX,@data
mov DS,AX

lea DX,msg_1
call printf
call read_8bit

mov cx,num        ;counter till nth number
sub cx,02h        ;decrement by 2 as first 2 are printed

lea DX,msg_2
call printf

loop1:                   ;Fibonacci loop
            mov AX,a            ;temporarily assigning to REG
            add AX,b            ;a = a + b ; c = AX
            mov a,AX            ;restoring a as c
            mov DI,CX                 ;storing the counter temporarily in DI
            mov DX,AX                       ;stored in DL to print Number
            call print_8bit           ;display nth number
            mov AX,a
            XCHG AX,b                       ;a=b and b=a(which is c)
            mov a,AX
            mov CX,DI                 ;restoring counter
loop loop1

mov AH,4Ch
int 21h

;procedure to display 8 bit character in decimal value in DL
print_8bit proc near
      mov ax,0000h
```

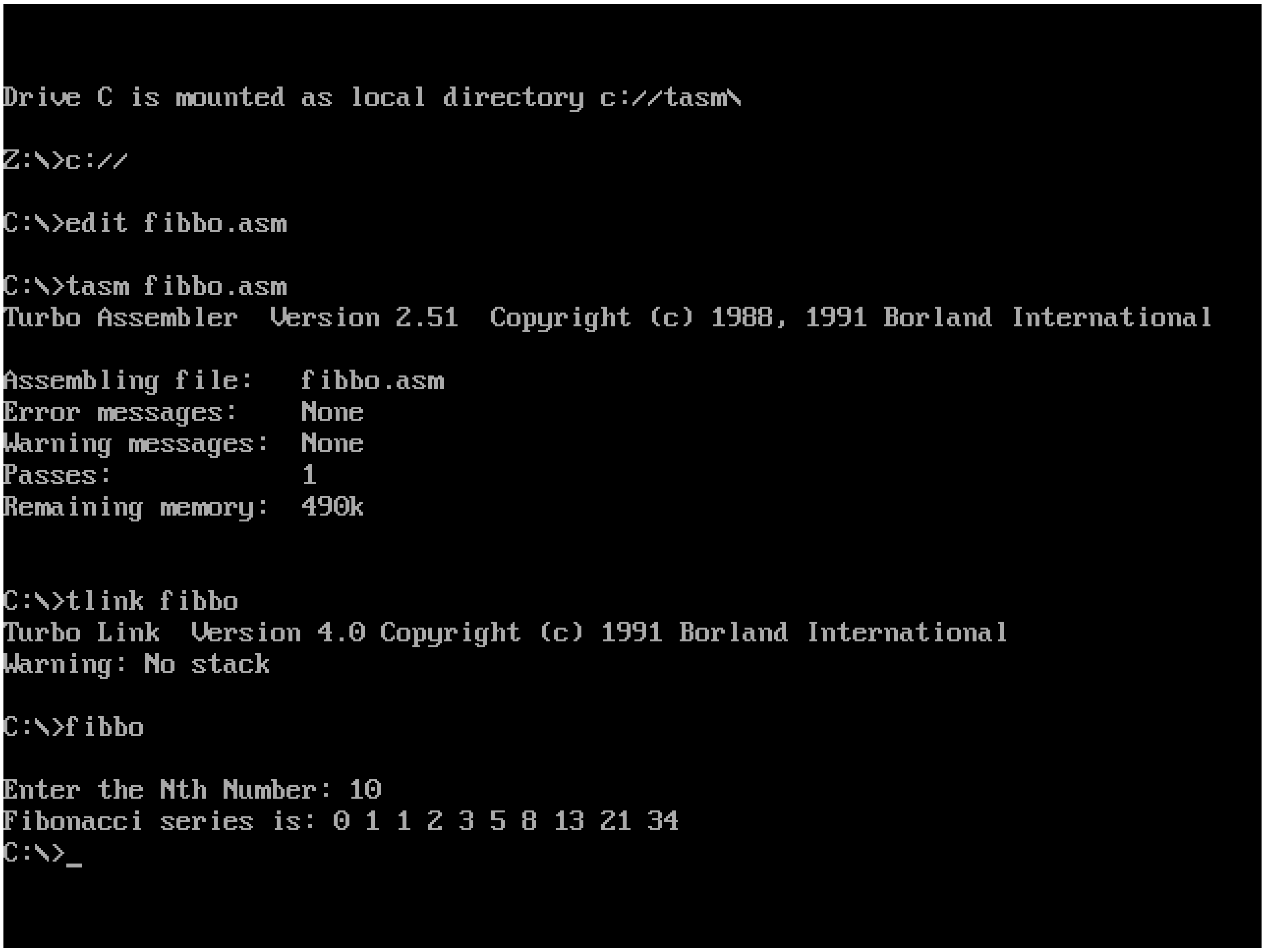
```
    mov al,dl
    mov bx,0010d
    mov CX,0000h

    Loop_push:
        mov DX,0000h
        div BX
        push DX
        inc CX
        cmp AX,0000h
    JNE Loop_push

    Loop_pop:
        pop DX
        add dx,0030h        ;converting the number to ASCII value
        mov ah,02h          ;character display
        int 21h
    loop Loop_pop

    mov dl,' '          ;printing space
    mov ah,02h
    int 21h

    ret
print_8bit endp

;procedure to read 8 bit numbers & store in AL
read_8bit proc near
    mov AH,01h              ;reading 1st nibble
    int 21h
    sub AL,30h
    mov BL,AL           ;temporary storage
    mov AH,01h              ;reading 2nd Nibble
    int 21h
```

```
        sub AL,30h
        mov AH,BL
        AAD                         ;ASCII adjust before division
        mov num,AX
        ret
read_8bit endp

;procedure to print message
printf proc near
        mov AH,09h
        int 21h
        ret
printf endp

end
```

Input:

Here we are finding/generating the first fifteen fibonacci numbers viz.
0,1, 1, 2, 3, 5, 8, 13, 21, 34

Output:



```
Drive C is mounted as local directory c://tasm\

Z:\>c://

C:\>edit fibbo.asm

C:\>tasm fibbo.asm
Turbo Assembler  Version 2.51  Copyright (c) 1988, 1991 Borland International

Assembling file:   fibbo.asm
Error messages:    None
Warning messages:  None
Passes:            1
Remaining memory:  490k


C:\>tlink fibbo
Turbo Link  Version 4.0 Copyright (c) 1991 Borland International
Warning: No stack

C:\>fibbo

Enter the Nth Number: 10
Fibonacci series is: 0 1 1 2 3 5 8 13 21 34
C:\>_
```

<u>Conclusion</u>:

We have understood the aim of this experiment and successfully executed it.