Marco Martinez
CISP 310
Assignment 2b

Pseudo Code

```
Main Module
Begin
	INCLUDE Irvine32.inc

	Declare fib as array[1…7] of integers
	Declare counter as integer = 0
	Declare fibStart as String = "fib("
	Declare fibEnd as String = ")="
	Declare space as String = " "

	Save eax
	Set eax as 10

	Do
		Save reg
		Set ebx as 0
		Set edx as 1
		Set esi as OFFSET fib
		Set [esi],edx
		Set esi as esi + 4
		Set ecx as 6
		Do
			Set eax as ebx
			Set eax as eax + edx
			Set [esi] as eax
			Set esi as esi + 4
			Set ebx as edx
			Set edx as eax
		While(ecx > 0)
		Load reg

		Save reg
		Set ebx as 0
		Set esi as OFFSET fib
		Set edx as OFFSET fibStart
		Call writeString
		Set eax as ebx
		Set eax as eax + 1
		Call writeInt
		Set edx as OFFSET fibEnd
		Call writeString
		Set eax as [esi]
		Call writeInt
		Set edx as OFFSET space
		Call writeString
		Set ebx as ebx + 1
		Set ecx as 6
		Do
			Set edx as OFFSET fibStart
			Call writeString
			Set eax as ebx
			Set eax as eax + 1
			Call writeInt
			Set edx as OFFSET fibEnd
			Call writeString
			Set eax as [esi+ebx*4]
			Call writeInt
```

```
                    Set edx as OFFSET space
                    Call writeString
                    Set ebx as ebx + 1
                    Set ecx as ecx - 1
                While(ecx<0)
                Load Reg

                Call crlf
                Set eax as eax - 1
        While (eax < 0)
        Load eax

End Main
```

Listing File

Microsoft (R) Macro Assembler Version 14.15.26732.1 12/06/18 09:59:26
..\..\..\..\Documents\School Work\P310\secondProject\ASM 2b\FibonacciNumbers.asm Page 1 - 1

```
                                ;;      Author: Marco Martinez
                                ;;      Program:        FibonacciNumbers.asm
                                ;;      Date:   10/14/2018
                                ;;      Purpose:        Write a program that uses a loop to calculate the first seven
values of the Fibonacci number sequence,
                                ;;                      described by the following formula: Fib(1) = 1, Fib(2) = 1, Fib(n)
= Fib(n – 1) + Fib(n – 2).
                                ;;
                                ;;      Software Change Record
                                ;;      Name            Date            What
                                ;;      Marco   10/14   Baseline for FibonacciNumbers.asm
                                ;;      Marco   12/3            Adjust for reporting

                                INCLUDE Irvine32.inc
                            C ; Include file for Irvine32.lib         (Irvine32.inc)
                            C
                            C ;OPTION CASEMAP:NONE          ; optional: make identifiers case-sensitive
                            C
                            C INCLUDE SmallWin.inc          ; MS-Windows prototypes, structures, and constants
                            C .NOLIST
                            C .LIST
                            C
                            C INCLUDE VirtualKeys.inc
                            C ; VirtualKeys.inc
                            C .NOLIST
                            C .LIST
                            C
                            C
                            C .NOLIST
                            C .LIST
                            C

00000000                        .data
00000000  00000007 [                    fib DWORD 7 DUP(0)
          00000000
```

```
            ]
0000001C 00000000                   counter DWORD 0
00000020 66 69 62 28 00             fibStart BYTE "fib(",0
00000025 29 3D 00                   fibEnd BYTE ")=",0
00000028 20 00                              space BYTE " ",0

00000000                    .code
00000000                    main proc

00000000  B8 0000000A              mov eax,10
00000005                    start:
00000005  60                        pushad
00000006  BB 00000000                       mov ebx,0
0000000B  BA 00000001                       mov edx,1
00000010  BE 00000000 R             mov esi,OFFSET fib
00000015  89 16                     mov [esi],edx
00000017  83 C6 04                  add esi,4
0000001A  B9 00000006                       mov ecx,6
0000001F                            L1:
0000001F  8B C3                             mov eax,ebx
00000021  03 C2                             add eax,edx
00000023  89 06                             mov [esi],eax
00000025  83 C6 04                          add esi,4
00000028  8B DA                             mov ebx,edx
0000002A  8B D0                             mov edx,eax
0000002C  E2 F1                     loop L1
0000002E  61                        popad

0000002F  60                        pushad
00000030  BB 00000000                       mov ebx,0
00000035  BE 00000000 R             mov esi,OFFSET fib
0000003A  BA 00000020 R             mov edx,OFFSET fibStart
0000003F  E8 00000000 E             call writeString
00000044  8B C3                     mov eax,ebx
00000046  83 C0 01                  add eax,1
00000049  E8 00000000 E             call writeInt
0000004E  BA 00000025 R             mov edx,OFFSET fibEnd
00000053  E8 00000000 E             call writeString
00000058  8B 06                     mov eax,[esi]
0000005A  E8 00000000 E             call writeInt
0000005F  BA 00000028 R             mov edx,OFFSET space
00000064  E8 00000000 E             call writeString
00000069  43                        inc ebx
0000006A  B9 00000006                       mov ecx,6
0000006F                            L2:
0000006F  BA 00000020 R                     mov edx,OFFSET fibStart
00000074  E8 00000000 E                     call writeString
00000079  8B C3                             mov eax,ebx
0000007B  83 C0 01                          add eax,1
0000007E  E8 00000000 E                     call writeInt
00000083  BA 00000025 R                     mov edx,OFFSET fibEnd
00000088  E8 00000000 E                     call writeString
0000008D  8B 04 9E                          mov eax,[esi+ebx*4]
```

```
00000090  E8 00000000 E                     call writeInt
00000095  BA 00000028 R                     mov edx,OFFSET space
0000009A  E8 00000000 E                     call writeString
0000009F  43                                inc ebx
000000A0  E2 CD                         loop L2
000000A2  61                            popad

000000A3  E8 00000000 E                 call crlf
000000A8  48                            dec eax
000000A9  83 F8 00                      cmp eax,0
000000AC  0F 8F FFFFFF53        jg start

                                        invoke ExitProcess,0
000000B2  6A 00             *           push   +000000000h
000000B4  E8 00000000 E     *           call   ExitProcess
000000B9                            main endp
                                    end main
```

Structures and Unions:

