

# PROJECT 2

## FIBONACCI NUMBERS

**CSCI 150  
ASSEMBLY LANGUAGE**

**MAI PHAM**

**DEVELOPMENT ENVIRONMENT  
VM - VISUAL STUDIO 2017**

### **TABLE OF CONTENTS**

- ❖ **Project Note**
- ❖ **Output**
- ❖ **Source Code (Fibonacci.asm)**

## PROJECT NOTE

### OBJECTIVE:

- ❖ Write a program that ask for an input n value between 2 to 20 and use a procedure to output the Fibonacci sequence as well as the sum and the last value.

### STATUS/EXTRA CREDIT:

- ❖ The project is fairly easy. I did not encounter major issues with it and had successfully run the main project as well as the extra credit. The extra credit is using a Fibonacci procedure to create an array of Fibonacci sequence and use that array to compute the sum and print out the last value.

## OUTPUT

```
C:\Windows\system32\cmd.exe
Project 2 - Fibonacci Number
Author: Mai Pham

Enter a positive number [2 - 20] --> 4
Fibonacci sequence: 0 1 1 2 3
Sum: 7
Last Value: 3

Extra credit using array and procedure:
Fibonacci sequence: 0 1 1 2 3
Sum: 7
Last Value: 3

Press any key to continue . . .
```

```
C:\Windows\system32\cmd.exe
Project 2 - Fibonacci Number
Author: Mai Pham

Enter a positive number [2 - 20] --> 20
Fibonacci sequence: 0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 6765
Sum: 17710
Last Value: 6765

Extra credit using array and procedure:
Fibonacci sequence: 0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 6765
Sum: 17710
Last Value: 6765

Press any key to continue . . .
```

# SOURCE CODE

```
TITLE ASM Template

INCLUDE Irvine32.inc

.data
project BYTE "Project 2 - Fibonacci Number", 0
author BYTE "Author: Mai Pham", 0
prompt1 BYTE "Enter a positive number [2 - 20] --> ", 0
prompt2 BYTE "Fibonacci sequence: ", 0
prompt3 BYTE "Sum: ", 0
prompt4 BYTE "Last Value: ", 0
prompt5 BYTE "Extra credit using array and procedure:", 0

sum DWORD ?
lValue DWORD ?
myArray DWORD 25 DUP(0), 0

.code
main PROC
    mov edx, OFFSET project          ; display project
    call WriteString
    call crlf
    mov edx, OFFSET author          ; display name
    call WriteString
    call crlf
    call crlf

    mov edx, OFFSET prompt1         ; ask for input value
    call WriteString
    call ReadInt                    ; read integer into EAX
    push eax                        ; save for extra credit

    mov edx, OFFSET prompt2         ; display fibonacci sequence
    call WriteString

    call fibNumbers                  ; call procedure
    mov sum, edx                    ; save sum
    mov lValue, eax                  ; save last value
    call crlf

    mov edx, OFFSET prompt3         ; output sum
    call WriteString
    mov eax, sum
    call WriteDec
    call crlf

    mov edx, OFFSET prompt4         ; output last value
    call WriteString
    mov eax, lValue
    call WriteDec
    call crlf
    call crlf

    ;;;;;;;;;;;;;;
    mov edx, OFFSET prompt5         ; display extra credit
    call WriteString
    call crlf
```

```

    mov esi, OFFSET myArray          ; get array address
    pop ecx                          ; get n times from stack
    call fibNumbers2

    mov edx, OFFSET prompt2          ; display fibonacci sequence
    call WriteString

    inc ecx                          ; print sequence
    call printArray
    call crlf

    mov edx, OFFSET prompt3          ; output sum
    call WriteString
    call arraySum                    ; get sum
    mov sum, eax                     ; save sum
    call WriteDec
    call crlf

    mov edx, OFFSET prompt4          ; output last value
    call WriteString
    call getLastValue                ; get last value
    mov lValue, eax                  ; save last value
    call WriteDec
    call crlf
    call crlf

    exit
main ENDP

fibNumbers PROC
    mov ecx, eax                     ; loop amount of n times
    mov edx, 0                       ; sum start at 0
    mov eax, 0
    mov ebx, 1
    call WriteDec                    ; display sequence
L1:
    add eax, ebx                     ; add last two numbers
    add edx, ebx                     ; add to sum
    xchg eax, ebx                    ; exchange the two last number
    push eax                         ; save eax
    mov al, ' '                      ; display space
    call WriteChar
    pop eax
    call WriteDec                    ; display sequence
    loop L1
    ret
fibNumbers ENDP

fibNumbers2 PROC
    push esi                         ; save address and n times
    push ecx
    mov eax, 0
    mov ebx, 1
    mov [esi], eax                   ; store into array
L1:
    add eax, ebx                     ; add last two numbers
    xchg eax, ebx                    ; exchange the two last number
    add esi, TYPE myArray            ; move to next index
    mov [esi], eax                   ; store into array
    loop L1

```

```

        pop ecx                ; restore address and n times
        pop esi
        ret
fibNumbers2 ENDP

arraySum PROC
        push esi              ; save address and n times
        push ecx
        mov eax, 0            ; set to 0
L1:      add eax, [esi]         ; add sum as it loop through index
        add esi, TYPE myARRAY
        loop L1

        pop ecx              ; restore address and n times
        pop esi
        ret
arraySum ENDP

printArray PROC
        push esi              ; save address and n times
        push ecx
L1:      mov eax, [esi]         ; print value
        call WriteDec
        mov al, ' '           ; print space
        call WriteChar
        add esi, TYPE myArray
        loop L1

        pop ecx              ; restore address and n times
        pop esi
        ret
printArray ENDP

getLastValue PROC
        mov eax, ecx          ; get the number of value in an array
        dec eax              ; dec to go to the last value
        mov ebx, TYPE myArray ; mul by the type of array to get
position
        mul ebx
        add esi, eax
        mov eax, [esi]        ; move that value into eax
        ret
getLastValue ENDP

END main

```