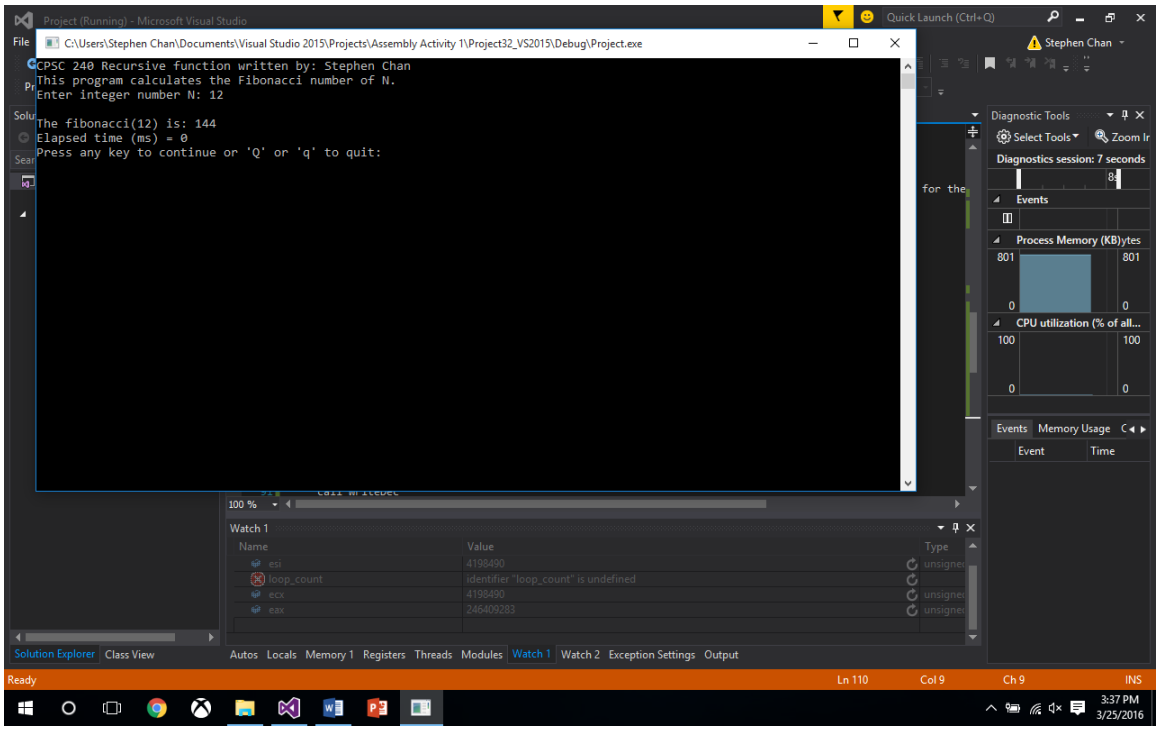
Stephen Chan
Team Number: 422
9419
CPSC 240 Assignment 2

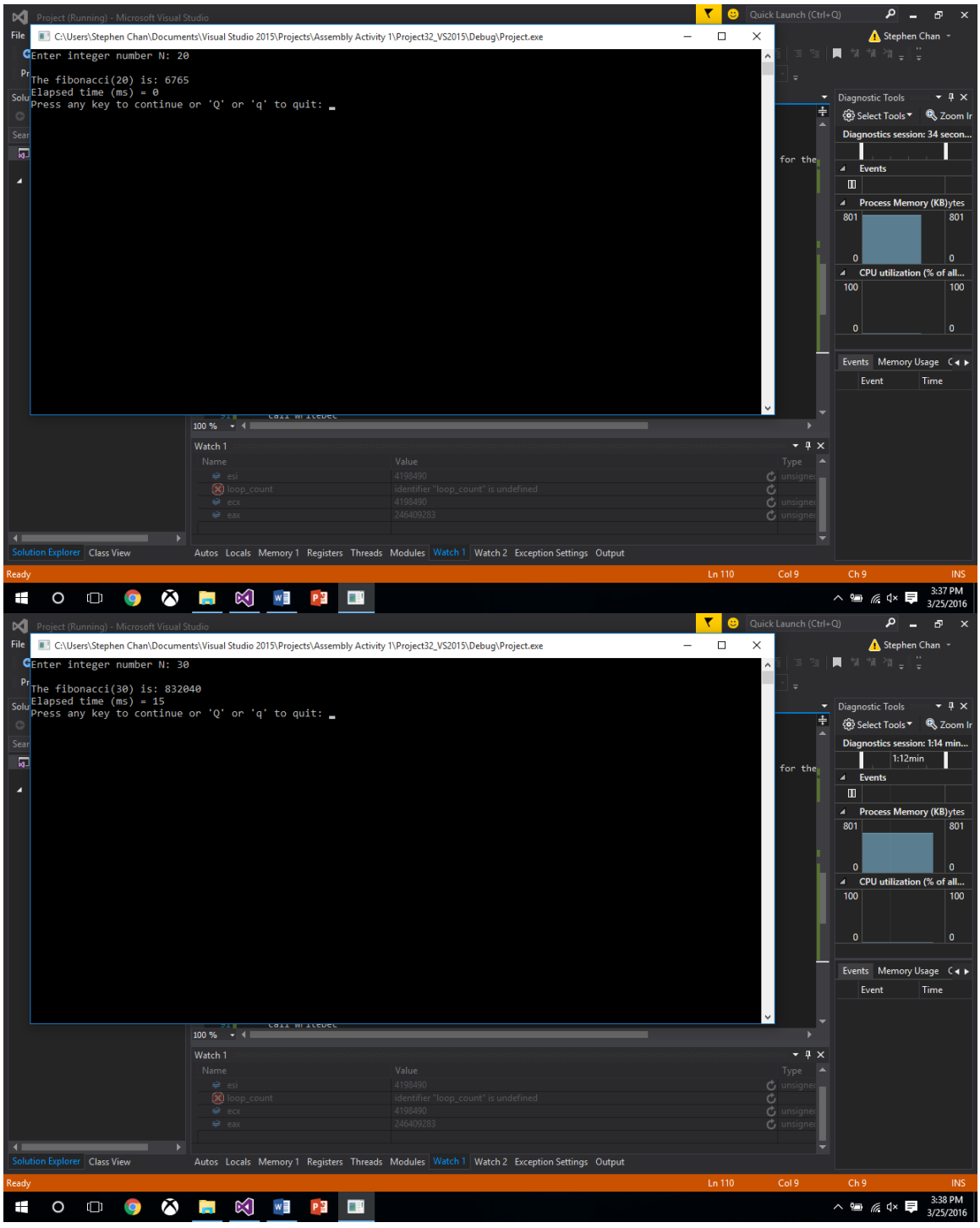
Program prompts user to enter a value in order to find the Fibonacci sum of all the values to that number. Time it takes to calculate the values is recorded and displayed to the user. User can continue finding more values or quit the program.