N a m e	Offset	Size	Type
CONSOLE_CURSOR_INFO		00000008	
dwSize	00000000	DWord	
bVisible	00000004	DWord	
CONSOLE_SCREEN_BUFFER_INFO . . .		00000016	
dwSize	00000000	DWord	
dwCursorPosition	00000004	DWord	
wAttributes	00000008	Word	
srWindow	0000000A	QWord	
dwMaximumWindowSize	00000012	DWord	
COORD	00000004		
X	00000000	Word	
Y	00000002	Word	
FILETIME	00000008		
loDateTime	00000000	DWord	
hiDateTime	00000004	DWord	
FOCUS_EVENT_RECORD		00000004	
bSetFocus	00000000	DWord	
FPU_ENVIRON		0000001C	
controlWord	00000000	Word	
statusWord	00000004	Word	
tagWord	00000008	Word	
instrPointerOffset	0000000C	DWord	
instrPointerSelector	00000010	DWord	
operandPointerOffset	00000014	DWord	
operandPointerSelector	00000018	Word	
INPUT_RECORD		00000014	
EventType	00000000	Word	
Event	00000004	XmmWord	
bKeyDown	00000000	DWord	
wRepeatCount	00000004	Word	
wVirtualKeyCode	00000006	Word	
wVirtualScanCode	00000008	Word	
uChar	0000000A	Word	
UnicodeChar	00000000	Word	
AsciiChar	00000000	Byte	
dwControlKeyState	0000000C	DWord	
dwMousePosition	00000000	DWord	
dwButtonState	00000004	DWord	
dwMouseControlKeyState	00000008	DWord	
dwEventFlags	0000000C	DWord	
dwSize	00000000	DWord	
dwCommandId	00000000	DWord	

```
  bSetFocus . . . . . 00000000       DWord
KEY_EVENT_RECORD . . . . . .  00000010
  bKeyDown . . . . . .        00000000       DWord
  wRepeatCount . . . . . .    00000004       Word
  wVirtualKeyCode . . . . .   00000006       Word
  wVirtualScanCode . . . . .  00000008       Word
  uChar . . . . . .  0000000A       Word
  UnicodeChar . . . . . .     00000000       Word
  AsciiChar . . . . . .  00000000       Byte
  dwControlKeyState . . . . . 0000000C       DWord
MENU_EVENT_RECORD . . . . . .       00000004
  dwCommandId . . . . . .     00000000       DWord
MOUSE_EVENT_RECORD . . . . . .      00000010
  dwMousePosition . . . . . .  00000000       DWord
  dwButtonState . . . . . .    00000004       DWord
  dwMouseControlKeyState . . . . 00000008       DWord
  dwEventFlags . . . . . .     0000000C       DWord
SMALL_RECT . . . . . .      00000008
  Left . . . . . .  00000000       Word
  Top . . . . . .  00000002       Word
  Right . . . . . .  00000004       Word
  Bottom . . . . . .  00000006       Word
SYSTEMTIME . . . . . .      00000010
  wYear . . . . . .  00000000       Word
  wMonth . . . . . .  00000002       Word
  wDayOfWeek . . . . . .      00000004       Word
  wDay . . . . . .  00000006       Word
  wHour . . . . . .  00000008       Word
  wMinute . . . . . .  0000000A       Word
  wSecond . . . . . .  0000000C       Word
  wMilliseconds . . . . . .   0000000E       Word
WINDOW_BUFFER_SIZE_RECORD . . .  00000004
  dwSize . . . . . .  00000000       DWord


Segments and Groups:

                N a m e         Size     Length   Align   Combine Class

FLAT . . . . . .  GROUP
STACK . . . . . .  32 Bit    00001000 Para   Stack    'STACK'
_DATA . . . . . .  32 Bit    0000002A Para   Public  'DATA'
_TEXT . . . . . .  32 Bit    000000B9 Para   Public  'CODE'


Procedures, parameters, and locals:

                N a m e         Type     Value    Attr

CloseFile . . . . . .  P Near   00000000 FLAT Length= 00000000 External STDCALL
CloseHandle . . . . . .       P Near   00000000 FLAT Length= 00000000 External STDCALL
Clrscr . . . . . .  P Near   00000000 FLAT Length= 00000000 External STDCALL
CreateFileA . . . . . .P Near   00000000 FLAT Length= 00000000 External STDCALL
```

```
CreateOutputFile . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
Crlf . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
Delay . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
DumpMem . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
DumpRegs . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
ExitProcess . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
FileTimeToDosDateTime . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
FileTimeToSystemTime . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
FlushConsoleInputBuffer . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
FormatMessageA . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
GetCommandLineA . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
GetCommandTail . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
GetConsoleCP . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
GetConsoleCursorInfo . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
GetConsoleMode . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
GetConsoleScreenBufferInfo . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
GetDateTime . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
GetFileTime . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
GetKeyState . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
GetLastError . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
GetLocalTime . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
GetMaxXY . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
GetMseconds . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
GetNumberOfConsoleInputEvents .  P Near   00000000 FLAT Length= 00000000 External STDCALL
GetProcessHeap . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
GetStdHandle . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
GetSystemTime . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
GetTextColor . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
GetTickCount . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
Gotoxy . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
HeapAlloc . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
HeapCreate . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
HeapDestroy . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
HeapFree . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
HeapSize . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
IsDigit . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
LocalFree . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
MessageBoxA . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
MsgBoxAsk . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
MsgBox . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
OpenInputFile . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
ParseDecimal32 . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
ParseInteger32 . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
PeekConsoleInputA . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
Random32 . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
RandomRange . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
Randomize . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
ReadChar . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
ReadConsoleA . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
ReadConsoleInputA . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
ReadDec . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
ReadFile . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
ReadFloat . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
```

```
ReadFromFile . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
ReadHex . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
ReadInt . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
ReadKeyFlush . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
ReadKey . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
ReadString . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
SetConsoleCursorInfo . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
SetConsoleCursorPosition . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
SetConsoleMode . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
SetConsoleScreenBufferSize . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
SetConsoleTextAttribute . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
SetConsoleTitleA . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
SetConsoleWindowInfo . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
SetFilePointer . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
SetLocalTime . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
SetTextColor . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
ShowFPUStack . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
Sleep . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
StrLength . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
Str_compare . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
Str_copy . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
Str_length . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
Str_trim . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
Str_ucase . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
SystemTimeToFileTime . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
WaitMsg . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
WriteBinB . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
WriteBin . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
WriteChar . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
WriteConsoleA . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
WriteConsoleOutputAttribute . . P Near   00000000 FLAT Length= 00000000 External STDCALL
WriteConsoleOutputCharacterA . . P Near   00000000 FLAT Length= 00000000 External STDCALL
WriteDec . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
WriteFile . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
WriteFloat . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
WriteHexB . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
WriteHex . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
WriteInt . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
WriteStackFrameName . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
WriteStackFrame . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
WriteString . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
WriteToFile . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
WriteWindowsMsg . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
main . . . . . P Near   00000000 _TEXT          Length= 000000B9 Public STDCALL
  start . . . . . L Near   00000005 _TEXT
  L1 . . . . . L Near   0000001F _TEXT
  L2 . . . . . L Near   0000006F _TEXT
printf . . . . . P Near   00000000 FLAT Length= 00000000 External C
scanf . . . . . P Near   00000000 FLAT Length= 00000000 External C
wsprintfA . . . . . P Near   00000000 FLAT Length= 00000000 External C
```

Symbols:

```
          N a m e            Type    Value    Attr

@CodeSize . . . . . .          Number          00000000h
@DataSize . . . . . .          Number          00000000h
@Interface . . . . . .         Number          00000003h
@Model . . . . . .Number          00000007h
@code . . . . . .  Text      _TEXT
@data . . . . . .  Text      FLAT
@fardata? . . . . . .Text      FLAT
@fardata . . . . . .  Text      FLAT
@stack . . . . . .  Text      FLAT
ALT_MASK . . . . . .           Number          00000003h
CAPSLOCK_ON . . . . . .        Number          00000080h
CREATE_ALWAYS . . . . . .      Number          00000002h
CREATE_NEW . . . . . .         Number          00000001h
CTRL_MASK . . . . . .          Number          0000000Ch
CreateFile . . . . . .  Text      CreateFileA
DO_NOT_SHARE . . . . . .       Number          00000000h
ENABLE_ECHO_INPUT . . . . . .  Number          00000004h
ENABLE_LINE_INPUT . . . . . .  Number          00000002h
ENABLE_MOUSE_INPUT . . . . . .         Number          00000010h
ENABLE_PROCESSED_INPUT . . . . .       Number          00000001h
ENABLE_PROCESSED_OUTPUT . . . .        Number          00000001h
ENABLE_WINDOW_INPUT . . . . . .        Number          00000008h
ENABLE_WRAP_AT_EOL_OUTPUT . . .  Number          00000002h
ENHANCED_KEY . . . . . .       Number          00000100h
FALSE . . . . . .   Number          00000000h
FILE_APPEND_DATA . . . . . .   Number          00000004h
FILE_ATTRIBUTE_ARCHIVE . . . . .       Number          00000020h
FILE_ATTRIBUTE_COMPRESSED . . .        Number          00000800h
FILE_ATTRIBUTE_DEVICE . . . . . Number          00000040h
FILE_ATTRIBUTE_DIRECTORY . . . .       Number          00000010h
FILE_ATTRIBUTE_ENCRYPTED . . . .       Number          00004000h
FILE_ATTRIBUTE_HIDDEN . . . . .Number          00000002h
FILE_ATTRIBUTE_NORMAL . . . . .        Number          00000080h
FILE_ATTRIBUTE_NOT_CONTENT_INDEXED .      Number          00002000h
FILE_ATTRIBUTE_OFFLINE . . . . . Number          00001000h
FILE_ATTRIBUTE_READONLY . . . .        Number          00000001h
FILE_ATTRIBUTE_REPARSE_POINT . .       Number          00000400h
FILE_ATTRIBUTE_SPARSE_FILE . . .       Number          00000200h
FILE_ATTRIBUTE_SYSTEM . . . . .Number          00000004h
FILE_ATTRIBUTE_TEMPORARY . . . .       Number          00000100h
FILE_BEGIN . . . . . .         Number          00000000h
FILE_CURRENT . . . . . .       Number          00000001h
FILE_DELETE_CHILD . . . . . .  Number          00000040h
FILE_END . . . . . . Number          00000002h
FILE_READ_DATA . . . . . .     Number          00000001h
FILE_SHARE_DELETE . . . . . .  Number          00000004h
FILE_SHARE_READ . . . . . .    Number          00000001h
FILE_SHARE_WRITE . . . . . .   Number          00000002h
FILE_WRITE_DATA . . . . . .    Number          00000002h
FOCUS_EVENT . . . . . .        Number          00000010h
```

FORMAT_MESSAGE_ALLOCATE_BUFFER . Number 00000100h
FORMAT_MESSAGE_FROM_SYSTEM . . . Number 00001000h
FormatMessage . . . . . Text FormatMessageA
GENERIC_ALL . . . . . Number 10000000h
GENERIC_EXECUTE . . . . . Number 20000000h
GENERIC_READ . . . . . Number -80000000h
GENERIC_WRITE . . . . . Number 40000000h
GetCommandLine . . . . . Text GetCommandLineA
HANDLE . . . . . Text DWORD
HEAP_GENERATE_EXCEPTIONS . . . . Number 00000004h
HEAP_GROWABLE . . . . . Number 00000002h
HEAP_NO_SERIALIZE . . . . . Number 00000001h
HEAP_REALLOC_IN_PLACE_ONLY . . . Number 00000010h
HEAP_ZERO_MEMORY . . . . . .Number 00000008h
IDABORT . . . . . .Number 00000003h
IDCANCEL . . . . . .Number 00000002h
IDCLOSE . . . . . Number 00000008h
IDCONTINUE . . . . . Number 0000000Bh
IDHELP . . . . . Number 00000009h
IDIGNORE . . . . . Number 00000005h
IDNO . . . . . Number 00000007h
IDOK . . . . . Number 00000001h
IDRETRY . . . . . Number 00000004h
IDTIMEOUT . . . . . Number 00007D00h
IDTRYAGAIN . . . . . Number 0000000Ah
IDYES . . . . . Number 00000006h
INVALID_HANDLE_VALUE . . . . . Number -00000001h
KBDOWN_FLAG . . . . . Number 00000001h
KEY_EVENT . . . . . Number 00000001h
KEY_MASKS . . . . . Number 0000001Fh
LEFT_ALT_PRESSED . . . . . Number 00000002h
LEFT_CTRL_PRESSED . . . . . Number 00000008h
MB_ABORTRETRYIGNORE . . . . . Number 00000002h
MB_APPLMODAL . . . . . Number 00000000h
MB_CANCELTRYCONTINUE . . . . . Number 00000006h
MB_DEFBUTTON1 . . . . . Number 00000000h
MB_DEFBUTTON2 . . . . . Number 00000100h
MB_DEFBUTTON3 . . . . . Number 00000200h
MB_DEFBUTTON4 . . . . . Number 00000300h
MB_HELP . . . . . Number 00004000h
MB_ICONASTERISK . . . . . Number 00000040h
MB_ICONERROR . . . . . Number 00000010h
MB_ICONEXCLAMATION . . . . . Number 00000030h
MB_ICONHAND . . . . . Number 00000010h
MB_ICONINFORMATION . . . . . Number 00000040h
MB_ICONQUESTION . . . . . Number 00000020h
MB_ICONSTOP . . . . . Number 00000010h
MB_ICONWARNING . . . . . Number 00000030h
MB_OKCANCEL . . . . . Number 00000001h
MB_OK . . . . . Number 00000000h
MB_RETRYCANCEL . . . . . Number 00000005h
MB_SYSTEMMODAL . . . . . Number 00001000h
MB_TASKMODAL . . . . . Number 00002000h

MB_USERICON . . . . . Number 00000080h
MB_YESNOCANCEL . . . . . Number 00000003h
MB_YESNO . . . . . Number 00000004h
MENU_EVENT . . . . . Number 00000008h
MOUSE_EVENT . . . . . Number 00000002h
MessageBox . . . . . Text MessageBoxA
NULL . . . . . Number 00000000h
NUMLOCK_ON . . . . . Number 00000020h
OPEN_ALWAYS . . . . . Number 00000004h
OPEN_EXISTING . . . . . Number 00000003h
PeekConsoleInput . . . . . Text PeekConsoleInputA
RIGHT_ALT_PRESSED . . . . . Number 00000001h
RIGHT_CTRL_PRESSED . . . . . Number 00000004h
ReadConsoleInput . . . . . Text ReadConsoleInputA
ReadConsole . . . . . Text ReadConsoleA
SCROLLLOCK_ON . . . . . Number 00000040h
SHIFT_MASK . . . . . Number 00000010h
SHIFT_PRESSED . . . . . Number 00000010h
STD_ERROR_HANDLE . . . . . Number -0000000Ch
STD_INPUT_HANDLE . . . . . Number -0000000Ah
STD_OUTPUT_HANDLE . . . . . Number -0000000Bh
SetConsoleTitle . . . . . Text SetConsoleTitleA
TAB . . . . . Number 00000009h
TRUE . . . . . Number 00000001h
TRUNCATE_EXISTING . . . . . Number 00000005h
VK_11 . . . . . Number 000000BDh
VK_12 . . . . . Number 000000BBh
VK_ADD . . . . . Number 0000006Bh
VK_BACK . . . . . Number 00000008h
VK_CANCEL . . . . . Number 00000003h
VK_CAPITAL . . . . . Number 00000014h
VK_CLEAR . . . . . Number 0000000Ch
VK_CONTROL . . . . . Number 00000011h
VK_DECIMAL . . . . . Number 0000006Eh
VK_DELETE . . . . . Number 0000002Eh
VK_DIVIDE . . . . . Number 0000006Fh
VK_DOWN . . . . . Number 00000028h
VK_END . . . . . Number 00000023h
VK_ESCAPE . . . . . Number 0000001Bh
VK_EXECUTE . . . . . Number 0000002Bh
VK_F10 . . . . . Number 00000079h
VK_F11 . . . . . Number 0000007Ah
VK_F12 . . . . . Number 0000007Bh
VK_F13 . . . . . Number 0000007Ch
VK_F14 . . . . . Number 0000007Dh
VK_F15 . . . . . Number 0000007Eh
VK_F16 . . . . . Number 0000007Fh
VK_F17 . . . . . Number 00000080h
VK_F18 . . . . . Number 00000081h
VK_F19 . . . . . Number 00000082h
VK_F1 . . . . . Number 00000070h
VK_F20 . . . . . Number 00000083h
VK_F21 . . . . . Number 00000084h