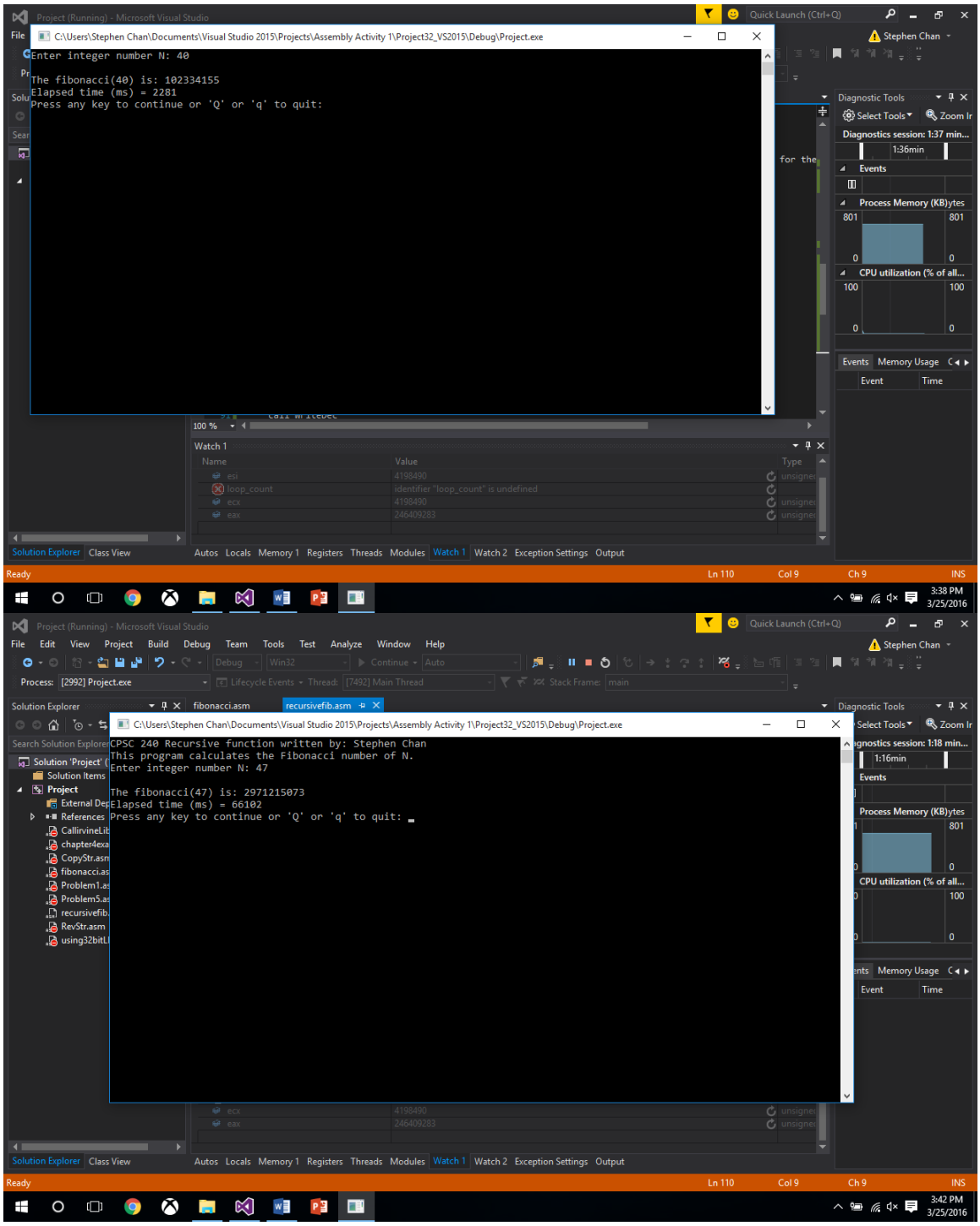
```
; Assignment 2 Recursive Fibonacci
; Stephen Chan
; 9419
; CPSC 240 Assignment 2
; Partner: Daniel Berumen
; Creates a recursive fibonacci array of values.