VK_F22 . . . . . Number 00000085h
VK_F23 . . . . . Number 00000086h
VK_F24 . . . . . Number 00000087h
VK_F2 . . . . . Number 00000071h
VK_F3 . . . . . Number 00000072h
VK_F4 . . . . . Number 00000073h
VK_F5 . . . . . Number 00000074h
VK_F6 . . . . . Number 00000075h
VK_F7 . . . . . Number 00000076h
VK_F8 . . . . . Number 00000077h
VK_F9 . . . . . Number 00000078h
VK_HELP . . . . . Number 0000002Fh
VK_HOME . . . . . Number 00000024h
VK_INSERT . . . . . Number 0000002Dh
VK_LBUTTON . . . . . Number 00000001h
VK_LCONTROL . . . . . Number 000000A2h
VK_LEFT . . . . . Number 00000025h
VK_LMENU . . . . . Number 000000A4h
VK_LSHIFT . . . . . Number 000000A0h
VK_MENU . . . . . Number 00000012h
VK_MULTIPLY . . . . . Number 0000006Ah
VK_NEXT . . . . . Number 00000022h
VK_NUMLOCK . . . . . Number 00000090h
VK_NUMPAD0 . . . . . Number 00000060h
VK_NUMPAD1 . . . . . Number 00000061h
VK_NUMPAD2 . . . . . Number 00000062h
VK_NUMPAD3 . . . . . Number 00000063h
VK_NUMPAD4 . . . . . Number 00000064h
VK_NUMPAD5 . . . . . Number 00000065h
VK_NUMPAD6 . . . . . Number 00000066h
VK_NUMPAD7 . . . . . Number 00000067h
VK_NUMPAD8 . . . . . Number 00000068h
VK_NUMPAD9 . . . . . Number 00000069h
VK_PAUSE . . . . . Number 00000013h
VK_PRINT . . . . . Number 0000002Ah
VK_PRIOR . . . . . Number 00000021h
VK_RBUTTON . . . . . Number 00000002h
VK_RCONTROL . . . . . Number 000000A3h
VK_RETURN . . . . . Number 0000000Dh
VK_RIGHT . . . . . Number 00000027h
VK_RMENU . . . . . Number 000000A5h
VK_RSHIFT . . . . . Number 000000A1h
VK_SCROLL . . . . . Number 00000091h
VK_SEPARATER . . . . . Number 0000006Ch
VK_SHIFT . . . . . Number 00000010h
VK_SNAPSHOT . . . . . Number 0000002Ch
VK_SPACE . . . . . Number 00000020h
VK_SUBTRACT . . . . . Number 0000006Dh
VK_TAB . . . . . Number 00000009h
VK_UP . . . . . Number 00000026h
WINDOW_BUFFER_SIZE_EVENT . . . . Number 00000004h
WriteConsoleOutputCharacter . . Text WriteConsoleOutputCharacterA
WriteConsole . . . . . Text WriteConsoleA

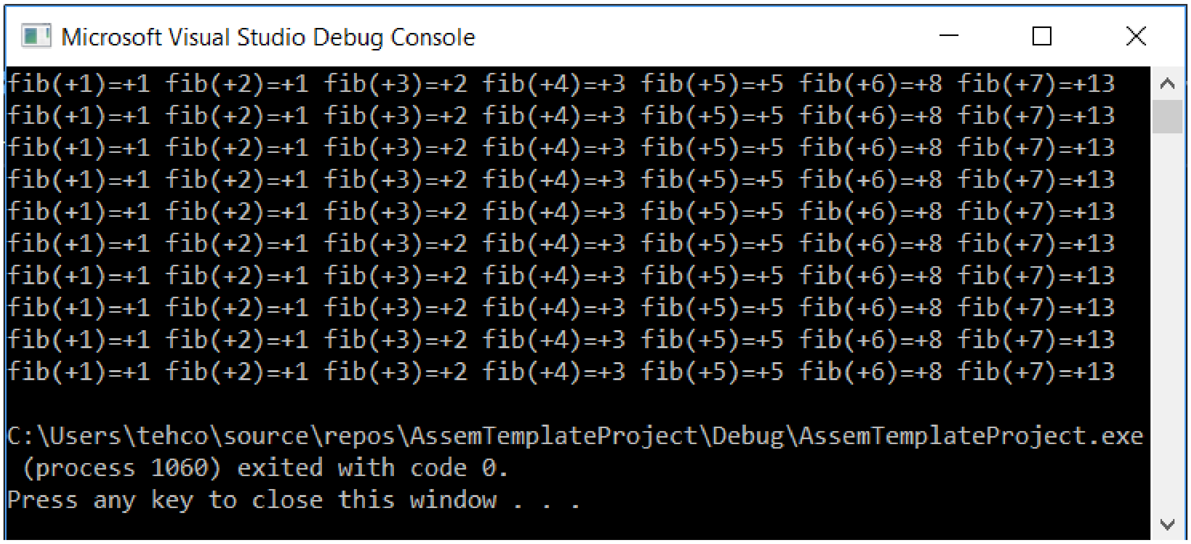
```
black . . . . . .  Number          00000000h
blue . . . . . .   Number          00000001h
brown . . . . . .  Number          00000006h
counter . . . . . . DWord  0000001C _DATA
cyan . . . . . .   Number          00000003h
exit . . . . . .   Text      INVOKE ExitProcess,0
fibEnd . . . . . . Byte    00000025 _DATA
fibStart . . . . . . Byte  00000020 _DATA
fib . . . . . .    DWord  00000000 _DATA
gray . . . . . .   Number          00000008h
green . . . . . .  Number          00000002h
lightBlue . . . . . . Number       00000009h
lightCyan . . . . . . Number       0000000Bh
lightGray . . . . . . Number       00000007h
lightGreen . . . . . . Number      0000000Ah
lightMagenta . . . . . .        Number          0000000Dh
lightRed . . . . . .  Number       0000000Ch
magenta . . . . . .  Number        00000005h
red . . . . . .    Number          00000004h
space . . . . . .  Byte    00000028 _DATA
white . . . . . .  Number          0000000Fh
wsprintf . . . . . .  Text     wsprintfA
yellow . . . . . .  Number         0000000Eh

          0 Warnings
          0 Errors
```

Source Code

```
;;     Author:       Marco Martinez
;;     Program:      FibonacciNumbers.asm
;;     Date:  10/14/2018
;;     Purpose:      Write a program that uses a loop to calculate the first seven values of the
Fibonacci number sequence,
;;                   described by the following formula: Fib(1) = 1, Fib(2) = 1, Fib(n) = Fib(n – 1) +
Fib(n – 2).
;;
;;     Software Change Record
;;     Name          Date          What
;;     Marco  10/14  Baseline for FibonacciNumbers.asm
;;     Marco  12/3          Adjust for reporting

INCLUDE Irvine32.inc

.data
       fib DWORD 7 DUP(0)
       counter DWORD 0
       fibStart BYTE "fib(",0
       fibEnd BYTE ")=",0
       space BYTE " ",0

.code
main proc

       push eax
       mov eax,10
       start:
              pushad
```

```
		mov ebx,0
		mov edx,1
		mov esi,OFFSET fib
		mov [esi],edx
		add esi,4
		mov ecx,6
		L1:
			mov eax,ebx
			add eax,edx
			mov [esi],eax
			add esi,4
			mov ebx,edx
			mov edx,eax
		loop L1
		popad

		pushad
		mov ebx,0
		mov esi,OFFSET fib
		mov edx,OFFSET fibStart
		call writeString
		mov eax,ebx
		add eax,1
		call writeInt
		mov edx,OFFSET fibEnd
		call writeString
		mov eax,[esi]
		call writeInt
		mov edx,OFFSET space
		call writeString
		inc ebx
		mov ecx,6
		L2:
			mov edx,OFFSET fibStart
			call writeString
			mov eax,ebx
			add eax,1
			call writeInt
			mov edx,OFFSET fibEnd
			call writeString
			mov eax,[esi+ebx*4]
			call writeInt
			mov edx,OFFSET space
			call writeString
			inc ebx
		loop L2
		popad

		call crlf
		dec eax
		cmp eax,0
	jg start
	pop eax

	invoke ExitProcess,0
main endp
end main
```

Screenshot



Pseudo Code

Main Module

Begin Main

```
	INCLUDE Irvine32.inc

	Declare array as array = {10,20,30,40}
	Declare counter as integer = 1
	Declare space as string = " "
	Declare msgOriginal as string = "Original array: "
	Declare msgShifted as string = "Shifted array: "

	Set ecx as 0
	Do
		Save reg
		Set esi as OFFSET array
		Set ecx as LENGTHOF array
		Set edx as OFFSET msgOriginal
		Call writeString
		Set edx as OFFSET space
		Do
			Set eax as [esi]
			Call writeInt
			Call writeString
			Set esi as esi + 4
			Set ecx as ecx - 1
		While (ecx > 0)
		Call crlf
		Load reg

		Save reg
		Set esi as OFFSET array + 3 * 4
		Set eax as [esi]
		Set esi as OFFSET array
		Set ebx as [esi]
		Set [esi] as eax
		Set eax as ebx
		Set esi as OFFSET array+1*4
		Set ecx as LENGTHOF array-1
		Do
			Set ebx as [esi]
```

```
                Set [esi] as eax
                Set eax as ebx
                Set edx as edx + 1
                Set counter as edx
                Set esi as esi + 4
                Set ecx as ecx - 1
        While (ecx > 0)
        Load reg

        Save reg
        Set esi as OFFSET array
        Set ecx as LENGTHOF array
        Set edx as OFFSET msgShifted
        Call writeString
        Set edx as OFFSET space
        Call writeString
        Do
                Set eax as [esi]
                Call writeInt
                Call writeString
                Set esi as esi + 4
                Set ecx as ecx + 1
        While (ecx > 0)
        Call crlf
        Load reg

        Call crlf
        Set ecx as ecx + 1
    While (ecx > 0)
End Main
```

Listing File

Microsoft (R) Macro Assembler Version 14.15.26732.1 12/06/18 12:01:52
..\..\..\..\Documents\School Work\P310\secondProject\ASM 2b\swapElements.asm Page 1 - 1

```
                        ;;      Author: Marco Martinez
                        ;;      Program:        ShiftElements.asm
                        ;;      Date:           10/14/2018
                        ;;      Purpose:        Using a loop and indexed addressing, write code that rotates
                        ;;                      the members of a 32-bit integer array forward one position. The
                        ;;                      value at the end of the array must wrap around to the first
                        ;;                      position.
                        ;;                      For example, the array [10,20,30,40] would be transformed into
                        ;;
                        ;;                      [40,10,20,30].
                        ;;
                        ;;      Software Change Record
                        ;;      Name    Date            What
                        ;;      Marco   10/14           Baseline for ShiftElements.asm
                        ;;      Marco   12/6            Implement reporting features.
                        ;;

                            INCLUDE Irvine32.inc
                        C ; Include file for Irvine32.lib       (Irvine32.inc)
                        C
                        C ;OPTION CASEMAP:NONE          ; optional: make identifiers case-sensitive
```

```
                        C
                        C INCLUDE SmallWin.inc          ; MS-Windows prototypes, structures, and constants
                        C .NOLIST
                        C .LIST
                        C
                        C INCLUDE VirtualKeys.inc
                        C ; VirtualKeys.inc
                        C .NOLIST
                        C .LIST
                        C
                        C
                        C .NOLIST
                        C .LIST
                        C

00000000                .data
00000000 0000000A               array DWORD 10,20,30,40
         00000014
         0000001E
         00000028
00000010 00000001               counter DWORD 1
00000014 20 00                      space BYTE " ",0
00000016 4F 72 69 67 69         msgOriginal BYTE "Original array: ",0
         6E 61 6C 20 61
         72 72 61 79 3A
         20 00
00000027 53 68 69 66 74         msgShifted BYTE "Shifted array: ",0
         65 64 20 61 72
         72 61 79 3A 20
         00

00000000                .code
00000000                main proc
00000000  B9 00000000               mov ecx,0

00000005                        program:
00000005  60                        pushad
00000006  BE 00000000 R             mov esi,OFFSET array
0000000B  B9 00000004                   mov ecx,LENGTHOF array
00000010  BA 00000016 R             mov edx,OFFSET msgOriginal
00000015  E8 00000000 E             call writeString
0000001A  BA 00000014 R             mov edx,OFFSET space
0000001F                            L1:
0000001F  8B 06                         mov eax,[esi]
00000021  E8 00000000 E                 call writeInt
00000026  E8 00000000 E                 call writeString
0000002B  83 C6 04                      add esi,4
0000002E  E2 EF                     loop L1
00000030  E8 00000000 E             call crlf
00000035  61                        popad

00000036  60                        pushad
00000037  BE 0000000C R             mov esi,OFFSET array+3*4
```

```
0000003C  8B 06                                   mov eax,[esi]
0000003E  BE 00000000 R                           mov esi,OFFSET array
00000043  8B 1E                                   mov ebx,[esi]
00000045  89 06                                   mov [esi],eax
00000047  8B C3                                   mov eax,ebx
00000049  BE 00000004 R                           mov esi,OFFSET array+1*4
0000004E  B9 00000003                                     mov ecx,LENGTHOF array-1
00000053                                          L2:
00000053  8B 1E                                           mov ebx,[esi]
00000055  89 06                                           mov [esi],eax
00000057  8B C3                                           mov eax,ebx
00000059  42                                              inc edx
0000005A  89 15 00000010 R                                mov counter,edx
00000060  83 C6 04                                        add esi,4
00000063  E2 EE                                   loop L2
00000065  61                                      popad

00000066  60                                      pushad
00000067  BE 00000000 R                           mov esi,OFFSET array
0000006C  B9 00000004                                     mov ecx,LENGTHOF array
00000071  BA 00000027 R                           mov edx,OFFSET msgShifted
00000076  E8 00000000 E                           call writeString
0000007B  BA 00000014 R                           mov edx,OFFSET space
00000080  E8 00000000 E                           call writeString
00000085                                          L3:
00000085  8B 06                                           mov eax,[esi]
00000087  E8 00000000 E                                   call writeInt
0000008C  E8 00000000 E                                   call writeString
00000091  83 C6 04                                        add esi,4
00000094  E2 EF                                   loop L3
00000096  E8 00000000 E                           call crlf
0000009B  61                                      popad

0000009C  E8 00000000 E                           call crlf
000000A1  41                                      inc ecx
000000A2  83 F9 04                                cmp ecx,4
000000A5  0F 8C FFFFFF5A                  jl program

                                                  invoke ExitProcess,0
000000AB  6A 00           *               push   +000000000h
000000AD  E8 00000000 E   *               call   ExitProcess
000000B2                                  main endp
                                          end main
```

Structures and Unions:

N a m e	Offset	Size	Type
CONSOLE_CURSOR_INFO		00000008	
dwSize	00000000	DWord	
bVisible	00000004	DWord	
CONSOLE_SCREEN_BUFFER_INFO . . .		00000016	
dwSize	00000000	DWord	
dwCursorPosition	00000004	DWord	
wAttributes	00000008	Word	
srWindow	0000000A	QWord	
dwMaximumWindowSize	00000012	DWord	
COORD	00000004		
X	00000000	Word	
Y	00000002	Word	
FILETIME	00000008		
loDateTime	00000000	DWord	
hiDateTime	00000004	DWord	
FOCUS_EVENT_RECORD		00000004	
bSetFocus	00000000	DWord	
FPU_ENVIRON		0000001C	
controlWord	00000000	Word	
statusWord	00000004	Word	
tagWord	00000008	Word	
instrPointerOffset	0000000C	DWord	
instrPointerSelector	00000010	DWord	
operandPointerOffset	00000014	DWord	
operandPointerSelector	00000018	Word	
INPUT_RECORD		00000014	
EventType	00000000	Word	
Event	00000004	XmmWord	
bKeyDown	00000000	DWord	
wRepeatCount	00000004	Word	
wVirtualKeyCode	00000006	Word	
wVirtualScanCode	00000008	Word	
uChar	0000000A	Word	
UnicodeChar	00000000	Word	
AsciiChar	00000000	Byte	
dwControlKeyState	0000000C	DWord	
dwMousePosition	00000000	DWord	
dwButtonState	00000004	DWord	
dwMouseControlKeyState	00000008	DWord	
dwEventFlags	0000000C	DWord	
dwSize	00000000	DWord	
dwCommandId	00000000	DWord	

```
  bSetFocus . . . . . 00000000       DWord
KEY_EVENT_RECORD . . . . . .  00000010
  bKeyDown . . . . . .        00000000       DWord
  wRepeatCount . . . . . .    00000004       Word
  wVirtualKeyCode . . . . .   00000006       Word
  wVirtualScanCode . . . . .  00000008       Word
  uChar . . . . . .  0000000A       Word
  UnicodeChar . . . . . .     00000000       Word
  AsciiChar . . . . . .  00000000       Byte
  dwControlKeyState . . . . . 0000000C       DWord
MENU_EVENT_RECORD . . . . . .       00000004
  dwCommandId . . . . . .     00000000       DWord
MOUSE_EVENT_RECORD . . . . . .      00000010
  dwMousePosition . . . . . .  00000000       DWord
  dwButtonState . . . . . .    00000004       DWord
  dwMouseControlKeyState . . . . 00000008       DWord
  dwEventFlags . . . . . .     0000000C       DWord
SMALL_RECT . . . . . .      00000008
  Left . . . . . .  00000000       Word
  Top . . . . . .  00000002       Word
  Right . . . . . .  00000004       Word
  Bottom . . . . . .  00000006       Word
SYSTEMTIME . . . . . .      00000010
  wYear . . . . . .  00000000       Word
  wMonth . . . . . .  00000002       Word
  wDayOfWeek . . . . . .      00000004       Word
  wDay . . . . . .  00000006       Word
  wHour . . . . . .  00000008       Word
  wMinute . . . . . .  0000000A       Word
  wSecond . . . . . .  0000000C       Word
  wMilliseconds . . . . . .   0000000E       Word
WINDOW_BUFFER_SIZE_RECORD . . .  00000004
  dwSize . . . . . .  00000000       DWord


Segments and Groups:

                N a m e         Size     Length   Align   Combine Class

FLAT . . . . . .  GROUP
STACK . . . . . .  32 Bit    00001000 Para   Stack    'STACK'
_DATA . . . . . .  32 Bit    00000037 Para   Public  'DATA'
_TEXT . . . . . .  32 Bit    000000B2 Para   Public  'CODE'


Procedures, parameters, and locals:

                N a m e         Type     Value    Attr

CloseFile . . . . . .  P Near   00000000 FLAT Length= 00000000 External STDCALL
CloseHandle . . . . . .       P Near   00000000 FLAT Length= 00000000 External STDCALL
Clrscr . . . . . .  P Near   00000000 FLAT Length= 00000000 External STDCALL
CreateFileA . . . . . .P Near   00000000 FLAT Length= 00000000 External STDCALL
```

CreateOutputFile . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL
Crlf . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL
Delay . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL
DumpMem . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL
DumpRegs . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL
ExitProcess . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL
FileTimeToDosDateTime . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL
FileTimeToSystemTime . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL
FlushConsoleInputBuffer . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL
FormatMessageA . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL
GetCommandLineA . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL
GetCommandTail . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL
GetConsoleCP . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL
GetConsoleCursorInfo . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL
GetConsoleMode . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL
GetConsoleScreenBufferInfo . . . P Near 00000000 FLAT Length= 00000000 External STDCALL
GetDateTime . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL
GetFileTime . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL
GetKeyState . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL
GetLastError . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL
GetLocalTime . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL
GetMaxXY . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL
GetMseconds . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL
GetNumberOfConsoleInputEvents . P Near 00000000 FLAT Length= 00000000 External STDCALL
GetProcessHeap . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL
GetStdHandle . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL
GetSystemTime . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL
GetTextColor . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL
GetTickCount . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL
Gotoxy . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL
HeapAlloc . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL
HeapCreate . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL
HeapDestroy . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL
HeapFree . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL
HeapSize . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL
IsDigit . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL
LocalFree . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL
MessageBoxA . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL
MsgBoxAsk . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL
MsgBox . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL
OpenInputFile . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL
ParseDecimal32 . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL
ParseInteger32 . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL
PeekConsoleInputA . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL
Random32 . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL
RandomRange . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL
Randomize . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL
ReadChar . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL
ReadConsoleA . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL
ReadConsoleInputA . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL
ReadDec . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL
ReadFile . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL
ReadFloat . . . . . P Near 00000000 FLAT Length= 00000000 External STDCALL

```
ReadFromFile . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
ReadHex . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
ReadInt . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
ReadKeyFlush . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
ReadKey . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
ReadString . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
SetConsoleCursorInfo . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
SetConsoleCursorPosition . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
SetConsoleMode . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
SetConsoleScreenBufferSize . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
SetConsoleTextAttribute . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
SetConsoleTitleA . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
SetConsoleWindowInfo . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
SetFilePointer . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
SetLocalTime . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
SetTextColor . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
ShowFPUStack . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
Sleep . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
StrLength . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
Str_compare . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
Str_copy . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
Str_length . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
Str_trim . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
Str_ucase . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
SystemTimeToFileTime . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
WaitMsg . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
WriteBinB . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
WriteBin . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
WriteChar . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
WriteConsoleA . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
WriteConsoleOutputAttribute . . P Near   00000000 FLAT Length= 00000000 External STDCALL
WriteConsoleOutputCharacterA . . P Near   00000000 FLAT Length= 00000000 External STDCALL
WriteDec . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
WriteFile . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
WriteFloat . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
WriteHexB . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
WriteHex . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
WriteInt . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
WriteStackFrameName . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
WriteStackFrame . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
WriteString . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
WriteToFile . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
WriteWindowsMsg . . . . . P Near   00000000 FLAT Length= 00000000 External STDCALL
main . . . . . P Near   00000000 _TEXT          Length= 000000B2 Public STDCALL
  program . . . . . L Near   00000005 _TEXT
  L1 . . . . . L Near   0000001F _TEXT
  L2 . . . . . L Near   00000053 _TEXT
  L3 . . . . . L Near   00000085 _TEXT
printf . . . . . P Near   00000000 FLAT Length= 00000000 External C
scanf . . . . . P Near   00000000 FLAT Length= 00000000 External C
wsprintfA . . . . . P Near   00000000 FLAT Length= 00000000 External C
```

Symbols:

N a m e	Type	Value	Attr
@CodeSize	Number	00000000h	
@DataSize	Number	00000000h	
@Interface	Number	00000003h	
@Model	Number	00000007h	
@code	Text	_TEXT	
@data	Text	FLAT	
@fardata?	Text	FLAT	
@fardata	Text	FLAT	
@stack	Text	FLAT	
ALT_MASK	Number	00000003h	
CAPSLOCK_ON	Number	00000080h	
CREATE_ALWAYS	Number	00000002h	
CREATE_NEW	Number	00000001h	
CTRL_MASK	Number	0000000Ch	
CreateFile	Text	CreateFileA	
DO_NOT_SHARE	Number	00000000h	
ENABLE_ECHO_INPUT	Number	00000004h	
ENABLE_LINE_INPUT	Number	00000002h	
ENABLE_MOUSE_INPUT	Number	00000010h	
ENABLE_PROCESSED_INPUT	Number	00000001h	
ENABLE_PROCESSED_OUTPUT	Number	00000001h	
ENABLE_WINDOW_INPUT	Number	00000008h	
ENABLE_WRAP_AT_EOL_OUTPUT . . .	Number	00000002h	
ENHANCED_KEY	Number	00000100h	
FALSE	Number	00000000h	
FILE_APPEND_DATA	Number	00000004h	
FILE_ATTRIBUTE_ARCHIVE	Number	00000020h	
FILE_ATTRIBUTE_COMPRESSED . . .	Number	00000800h	
FILE_ATTRIBUTE_DEVICE	Number	00000040h	
FILE_ATTRIBUTE_DIRECTORY	Number	00000010h	
FILE_ATTRIBUTE_ENCRYPTED	Number	00004000h	
FILE_ATTRIBUTE_HIDDEN	Number	00000002h	
FILE_ATTRIBUTE_NORMAL	Number	00000080h	
FILE_ATTRIBUTE_NOT_CONTENT_INDEXED .	Number	00002000h	
FILE_ATTRIBUTE_OFFLINE	Number	00001000h	
FILE_ATTRIBUTE_READONLY	Number	00000001h	
FILE_ATTRIBUTE_REPARSE_POINT . .	Number	00000400h	
FILE_ATTRIBUTE_SPARSE_FILE . . .	Number	00000200h	
FILE_ATTRIBUTE_SYSTEM	Number	00000004h	
FILE_ATTRIBUTE_TEMPORARY	Number	00000100h	
FILE_BEGIN	Number	00000000h	
FILE_CURRENT	Number	00000001h	
FILE_DELETE_CHILD	Number	00000040h	
FILE_END	Number	00000002h	
FILE_READ_DATA	Number	00000001h	
FILE_SHARE_DELETE	Number	00000004h	
FILE_SHARE_READ	Number	00000001h	
FILE_SHARE_WRITE	Number	00000002h	
FILE_WRITE_DATA	Number	00000002h	