INCLUDE Irvine32.inc

.386
.model flat,stdcall
.stack 4096
ExitProcess proto,dwExitCode:dword

.data
str1 BYTE "CPSC 240 Recursive function written by: Stephen Chan",0
str2 BYTE "This program calculates the Fibonacci number of N.",0
str3 BYTE "Enter integer number N: ",0
str4 BYTE "The fibonacci(",0
str5 BYTE ") is: ",0
str6 BYTE "Elapsed time (ms) = ",0
str7 BYTE "Press any key to continue or 'Q' or 'q' to quit: ",0
           ;------------------


.code
;-----------------------------------------------------
; generate_fibonacci proc uses ebx ecx
; Generates fibonacci values and sums them to register eax
; Receives: ebx,ecx
; Returns: eax
;-----------------------------------------------------

generate_fibonacci proc uses ebx ecx
      cmp eax,1
      je one                                          ;if f(n)==1
      jl zero                                         ;if f(n)==0
      push eax                                        ;save n
      dec eax                                         ;n-1
      call generate_fibonacci                         ;f(n-1)
      mov ebx,eax                                     ;mov value from
                                                 eax to ebx
      pop eax                                         ;restore n
      sub eax,2                                       ;n-2
      call generate_fibonacci                         ;f(n-2)
```

```
      mov ecx,eax                                    ;mov value from
eax to ecx
      add ebx,ecx                                    ;f(n-1)+ f(n-2)
      mov eax,ebx
      ret
one:
      mov eax,1
      ret
zero:
      mov eax,0
      ret
generate_fibonacci endp


main proc
      mov edx, OFFSET str1          ;Header
      call WriteString
      call Crlf

      mov edx, OFFSET str2          ;Header
      call WriteString
      call Crlf
L1:
      mov edx, OFFSET str3          ;Prompt user to enter a value
      call WriteString
      call ReadInt                  ;Reads User value
      call Crlf

      push eax                      ;Pushes user value in order to save
                                    the value and open the register for
                                    the start time.
      call GetMSeconds              ;Get start time
      mov ebx,eax                   ;Moves the start time to register
                                    ebx
      pop eax                       ;Pops value user entered

      mov edx, OFFSET str4          ;Header for fibonacci count
      call WriteString
      call WriteDec

      call generate_fibonacci       ;Calls the procedure

      mov edx, OFFSET str5          ;Header for fibonacci sum
      call WriteString
      call WriteDec
      call Crlf

      call GetMSeconds              ;End time of calculation
      sub eax,ebx                   ;Calculates the time elapsed from
                                    start to end

      mov edx, OFFSET str6          ;Header for the elapsed time
      call WriteString
      call WriteDec
      call Crlf
```

```
	mov edx, OFFSET str7		;Header prompting the user to enter
					q to quit or enter any key to
					continue
	call WriteString
	call ReadChar			;Reads the user input

	cmp al,'q'			;Compares user with lower case q
					character
	je quit				;Jumps to loop to exit program if
					equal
	cmp al,'Q'			;Compares with Upper case Q
	je quit				;Jumps to loop to exit program if
				equal

	call Crlf			;endline
	call Clrscr			;Clear the screen
	jnz L1				;Loop back to beginning
quit:
	invoke ExitProcess,0

main endp
end main
```




Project (Running) - Microsoft Visual Studio
C:\Users\Stephen Chan\Documents\Visual Studio 2015\Projects\Assembly Activity 1\Project32_VS2015\Debug\Project.exe
Enter integer number N: 20
The fibonacci(20) is: 6765
Elapsed time (ms) = 0
Press any key to continue or 'Q' or 'q' to quit:
Diagnostic Tools
Diagnostics session: 34 secon...
Events
Process Memory (KB)ytes
CPU utilization (% of all...
Watch 1
Name
Value
Type
esi
loop_count
identifier "loop_count" is undefined
ecx
eax
Solution Explorer
Class View
Autos
Locals
Memory 1
Registers
Threads
Modules
Watch 1
Watch 2
Exception Settings
Output
Ready
Ln 110
Col 9
Ch 9
INS
3:37 PM
3/25/2016
C:\Users\Stephen Chan\Documents\Visual Studio 2015\Projects\Assembly Activity 1\Project32_VS2015\Debug\Project.exe
Enter integer number N: 30
The fibonacci(30) is: 832040
Elapsed time (ms) = 15
Press any key to continue or 'Q' or 'q' to quit:
Diagnostics session: 1:14 min...
1:12min
3:38 PM
3/25/2016



Conclusion: I learned the methods used for recursion in assembly as well as comparison operators. Incorporating the stack was also important in order to save values and pop them back into specific registers. Jumps also allows for conditions to be met based upon the values at each register allowing for specific situations to be dealt with. Calling Irvine library procedures also was important for getting specific input from user and calculate the time elapsed.