FOCUS_EVENT . . . . . Number 00000010h
FORMAT_MESSAGE_ALLOCATE_BUFFER . Number 00000100h
FORMAT_MESSAGE_FROM_SYSTEM . . . Number 00001000h
FormatMessage . . . . . Text FormatMessageA
GENERIC_ALL . . . . . Number 10000000h
GENERIC_EXECUTE . . . . . Number 20000000h
GENERIC_READ . . . . . Number -80000000h
GENERIC_WRITE . . . . . Number 40000000h
GetCommandLine . . . . . Text GetCommandLineA
HANDLE . . . . . Text DWORD
HEAP_GENERATE_EXCEPTIONS . . . . Number 00000004h
HEAP_GROWABLE . . . . . Number 00000002h
HEAP_NO_SERIALIZE . . . . . Number 00000001h
HEAP_REALLOC_IN_PLACE_ONLY . . . Number 00000010h
HEAP_ZERO_MEMORY . . . . . Number 00000008h
IDABORT . . . . . Number 00000003h
IDCANCEL . . . . . Number 00000002h
IDCLOSE . . . . . Number 00000008h
IDCONTINUE . . . . . Number 0000000Bh
IDHELP . . . . . Number 00000009h
IDIGNORE . . . . . Number 00000005h
IDNO . . . . . Number 00000007h
IDOK . . . . . Number 00000001h
IDRETRY . . . . . Number 00000004h
IDTIMEOUT . . . . . Number 00007D00h
IDTRYAGAIN . . . . . Number 0000000Ah
IDYES . . . . . Number 00000006h
INVALID_HANDLE_VALUE . . . . . Number -00000001h
KBDOWN_FLAG . . . . . Number 00000001h
KEY_EVENT . . . . . Number 00000001h
KEY_MASKS . . . . . Number 0000001Fh
LEFT_ALT_PRESSED . . . . . Number 00000002h
LEFT_CTRL_PRESSED . . . . . Number 00000008h
MB_ABORTRETRYIGNORE . . . . . Number 00000002h
MB_APPLMODAL . . . . . Number 00000000h
MB_CANCELTRYCONTINUE . . . . . Number 00000006h
MB_DEFBUTTON1 . . . . . Number 00000000h
MB_DEFBUTTON2 . . . . . Number 00000100h
MB_DEFBUTTON3 . . . . . Number 00000200h
MB_DEFBUTTON4 . . . . . Number 00000300h
MB_HELP . . . . . Number 00004000h
MB_ICONASTERISK . . . . . Number 00000040h
MB_ICONERROR . . . . . Number 00000010h
MB_ICONEXCLAMATION . . . . . Number 00000030h
MB_ICONHAND . . . . . Number 00000010h
MB_ICONINFORMATION . . . . . Number 00000040h
MB_ICONQUESTION . . . . . Number 00000020h
MB_ICONSTOP . . . . . Number 00000010h
MB_ICONWARNING . . . . . Number 00000030h
MB_OKCANCEL . . . . . Number 00000001h
MB_OK . . . . . Number 00000000h
MB_RETRYCANCEL . . . . . Number 00000005h
MB_SYSTEMMODAL . . . . . Number 00001000h

MB_TASKMODAL . . . . . Number 00002000h
MB_USERICON . . . . . Number 00000080h
MB_YESNOCANCEL . . . . . Number 00000003h
MB_YESNO . . . . . Number 00000004h
MENU_EVENT . . . . . Number 00000008h
MOUSE_EVENT . . . . . Number 00000002h
MessageBox . . . . . Text MessageBoxA
NULL . . . . . Number 00000000h
NUMLOCK_ON . . . . . Number 00000020h
OPEN_ALWAYS . . . . . Number 00000004h
OPEN_EXISTING . . . . . Number 00000003h
PeekConsoleInput . . . . . Text PeekConsoleInputA
RIGHT_ALT_PRESSED . . . . . Number 00000001h
RIGHT_CTRL_PRESSED . . . . . Number 00000004h
ReadConsoleInput . . . . . Text ReadConsoleInputA
ReadConsole . . . . . Text ReadConsoleA
SCROLLLOCK_ON . . . . . Number 00000040h
SHIFT_MASK . . . . . Number 00000010h
SHIFT_PRESSED . . . . . Number 00000010h
STD_ERROR_HANDLE . . . . . Number -0000000Ch
STD_INPUT_HANDLE . . . . . Number -0000000Ah
STD_OUTPUT_HANDLE . . . . . Number -0000000Bh
SetConsoleTitle . . . . . Text SetConsoleTitleA
TAB . . . . . Number 00000009h
TRUE . . . . . Number 00000001h
TRUNCATE_EXISTING . . . . . Number 00000005h
VK_11 . . . . . Number 000000BDh
VK_12 . . . . . Number 000000BBh
VK_ADD . . . . . Number 0000006Bh
VK_BACK . . . . . Number 00000008h
VK_CANCEL . . . . . Number 00000003h
VK_CAPITAL . . . . . Number 00000014h
VK_CLEAR . . . . . Number 0000000Ch
VK_CONTROL . . . . . Number 00000011h
VK_DECIMAL . . . . . Number 0000006Eh
VK_DELETE . . . . . Number 0000002Eh
VK_DIVIDE . . . . . Number 0000006Fh
VK_DOWN . . . . . Number 00000028h
VK_END . . . . . Number 00000023h
VK_ESCAPE . . . . . Number 0000001Bh
VK_EXECUTE . . . . . Number 0000002Bh
VK_F10 . . . . . Number 00000079h
VK_F11 . . . . . Number 0000007Ah
VK_F12 . . . . . Number 0000007Bh
VK_F13 . . . . . Number 0000007Ch
VK_F14 . . . . . Number 0000007Dh
VK_F15 . . . . . Number 0000007Eh
VK_F16 . . . . . Number 0000007Fh
VK_F17 . . . . . Number 00000080h
VK_F18 . . . . . Number 00000081h
VK_F19 . . . . . Number 00000082h
VK_F1 . . . . . Number 00000070h
VK_F20 . . . . . Number 00000083h

VK_F21 . . . . . Number 00000084h
VK_F22 . . . . . Number 00000085h
VK_F23 . . . . . Number 00000086h
VK_F24 . . . . . Number 00000087h
VK_F2 . . . . . Number 00000071h
VK_F3 . . . . . Number 00000072h
VK_F4 . . . . . Number 00000073h
VK_F5 . . . . . Number 00000074h
VK_F6 . . . . . Number 00000075h
VK_F7 . . . . . Number 00000076h
VK_F8 . . . . . Number 00000077h
VK_F9 . . . . . Number 00000078h
VK_HELP . . . . . Number 0000002Fh
VK_HOME . . . . . Number 00000024h
VK_INSERT . . . . . Number 0000002Dh
VK_LBUTTON . . . . . Number 00000001h
VK_LCONTROL . . . . . Number 000000A2h
VK_LEFT . . . . . Number 00000025h
VK_LMENU . . . . . Number 000000A4h
VK_LSHIFT . . . . . Number 000000A0h
VK_MENU . . . . . Number 00000012h
VK_MULTIPLY . . . . . Number 0000006Ah
VK_NEXT . . . . . Number 00000022h
VK_NUMLOCK . . . . . Number 00000090h
VK_NUMPAD0 . . . . . Number 00000060h
VK_NUMPAD1 . . . . . Number 00000061h
VK_NUMPAD2 . . . . . Number 00000062h
VK_NUMPAD3 . . . . . Number 00000063h
VK_NUMPAD4 . . . . . Number 00000064h
VK_NUMPAD5 . . . . . Number 00000065h
VK_NUMPAD6 . . . . . Number 00000066h
VK_NUMPAD7 . . . . . Number 00000067h
VK_NUMPAD8 . . . . . Number 00000068h
VK_NUMPAD9 . . . . . Number 00000069h
VK_PAUSE . . . . . Number 00000013h
VK_PRINT . . . . . Number 0000002Ah
VK_PRIOR . . . . . Number 00000021h
VK_RBUTTON . . . . . Number 00000002h
VK_RCONTROL . . . . . Number 000000A3h
VK_RETURN . . . . . Number 0000000Dh
VK_RIGHT . . . . . Number 00000027h
VK_RMENU . . . . . Number 000000A5h
VK_RSHIFT . . . . . Number 000000A1h
VK_SCROLL . . . . . Number 00000091h
VK_SEPARATER . . . . . Number 0000006Ch
VK_SHIFT . . . . . Number 00000010h
VK_SNAPSHOT . . . . . Number 0000002Ch
VK_SPACE . . . . . Number 00000020h
VK_SUBTRACT . . . . . Number 0000006Dh
VK_TAB . . . . . Number 00000009h
VK_UP . . . . . Number 00000026h
WINDOW_BUFFER_SIZE_EVENT . . . . Number 00000004h
WriteConsoleOutputCharacter . . Text WriteConsoleOutputCharacterA

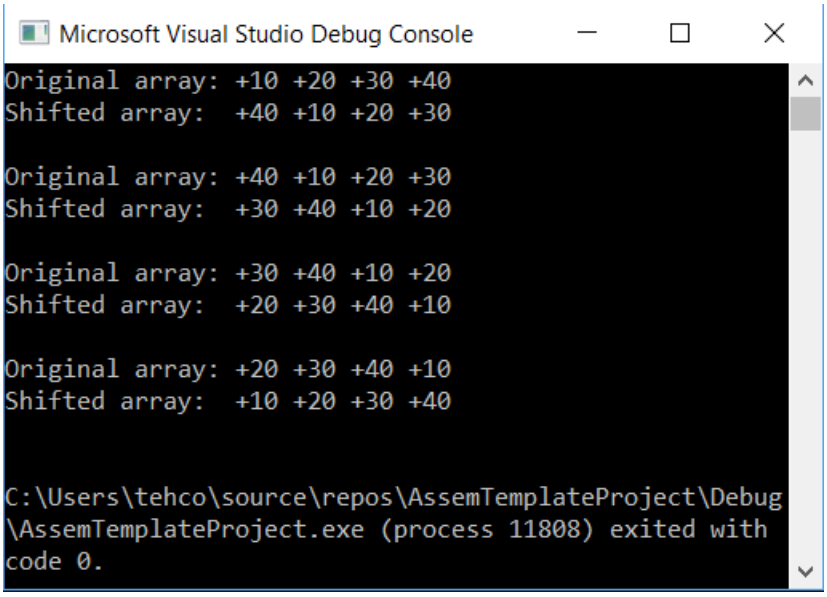
```
WriteConsole . . . . . .        Text     WriteConsoleA
array . . . . . .   DWord  00000000 _DATA
black . . . . . .   Number          00000000h
blue . . . . . .    Number          00000001h
brown . . . . . .   Number          00000006h
counter . . . . . .  DWord  00000010 _DATA
cyan . . . . . .    Number          00000003h
exit . . . . . .    Text     INVOKE ExitProcess,0
gray . . . . . .    Number          00000008h
green . . . . . .   Number          00000002h
lightBlue . . . . . .  Number          00000009h
lightCyan . . . . . .  Number          0000000Bh
lightGray . . . . . .  Number          00000007h
lightGreen . . . . . . Number          0000000Ah
lightMagenta . . . . . .        Number          0000000Dh
lightRed . . . . . .  Number          0000000Ch
magenta . . . . . . Number          00000005h
msgOriginal . . . . . .         Byte     00000016 _DATA
msgShifted . . . . . .Byte     00000027 _DATA
red . . . . . .     Number          00000004h
space . . . . . .   Byte     00000014 _DATA
white . . . . . .   Number          0000000Fh
wsprintf . . . . . .  Text     wsprintfA
yellow . . . . . .  Number          0000000Eh

          0 Warnings
          0 Errors
```

Assembly Source Code

```
;;     Author:       Marco Martinez
;;     Program:      ShiftElements.asm
;;     Date:         10/14/2018
;;     Purpose:      Using a loop and indexed addressing, write code that rotates the members
;;                   of a 32-bit integer array forward one position. The value at the end of
;;                   the array must wrap around to the first position.
;;                   For example, the array [10,20,30,40] would be transformed into
;;                   [40,10,20,30].
;;
;;     Software Change Record
;;     Name        Date            What
;;     Marco    10/14              Baseline for ShiftElements.asm
;;     Marco    12/6        Implement reporting features.
;;

INCLUDE Irvine32.inc

.data
       array DWORD 10,20,30,40
       counter DWORD 1
       space BYTE " ",0
       msgOriginal BYTE "Original array: ",0
       msgShifted BYTE "Shifted array: ",0

.code
main proc
       mov ecx,0
```

```
	program:
		pushad
		mov esi,OFFSET array
		mov ecx,LENGTHOF array
		mov edx,OFFSET msgOriginal
		call writeString
		mov edx,OFFSET space
		L1:
			mov eax,[esi]
			call writeInt
			call writeString
			add esi,4
		loop L1
		call crlf
		popad

		pushad
		mov esi,OFFSET array+3*4
		mov eax,[esi]
		mov esi,OFFSET array
		mov ebx,[esi]
		mov [esi],eax
		mov eax,ebx
		mov esi,OFFSET array+1*4
		mov ecx,LENGTHOF array-1
		L2:
			mov ebx,[esi]
			mov [esi],eax
			mov eax,ebx
			inc edx
			mov counter,edx
			add esi,4
		loop L2
		popad

		pushad
		mov esi,OFFSET array
		mov ecx,LENGTHOF array
		mov edx,OFFSET msgShifted
		call writeString
		mov edx,OFFSET space
		call writeString
		L3:
			mov eax,[esi]
			call writeInt
			call writeString
			add esi,4
		loop L3
		call crlf
		popad

		call crlf
		inc ecx
		cmp ecx,4
	jl program

	invoke ExitProcess,0
main endp
end main
```

Screenshot


Microsoft Visual Studio Debug Console
Original array: +10 +20 +30 +40
Shifted array: +40 +10 +20 +30
Original array: +40 +10 +20 +30
Shifted array: +30 +40 +10 +20
Original array: +30 +40 +10 +20
Shifted array: +20 +30 +40 +10
Original array: +20 +30 +40 +10
Shifted array: +10 +20 +30 +40
C:\Users\tehco\source\repos\AssemTemplateProject\Debug
\AssemTemplateProject.exe (process 11808) exited with
code 0.